

UNIVERSITÀ DEGLI STUDI DI TORINO
SCUOLA DI SCIENZE DELLA NATURA
Corso di Laurea Magistrale in Fisica dei Sistemi Complessi



Tesi di Laurea Magistrale

THE FANCY TITLE OF MY FANCY THESIS

Relatore:
Prof. Michele Caselle

Co-relatore:
Dott. Matteo Osella

Candidato:
Filippo Valle

Controrelatore:
Dott. Matteo Cereda

Anno Accademico 2018/2019

Computers are incredibly fast, accurate, and stupid.
Human beings are incredibly slow, inaccurate, and brilliant.
Together they are powerful beyond imagination Albert Einstein

Abstract

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

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

In recent years datasets with a large amount of cancer samples' data cancer are available, the most complete is The Cancer Genome Atlas (TCGA). From this dataset it is easy to get, for example, gene expression data from RNA sequencing experiments together with a lot of information about the sample itself.

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. This exact 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, they can be classified by their specific subject. In a similar way a set of samples can be classified, for example, by the tissue it comes from or by the type of the disease it is referred to.

The similarities between gene expression data and linguistics 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, it describes documents as a mixture of topics. Topics are nothing but communities of words each 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. In fact once a community structure of genes emerges, it is possible to run a hypergeometric test on the whole set in order to verify if they reveal some type of enrichment and to inspect their common properties.

The specific algorithm used in this work is particularly unique because it allows overlapping clusters; so it is possible to find genes belonging to different topics, this empowers a lot of new possibilities to investigate the network.

Furthermore a hierarchic approach can be used in finding topics, this let 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 by tissue, then by tumour type and so on.

Contents

Introduction	1
1 Data presentation	3
1.1 RNA-Sequencing	3
1.2 Datasets	5
2 Data structure	6
2.1 Component systems	6
2.2 Universal laws in gene expression	7
2.3 Null model construction	10
2.4 Statistical laws differentiate by tissue	13
3 Scaling laws	17
4 Topic modelling	22
4.1 Metrics and benchmarks	25
4.1.1 Hierarchical clustering	27
4.1.2 LDA	28
4.2 Pre-process	29
4.3 Run	31
4.3.1 Run on Gene Tissues Expression dataset	31
4.3.2 Shuffling	34
4.3.3 Standard algorithms	40
4.3.4 Topics	41
4.3.5 Run on The Cancer Genomics Atlas	44
4.3.6 Healthy and diseased together	46
4.4 Results	52
5 Conclusions	53
Appendices	54
A Hierarchical stochastic block model	55
B Homogeneity, completeness and V-measure	60
Bibliography	66

Introduction

In recent years the study of complex systems is becoming more interesting especially when some different systems that share interesting and 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 kind 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 study and gain information about each other. 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 discriminate healthy and diseased tissues at the beginning. Once diseased tissues are found, the next goal is to separate cancer types and ultimately sub-types, which is not always easy clinically. The interest in the development of a method able to separate well this kind of data is increasing [9].

The methods to gain this goal are derived firstly from linguistics; in particular, a topic model approach will be widely described.

In chapter 1, 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 diseased tissues and healthy tissues.

In chapter 2, I will describe the basics of component systems and give some useful mathematical definitions. Here it will be shown 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 are shown to explain the different behaviour of different tissues.

In chapter 3, I will study the gene expression across samples of all the genes. This analysis is preparatory to the following sections where some gene selection would be necessary.

Demonstrated that linguistics and biological data share some statistical laws, in section 4, the main one, I will use topic modelling to perform network analysis on the 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 [10] to some new proposal [11, 12, 27]. 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 this algorithm will be widely discussed.

In chapter 5, 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 [1], are encapsulated in a tool available at <https://github.com/fvalle1/tacos>. During this work I used different python libraries such as pandas [2], scipy [3], numpy [4] and matplotlib [5]. Some advanced analysis required tensorflow [6] and pySpark [7]. The topic modelling stochastic block model's minimization functions are implemented in the graph-tool library [8]. The full work repository is on GitHub® at https://github.com/fvalle1/master_thesis.

Chapter 1

Data presentation

1.1 RNA-Sequencing

Data come from a RNA-sequencing [13] experiments.

RNA-Sequencing data provide a unique snapshot of the transcriptomic status of the disease and look at an unbiased population of transcripts that allows the identification of novel transcripts, fusion transcripts and non-coding RNAs that could be undetected with different technologies.

Briefly, long RNAs are first converted into a library of cDNA fragments through either RNA fragmentation or DNA fragmentation (see main text). Sequencing adaptors (blue) 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, as illustrated at the bottom; a yeast ORF with one intron is shown.

The general steps to prepare a complementary DNA (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 analyze 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, where either small sequences are removed or a tight range of sequence lengths are selected. Because small RNAs like miRNAs are lost, these are analyzed 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.

In order to collect Gene expression data is sufficient to count how many reads are mapped to a specific exon or gene.

normalization Usually gene expression data can be normalized in different ways

- Counts
- RPK
- TPM
- FPKM

Count reads correspond to raw data. Counts needs no normalisation to be treated but may have some biases. For example, longer genes may have more reads than shorter gene just because they are longer. That's why other kind of normalization can be used.

RPK¹ remove the length bias by dividing counts by the gene length L : $RPK = \frac{\text{counts}}{L}$. This solves some problems but doesn't take care of the different sizes of the transcript in different samples.

FPKM² calculation normalizes read count by dividing it by the gene length and by the total number of reads mapped to protein-coding genes in that sample.

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

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 some thresholds are put in particular $FPKM$ below 0,1 and above 10^5 will not be considered, maybe they come from some kind of experimental error.

TPM³ tries to unify the sizes of the samples: $TPM = \frac{RC_g * 10^9}{\sum_{g'} \left(\frac{RC_{g'}}{l_{g'}} \right) RC_{pc} * L}$.

In this work the idea is not to 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 sizes distribution needs not to be made with TPM, because the quantity that's going to be studied is the quantity the normalization trashes out.

¹Reads Per Kilobase of transcript

²Fragments Per Kilobase of transcript per Million mapped reads

³Transcript Per Million

1.2 Datasets

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 [18]. GTEx 2016-01-15 v7 RNASeqCv1.1.8 version was downloaded⁴. GTEx contains 11688 samples of 53 tissues. For many of them a sub-tissue label is available. As highlighted in [15] 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 different tissues.

The other dataset considered is The Cancer Genome Atlas (TCGA) dataset [14]. Data were collected via Genomic Data Commons tools⁵ 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 site or tissues. On this dataset there are a great quantity of metadata, in particular the disease type will be considered in this work.

Onother source of data used in this work is [17] which authors tried to unify GTEx and TCGA [16], when possible.

Protein coding The whole dataset contains infos on approximately 60000 elements with different *ENSG* identifier. Only $\simeq 20000$ of this are protein coding genes, using Ensemble⁶ protein coding genes are selected.

⁴<https://gtexportal.org/home/datasets>

⁵<https://portal.gdc.cancer.gov>

⁶<https://ensembl.org>

Chapter 2

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 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 2.1 an example of this kind of matrices.

2.1 Component systems

$$\begin{array}{c} \text{Components} \end{array} \left(\begin{array}{ccccc} & \text{Realizations} & & & \\ n_{11} & n_{12} & n_{13} & \dots & n_{1R} \\ n_{21} & n_{22} & n_{23} & \dots & n_{2R} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ n_{N1} & n_{N2} & n_{N3} & \dots & n_{NR} \end{array} \right)$$

Figure 2.1: Structure of a matrix representing component systems with $i = 0 \dots N$ rows and $j = 0 \dots R$ columns.

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 correspond to row i and column j is the number of times the word i appears in book j . The same happens if one considers Wikipedia's pages. Other examples are Lego[®] sets where components are the Lego[®] bricks and realizations Lego[®] packages and protein domains; all these were described and well studied in [19, 20].

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, defined as

$$o_i = \frac{\sum_{j=1}^R (1 - \delta_{n_{ij},0})}{R}, \quad (2.1)$$

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 all components with $O_i = 1$ is known as **core**. Components with high ($\simeq 1$) occurrence are present in mostly all realizations of the datasets, in linguistics these components are articles. Components with low occurrence $\simeq 0$ are present only in a few realizations and are the most specific ones [21].

The sum across all columns is called **abundance** of a component and is defined as:

$$a_i = \sum_{j=1}^R n_{ij}; \quad (2.2)$$

dividing this by the global abundance

$$a = \sum_{i=1}^N a_i \quad (2.3)$$

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

$$f_i = \frac{a_i}{\sum_{c=1}^N a_c}. \quad (2.4)$$

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

$$f_{ij} = \frac{n_{ij}}{\sum_{c=1}^N n_{cj}}. \quad (2.5)$$

The sum of all abundances in the same realization

$$M_j = \sum_{c=1}^N n_{cj} \quad (2.6)$$

is called **size**.

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

$$f_i \propto r_i^{-\alpha} \quad (2.7)$$

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

2.2 Universal laws in gene expression

TCGA

Analysing TCGA dataset the first interesting analysis is to plot the sorted abundance, this gives the so called Zipf's law. In figure 2.2, it is shown the frequency rank plot. It is interesting that this kind of data distribute according a power law with, this same behaviour can be found in many systems such as linguistics' ones [21]. Another

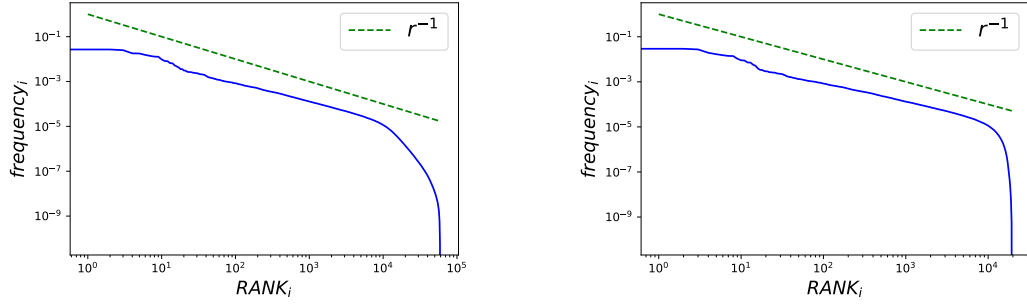


Figure 2.2: Zipf's law from FPKM normalised data. On the right considering only protein-coding genes.

interesting fact is that considering in the analysis both coding and non-coding genes this gives a double-scaled power law. This is due to the fact that 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.

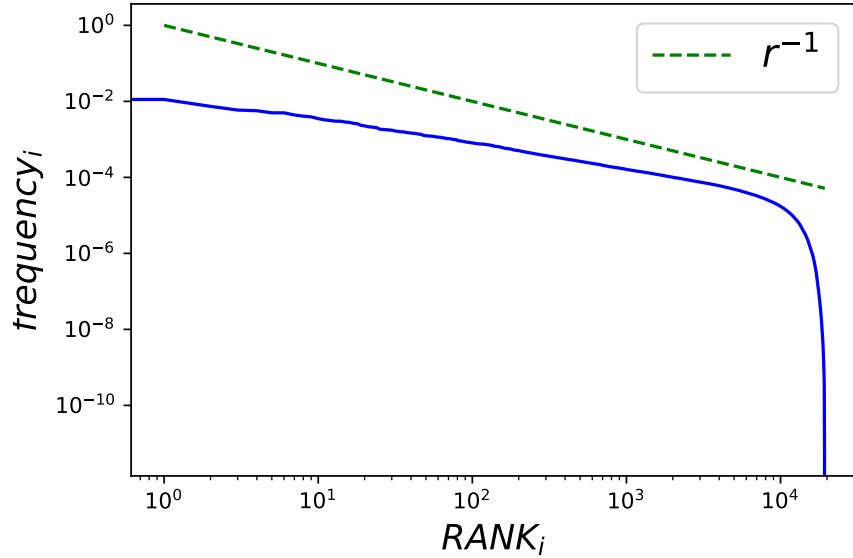


Figure 2.3: Zipf's law of protein-coding genes considering counts.

GTE_x

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

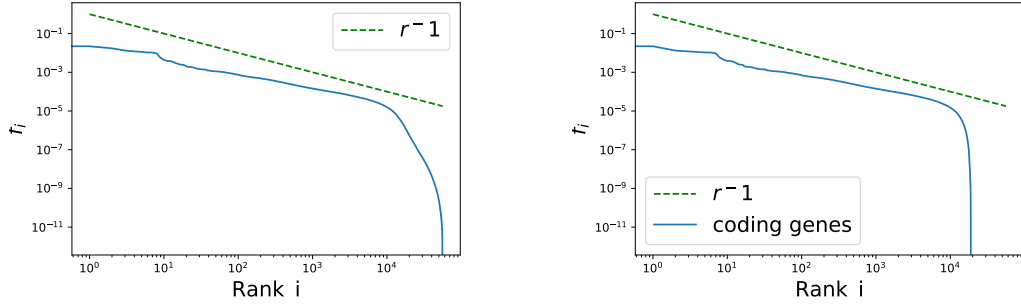


Figure 2.4: Zipf's law from GTEx count data. On the left all genes considered, on the right only protein coding ones.

this point of view. The power law is found and considering non coding genes lead to a knee in the power law.

Going further in the analysis it is possible make a histogram of occurrences defined by 2.1, also known as U_s . Even in this kind of analysis it is possible to see the different

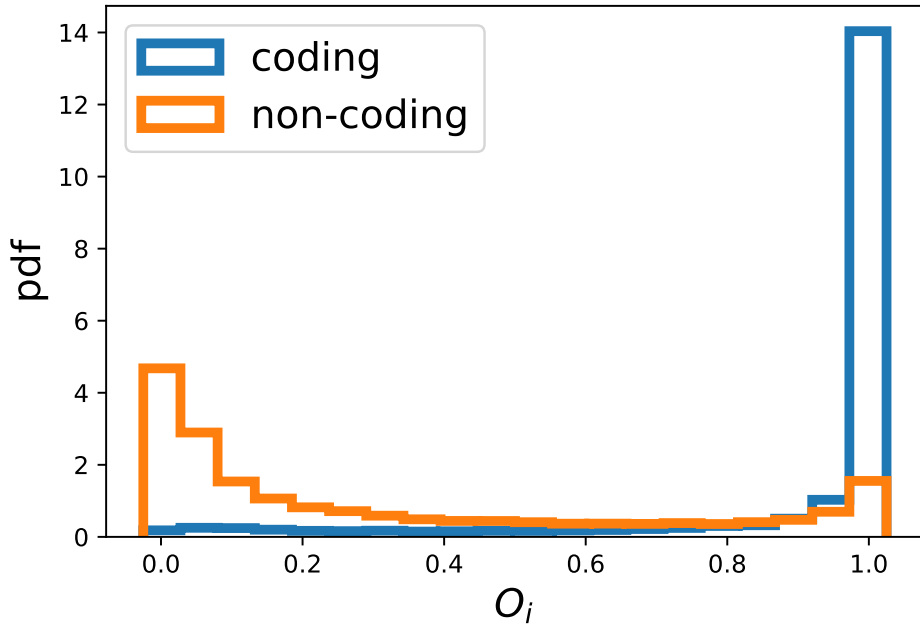


Figure 2.5: The histogram of the occurrences O_i . Coding and non-coding entries are coloured.

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.

The same behaviour can be observed considering not all samples but just a given tissue. In this case $O_i = 0$ means that a gene has a non-zero expression in just one of

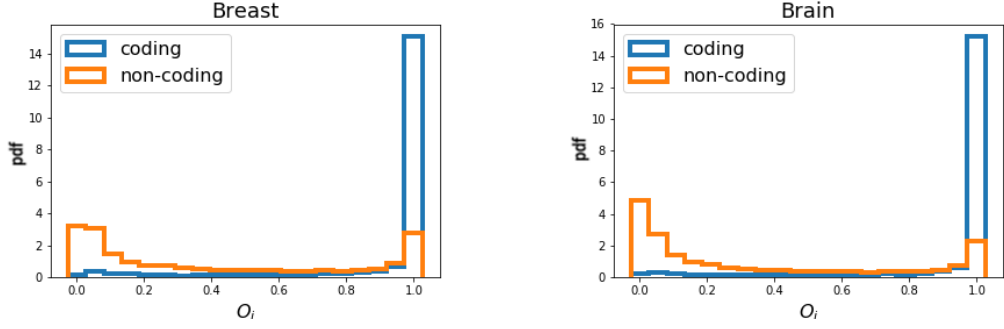


Figure 2.6: Same behaviour is observed looking at one tissue a time.

the samples of the tissue considered; in other words if a gene never express in a tissue it is not considered when constructing this tissue specific U distribution.

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

2.3 Null model construction

The kind of data considered in this work comes from RNA Sequencing experiments. These experiments use 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 [22], a random matrix has to be created. This matrix is a collection of components and realizations exactly as 2.1. 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 2.2. Values of each column are extracted until the size 2.6 is reached. Strictly speaking it is a multinomial process

$$P(\{n_i\}; M) = \frac{M!}{\prod_{i=1}^N n_i} \prod_{i=1}^N f_i^{n_i} \quad (2.8)$$

where n_i is the number of components with frequency f_i , being $f_i = \frac{a_i}{\sum_{c=1}^N a_c}$ as defined in 2.4.

Figure 2.7 shows an example of this, M components are picked up with respect to their frequency in the dataset. The most abundant components, which are also the ones with higher frequency (frequency is nothing but the normalised abundance), have a greater probability to be picked up.

Using this construction on real counts data, by definition the Zipf's law sampled is identical to the data's one. By construction the distribution of the sizes of the sampling and of the data are also identical.

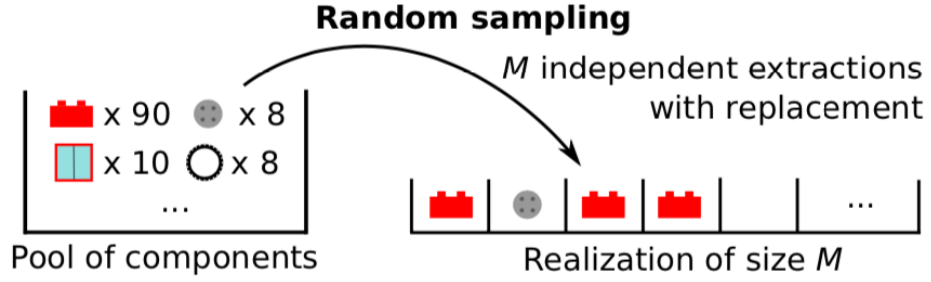


Figure 2.7: Random sampling of components to build a realization of size M .

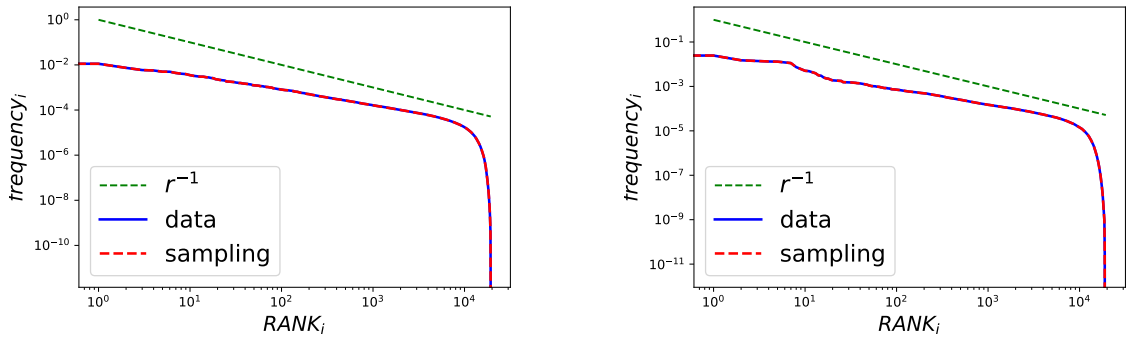


Figure 2.8: Zipf's law sampled; TCGA(left) and GTEx (right). Sampling is reported for comparison.

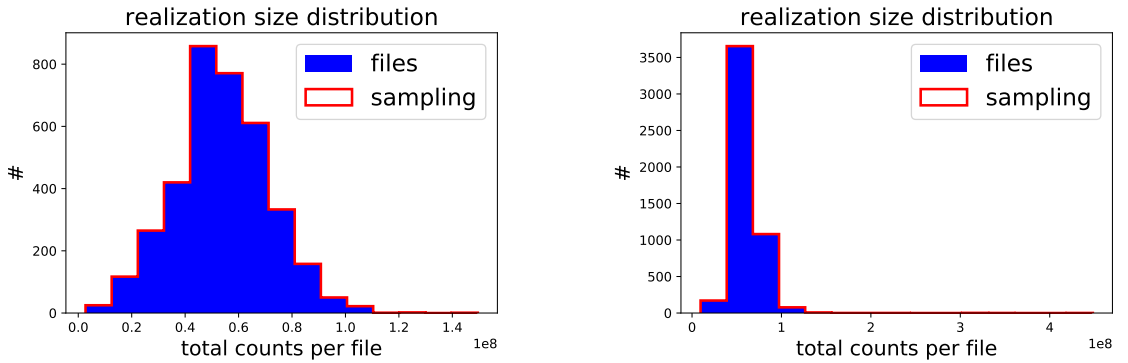


Figure 2.9: Distribution of sizes M ; TCGA(left) and GTEx (right). Sampling is reported for comparison.

Looking at the U s in figure 2.10, it is evident that data behave differently from sampling. This is a signal that the null model is not enough to explain the data matrices. In particular, it is evident that the null model generates the matrices in a manner such that more components have high occurrence with respect to the original data. This can be easily explained, in fact in real world there are some genes that 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 high abundance) and so samples these quite often (components

with high abundance have a greater chance to be picked up by the null model sampling). The set of the genes with $O_i = 1$ is called **core**.

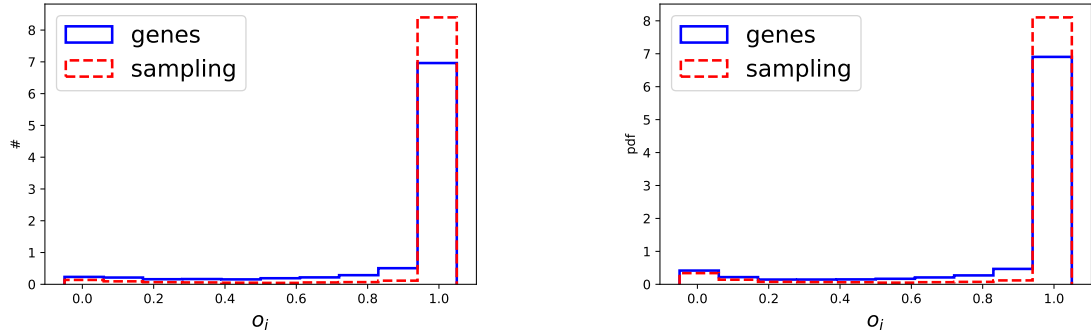


Figure 2.10: Occurrence distributions; TCGA(left) and GTEx (right). Sampling is reported for comparison.

Plotting on the abscissa the size of samples, on the ordinate the number of genes expressed one point per sample, one obtains the so-called Heaps' law [23]. In figure 2.11 the Heaps' law is presented compared to the one obtained by sampling. Again the curves differ and the null model is not enough complete to explain the trend. Note that each data point share the abscissa with a sampling one (figures 2.9 are nothing but the histograms of the abscissas of 2.11). Moreover it is interesting that the ordinate does not start from zero, this happens because there are lot of genes that express everywhere, the core. It happens that the sampling curve is 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 real world are expressed only the genes that are really useful in the sample, and this is not derivable just by sampling. This fact is coherent with the fact that the U s differ. Another way to see this is looking at the histograms

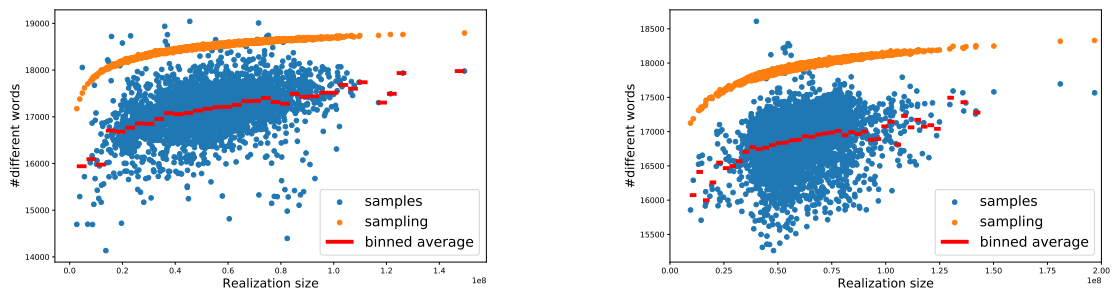


Figure 2.11: Heaps' law; TCGA(left) and GTEx (right). Sampling is reported for comparison.

of the number of different genes expressed, actually the distribution of the 2.11 y-axis. Figure 2.12 shows that these distributions are completely different if one looks at the data and at the samples.

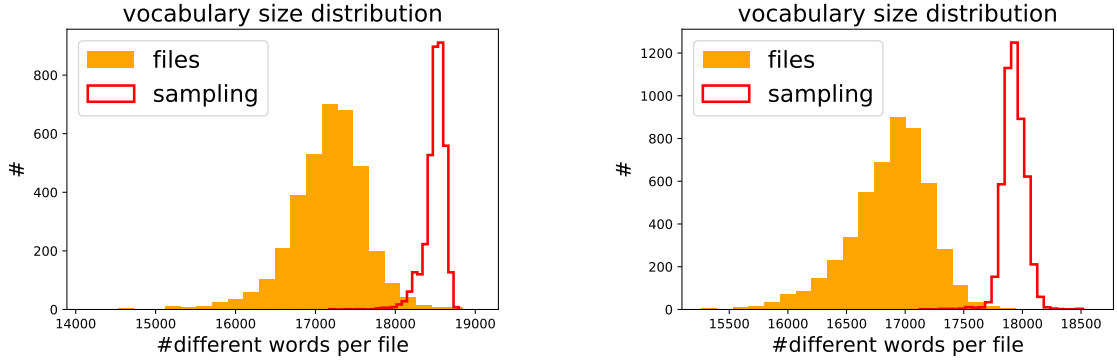


Figure 2.12: Number of genes expressed in a sample; TCGA(left) and GTEx (right). Sampling is reported for comparison.

2.4 Statistical laws differentiate by tissue

Observing the GTEx dataset of healthy samples it is possible to study how it is possible to see the tissue differentiation and how to study tissues' differences, [24] 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 it is necessary to get all the samples of a given tissue. Then one estimates the average expression per 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 and integrating (cumulative summing) one have the fraction of transcript due to 1, 2, 3... genes. This is reported in figure 2.13. This analysis is done using TPM to avoid biases due to gene lengths or to samples sizes. Where a curve is steep it means that a few genes'

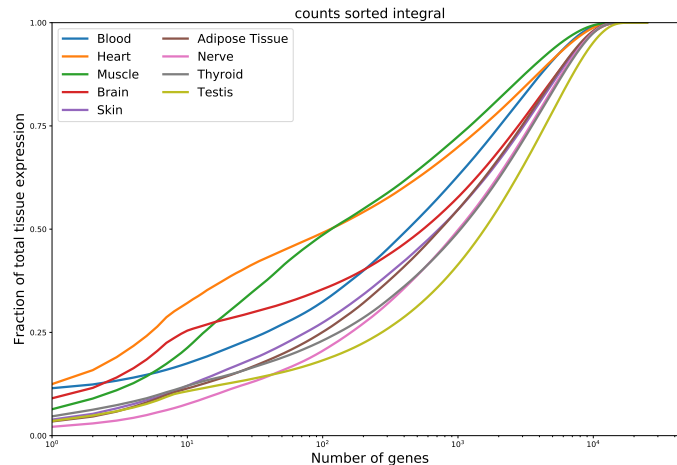


Figure 2.13: The integral of the sorted abundances for each tissue.

expression represent 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 2.14 the same analysis is done for the sub-tissues of Brain, also this sub-type present a great separation.

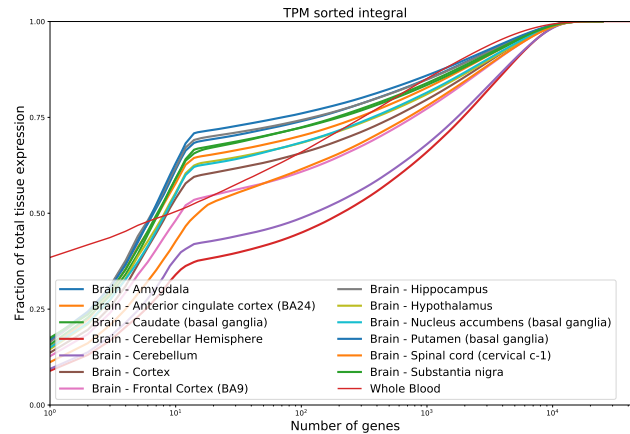


Figure 2.14: The integral of the sorted abundances for sub-types of Brain. This is done using TPM to avoid biases due to gene lengths. Blood is plotted for reference.

Another way to interpret this analysis is thinking 2.13 as the integral of the Zipf's law. So it could be interesting to examine the Zipf one tissue a time. In figure 2.15 are reported the Zipf's law for some tissues with an extremal behaviour. From this point

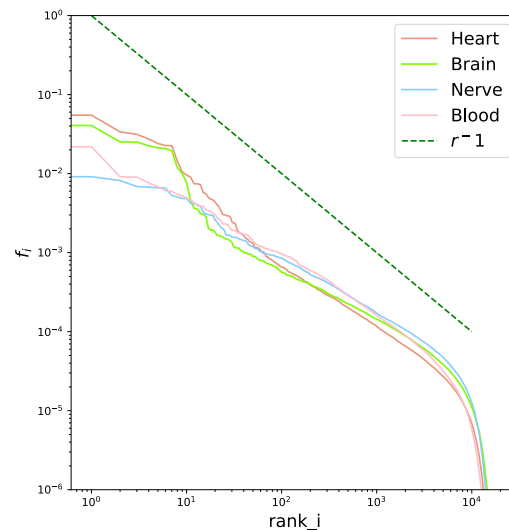


Figure 2.15: The integral of the sorted abundances for each tissue

of view each tissue has its particular slope. The steeper the Zipf the simplest is the tissue: the transcript can be described with few genes.

Coming back to the transcriptome analysis. In figure 2.13 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; in fact the number of genes expressed seen in the Heaps' law plot is the number of genes necessary to explain the whole transcriptome. In figure 2.16, 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 looking at the

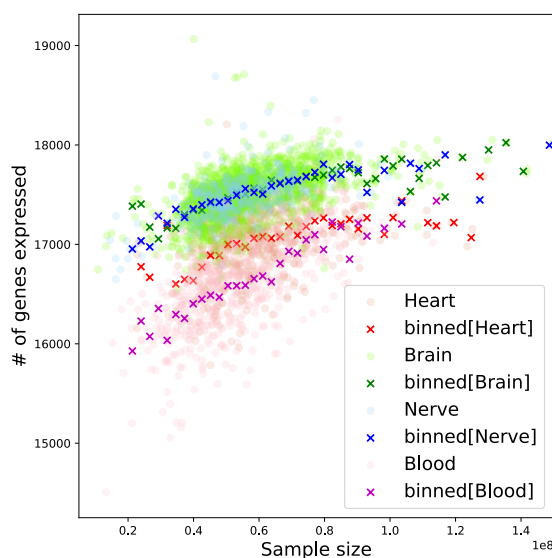


Figure 2.16: The integral of the sorted abundances for each tissue. This plot was done using counts, using TPM it is not possible because the size on the x-axis is a would be a constant.

disease type of cancer samples. In this case there is no evident differentiation as shown in figure 2.17. The only disease that behave differently are *Parangliomas*, but this are associated only to Brain, so the differentiation seen is just a Brain separation. This means that separate diseases would be tricky and much difficult than just separate tissues.

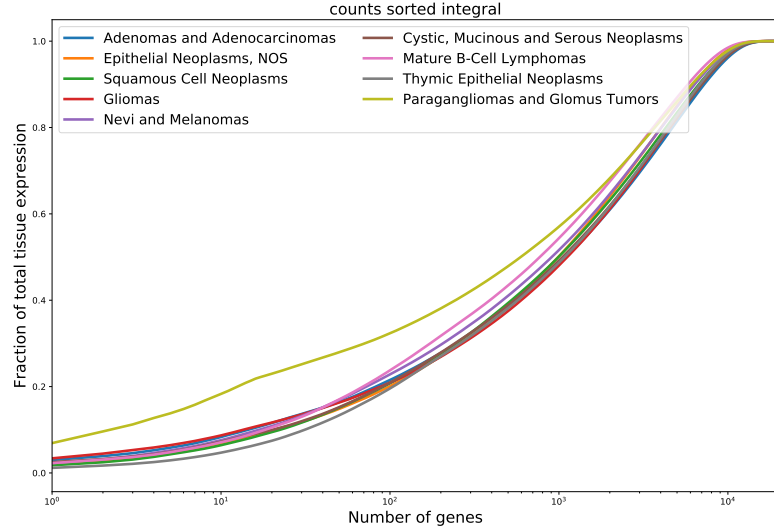


Figure 2.17: The integral of the sorted abundances for each disease type.

All these analyses suggest that there must be a sort of hidden structure in the data that is somehow related with the tissue each sample comes from. In particular there are many Zipf's laws hidden behind the data and each sample is build looking at one of these a time. Also 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 tissue of the sample.

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 3

Scaling laws

One of the goals of this work is to search, reveal, study and use universal laws in bulk gene expression data. As was done in chapter 2 approaches from different field of science are considered at this point.

It can be interesting to study the behaviour of the gene expression across samples.

Given a matrix of components and realisations as 2.1 with expression entries n_{ij} it is possible to estimate the mean of a row $m_i = \langle n_{ij} \rangle_j$ and its variance $\sigma_i^2 = \langle n_{ij}^2 \rangle_j - \langle n_{ij} \rangle_j^2$.

Variance versus mean

First of all it could be interesting to study the variance of expression σ_{counts}^2 versus the average $\langle \text{counts} \rangle$ across tissues. In figure 3.3 the scatter plot of variance versus

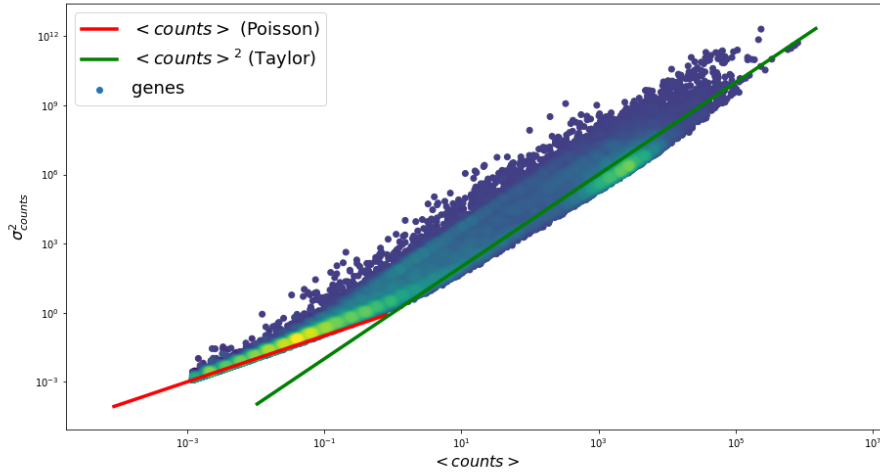


Figure 3.1: Variance versus average. In red the Poisson-like scaling, in green the Taylor-like scaling. All genes are considered

mean reveal some interesting facts. First of all it is evident that data have a double scaling behaviour: when the mean is small ($\lesssim 1$) data have a Poisson-like scaling ($\sigma_{\text{counts}}^2 \sim \langle \text{counts} \rangle$), at higher means instead data have a quadratic scaling ($\sigma_{\text{counts}}^2 \sim$

$\langle \text{counts} \rangle^2$) known in ecology as Taylor's law [25]. This means that at low averages data behaviour is just due to the sampling experimental process, on the contrary the 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 3.1) are evident two clouds of points, one at low averages, one at high averages. These correspond to coding and non coding genes, remembering section 2.2 these two kind of genes have a different behaviour: coding genes are highly expressed in the majority of the samples, non coding ones are less expressed (and so less sampled) in few samples.

Coefficient of Variation

A similar analysis, common in literature, is the analysis of the coefficient of variation squared $CV^2 = \frac{\sigma_{\text{counts}}^2}{\langle \text{counts} \rangle^2}$ represented in figure 3.2. The behaviour is complementary to

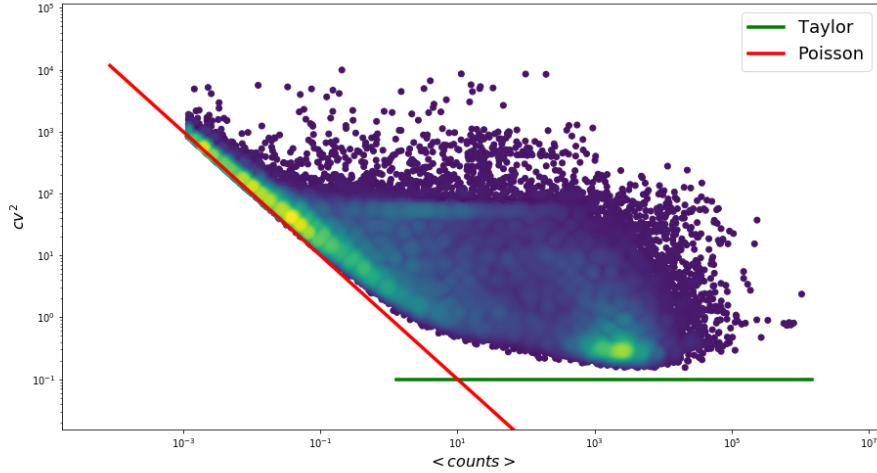


Figure 3.2: Coefficient of variation squared versus average. In red the Poisson-like scaling, in green the Taylor-like scaling

the above discussed double scaling and is quite common in literature looking at single cell RNA sequencing data [26]. Even looking at CV^2 it is evident the presence of the coding and non-coding clouds of points. The non coding genes are on the Poisson-like scaling, $\sigma_{\text{counts}}^2 \sim \langle \text{counts} \rangle$ so $CV^2 = \frac{\sigma_{\text{counts}}^2}{\langle \text{counts} \rangle^2} \sim \frac{1}{\langle \text{counts} \rangle}$, otherwise the protein coding genes are on the Taylor-like curve $CV^2 = \frac{\sigma_{\text{counts}}^2}{\langle \text{counts} \rangle^2} \sim 1$.

Protein coding genes can be isolated and considered on their own. The same analysis confirms that the cloud of genes' points on the Taylor-like scaling are effective the protein coding genes. Following the sampling model of [22] sum up in section 2.3 the averages and variances can be estimated on null matrices. In figure 3.4 the comparison between real genes and sampling ones. 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

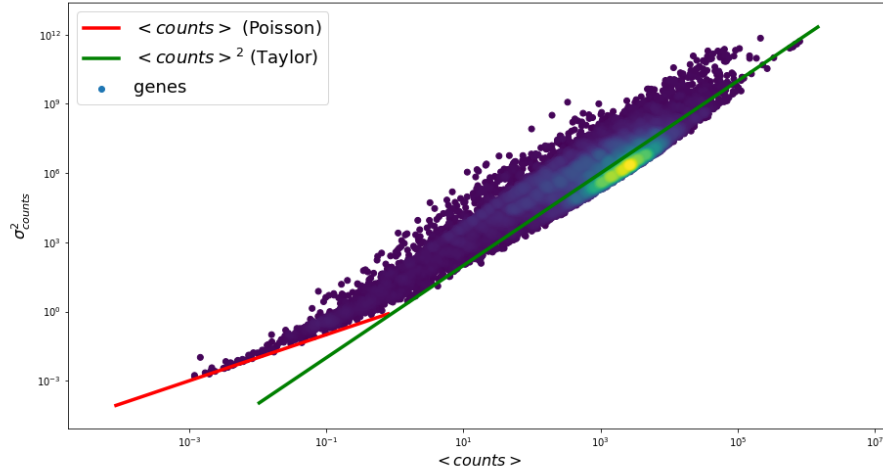


Figure 3.3: Variance versus average. In red the Poisson-like scaling, in green the Taylor-like scaling. Only protein coding genes are considered

and the sizes distribution themselves, they are identical in data and sampling by definition. 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.

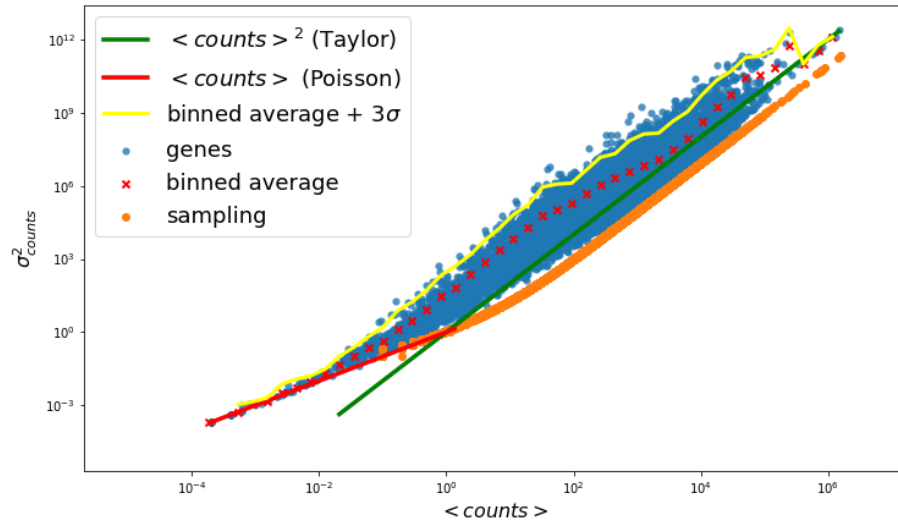


Figure 3.4: Variance versus average. In red the Poisson-like scaling, in green the Taylor-like scaling. In orange the sampling components. Only protein coding genes are considered

Again it is possible to analyse the CV^2 , at this time considering only protein coding genes. Figure 3.2 confirms that the cloud of points near the Taylor-like scaling are the protein coding genes and a double scaling is seen once again.

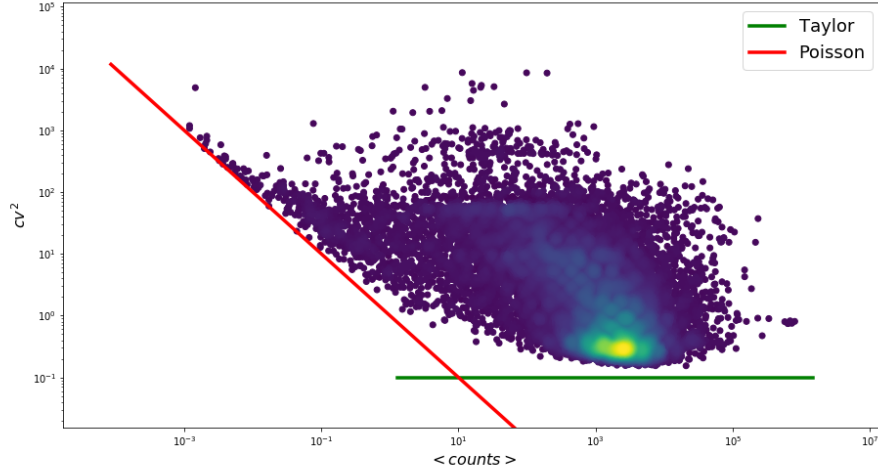


Figure 3.5: Variance versus occurrence

In figure 3.6 the same plot compared to the sampling points. 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 if all expression are identical in all samples (σ_{counts}^2) and an upper bound at $N - 1$ with N the number of realisations and corresponds to the most variable case where a component express in only one realisation and is 0 elsewhere.

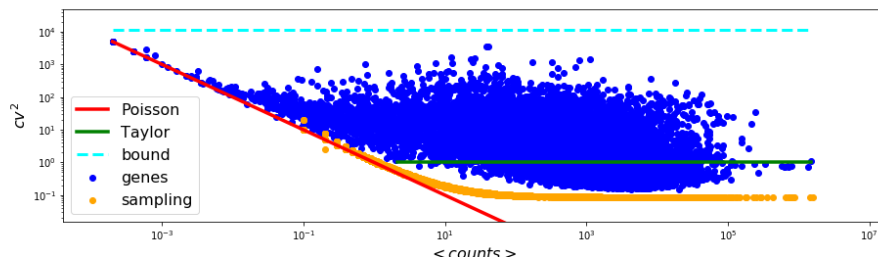


Figure 3.6: Caption

Finally the data have a double scaling when looking at their global variance across realisations, a Poisson-like where the sampling experimental process is more important and a Taylor-like 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 following 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 relation between the occurrence and the average. In figure 3.7 it is shown the result, it is clear that there is a relation between occurrence and average, genes that express in more realisations (higher occurrence and right in the figure) have an higher average. Moreover doesn't exist genes that have high expression in few realisations; genes that are rare are also difficult to find so have a small average. Note that the average has got a bound due to the fact that 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 is at least $\langle \text{counts} \rangle = \frac{1*n}{R}$

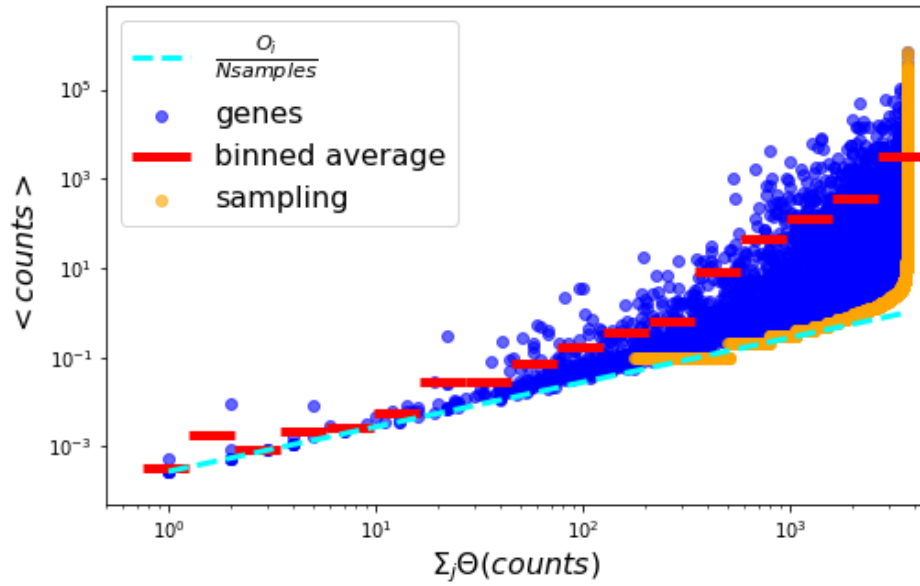


Figure 3.7: Relation between the occurrence of a gene and its average across realisations

Chapter 4

Topic modelling

Once extensively analyzed the structure of the dataset, the goal becomes to develop a machine learning method to learn the hidden structure of the data.

Remembering that in chapter 2 it emerged some kind of structure behind data, where each tissue seemed to be sampled by a different power law, a topic modelling approach it is here proposed.

The idea is that behind 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 model 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 $P(\text{topic}|\text{sample})$, both gives the relation between a *sample* and a *gene*.

In this work it is used an innovative and recent approach to topic modelling, the algorithm was presented by [27] and well explained in details in [28], this belongs to the so-called stochastic block models [29]. Topic modelling is being developed and studied to approach linguistics problems, so this algorithm was developed considering words and books as input, links represents the abundance of a word in a book. In chapter 2 was evident that there are many similarities between data considered in this work and linguistics' dataset. 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 samples due to the gene expression distribution in samples.

The ultimate goal would be to be able to separate healthy and diseased samples than separate and find well-known tumour types, then 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. So the output has got different resolution, the extreme one is the separation, by definition, between samples and genes, at other layers it is possible to have few big clusters until the other extreme were the number of clusters is comparable with the number of nodes.

What the algorithm does is to run a sort of MonteCarlo simulation and find the best partition of the data. The probability that the hidden variables θ describe the

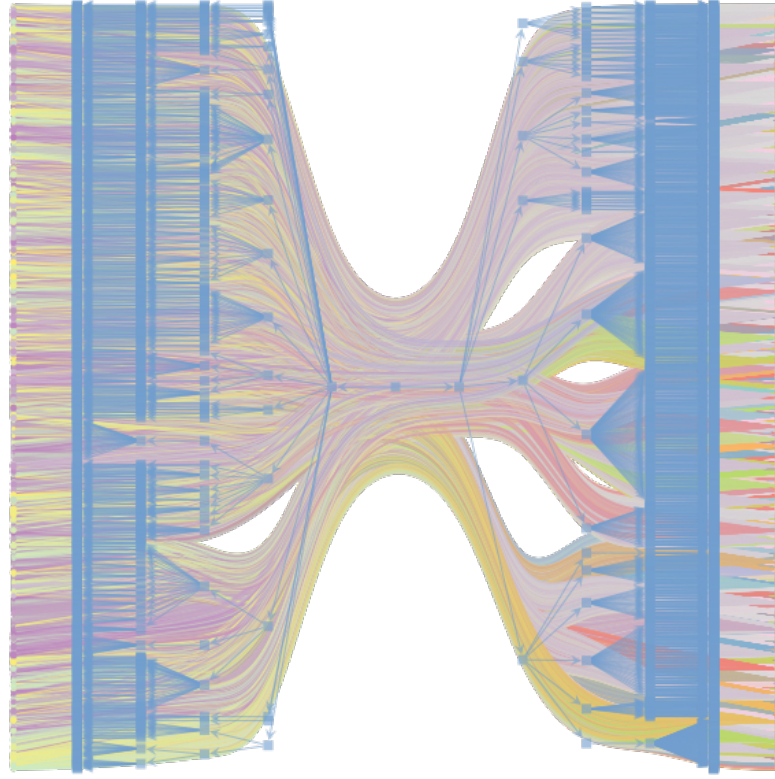


Figure 4.1: An example of a bipartite network. Samples are on the left, genes are on the right. Each link is weighted by gene expression value. On the left side, all nodes of the same colour are clusters of samples. On the right side, all nodes with the same colour are set of genes, also known as topics.

Blue lines represent the cluster structure, each blue square is a set of nodes, lines delineate the hierarchical structure.

It is clear in the middle the network separation in genes and samples.

data G $P(\theta|G)$ can be written as a likelihood times a prior probability as

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

It is possible to define a description length

$$\Sigma = -\ln P(G|\theta) - \ln P(\theta),$$

so that $P(\theta|G) \propto \exp -\Sigma$. Moreover, the likelihood $P(G|\theta)$, can be written as $\frac{1}{\Omega}$ where Ω is the number of possible realisations given θ . This can be represented as a microcanonical ensemble with entropy $S = \ln(\Omega)$. Note that $\Sigma = S - \ln P(\theta)$. According to [30] entropy S can be written as

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

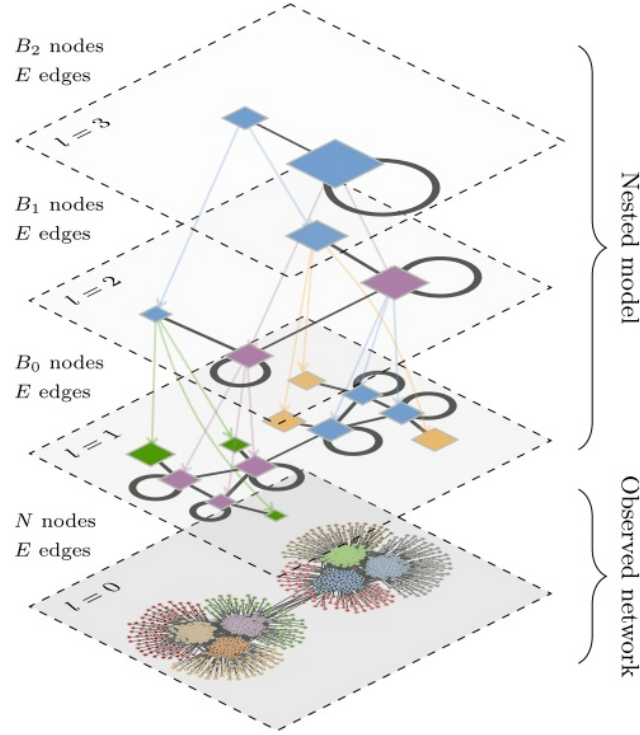


Figure 4.2: Example of a hierarchical structure.

where n_r is the number of nodes in block r , e_{rs} the number of links between nodes of group r and group s and H is the Shannon entropy $H(X) = x \log_2(x) + (1-x) \log_2(1-x)$. 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 to find groups with nodes very disconnected or topic and clusters with a lot of connections. The algorithm tries to minimise S , so that Σ is minimised, so $\exp -\Sigma$ is maximised, but this is $P(\theta|G)$ that is the required probability to maximise.

The MonteCarlo 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 $P(r \rightarrow s|t) = \frac{e_{ts} + \epsilon}{e_t + \epsilon B}$
- 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

in figure 4.3 an example of these steps. Once the model run it is possible to estimate

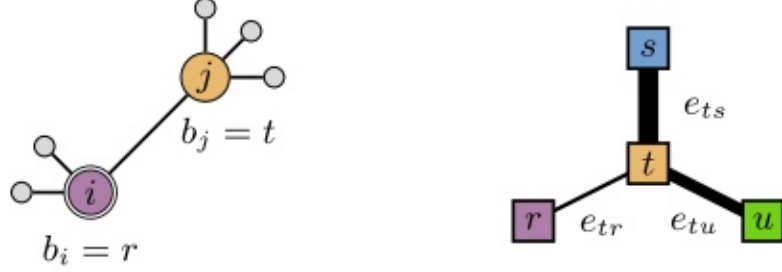


Figure 4.3: Left: Local neighbourhood of node i belonging to block r , and a randomly chosen neighbour j belonging to block t . Right: Block multi graph, indicating the number of edges between blocks, represented as the edge thickness. In this example, the attempted move $bi \rightarrow s$ is made with a larger probability than either $bi \rightarrow u$ or $bi \rightarrow r$ (no movement), since $e_{ts} > e_{tu}$ and $e_{ts} > e_{tr}$.

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

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

In the case of overlapping partitions, the presence of a word in a topic is not trivial and can be estimated as

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

or the presence of a document in a cluster

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

See appendix A for detailed analysis of the math behind the algorithm and <https://cloud.docker.com/repository/docker/fvalle01/hsbm> for the extension of [27] to non linguistics component systems datasets.

4.1 Metrics and benchmarks

Before running topic modelling, it is useful to define some metrics to test and benchmark the model. In particular the model search 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 necessary.

Looking at the samples side of the network, the outputs are sets of samples, the clusters. One can state the model works if all, or at least the majority, of samples in the same cluster share some label. Here the tissue is considered as the main label.

Note that this work's model is a non supervised one, but 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 some quantities, the homogeneity

$$h = 1 - \frac{H(C|K)}{H(C)} \quad (4.1)$$

defining the entropy

$$H(C|K) = \sum_{c \in \text{tissues}, k \in \text{clusters}} \frac{n_{ck}}{N} \text{Log} \left(\frac{n_{ck}}{n_k} \right) \quad (4.2)$$

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 complete homogeneous situation.

Another quantity can be defined and it is completeness:

$$c = 1 - \frac{H(K|C)}{H(K)}, \quad (4.3)$$

$H(K|C)$ is defined in the same way as 4.2. 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:

$$\text{V-measure} = 2 \frac{hc}{h+c}, \quad (4.4)$$

which is actually the normalized mutual information between $P(C)$ and $P(K)$ [31]. Please refer to appendix B for the detailed maths. In figure 4.4 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 increase.

In the next sections will be studied also the maximum fraction of label in the same cluster defined as

$$\max_{c \in k} \frac{n_{ck}}{n_k}.$$

Also the number of different labels in the same cluster will be studied.

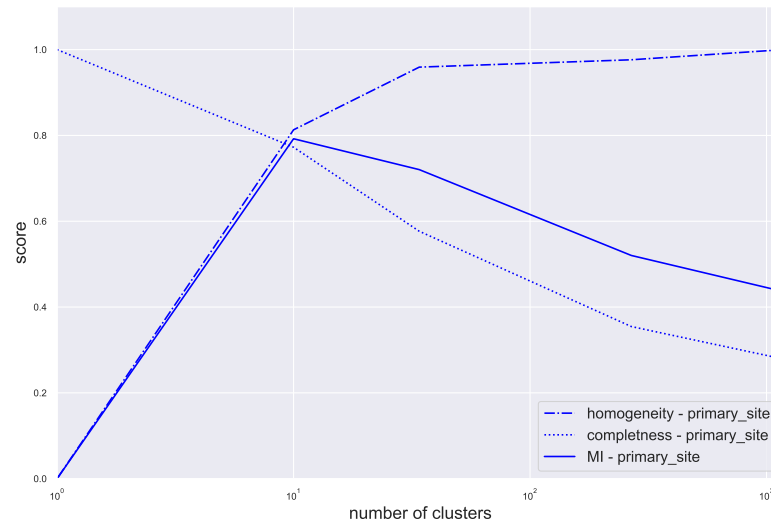


Figure 4.4: Score across hierarchy. The V-measure or normalized mutual information MI is the harmonic average between homogeneity and completeness.

4.1.1 Hierarchical clustering

Hierarchical clustering is a general family of clustering algorithms that builds 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 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 was used standard euclidean distance. 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 actually a variance-minimizing approach. Here the specific setting used in this work.

```

1 from sklearn.cluster import AgglomerativeClustering
2 AgglomerativeClustering(
3     affinity='euclidean',
4     compute_full_tree='auto',
5     linkage='ward',
6     n_clusters=x,
7 )

```

In figure 4.5 an example of a hierarchical clustering. Note that nodes with short distance are linked firstly.

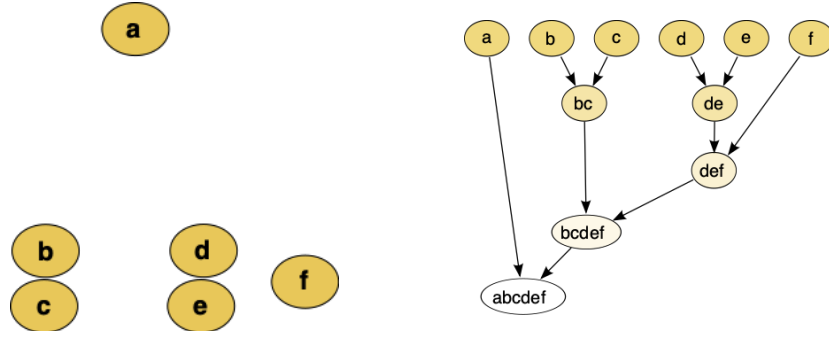


Figure 4.5: Example of hierarchical clustering

4.1.2 LDA

Latent Dirichlet Allocation is the standard and well-known approach to topic models. It has got more restrictive priors and needs some parameters to be set than hierarchical stochastic block model. It uses some different methods to maximize the posterior probability to observe some latent variables given the data. As well described in [10] LDA is a generative model and can be summarised as follows:

- set number of topic K and parameters η and α
- 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 $w \sim \text{Multinomial}(\bullet|\beta_k)$

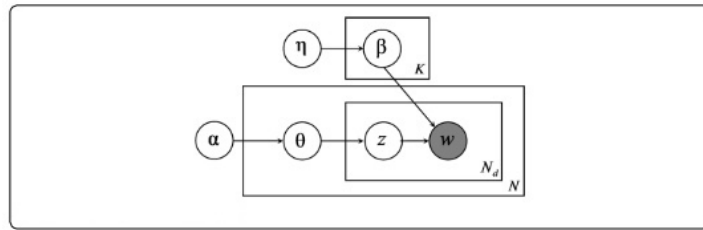


Figure 4.6: LAD scheme

this process is summarized in figure 4.6. These steps are useful to maximize the posterior probability

$$P(w, z, \beta, \theta | \alpha, \eta) = \prod_{d=1}^N P(\theta_d | \alpha) \prod_{n=1}^{N_d} P(w_{dn} | z_{dn}, \beta_k) P(z_{dn} | \theta_d) \prod_{k=1}^K P(\beta_k | \eta) \quad (4.5)$$

where

-
- N is the number of documents
 - K is the number of topics as set by the user
 - w are words
 - N_d is the number of words in document d
 - α and η are parameters of the model (usually $\eta = 0.01$ and $\alpha = 50/K$)
 - $P(\theta|\alpha)$ and $P(\beta|\eta)$ are Dirichlet distributions

When β , θ and z are estimated, the outputs are the topic distribution in documents $P(z|d)$ and the word distribution in topics $P(w|z)$.

4.2 Pre-process

This work aim to use the hierarchical Stochastic Block Model (hSBM) described before. Data considered in this work have ~ 11000 samples as documents and ~ 60000 genes as words (~ 20000 if considering only protein coding genes). In comparison in the original paper [27] 63 Wikipedia articles were considered 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-processing the data, all of them involving the quantities defined in 2. The goal is to identify components which are able to best separate the realizations or isolate the most interesting genes.

Low occurrence genes were selected firstly to approach topic modelling. A 0.5 threshold was set on occurrence. This method selects genes that appears (have expression greater than zero) only in less than half samples. This approach has some limitations, for instance it doesn't consider genes that appear everywhere (with occurrence $\simeq 1$) but changes their behaviour across realizations.

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

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

which increases the importance of components with small occurrence o_i . This approach doesn't actually select components, which is still an issue.

Highly variable genes can be selected. This is done using the CV^2 analysis done in chapter 3. Plotting the coefficient of variation versus the mean for each component reveals which components have higher variance with respect to components which, on average, have a similar behaviour. Binned averages and variances were estimated, and only genes with a CV^2 over a σ greater than the bin's mean were considered. This method seems to select useful genes even if the binned average bound is quite noisy.

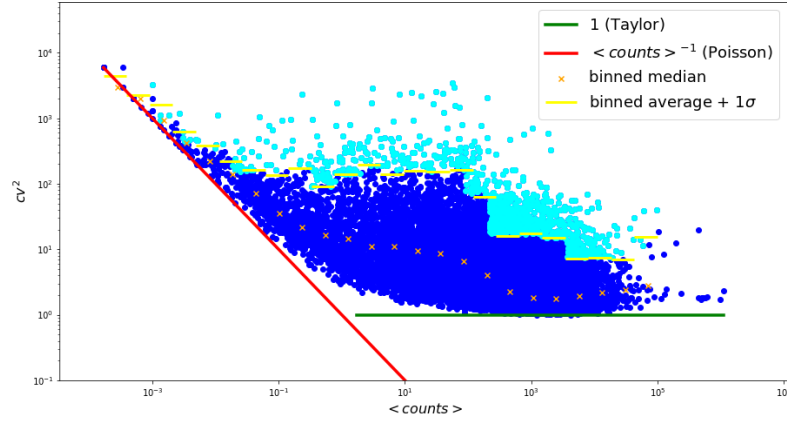


Figure 4.7: Highly variable genes. In cyan genes that are 1σ over the average of their bin.

Distance from boundaries can be a similar and alternative method to select highly variable genes. In this case the bound is smooth and well-defined. The distribution as

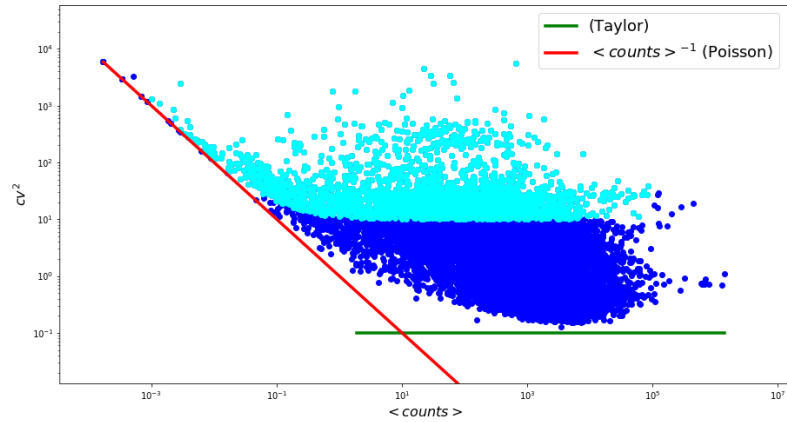


Figure 4.8: In cyan genes which distance from boundaries is greater than $10CV^2$.

discussed in 3 have a Poisson-like and a Taylor-like boundaries. So can be considered only components that are the most distant from these boundaries. Moreover, this boundaries can be found with a simple null model, as shown in figure 3.6 the sampling model defines the lower bound of the data.

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

4.3 Run

4.3.1 Run on Gene Tissues Expression dataset

Once the model was tuned and adapted to RNA-Sequencing data, it was run on a subset of the GTEx dataset. A subset was chosen randomly in order 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 of the oncoming result the metric above described were considered. In figure 4.9 it is represented the V-measure score versus the number of clusters found at different layers. The result is quite good,

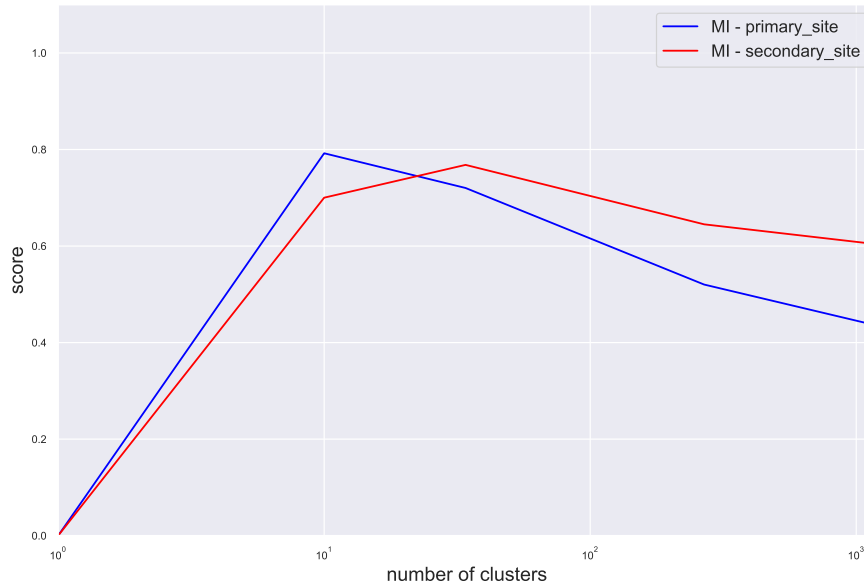


Figure 4.9: Scores across the hierarchy. The primary site and secondary site labels are compared.

the maximum score is over 0.8. Considering that, for example, [9] 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 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.

In figure 4.10 the relation between the clusters at different layers it is evident.

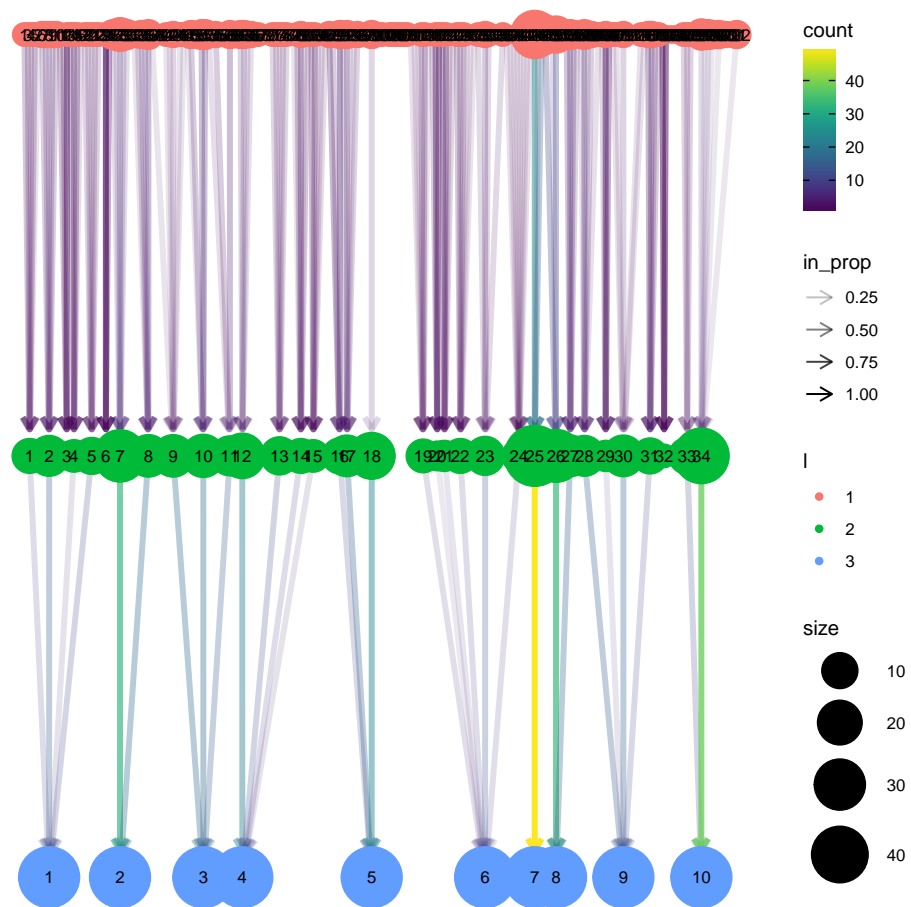


Figure 4.10: Hierarchy of the files' nodes.

In figure 4.11 each column is a cluster and each colour is a tissue of the dataset. It is evident that the majority of the tissue are identified: the first, second, fifth, sixth, eighth and tenth columns are fully and uniformly coloured of the same colour. These correspond to an identification of brain, blood, lung, testis and bladder. A

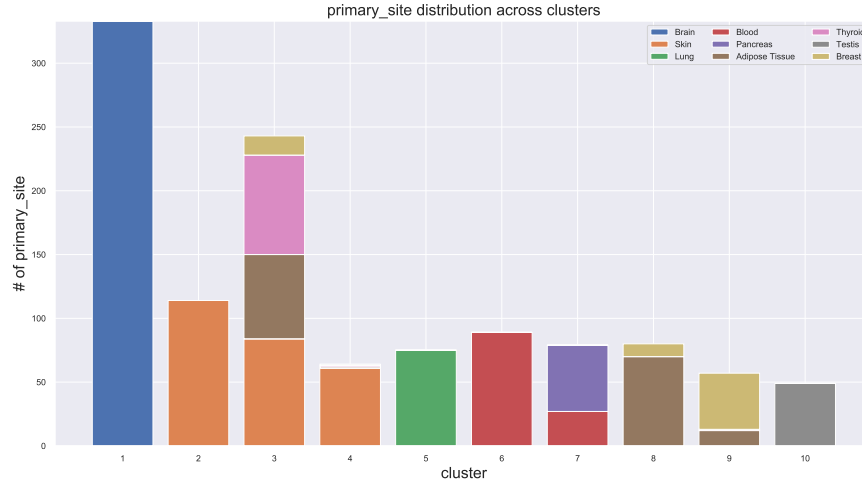


Figure 4.11: Clusters composition at the level of the hierarchy with higher score. Each column is a cluster, each colour is a label.

normalised representation of the same clusters the result is still quite interesting and the homogeneity of the clusters is more evident. Going deeper in the hierarchy and

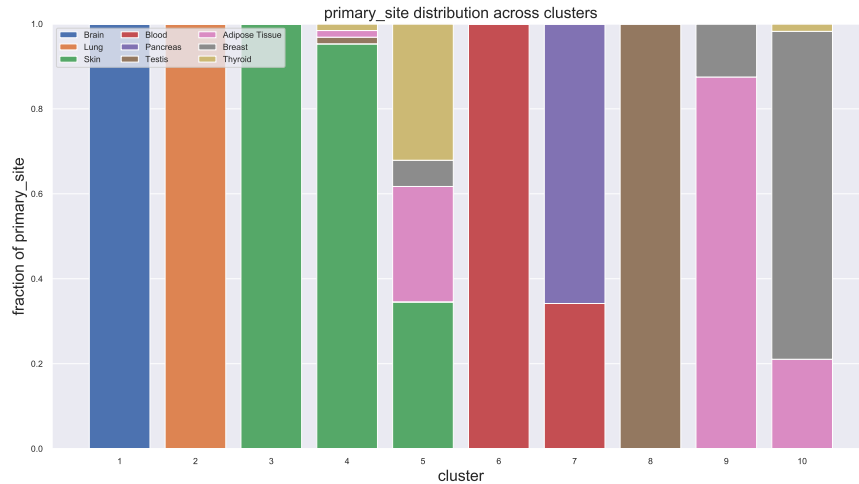


Figure 4.12: Normalised composition of clusters.

looking at a layer with more cluster the result, shown in figure 4.13, demonstrates that at this point all the tissues are separated and each cluster is full of nodes sharing the

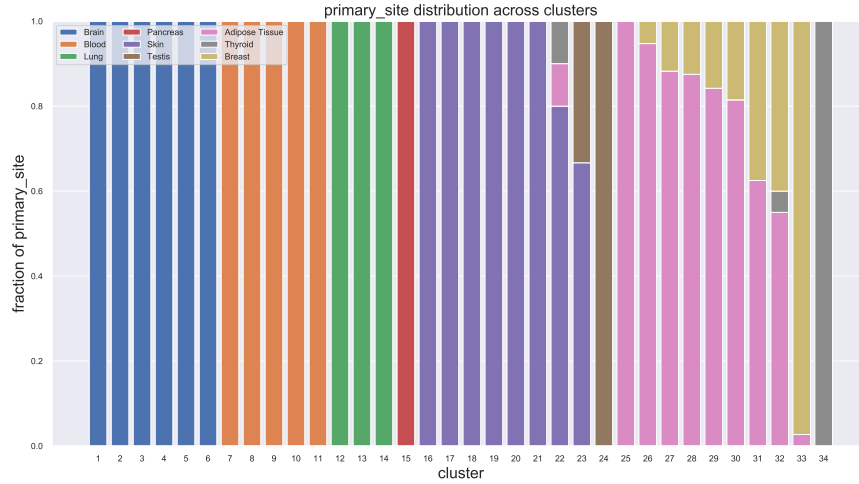


Figure 4.13: Normalised composition of clusters at a deeper level.

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 13) and blood is distinguished in whole blood (columns 1-4) and lymphocytes (column 10).

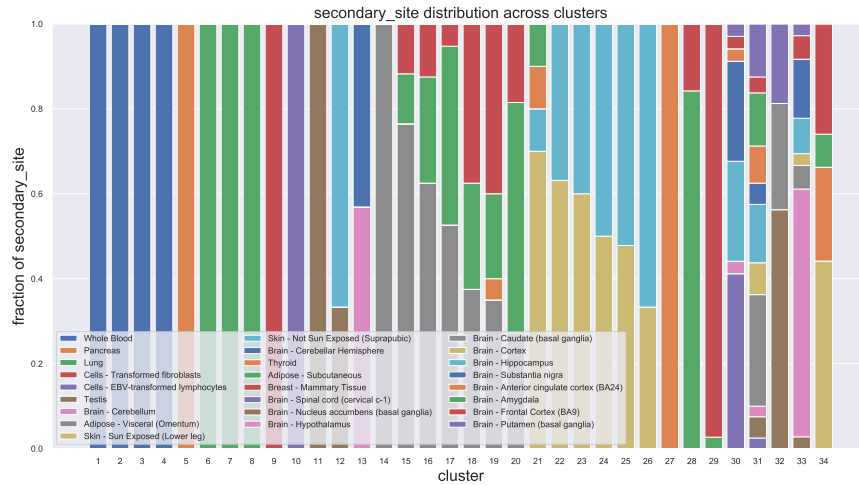


Figure 4.14: Normalised composition of clusters with respect to the secondary site sub-tissue labels.

4.3.2 Shuffling

A null model of cluster composition is necessary In order to 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. Doing so the number of clusters and the cluster sizes are maintained. In figure 4.15 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, blood is more represented than other tissues.

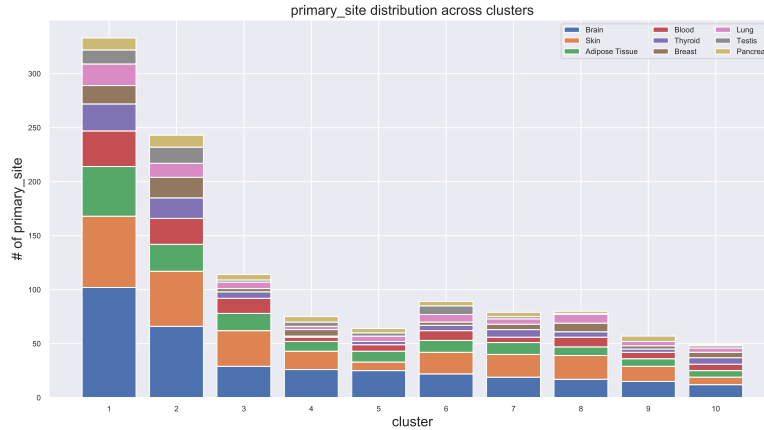


Figure 4.15: Example of visualization of clusters with reshuffled labels.

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

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

with n_{lk} is the numbers of nodes labelled l in cluster k and n_k is the number of nodes in cluster k . This is represented in figure 4.16 for the level where the V-measure is maximized (best results are expected here). In figure 4.16 on the left is shown the most representative label fraction versus for each cluster, on the right the histograms of the same quantity. It is evident that models' clusters are very homogeneous with the majority of cluster with almost 100% of the same tissue. It is also clear that reshuffling the labels the result is very different and so the models behave better than expected. In figure 4.17 the same analysis is done for every level of the hierarchy. It is interesting to notice that at deeper levels (upper left in the figure) the random reshuffling and the real labels have the same behaviour. This is due to the fact that at this level clusters are very small and so it is easier to pick up nodes with the same level (in the extreme case of a cluster with size 1 it is always full with the same label). This shows that the deeper level it is not interesting, results are the same with random labels; moreover the reshuffling null model it is good to show up eventual biases due to small cluster sizes.

A similar analysis can be made considering not just the number of the cluster but the cluster size, this is shown in figure 4.18. It is interesting to notice that the shuffle null model and the real labels clusters are evidently different, so there must be some kind of signal. It is clear that the model is able to output big clusters full of the same

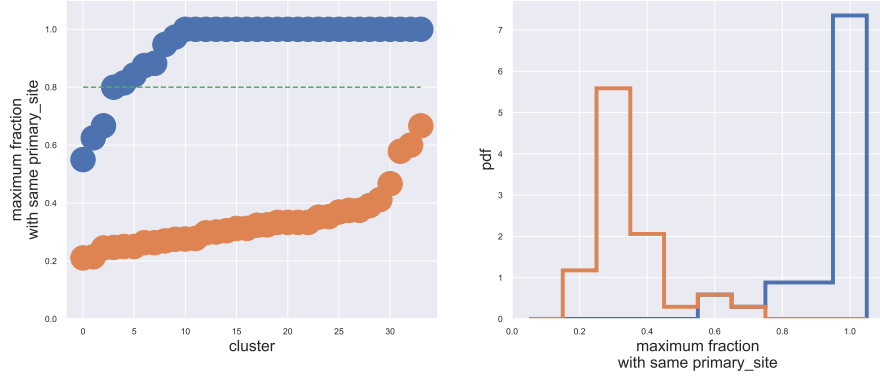


Figure 4.16: Most representative label versus cluster size.

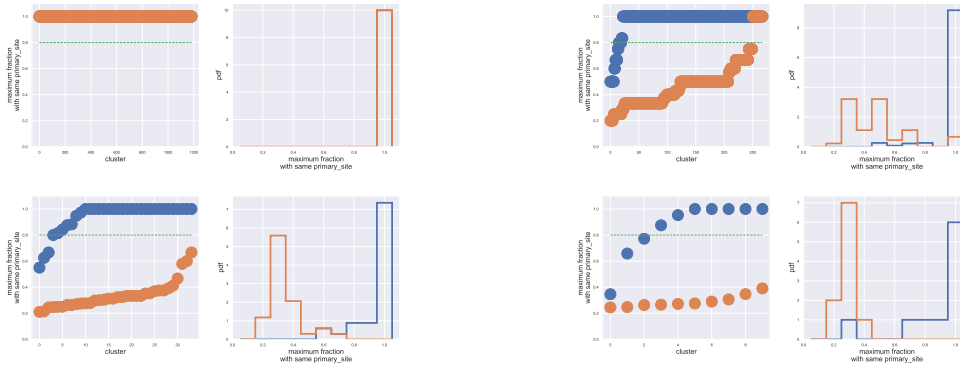


Figure 4.17: The fraction of the most representative label in all clusters for different levels of the hierarchy. From upper left the deeper layer than down right the upper one.

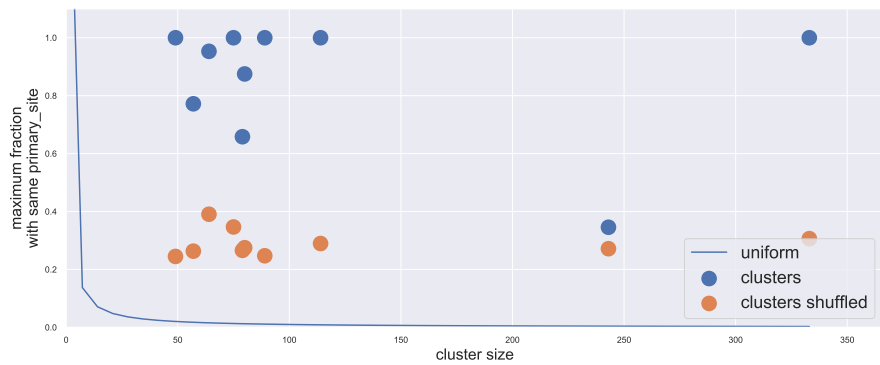


Figure 4.18: Fraction of most representative label versus cluster size.

label. In figure 4.19 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 deeper layer (upper left in the image) is not different from null model and so it is

not interesting.

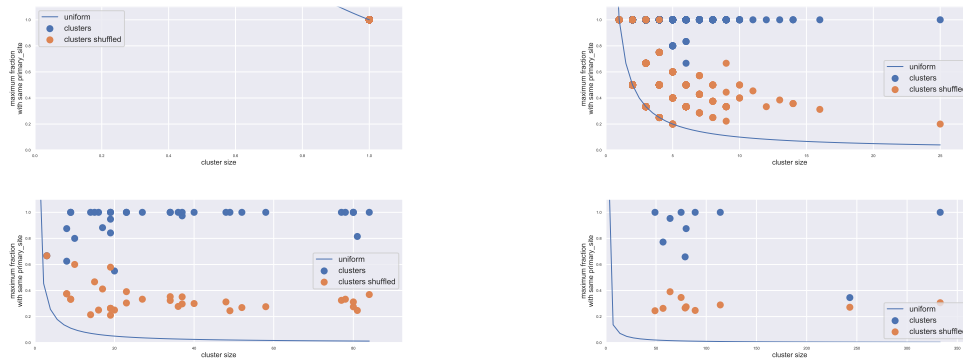


Figure 4.19: Fraction of most representative label versus cluster size across the hierarchy. From upper left the deeper layer than downright the upper one.

At this point to deepen investigate the structure of the clusters it can be interesting to study how many labels are present in each cluster. In fact, the fraction of 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 sort of effects. In figure 4.20 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 4.21 the same analysis for all the layer of the hierarchy. Even here the deeper 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.

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 4.22. Moreover remembering the V-measure or normalized mutual information defined in 4.4 it is possible to estimate a mixed score which considers the homogeneity of primary site and the completeness of 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 with the same label into two small ones) but gain information at the next magnification. 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 4.20: The number of different labels in each cluster versus cluster size.

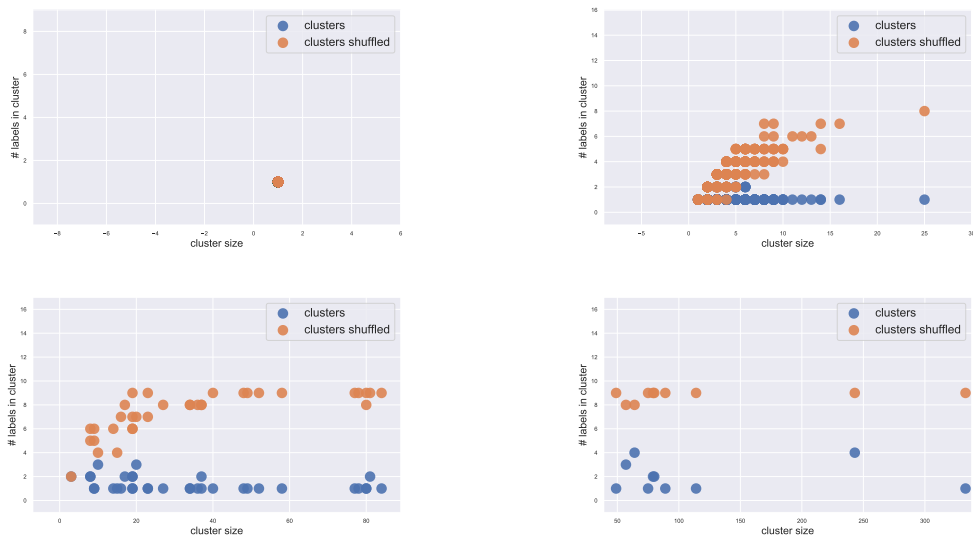


Figure 4.21: Number of different labels in each cluster versus cluster size. From upper left the deeper layer than downright the upper one.

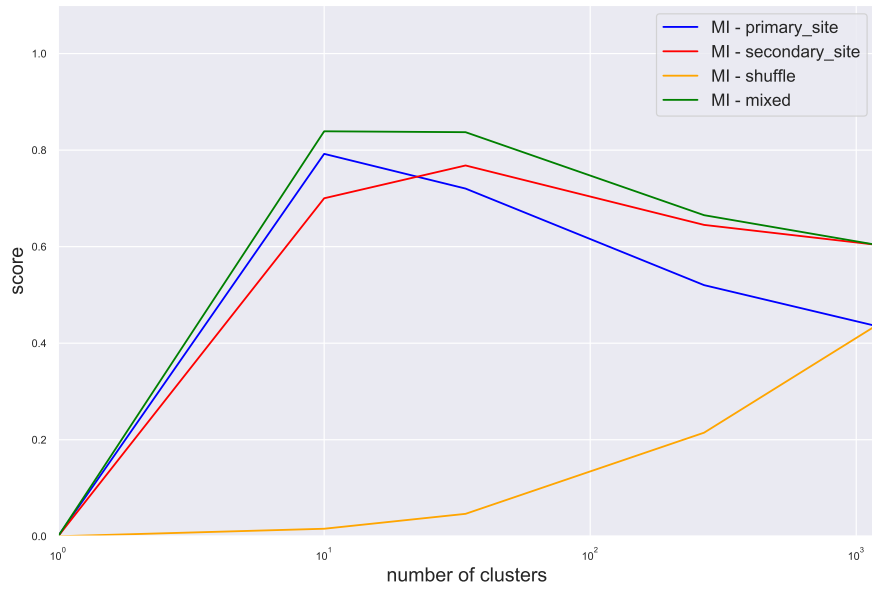


Figure 4.22: Scores across the hierarchy. The scored is compared with some random labels. In blue the score for the primary site labels, in red for the secondary site labels, in yellow the shuffled labels, in green the mixed score with primary homogeneity and secondary completeness.

4.3.3 Standard algorithms

At this point 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-known similar algorithms. First of all, a comparison is made with hierarchical clustering. This is done using the standard scipy [3] package, the metrics used was the euclidean one and the linkage method was set to Ward. This is quite fast, it needs a couple of minutes on a dual core, 8GB memory machine. In figure 4.23 the comparison between this scores, the hierarchical algorithm performs worse than hierarchical stochastic block model and as highly expected better than the random model.

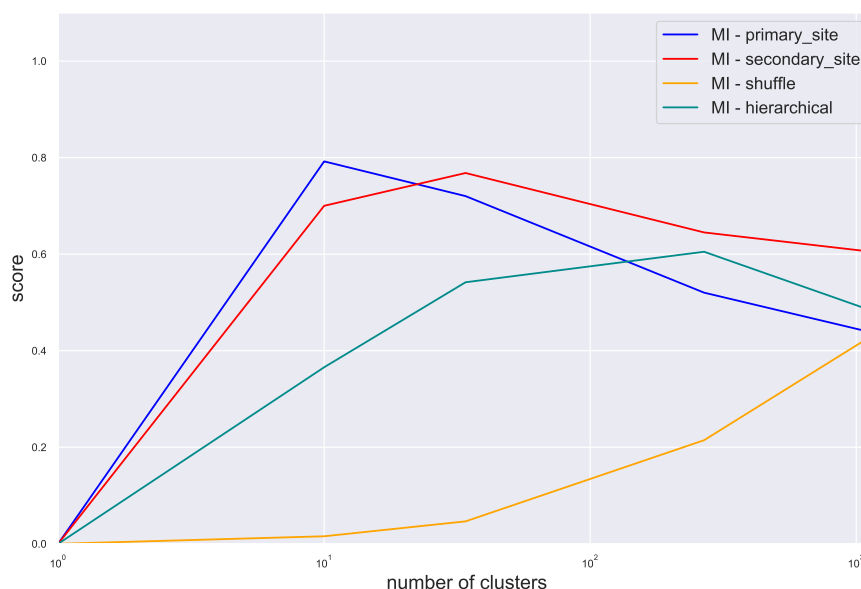


Figure 4.23: Scores across hierarchy. The scored is compared with some random labels

Another very used and well-studied algorithm is Latent Dirichlet Allocation briefly described in 4.1.2. Running LDA in 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 4.24 are reported the V-measure score for all the algorithms described until this point. What is clear is that the hierarchical Stochastic Block Model performs better than all the others, LDA gain a little worst score and hierarchical clustering is the worst of the three. As highly expected all models are quite different than random. 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 cluster models were not fine tuned and default parameters were used. Maybe a fine tuning of this packages can lead to better and more satisfying results. This analysis,

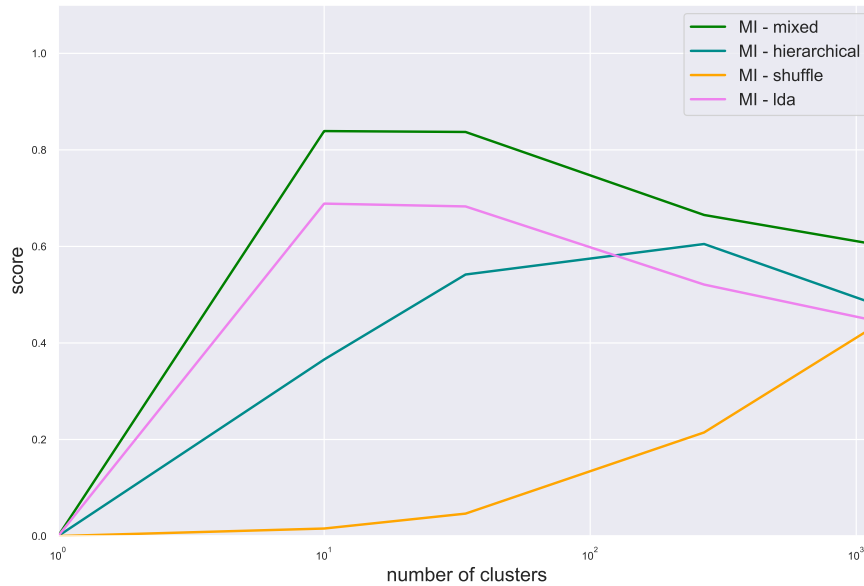


Figure 4.24: Scores across hierarchy. The scored is compared with some random labels

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 parameter were set. The fact reveals another good point of hSBM, it extracts not only better clusters, but also the parameters necessary to this kind of model.

4.3.4 Topics

The analysis up to this point considered only one of the two sides of the bipartite network, in fact nothing was told about the genes (words in topic models). Not less the model output some clusters of genes. From now on I'll refer to these clusters of genes as topics.

If one has got a set of genes in order to catch any important information and discover if there is any biological meaning behind it, it is possible to perform an enrichment test. 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 [32] python tool [33], which performs a Fisher exact test (hypergeometric test), the Benjamini-Hochberg corrected P-values is reported. Genes' annotation terms were searched in the following sets:

- GO Molecular Function 2018
- GO Biological Process 2018
- GO Cellular Component 2018
- Human Phenotype Ontology

- Tissue Protein Expression from Human Proteome Map
- KEGG 2019 Human
- NCI-60 Cancer Cell Lines
- GTEx Tissue Sample Gene Expression Profiles up
- GTEx Tissue Sample Gene Expression Profiles down,

in particular the two latter contains annotation specific for GTEx dataset [34].

In tables 4.1, 4.2 and 4.3 are reported examples of enrichment test results for topics output by hSBM. On the results is put a P-value cut at 0.05 and terms are sorted by adjusted P-value. Test were made on the topics at the level of the hierarchy which

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 4.1: 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 4.2: Enrichment test of a topic. It is clear the enrichment for brain related gene sets.

obtained the higher V-measure score on the sample side clustering. These results are

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

very interesting, these enrichment tests demonstrate that not only the sample side of the network is well clustered but only the topics have a non trivial meaning.

So also the topics are related to the tissues and somehow are tissue-specific. In the next examples the relation between the topics and the samples will be further investigated. In particular following what was done by [15] the importance of each topic inside each sample the $P(\text{topic}|\text{sample})$.

Separate healthy tissues is a good exercise and a good benchmark for models, but the real goals would be to be able to classify diseased samples. In fact 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 diseased samples.

4.3.5 Run on The Cancer Genomics Atlas

The exact same pipeline described so far can be applied at other dataset. 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 highly interesting and useful for the scientific community [9].

First of all let's take a look at the V-measure scores. As shown in figure 4.25 the score is almost 0.7, which is good, but worse than the healthy GTEx scenario. In this dataset there is no a sub-tissue label as before, but a disease type cancer information is available. The disease type separation is not so good; the fact that there is no evident difference between Zipf's laws when separating data by disease type 2.17 means that all genes contribute to define this specific label so the pre-process filter is not a good option in this case. 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.

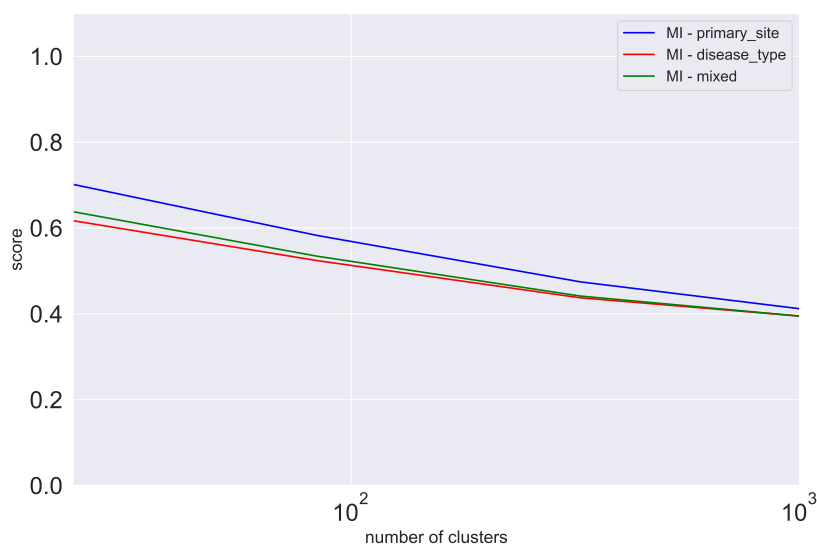


Figure 4.25: Score across the hierarchy for TCGA. In blue the primary site labels were considered, in red the disease types and in green the mix of the two.

Looking directly into the cluster composition the tissue separation is quite good and visually appreciable. In figure 4.26 clusters at the higher level of the hierarchy. Some tissues are well separated at this point, at the same time when possible samples are grouped by system, digestive system is the more evident. Going deeper in the hierarchy the tissue separation becomes visually appreciable and all the clusters are almost tissue-specific.

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

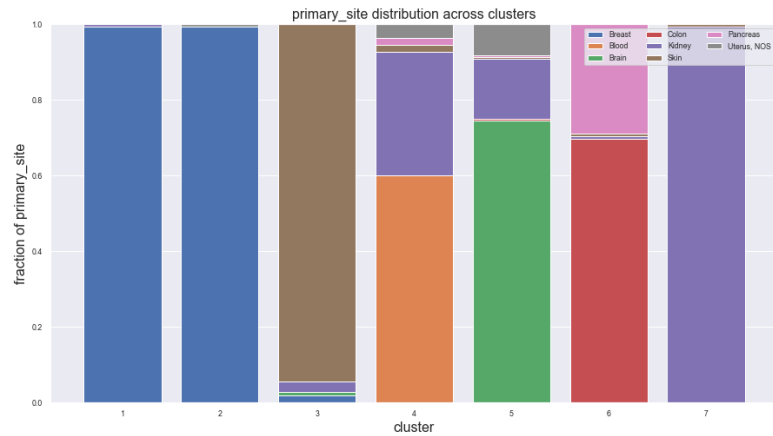


Figure 4.26: Clusters of diseased tissues at the higher level of the hierarchy. Breast is well separated, such as skin and brain. Cluster 6 contains digestive systems samples from pancreas and colon.

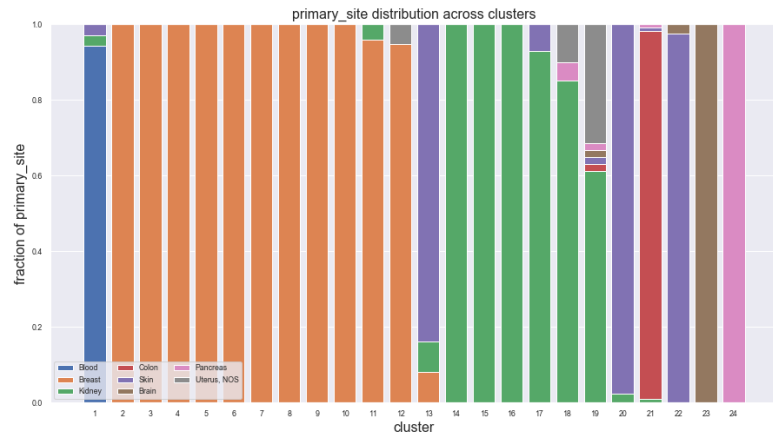


Figure 4.27: text

how the model behave when healthy and cancer samples are merged.

4.3.6 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 were still analysed data from GTEx and TCGA, but from a particular dataset available from [17] where authors tried to unify the normalization process from different dataset and sources [16]. These are, 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 4.28 the result for the primary site is quite satisfying: clusters are very homogeneous and V-measure's peak is near 0.8.

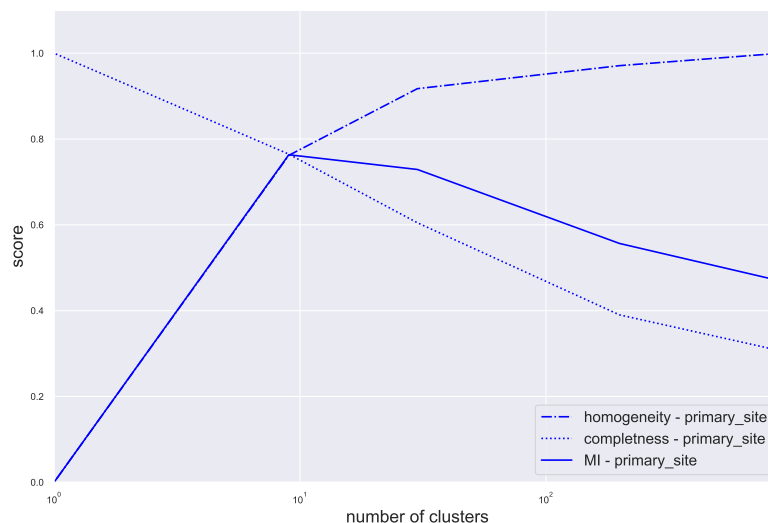


Figure 4.28: V-measure score for run with merged healthy and diseased samples. Homogeneity, completeness and mutual information are represented.

Estimating the score also for the secondary site, or rather for the tissues with the health state and for just the healthy/disease label lead to figure 4.29. This result is quite interesting, first of all even the secondary label is well classified and this happens at a deeper level 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 is an evidence that the model actually recognize tissues never mind where they come from, moreover the difference between datasets are not important here and so the normalization made by [16] 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 is

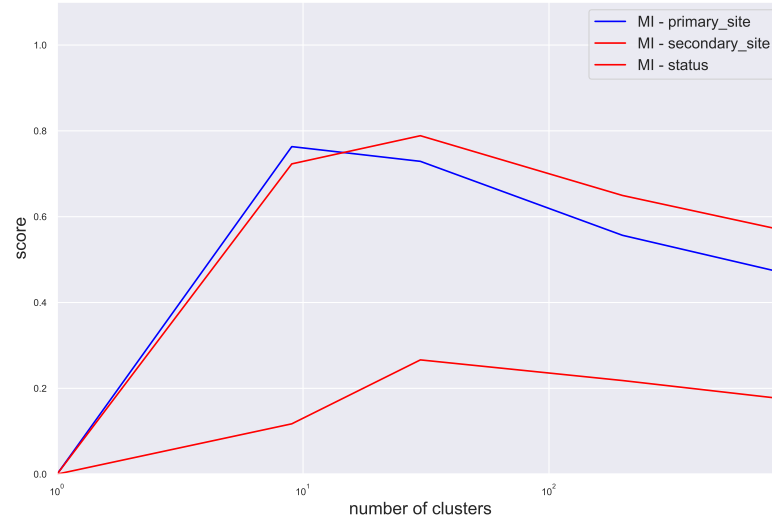


Figure 4.29: V-measure score for run with merged healthy and diseased samples. Primary site (brain, blood, pancreas..) labels are compared with secondary labels (healthy brain, brain cancer, healthy blood, blood cancer, healthy pancreas, pancreas cancer..). The health status label (healthy / diseased) is plotted.

considered (the homogeneity of primary site is considered with the completeness of secondary label) so that the score increase if going deeper in the hierarchy the separation of a homogeneous cluster brings to separation of the refined labels. In figure 4.30 this score is compared with the one 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 have a quite bad score and all are better than shuffling null model.

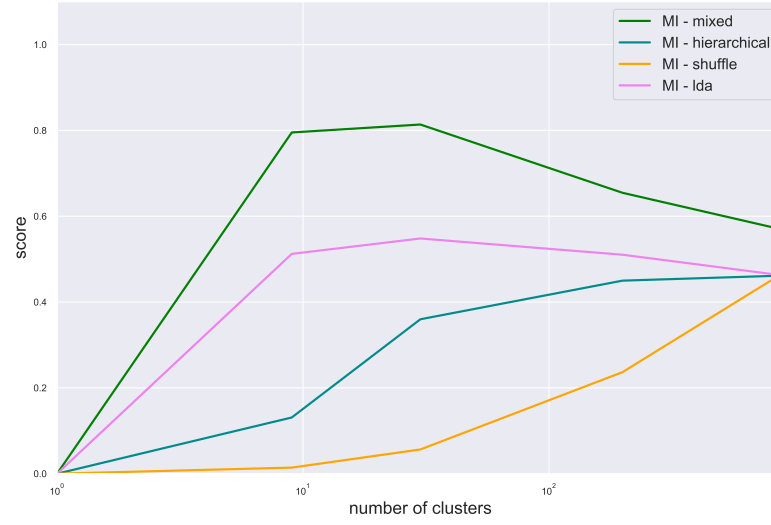


Figure 4.30: V-measure score for run with merged healthy and diseased samples. LDA, hierarchical clustering and null model for comparison.

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 $-\log_{10}(\text{P-value})$ 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 genes at random. In figure 4.32 are shown

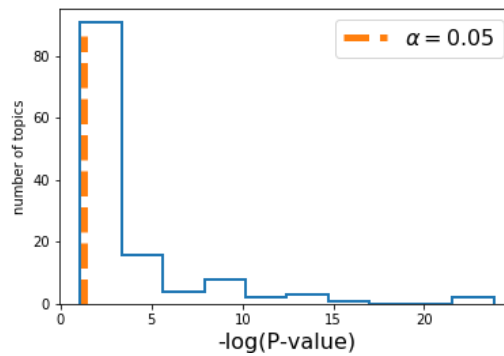


Figure 4.31: $-\log_{10}(\text{P-value})$ of the term with the lowest P-value in each topic. In orange the classical 0.05 threshold.

the categories of the terms with lower P-values. This explains what aspect of the samples topic describes. The majority of terms found in topics comes from the GTEx annotation for tissue expression, many are from GO biological process, GO molecular function and some from Human phenotype ontology.

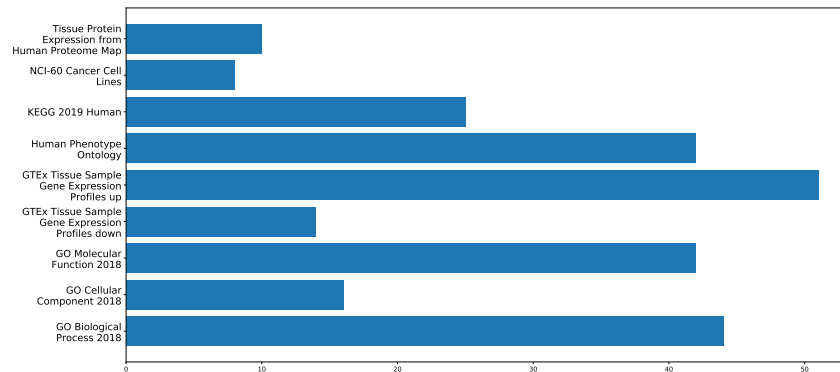


Figure 4.32: Categories of the terms with lower P-values in each topic.

Going forward in the analysis it is possible to perform enrichment test with other tools such as DAVID [35, 36]. Results are similar to the ones retrieved before. Tissues related terms are found also using this tool and this confirms the absence of tool or categories related biases. In figures 4.33, 4.34 and 4.35 the result from DAVID enrichment analysis. Finally it is interesting to notice that topics are quite small (order ≈ 20 genes), so there are no biases that can appear doing enrichment tests on big sets.

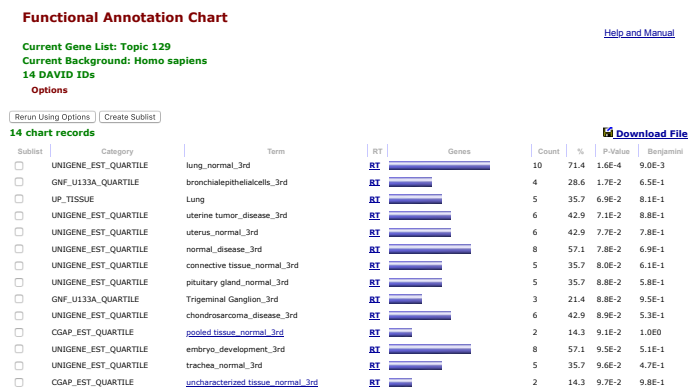


Figure 4.33: Caption

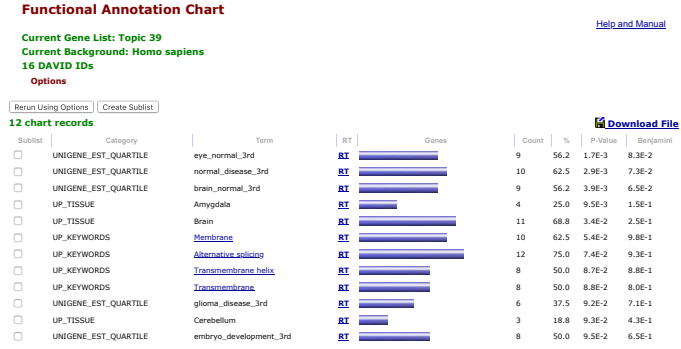


Figure 4.34: Caption

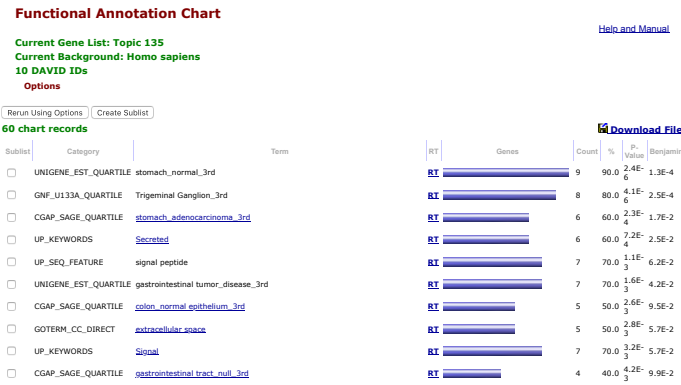


Figure 4.35: Caption

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 $P(\text{topic}|\text{tissue}) = \frac{1}{|\text{tissue}|} \sum_{\text{sample} \in \text{tissue}} P(\text{topic}|\text{sample})$.

In figure 4.36 it is plotted $P(\text{topic}|\text{tissue})$ for the first topics. What is clear is that in all samples there is a global trend that present no much differences between tissues, the topic expression differences between tissues are slightly appreciable at this point. This carries a profound and very informative message: under this new point of view 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 suggest 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 genes. 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

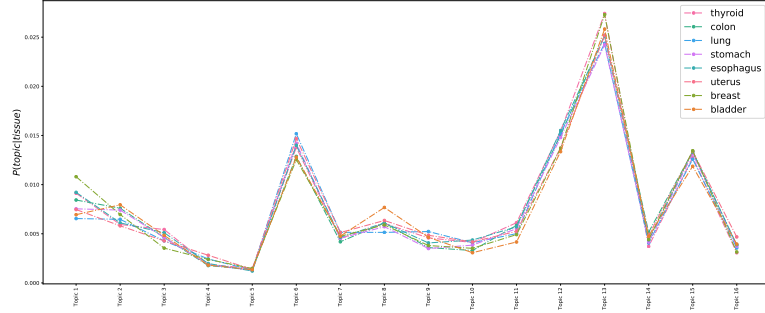


Figure 4.36: $P(\text{topic}|\text{tissue})$ for some topic colored by tissue. It is clear a global trend and in some topics there are little differences between tissues.

are most differently expressed than average. To do so from each $P(\text{topic}|\text{tissue})$ was subtracted the average topic expression $\text{mean}_{\text{tissue}}(\text{topic}) = \langle P(\text{topic}|\text{tissue}) \rangle_{\text{tissue}}$ and the result was divided by the standard deviation $\sigma_{\text{tissue}}(\text{topic})$. In figure 4.37 some most characteristic topics are reported. This analysis reveals that tissue have a different behaviour than average in different topics, different tissues are distant from the average in different topics. Moreover, the healthy and the disease sample from the two different sources seems to have a similar behaviour, in fact if a healthy tissue is far from the average in a topic, the diseased counterpart is also distant from the average as evident in the topics reported in the figure.

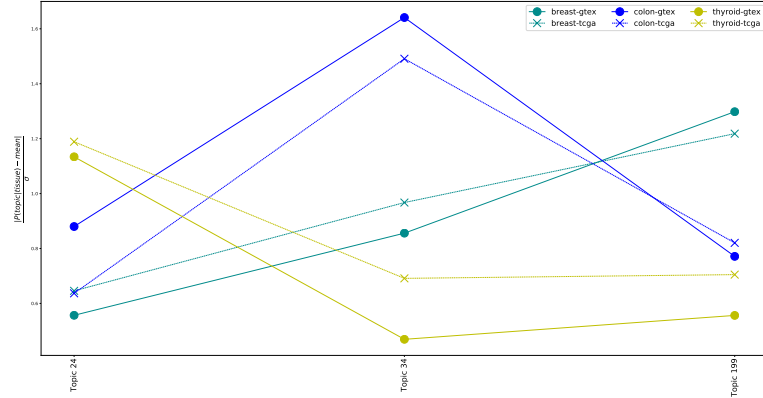


Figure 4.37: $\frac{|P(\text{topic}|\text{tissue}) - \langle P(\text{topic}|\text{tissue}) \rangle_{\text{tissue}}|}{\sigma_{\text{tissue}}(\text{topic})}$ or the distance of each tissue from the average tissue expression in each topic. Some interesting topics are reported.

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

4.4 Results

The analysis using topic modelling leads to many interesting results.

The first result achieved is the development of a model that is able to 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 the entropy 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 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 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 more standard approaches such as Latent Dirichlet Allocation and hierarchical clustering. hSBM outputs better results with respect to standard approaches, moreover, it gains higher scores. An interesting fact is that topic modelling (both LDA and hSBM) 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 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 a LDA approach where genes can be either present or absent in a sample. 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 [17] 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 is no, evident, bias between datasets.

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

Chapter 5

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 *Us* 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 many different Zipf's law if one considers each tissue independently. Very similar results were obtained considering TCGA, a dataset containing thousands of samples of cancer tissues.

The power law distributions and the similarities with what was found in linguistics 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 both to find clusters of samples that share some properties and to find the relation between genes and samples.

Many goals were achieved during the topic modelling analysis. The pipeline begins filtering the data and selecting useful 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 property, in particular, the tissue they are related; enrichment tests found tissue related terms in the topics along with Gene Ontology terms. The relation 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 model reveals itself as a useful approach to find the hidden structure of gene expression data. The prior analysis to select highly variable genes makes it possible to run the algorithm faster without losing necessary information.

During this work the foundations have been laid for other analysis; for example, it should be interesting to study the variability of gene expression between tissues and between individual, 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 [37] 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. 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 [38] could be an interesting starting point. Maybe it worth to run the model on 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 to a specific cancer tissue and find cancer subtypes, for instance, the Breast or Colon one. Obtaining a hierarchy that at some level is able to identify cancer sub-types would be an ideal and great goal.

Appendix A

Hierarchical stochastic block model

The algorithm is called hierarchic Stochastic Block Model.

The first step of hierarchical stochastic block model, as discussed in [39], is to create a bipartite network G with two kind 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 is ignored. The aim is to find a partition $b \in \{b_i\}$ with $B = |\{b_i\}|$ blocks.

These kind of models are 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 θ any additional parameter of the model.

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

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

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

$$\Sigma = -\ln P(G|b, \theta) - \ln P(b, \theta) \quad (\text{A.2})$$

the A.1 can be written as $\frac{e^{-\Sigma}}{P(G)}$, so maximising that is equivalent to minimise the description length 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 graph that is possible to generate with audience matrix A and n_r the counts of block partition $\{b_i\}$

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.

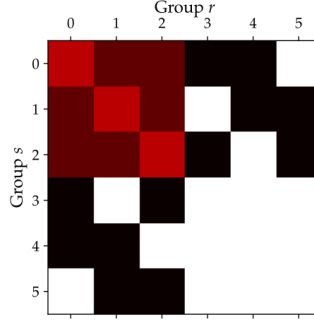


Figure A.1: Example of a edge's matrix from [8]

It is useful to define a traditional entropy:

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

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 $H(x) = -x \ln(x) - (1-x) \ln(1-x)$. This entropy is equivalent to the micro-canonical entropy of a system with $\Omega(A, \{n_r\})$ accessible states $S_t = L \ln \Omega$.

The algorithm uses a Markov Chain Monte Carlo to minimise this entropy. At each step a node changes block and the new configuration is accepted if S is decreased.

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

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

How to change group of a node? At each step according to [39] node i can change group from r to s with a probability

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

where j is a random neighbour of i : $j \in N_i$, $t \in \{b_j\}$ its block as defined in [39]. ϵ is a parameter that according to [30] has no significant impact in the algorithm, provided it is sufficiently small.

A.5 can be rewritten as

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

defining $R_t = \frac{\epsilon B}{e_t + \epsilon B}$

This is done in 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

This steps mime probability A.5; note that for $\epsilon \rightarrow \infty$ this gives a uniform probability.

To enchant the probability to go into a minimum, a bounce of these moves is made, only the set of moves with the minimum S is accepted.

How many blocks B ? Note that the number of blocks B is a free parameter and must be inferred as described in [30]. This implies a slight modification of the algorithm such that it became possible to admit that a new group is created. When a group s is chosen, the algorithm can now accept a **new group** and A.5 became

$$P(r \rightarrow s) = \sum_t P(t|i) \frac{e_{ts} + \epsilon}{e_t + \epsilon(B + 1)} \quad (\text{A.6})$$

being $P(t|i) = \sum_j \frac{A_{ij} \delta(b_j, t)}{k_i}$ 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 hierarchic layers? After the algorithm is run, one may would to 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 A.1 is created, where edges are considered between groups of the previous layer.

The posterior probability became

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

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

Adding a layer is done in 3 steps described in [40]:

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 $l = L$ [40]. 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 succeeds in decreasing the description length Σ 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 $l - 1$. If the lowest level $l = 0$ is reached and cannot be improved, the algorithm ends.

Overlapping partitions As described in [41] 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 audience matrix A and a partition $\{\vec{b}_i\}$ is

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

if Ω is the number of possible graphs. Entropy A.3 is $S_t = Ln\Omega$. This corresponds to an augmented graph generated via a non overlapping block model with $N' = \sum_r n_r > N$ nodes and the same audience 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 . $\binom{B}{N}$ 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

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

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

$$P(\{\vec{b}_i\}_d | \{n_{\vec{b}}\}_d) = \frac{\prod_{|\vec{b}_i|=d} n_{\vec{b}}!}{n_d!} \quad (\text{A.9})$$

the global posterior as defined in [41] is

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

At this time it is necessary to obtain the distribution of the edges between mixtures. Defined $e_r = \sum_s e_{rs}$ the number of half-edges labelled r , $m_r = \sum_{\vec{b}} b_r$ 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 labeled degree sequence $\{\vec{k}_i\}$ from

$$P(\{\vec{k}_i\}_{\vec{b}} | \{e_{\vec{b}}\}, \{\vec{b}_i\}) = \frac{\prod_k n_{\vec{b}_k}^{\vec{b}_k!}}{n_{\vec{b}}!}$$

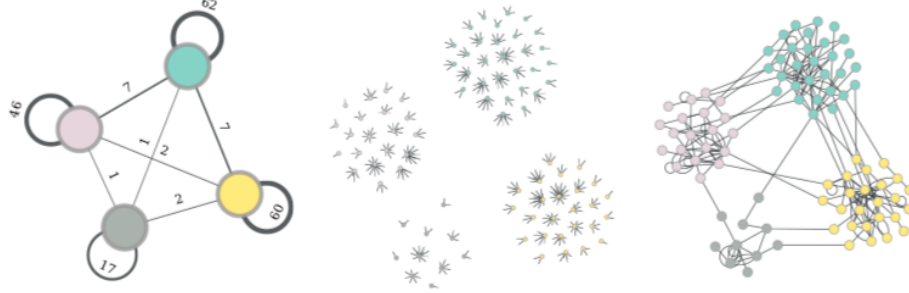


Figure A.2: Illustration of the generative process of the microcanonical SBM. Given a partition of the nodes, the edge counts between groups are sampled (left), followed by the degrees of the nodes (center) and finally the network itself (right). From [28].

Word documents separation Following what is done in [27], 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.

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

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 [39] 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.

The homogeneity

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

defining the entropy

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

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:

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

$H(K|C)$ 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 [31], which is the harmonic average of the two:

$$2 \frac{hc}{h+c}. \quad (\text{B.4})$$

The product hc is equal to

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

the sum $h+c$ is

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

Expressing the conditional entropy

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

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

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

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

2.1	Structure of a matrix representing component systems with $i = 0 \dots N$ rows and $j = 0 \dots R$ columns.	6
2.2	Zipf's law from FPKM normalised data. On the right considering only protein-coding genes.	8
2.3	Zipf's law of protein-coding genes considering counts.	8
2.4	Zipf's law from GTEx count data. On the left all genes considered, on the right only protein coding ones.	9
2.5	The histogram of the occurrences O_i . Coding and non-coding entries are coloured.	9
2.6	Same behaviour is observed looking at one tissue a time.	10
2.7	Random sampling of components to build a realization of size M	11
2.8	Zipf's law sampled; TCGA(left) and GTEx (right). Sampling is reported for comparison.	11
2.9	Distribution of sizes M ; TCGA(left) and GTEx (right).Sampling is reported for comparison.	11
2.10	Occurrence distributions; TCGA(left) and GTEx (right). Sampling is reported for comparison.	12
2.11	Heaps' law; TCGA(left) and GTEx (right). Sampling is reported for comparison.	12
2.12	Number of genes expressed in a sample; TCGA(left) and GTEx (right). Sampling is reported for comparison.	13
2.13	The integral of the sorted abundances for each tissue.	13
2.14	The integral of the sorted abundances for sub-types of Brain. This is done using TPM to avoid biases due to gene lengths. Blood is plotted for reference.	14
2.15	The integral of the sorted abundances for each tissue	14
2.16	The integral of the sorted abundances for each tissue. 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.	15
2.17	The integral of the sorted abundances for each disease type.	16
3.1	Variance versus average. In red the Poisson-like scaling, in green the Taylor-like scaling. All genes are considered	17
3.2	Coefficient of variation squared versus average. In red the Poisson-like scaling, in green the Taylor-like scaling	18

3.3	Variance versus average. In red the Poisson-like scaling, in green the Taylor-like scaling. Only protein coding genes are considered	19
3.4	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	19
3.5	Variance versus occurrence	20
3.6	Caption	20
3.7	Relation between the occurrence of a gene and its average across realisations	21
4.1	An example of a bipartite network. Samples are on the left, genes are on the right. Each link is weighted by 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 set of genes, also known as topics. Blue lines represent the cluster structure, each blue square is a set of nodes, lines delineate the hierarchical structure. It is clear in the middle the network separation in genes and samples.	23
4.2	Example of a hierarchical structure.	24
4.3	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 $bi \rightarrow s$ is made with a larger probability than either $bi \rightarrow u$ or $bi \rightarrow r$ (no movement), since $e_{ts} > e_{tu}$ and $e_{ts} > e_{tr}$	25
4.4	Score across hierarchy. The V-measure or normalized mutual information MI is the harmonic average between homogeneity and completeness.	27
4.5	Example of hierarchical clustering	28
4.6	LAD scheme	28
4.7	Highly variable genes. In cyan genes that are 1σ over the average of their bin.	30
4.8	In cyan genes which distance from boundaries is greater than $10CV^2$	30
4.9	Scores across the hierarchy. The primary site and secondary site labels are compared.	31
4.10	Hierarchy of the files' nodes.	32
4.11	Clusters composition at the level of the hierarchy with higher score. Each column is a cluster, each colour is a label.	33
4.12	Normalised composition of clusters.	33
4.13	Normalised composition of clusters at a deeper level.	34
4.14	Normalised composition of clusters with respect to the secondary site sub-tissue labels.	34
4.15	Example of visualization of clusters with reshuffled labels.	35
4.16	Most representative label versus cluster size.	36
4.17	The fraction of the most representative label in all clusters for different levels of the hierarchy. From upper left the deeper layer than down right the upper one.	36
4.18	Fraction of most representative label versus cluster size.	36

4.19	Fraction of most representative label versus cluster size across the hierarchy. From upper left the deeper layer than downright the upper one.	37
4.20	The number of different labels in each cluster versus cluster size.	38
4.21	Number of different labels in each cluster versus cluster size. From upper left the deeper layer than downright the upper one.	38
4.22	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.	39
4.23	Scores across hierarchy. The scored is compared with some random labels	40
4.24	Scores across hierarchy. The scored is compared with some random labels	41
4.25	Score across the hierarchy for TCGA. In blue the primary site labels were considered, in red the disease types and in green the mix of the two.	44
4.26	Clusters of diseased tissues at the higher level of the hierarchy. Breast is well separated, such as skin and brain. Cluster 6 contains digestive systems samples from pancreas and colon.	45
4.27	text	45
4.28	V-measure score for run with merged healthy and diseased samples. Homogeneity, completeness and mutual information are represented. . .	46
4.29	V-measure score for 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 / diseased) is plotted.	47
4.30	V-measure score for run with merged healthy and diseased samples. LDA, hierarchical clustering and null model for comparison.	48
4.31	$-\log_{10}(P - \text{value})$ of the term with the lowest P-value in each topic. In orange the classical 0.05 threshold.	48
4.32	Categories of the terms with lower P-values in each topic.	49
4.33	Caption	49
4.34	Caption	50
4.35	Caption	50
4.36	$P(\text{topic} \text{tissue})$ for some topic colored by tissue. It is clear a global trend and in some topics there are little differences between tissues.	51
4.37	$\frac{ P(\text{topic} \text{tissue}) - \langle P(\text{topic} \text{tissue}) \rangle_{\text{tissue}} }{\sigma_{\text{tissue}}(\text{topic})}$ or the distance of each tissue from the average tissue expression in each topic. Some interesting topics are reported.	51
A.1	Example of a edge's matrix from [8]	56
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 (center) and finally the network itself (right). From [28].	59

List of Tables

4.1	Enrichment test of a topic. It is clear the enrichment for pancreas related gene sets.	42
4.2	Enrichment test of a topic. It is clear the enrichment for brain related gene sets.	42
4.3	Enrichment test of a topic. It is clear the enrichment for blood related gene sets.	43

Bibliography

- [1] J. Siek, A. Lumsdaine, and L.-Q. Lee, *The boost graph library: user guide and reference manual*. Addison-Wesley, 2002.
- [2] 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.
- [3] E. Jones, T. Oliphant, and P. Peterson, “{SciPy}: Open source scientific tools for {Python},” [*Online; accessed 2019-07-12*], 2014.
- [4] T. E. Oliphant, *A guide to NumPy*, vol. 1. Trelgol Publishing USA, 2006.
- [5] J. D. Hunter, “Matplotlib: A 2d graphics environment,” *Computing in science & engineering*, vol. 9, no. 3, pp. 90–95, 2007.
- [6] M. A. et al., “TensorFlow: Large-scale machine learning on heterogeneous systems,” 2015. Software available from tensorflow.org.
- [7] Z. et al., “Apache spark: A unified engine for big data processing,” *Commun. ACM*, vol. 59, pp. 56–65, Oct. 2016.
- [8] T. P. Peixoto, “The graph-tool python library,” *figshare*, 2014.
- [9] 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.
- [10] W. Zhou, S. Yao, L. Liu, L. Tang, and W. Dong, “An overview of topic modeling and its current applications in bioinformatics,” *Springerplus*, 2016.
- [11] 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.
- [12] A. Martini, A. Cardillo, and P. D. L. Rios, “Entropic selection of concepts unveils hidden topics in documents corpora,” *arXiv*, 2017.
- [13] Z. Wang, M. Gerstein, and M. Snyder, “Rna-seq: a revolutionary tool for transcriptomics,” *Nature reviews genetics*, vol. 10, no. 1, p. 57, 2009.

-
- [14] 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.
 - [15] 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.
 - [16] 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.
 - [17] Q. Wang, J. Gao, and N. Schultz, "Unified RNA-seq datasets in human cancers and normal tissues - normalized data," *figshare*, 2017.
 - [18] 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.
 - [19] 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.
 - [20] 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.
 - [21] E. G. Altmann and M. Gerlach, "Statistical laws in linguistics," in *Creativity and Universality in Language*, pp. 7–26, Springer, 2016.
 - [22] 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.
 - [23] H. S. Heaps, *Information Retrieval: Computational and Theoretical Aspects*. Orlando, FL, USA: Academic Press, Inc., 1978.
 - [24] 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.
 - [25] 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.
 - [26] 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.

-
- [27] M. Gerlach, T. P. Peixoto, and E. G. Altmann, “A network approach to topic models,” *Science advances*, vol. 4, no. 7, p. eaaq1360, 2018.
 - [28] T. P. Peixoto, “Bayesian stochastic blockmodeling,” *arXiv*, may 2017.
 - [29] P. W. Holland, K. Blackmond, and S. Leinhardt, “STOCHASTIC BLOCKMODELS: FIRST STEPS,” tech. rep., CMU, 1983.
 - [30] T. P. Peixoto, “Nonparametric bayesian inference of the microcanonical stochastic block model,” *Physical Review E*, vol. 95, no. 1, p. 012317, 2017.
 - [31] 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.
 - [32] 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.
 - [33] 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.
 - [34] A. et al., “The Genotype-Tissue Expression (GTEx) pilot analysis: Multitissue gene regulation in humans,” *Science (80-.)*, vol. 348, pp. 648–660, may 2015.
 - [35] 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.
 - [36] 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.
 - [37] J. Grilli, “Multiplicative growth model explains macroecological patterns across microbiomes — Supplementary Material —,” -, pp. 1–15, 2019.
 - [38] 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.
 - [39] 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.

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

L'intero gruppo <http://personalpages.to.infn.it/~caselle/BioPhys/BioPhys.html>