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

Corelatore:
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

Chapter 1

Introduction

In recent years the study of complex systems is becoming more interesting especially because some different systems can share interesting and fundamental properties. Network theory has 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 important laws and data structure, therefore it is possible to use tool from different fields to study and gain information about each others. In particular two datasets containing information about some human healthy and diseased tissues will be analysed. This data come from biological experiments of RNA sequencing.

The ultimate goal of this work would be to study, develop and build machine learning's methods able to discriminate healthy and diseased tissues. Once diseased tissue are found, the next goal is to separate cancer types and ultimately sub-types, which is not always easy clinically.

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

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

In chapter 3 I will describe the basics of component systems and give some basic mathematical definitions of quantities useful in general. This chapter refers in particular to the so called component systems.

In chapter 4 I will represent the gene expression from one sample from TCGA with respect to the genes' rank, one can easily obtain a Zipf's law. This law is well-known and in-depth studied in linguistics, demonstrating that different sources of data (genomic and linguistics) can share some statistical properties.

Demonstrated that linguistics and biological data share some laws in section 5 I will use topic modelling to perform network analysis on datasets. Using topic modelling one would find the inner structure of the samples. One would find clusters such that all samples in a cluster share the tissue and tumour type. As far as a document can contain a mixture of similar topics a single tumour can be very heterogeneous.

In chapter 6 I will discuss the results and the future developments

Many methods of the pipeline written in c++ using openMP and Boost [1] are 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 analysis required tensorflow [6] and pySpark [7]. The topic modelling require graph-tool library [8]. The full work repository is on Github at http://github.com/fvalle1/master_thesis.

Chapter 2

Data presentation

2.1 Dataset

The goal of [9] [10] [11] [12]

2.2 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 tecnology 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.

Data was collected from TCGA[1] dataset at https://portal.gdc.cancer.gov [14].

The particular datatype considered was *Gene Expression Quantification*, with experimental strategy RNA-Sequencing. At the end the downloaded dataset consisted of:

- N = 60483 genes as **components**
- R = 10672 files as **realizations**

This type of data is just a small portion of all data available on the portal, this are the most useful data for this type of analysis.

As highlighted in [15] these data present a challange to clustering tools, because of both the relatively large number of samples and the complex structure created by the inclusion of many different tissues

Attemts were made from GTEx [16] which is a similar source of data from healty tissues. [17] tried to unify data from this two different sources and data are available from [18]. Anyway gene expression data were downloaded directly from GTEx v7[2]

2.2.1 normalization

Ususally gene expression data can be normalized in different ways

- Counts
- RPK
- FPKM
- FPKM-UQ

Counts correspond to raw data. Anyway longer genes may have much reads than smaller gene, so it can be useful to normalize this data.

RPK[3] solves this by dividing counts by gene length L,

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

.

[1] The Cancer Genome Atlas

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

[3] Reads Per Kilobase of transcript

FPKM[4] are provided. FPKM calculation normalizes read count by dividing it by the gene length and the total number of reads mapped to protein-coding genes.

$$FPKM = \frac{RC_g * 10^9}{RC_{pc} * L} \tag{2.1}$$

where

- 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

FPKM can be normalized to the 75th percentile read count value for the sample, in this case it is called FPKM-UQ. FPKM-UQ is obtained by:

$$FPKM - UQ = \frac{RC_g * 10^9}{RC_{g75} * L} \tag{2.2}$$

where

- RC_{g75}: 75th percentile read count value for genes in the sample

2.3 Clean data

2.3.1 Protein coding

The whole dataset contains infos on approximatly 60000 elements with different *ENSG* identifier. Only $\simeq$ 20000 of this are protein coding genes, using Ensemble[5] protein coding genes are selected.

2.3.2 Thresholds

In order to filter out noise, it is useful to put a threshold on the data. Considering data in $FPKM$ format, it is common opinion that values beleow 0.1 can be considered noise. Moreover data above 10^5 are trashed out, because the are syntom of some kind of errors during experiment.

Given this thresholds 3.1 becomes

$$o_i = \frac{\sum_{j=1}^{R} \theta(n_{ij} - 0.1) * \theta(10^5 - n_{ij})}{R} \tag{2.3}$$

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

[5] https://ensemble.org

Chapter 3

Data structure

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

metti citazioni [19] [20] [21]

3.1 Component systems

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

Figure 3.1: 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].

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

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

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 realisations of the datasets, in linguistics this components are articles. Components with low occurrence $\simeq 0$ are present only in a few realisations and are the most specific ones.

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

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

dividing this by the global abundance

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

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

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

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

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

The sum of all abundances in the same realization

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

is called **size**.

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

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

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

3.2 Universal laws in RNA-Seq

3.2.1 TCGA

Analysing TCGA dataset [14] the first interesting analysis is to plot the sorted abundance, this gives the so called Zipf's law. The analysis were made considering *Gene Expression Quantification* as data type, *Transcriptome Profiling* as data, *RNA-Seq* as experimental strategy, *HTSeq - Counts* or *HTSeq - FPKM* as workflow type. 5000

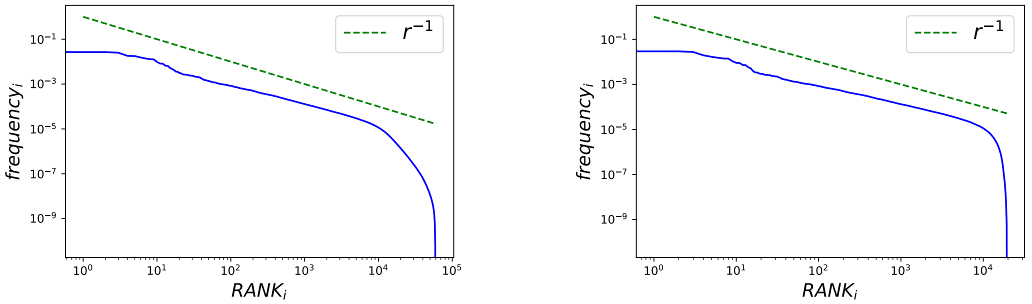


Figure 3.2: Zipf's law from FPKM normalised data. On the right considering only protein coding genes

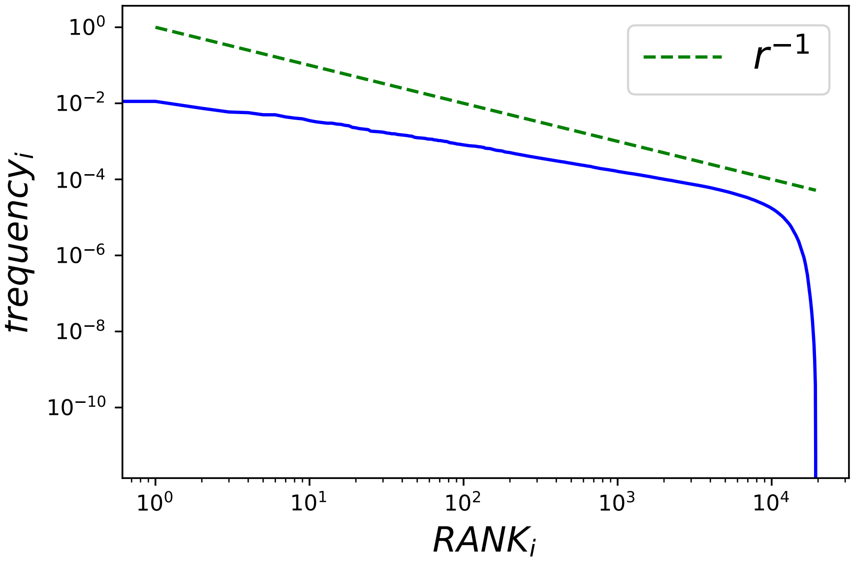


Figure 3.3: Zipf's law of protein coding genes considering counts

samples were downloaded and analysed. In figure 3.2 it is shown the frequency ranked plot. It is interesting that this kind of data distribute according a power law with exponent close to 1, this same behaviour can be found in completely different systems such as linguistics' ones [21]. Another interesting fact is that considering in the analysis also non-coding genes gives a double-scaled power law. This is due to the fact that non coding genes are also more specific and rare, so their frequencies are quite small compared to protein coding genes.

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

3.2.2 GTEx

A pretty similar analysis can be made on GTEx's [18] healthy samples. RNA sequencing raw counts data were download from file version *2016-01-15 v7 RNASeQCv1.1.8*. All ~ 11000 samples available were considered at this time.

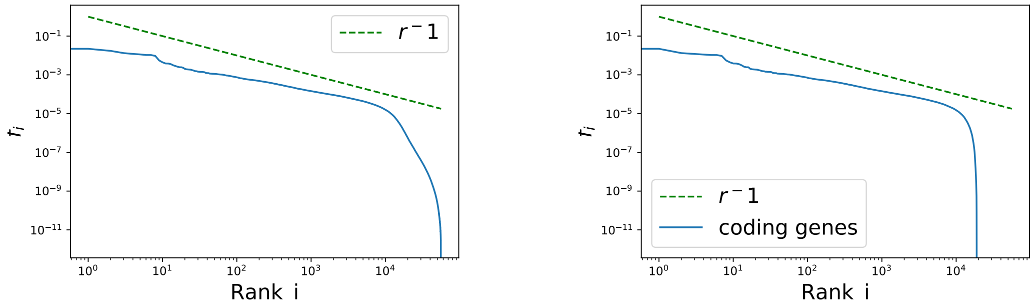


Figure 3.4: Zipf's law from GTEx count data. On the left all genes considered, on the right only protein coding ones

Not surprisingly in the GTEx dataset it is retrieved the same behaviour at this time. The power law with exponent $\simeq 1$ is found and considering non coding genes lead to a knee in the power law.

Going further in the analysis it is possible make an histogram of occurrences defined by 3.1, also known as Us.

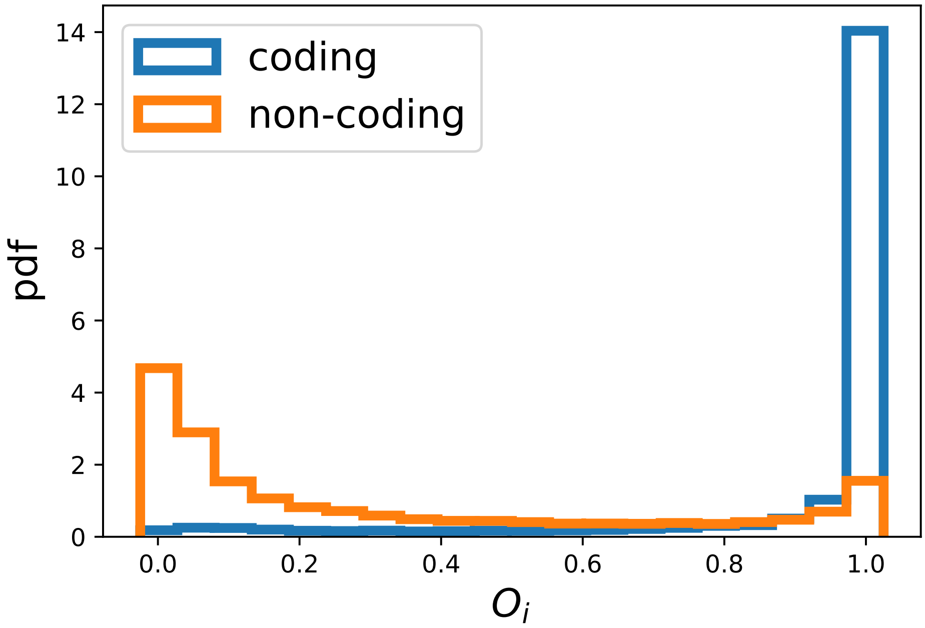


Figure 3.5: The histogram of the occurrences O_i

Also in this kind of analysis it is possible to see the different behaviour of coding and not 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 the have small occurrence. The same behaviour can be observed considering just all samples of a given tissue. In this case $O_i = 0$ means that the genes has a non zero expression in just one of the samples of the tissue considered; in other words if a gene never express in a tissue it is not considered when constructing these tissue specific U distributions.

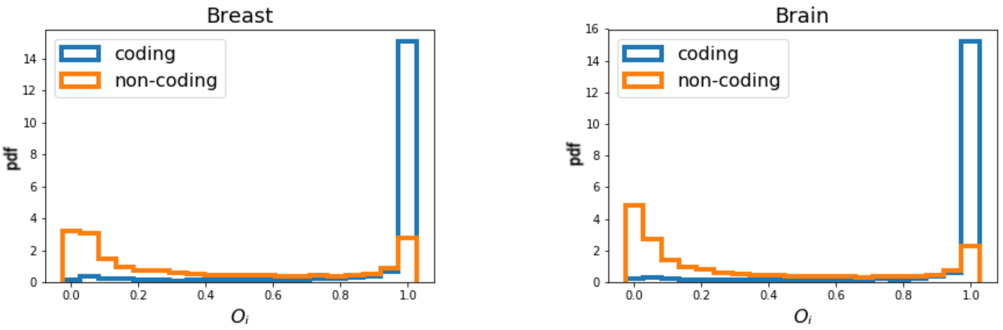


Figure 3.6: Same behaviour is observed looking at one tissue a time.

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

3.3 Null model construction

The kind of data considered in this work comes from RNA Sequencing experiments. This 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 experimenters 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 3.1. The values of abundances of each component in each realization n_{ij} are randomly assigned with a probability determined by the global abundance in the whole dataset 3.2. Values of each column are extracted until the size 3.6 is reached. Strictly speaking it is a multinomial process

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

where n_i is the number of components with frequency f_i, being $f_i = \frac{a_i}{\sum_{i=1}^{N} a_i}$ as defined in 3.4.

Figure 3.7 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

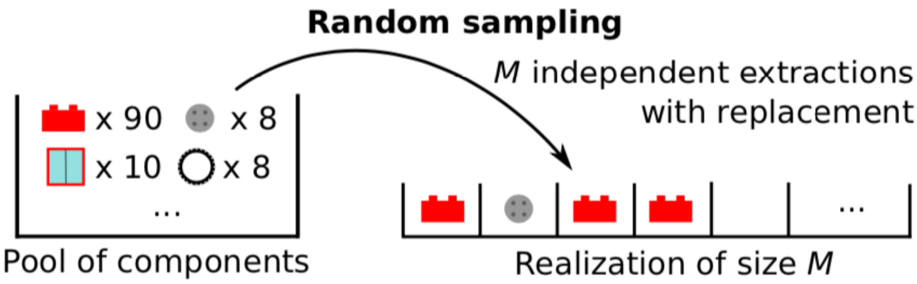


Figure 3.7: Random sampling of components to build a realization of size M

ones with higher frequency (frequency is nothing but the normalised abundance), have a greater probability to be picked up.

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

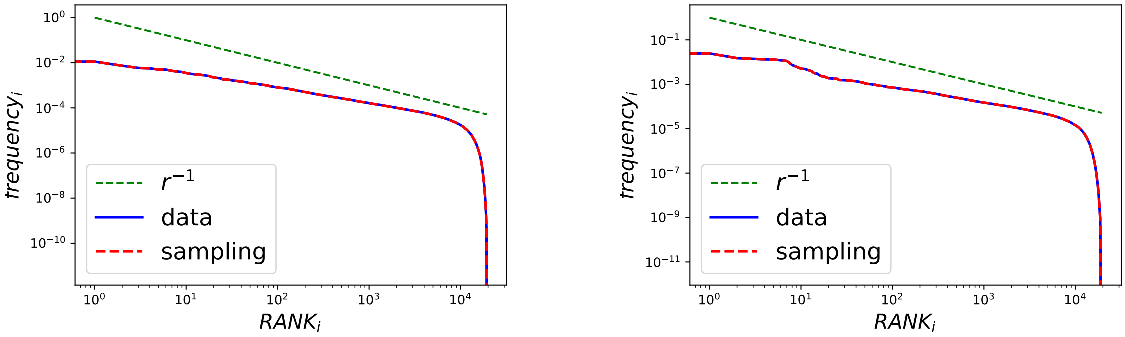


Figure 3.8: Zipf's law sampled; TCGA(left) and GTEx (right)

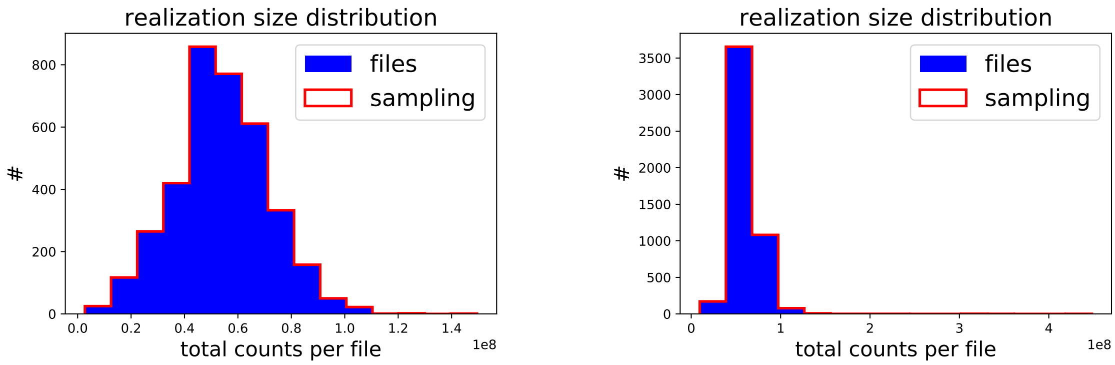


Figure 3.9: Distribution of sizes M; TCGA(left) and GTEx (right)

Looking at the Us, it is evident that data is different from sampling. This is a signal that the null model is not enough to explain the data matrices. In particular from figure 3.10 it is evident that the null model generate 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 the such genes are highly expressed from the abundance and so samples these quite often (components with high abundance have a greater chance to be picked up by the null model sampling).

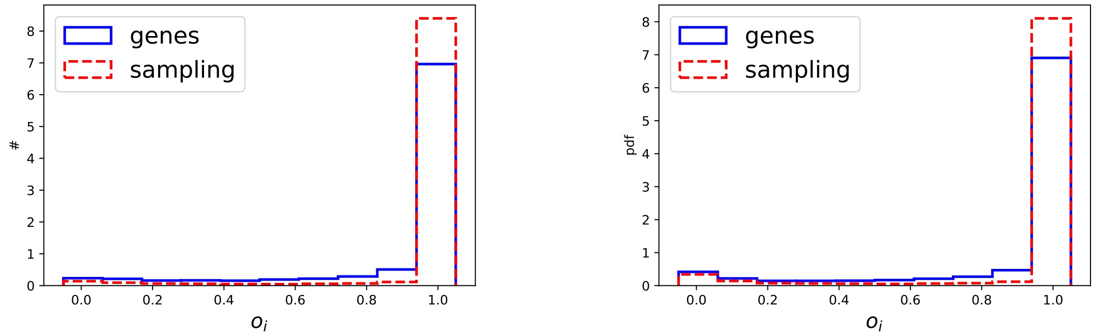


Figure 3.10: Occurrence distributions; TCGA(left) and GTEx (right)

Looking at the Heaps's law [23] , again the curves differ and the null model is not enough complete to explain the trend. In figure 3.11 the Heaps's law is presented compared to the one obtained by sampling, note that each data point share the abscissa with a sampling one (figures 3.9 are nothing but the histograms of the abscissas of 3.11). 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 describable just by sampling. This fact is coherent with the fact that the Us differ. Another way to see this is looking at the histograms of the number of different genes expressed, actually the distribution of the 3.11 y axis. Figure 3.12 shows that these distributions are completely different if one looks at the data and at the samples.

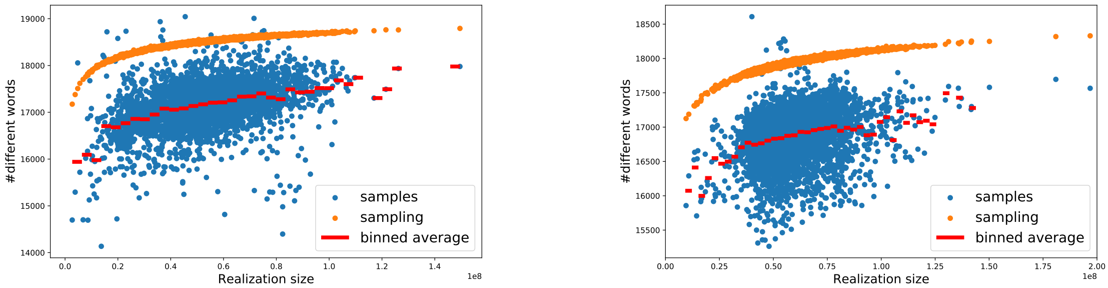


Figure 3.11: Heaps' law; TCGA(left) and GTEx (right)

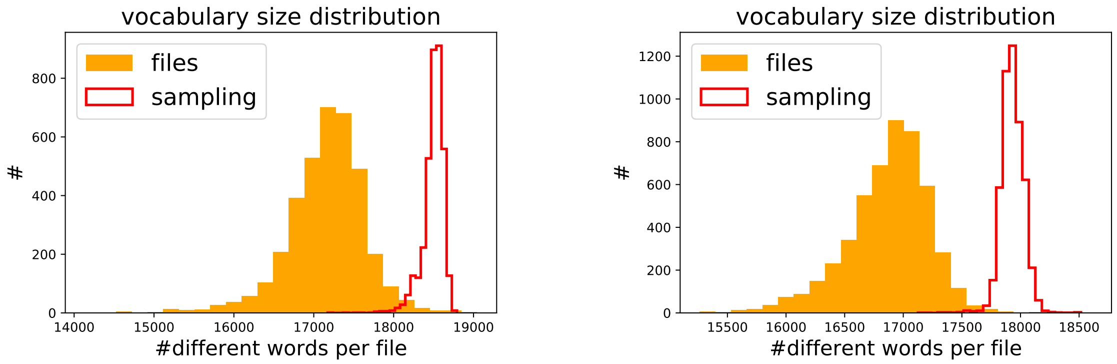


Figure 3.12: Occurrence distributions; TCGA(left) and GTEx (right)

3.4 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 the approach.

First of all could be interesting to study which is the fraction of transcript that can be explained by a certain number of genes. One can reduce the realisations to the ones that share the tissue. Than one estimates the average 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 $1, 2, 3 \ldots$ genes. This is plot in 3.13. Here, if a curve is steep it means that a few genes' counts represent a great fraction of the total. If a curve is smooth it means that many genes are necessary to describe the whole trascriptome for that particular tissue. This analisys shows that different tissues have a different complexity in terms of the number of genes necessary to build the trascriptome (in average). In figure 3.14 the same analisys is done for the sub-tissues of Brain, also this sub-type separate by tissue.

Coming back to the Zipf's law 3.7, it is now obvious that 3.13 represents nothing but the integral of the Zipf's law. So estimating the Zipf looking at a tissue a time, it is evident that each tissue has its particular slope. The steeper the Zipf the simplest is the tissue: the transcript can be described with a few genes. In figure 3.15 the tissue with an extreme behaviour.

The point where the 3.13 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 visualised again with the Heaps' law. In figure 3.16 it is evident that there is some kind of tissue differentiation even when looking at the Heaps' law.

All these analysis 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 different 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 realisation is not always the same (shown by Heaps' law) and it is somehow related to the tissue of the sample.

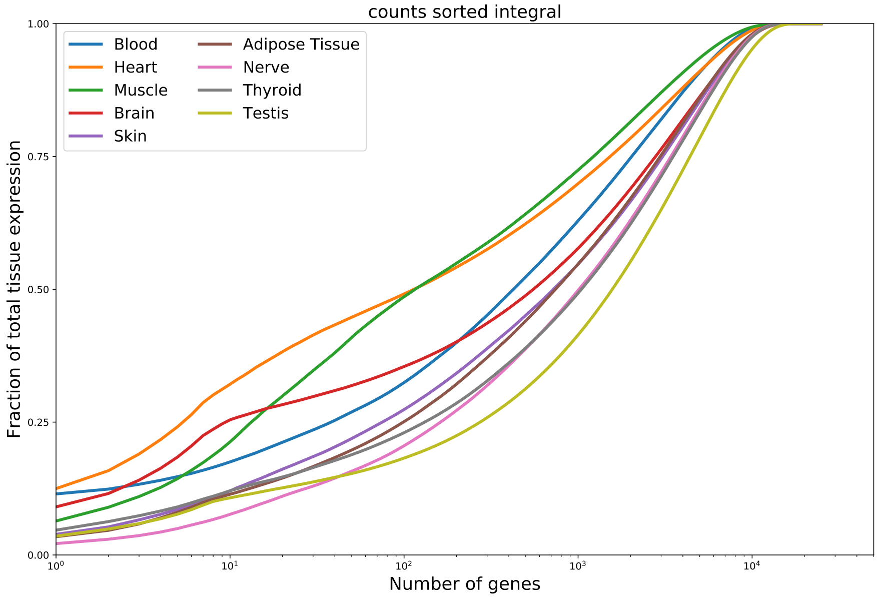


Figure 3.13: The integral of the sorted abundances for each tissue

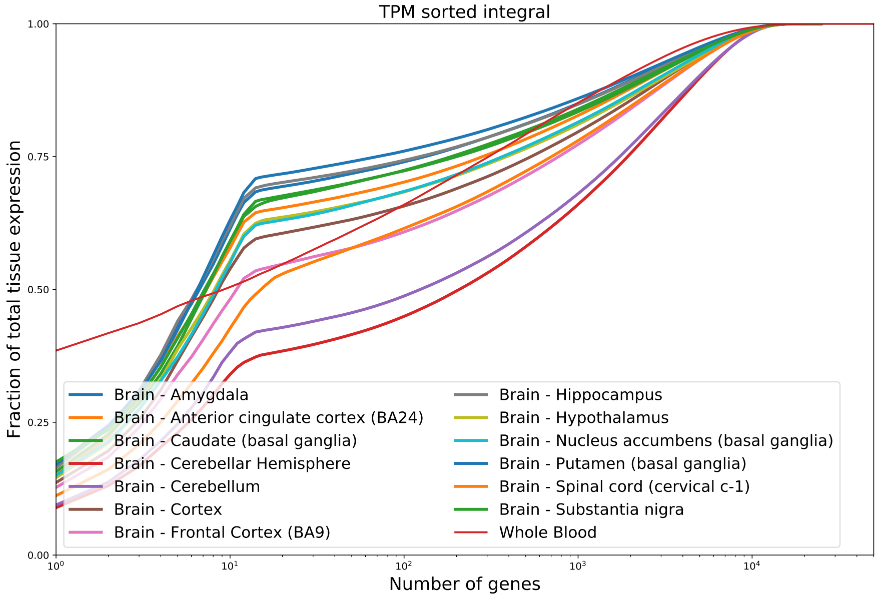


Figure 3.14: The integral of the sorted abundances for sub-types of Brain. This is done using TPM to avoid biases due to gene lengths. Blood is plotted for reference.

In conclusion, some interesting laws were found that some statistical.

...

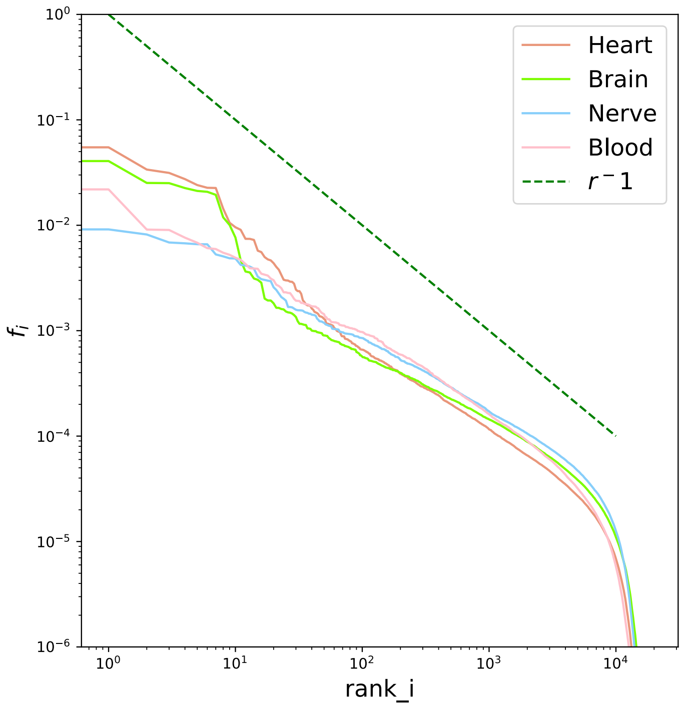


Figure 3.15: The integral of the sorted abundances for each tissue

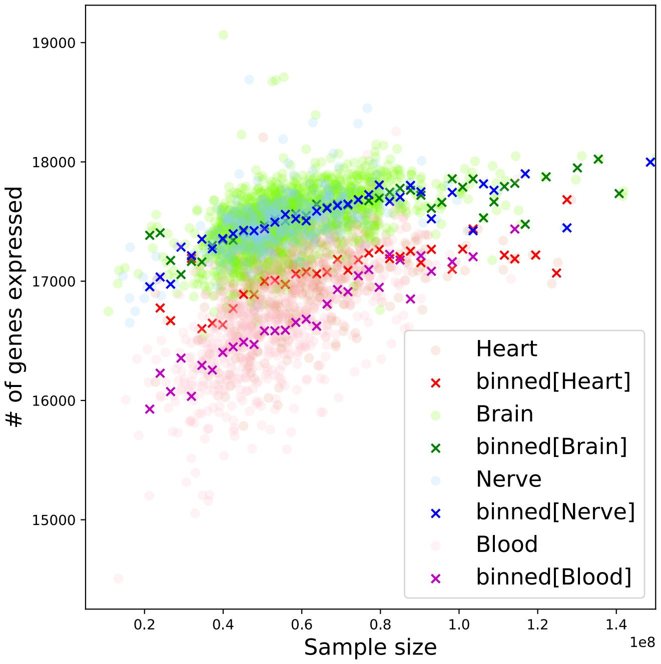


Figure 3.16: The integral of the sorted abundances for each tissue

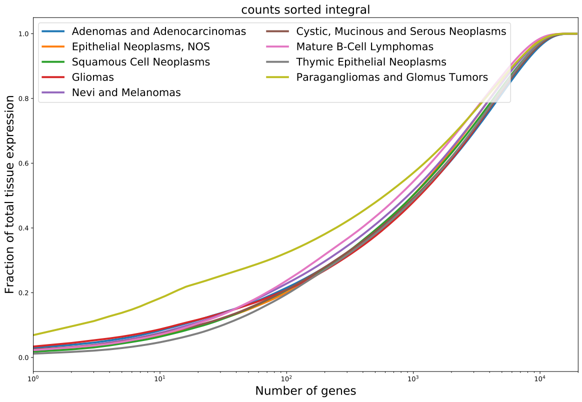


Figure 3.17: The integral of the sorted abundances for each tissue

Chapter 4

Scale laws

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

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

4.1 Scaling

gene expression across samples? gamma?

Given a matrix of components and realisations as 3.1 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 $\sigma_i^2 = \left\langle n_{ij}^2 \right\rangle_j - \langle n_{ij} \rangle_j^2$.

First of all it could be interesting to study the variance of expression σ^2_{counts} versus the average $\langle \text{counts} \rangle$ across tissues. In figure 4.3 the scatter plot of variance versus

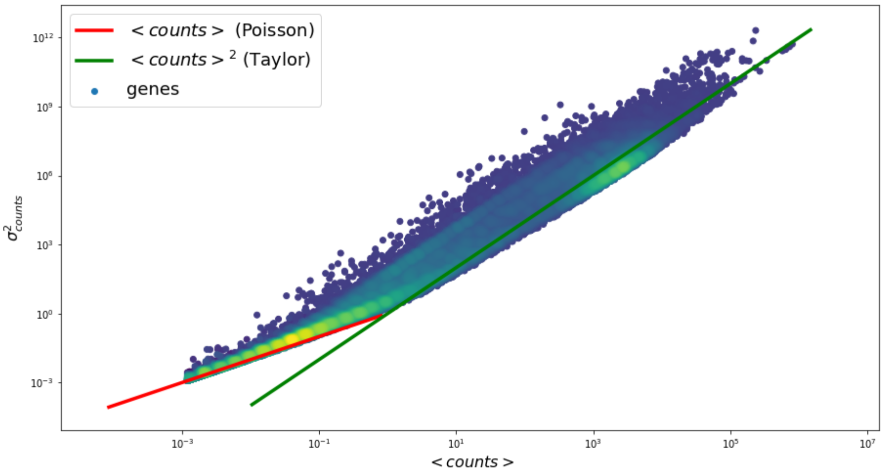


Figure 4.1: Variance versus average. In red the Poisson-like scaling, in green the Taylor-like scaling. All genes are considered

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 ($\sigma^2_{\text{counts}} \sim \langle\text{counts}\rangle$), at higher means instead data have a quadratic scaling ($\sigma^2_{\text{counts}} \sim \langle\text{counts}\rangle^2$) 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 4.1) are evident two clouds of points, one at low averages, one at high averages. These correspond to coding and non coding genes, remembering section 3.2 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.

A similar analysis, common in literature, is the analysis of the coefficient of variation squared $CV^2 = \frac{\sigma^2_{\text{counts}}}{\langle\text{counts}\rangle^2}$ represented in figure 4.2. The behaviour is complementary to

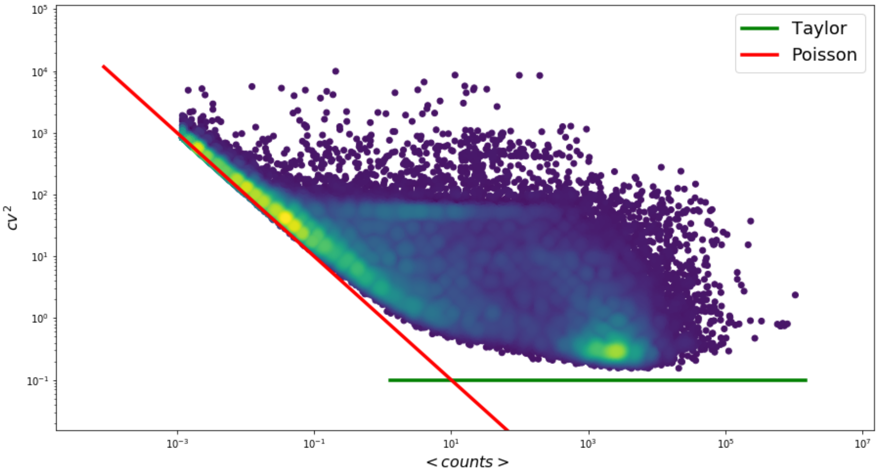


Figure 4.2: Coefficient of variation squared versus average. In red the Poisson-like scaling, in green the Taylor-like scaling

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, $\sigma^2_{\text{counts}} \sim \langle\text{counts}\rangle$ so $CV^2 = \frac{\sigma^2_{\text{counts}}}{\langle\text{counts}\rangle^2} \sim \frac{1}{\langle\text{counts}\rangle}$, otherwise the protein coding genes are on the Taylor-like curve $CV^2 = \frac{\sigma^2_{\text{counts}}}{\langle\text{counts}\rangle^2} \sim 1$.

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 3.3 the averages and variances can be estimated on null matrices. In figure 4.4 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 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

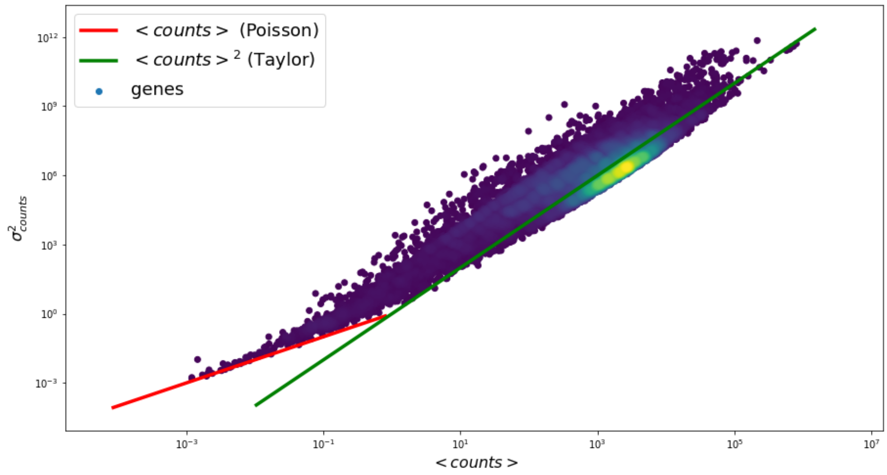


Figure 4.3: Variance versus average. In red the Poisson-like scaling, in green the Taylor-like scaling. Only protein coding genes are considered

information that the data are more variable (have higher variance) than just sampling, so there must be some biological information hidden that causes this over variable behaviour.

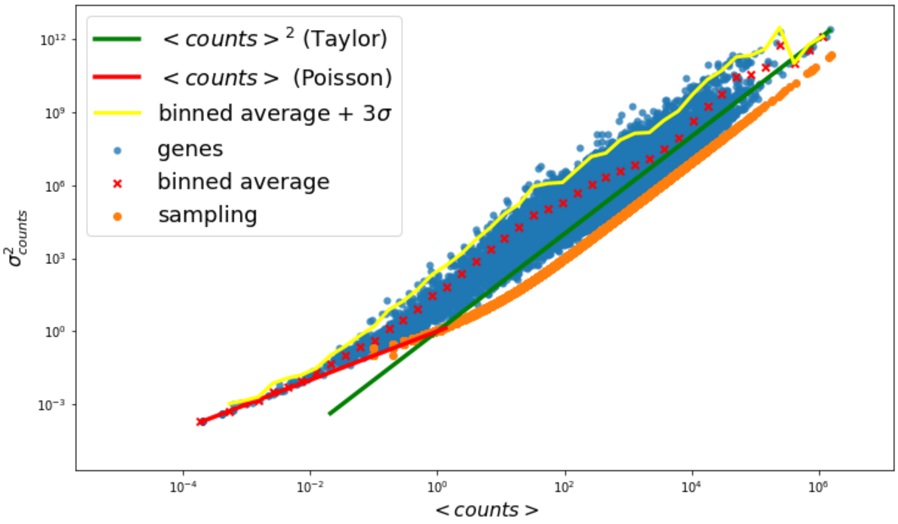


Figure 4.4: Variance versus average. In red the Poisson-like scaling, in green the Taylor-like scaling. In orange the sampling components. Only protein coding genes are considered

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

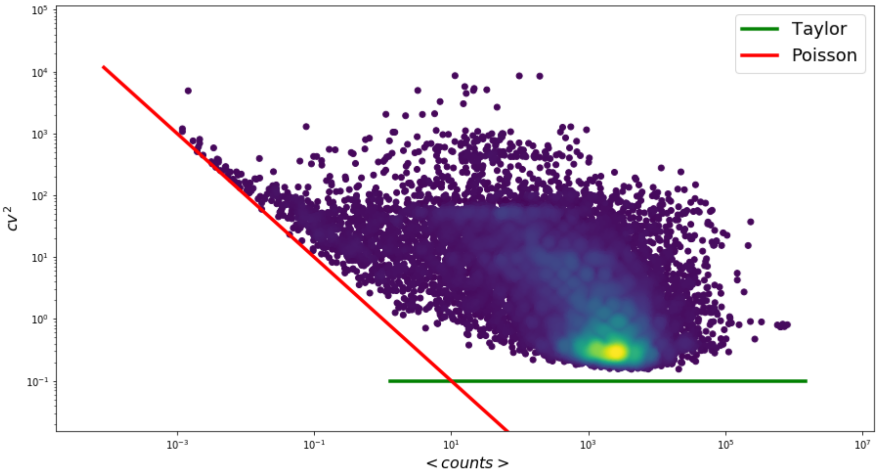


Figure 4.5: Variance versus occurrence

In figure 4.6 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^2_{counts}) 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.

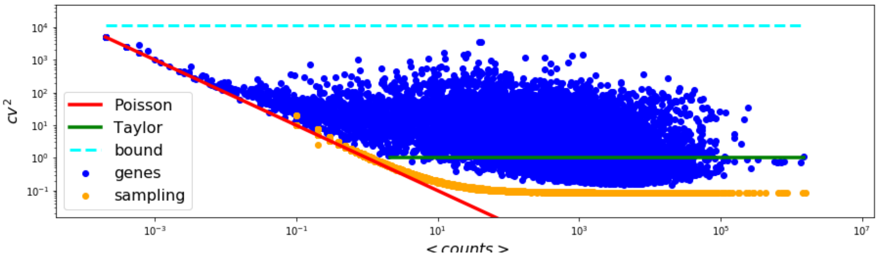


Figure 4.6: Caption

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.

$< FPKM >$ normalisation One can be interested in finding genes that are expressed often, and what is the average expression of them. To manage this it is plotted the average expression $< FPKM >$ versus the number of samples in which that gene is expressed that is, considering the thresholds 2.3.2, $\Sigma_j\theta(FPKM_{ij} - 0,1)\theta(10^5 - FPKM_{ij})$

4.1.1 Average versus occurrence

Another interesting analysis can be the relation between the occurrence and the average. In figure 4.7 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 $\langle \text{counts} \rangle = \frac{1*n}{R}$

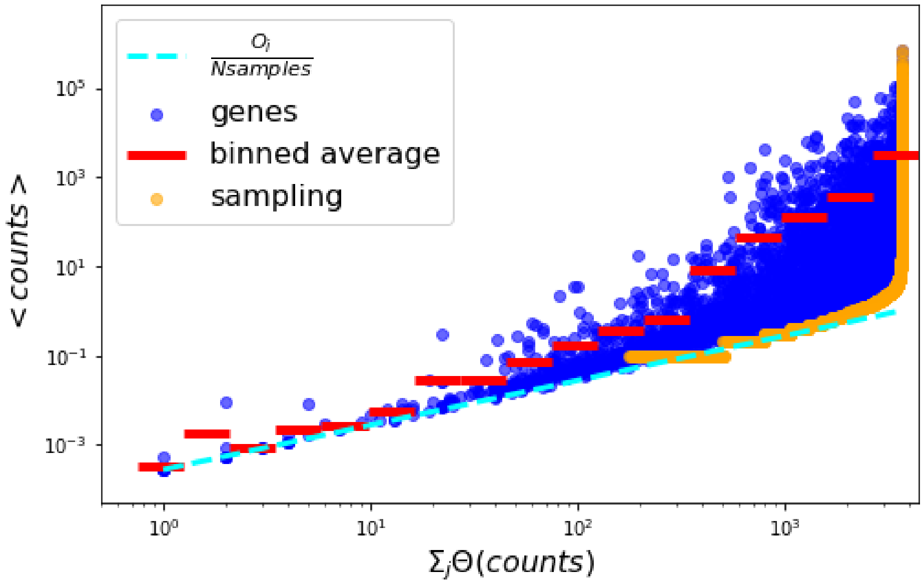


Figure 4.7: Relation between the occurrence of a gene and its average across realisations

4.1.2 Tissue differentiation

Per gene type scaling

Chapter 5

Topic modelling

Once extensively analysed 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 3 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 relation 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, than nodes are linked considering the gene expression value in the dataset.

The output of this kind of model are set of genes, the topics, with a probability distribution $P(gene|topic)$ and probability distributions of these topics inside each sample $P(topic|sample)$, 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 extends the so called stochastic block models [28]. 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 3 was evident that there are many similarities between data considered in this work and linguistics' dataset. Referring to data used in this project documents will be **samples**, words will be **genes**. 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, than 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 apply community detection at different layers. So the output has got different resolution, the extreme one is the separation, by definition, between samples and genes, than 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 data G

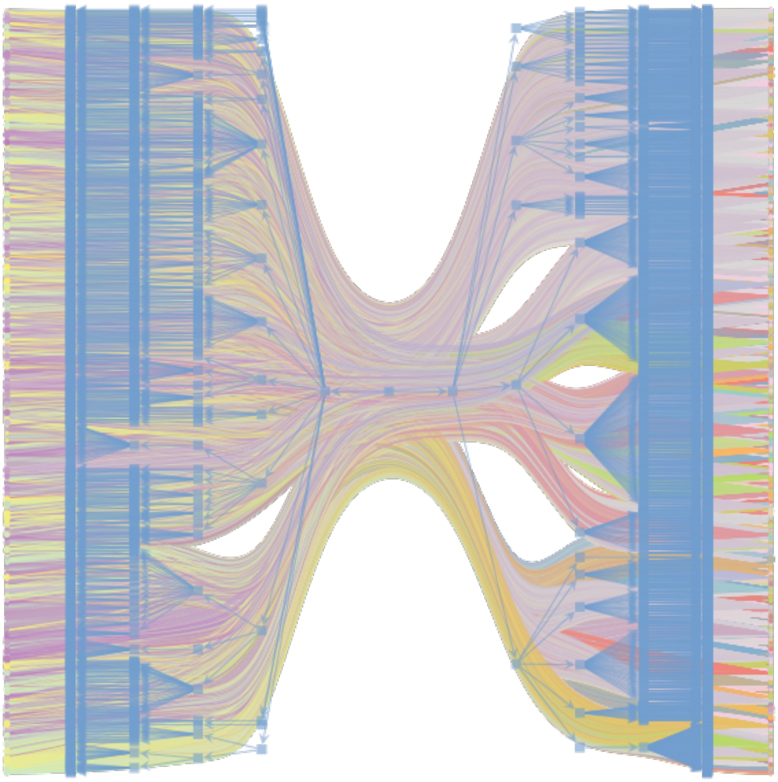


Figure 5.1: An example of a bipartite network. Samples are on the left, genes are on the right. Each link is weighted by gene expression value. On the left side all nodes of the same colour are clusters of samples. On the right side all nodes with same colour are set of genes, also known as topics.
Blue 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.

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

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

It is possible to define a description length

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

so that $P(\theta|G) \propto \exp -\Sigma$. Moreover the likelihood $P(G|\theta)$, can be written as $\frac{1}{\Omega}$ where Ω is the number of possibles realisations given θ. This can be represented as a microcanonical ensemble with entropy $S = Ln\,(\Omega)$. Note that $\Sigma = S - lnP(\theta)$ According to [29] entropy S can be written as

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

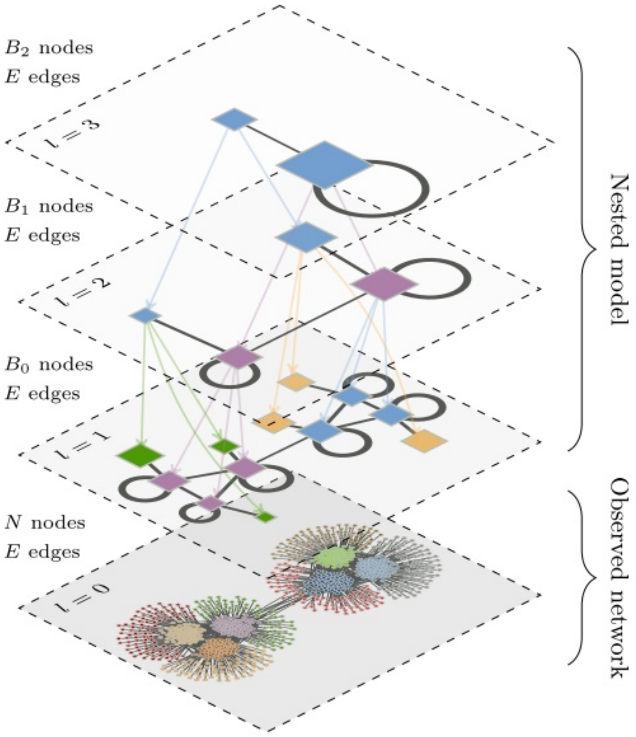


Figure 5.2: Hierarchical structure

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 $H(X) = xLog_2(x)+(1-x)Log_2(1-x)$. Note that S is minimal if $\frac{e_{rs}}{n_r n_s}$ is close to zero, r and s are two completely separated blocks or if it is close to 1, r and s are groups with many connections; this allows 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 Monte Carlo works in few steps:

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

in figure 5.3 an example of these steps.

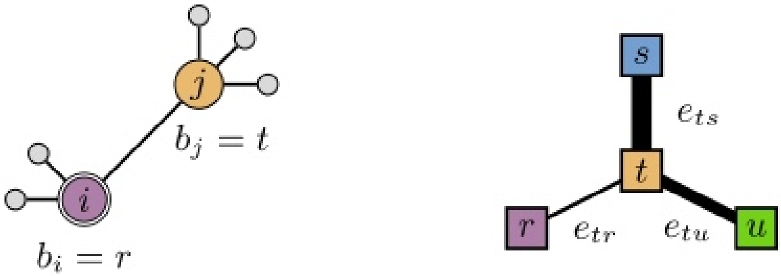


Figure 5.3: Left: Local neighbourhood of node i belonging to block r, and a randomly chosen neighbour j belonging to block t. Right: Block multigraph, indicating the number of edges between blocks, represented as the edge thickness. In this example, the attempted move $bi \to s$ is made with a larger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

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

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

and the topic distribution inside a document

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

In case of overlapping partitions the presence of a word in a topic is not trivial and can be extimated 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 7 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.

5.1 Metrics and benchmarks

Before put the hands on topic modelling, it is useful to define some metrics to test and benchmark the model.

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

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

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

defining the entropy

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

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

Another quantity can be defined and it is completeness:

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

$H(K|C)$ is defined in the same way as 5.2. Completeness measures if all nodes of the same type are 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, which is the harmonic average of the two:

$$MI = 2\frac{hc}{h+c}, \tag{5.4}$$

which is actually the mutual information between $P(C)$ and $P(K)$ [30].

In the next sections will be studied also the maximum fraction of label in the same cluster $max_{c \in k} \frac{n_{ck}}{n_k}$. Also the number of different labels in the same cluster will be studied.

5.1.1 LDA

commenti vari

As in [10]

$$P(w, z, \beta, \theta | \alpha, \eta) = \prod_n^{N_d} P(w|z, \beta) P(z|\theta) \prod_k^K P(\beta|\eta) \prod_d^N P(\theta|\alpha) \tag{5.5}$$

where

- N number of documents
- K number of topics

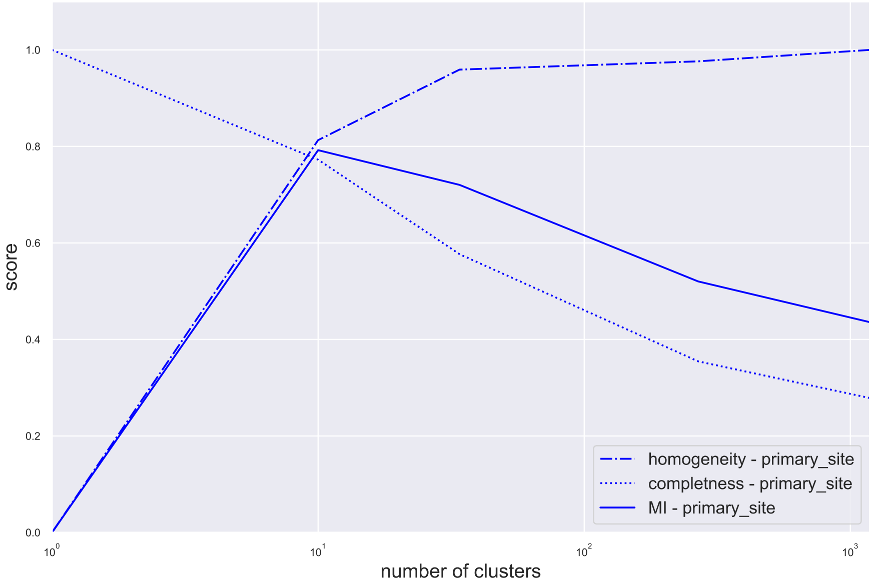


Figure 5.4: Scores across hierarchy. The mutual information is the armonic average between homogeneity and completeness

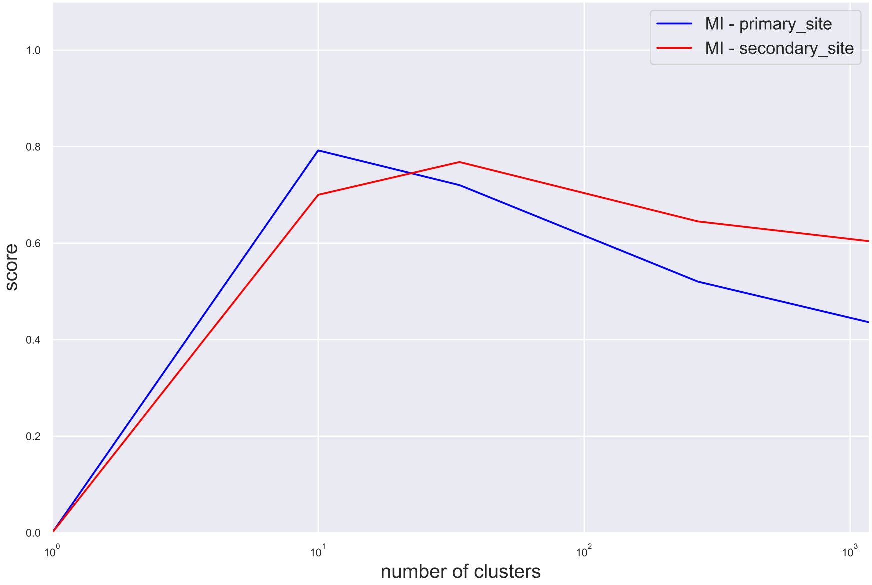


Figure 5.5: Scores across hierarchy. The scored is compared with some random labels

- w words
- N_d number of words in document d

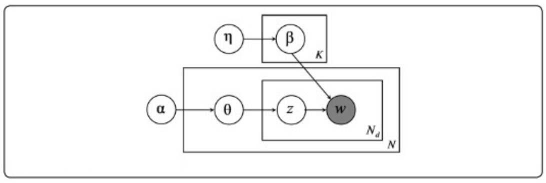


Figure 5.6: LAD scheme

- η and α are parameters of the model

in 5.5 $P(\theta|\alpha)$ and $P(\beta|\eta)$ are Dirichlet distributions the outputs are the topic distribution in documents $P(z|d)$ and the word distribution in topics $P(w|z)$

5.1.2 Hierarchical clustering

da scrivere

hierarchical clustering Hierarchical clustering is a general family of clustering algorithms that build nested clusters by merging or splitting them successively. This hierarchy of clusters is 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. See the Wikipedia page for more details.

The AgglomerativeClustering object performs a hierarchical clustering using a bottom up approach: each observation starts in its own cluster, and clusters are successively merged together. The linkage criteria determines the metric used for the merge strategy:

Ward minimizes the sum of squared differences within all clusters. It is a variance-minimizing approach and in this sense is similar to the k-means objective function but tackled with an agglomerative hierarchical approach. Maximum or complete linkage minimizes the maximum distance between observations of pairs of clusters. Average linkage minimizes the average of the distances between all observations of pairs of clusters. Single linkage minimizes the distance between the closest observations of pairs of clusters. AgglomerativeClustering can also scale to large number of samples when it is used jointly with a connectivity matrix, but is computationally expensive when no connectivity constraints are added between samples: it considers at each step all the possible merges.

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

5.2 Pre-process

To make the algorithm faster, could be interesting to do a pre-processing of the data. Vary approaches were tested, all of them involving the quantities defined in 3. The goal is to identify components which are able to best separate the realisations.

Low occurrence genes were selected firstly to make topic modelling. A 0.5 threshold was set. This method select genes that appears (have expression greater than zero) only in less than half samples. This doesn't consider genes that appear everywhere (whith occurrence $\simeq 1$), but changes their behaviour across realisations.

tf-idf (term frequency–inverse document frequency) should help. This approach doesn't consider values, but a transformed version

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

which increases the importance of components with small occurrence o_i. This 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 from metti referenza giusta. Plotting the coefficient of variation versus the mean for each

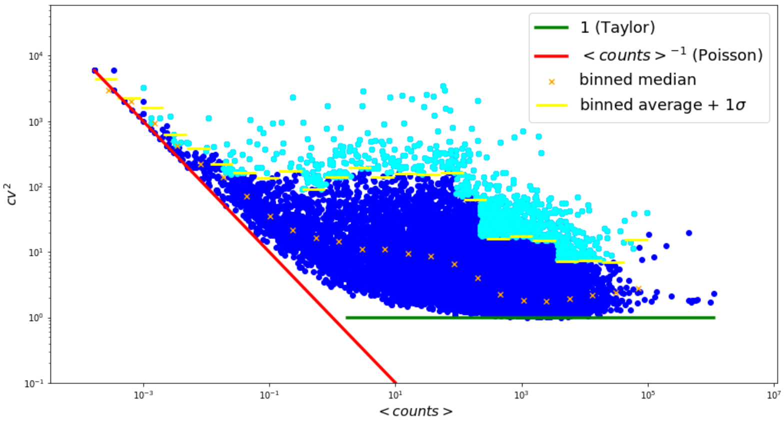


Figure 5.7: Highly variable genes

component reveals which components have higher variance with respect to components which, in 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.

Distance from boundaries can be a similar and alternative method to select highly variable genes. metti figura giusta!! The distribution as discussed in disctuti e linka

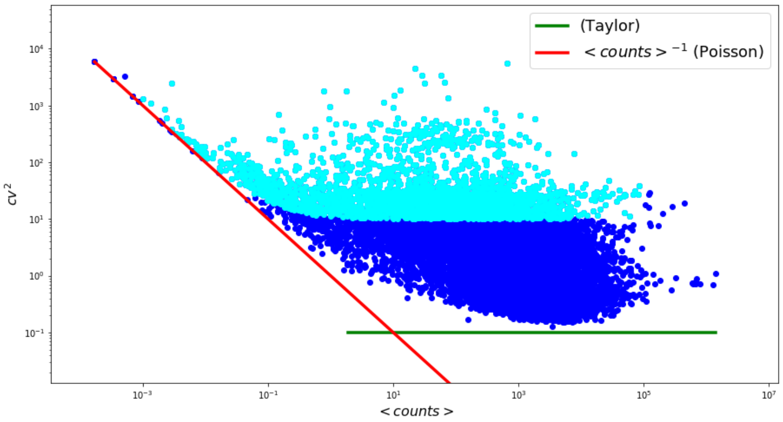


Figure 5.8: Genes distant from the boundaries

have a Poisson-like and a Taylor-like boundaries. So can be considered only components that are the most distant from this boundaries.

Using the last two approaches got the point and actually help topic modelling to succeed.

5.3 Run

5.3.1 Run on GTEx

Firstly the algorithm is run on a subset of 5 tissues of GTEx

5.3.2 Shuffling

5.3.3 Standard algorithms

5.3.4 Run on TCGA

5.3.5 Mixed runs

5.4 Results

Using as gene set [31] enrichment test can be made [32]

Enrichment test are made once for each topic, starting from the layer with more genes per single topic. Test are made across multiple categories.

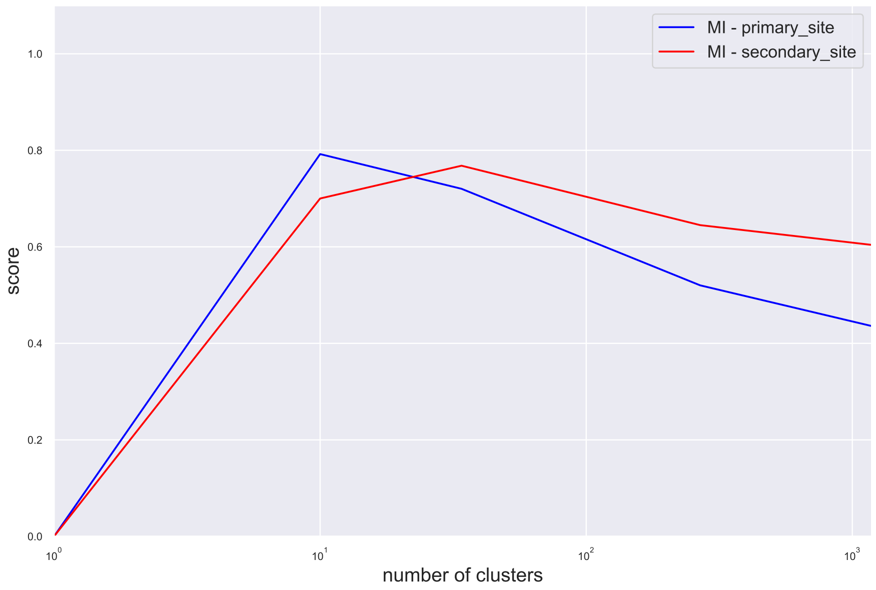


Figure 5.9: Scores across hierarchy

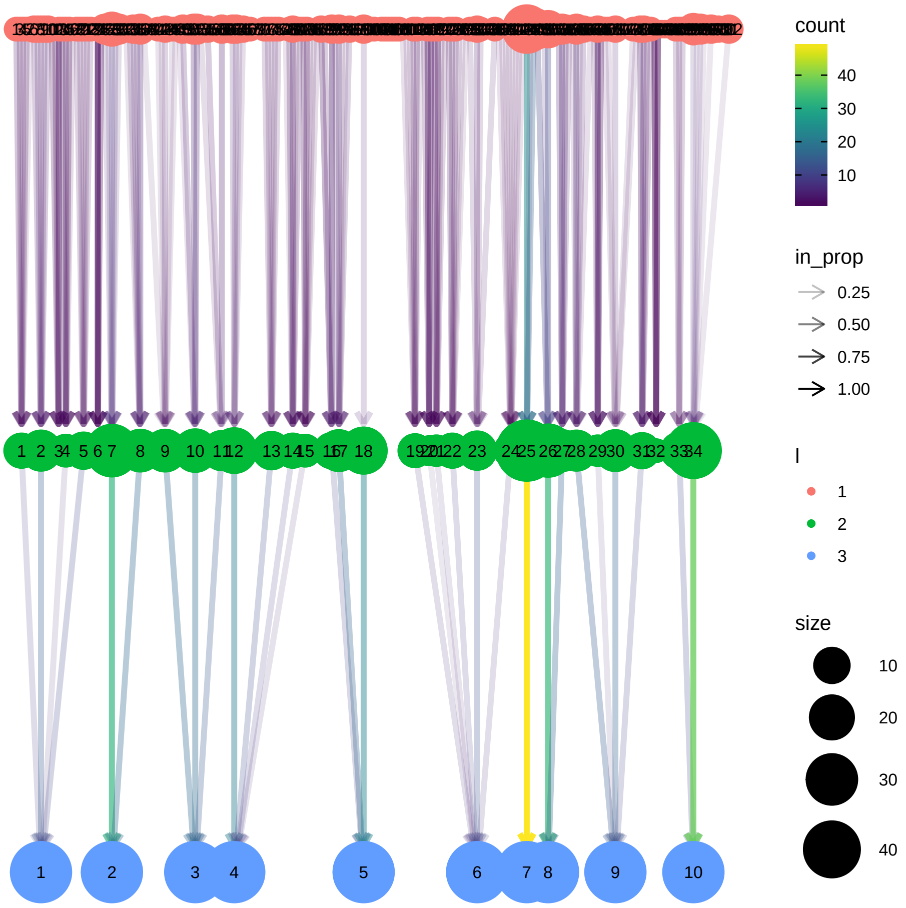


Figure 5.10: Hierarchy of the files' nodes

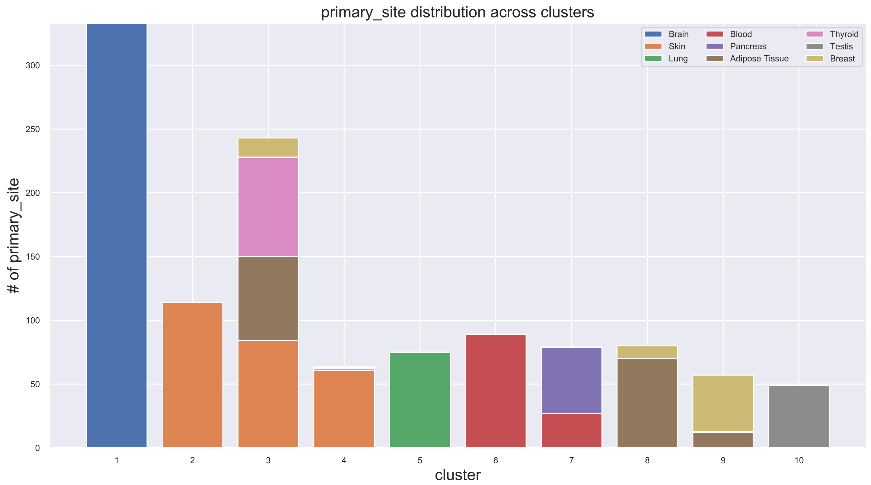


Figure 5.11: Caption

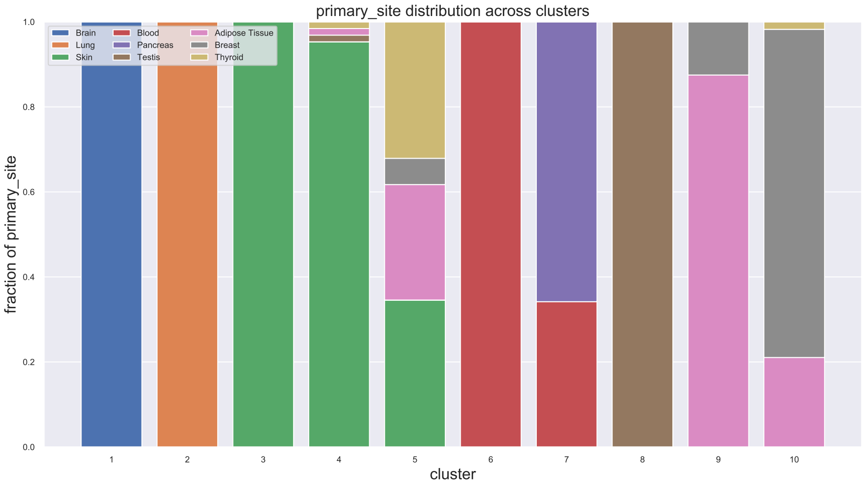


Figure 5.12: Caption

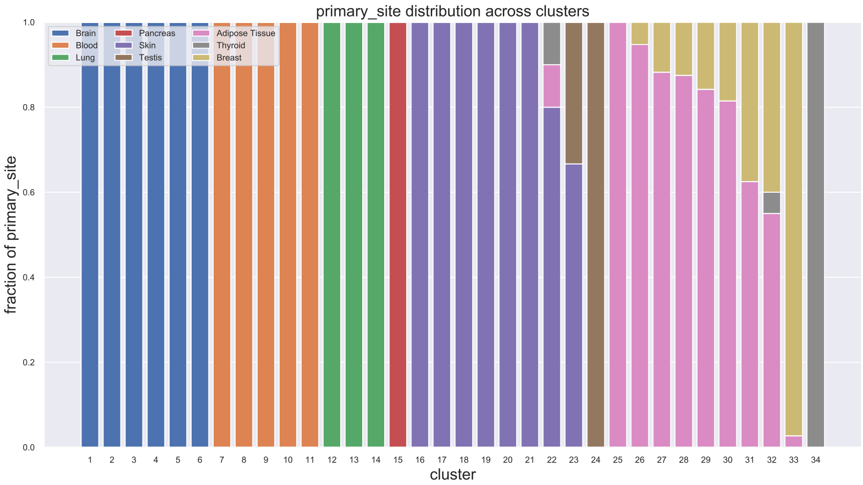


Figure 5.13: Caption

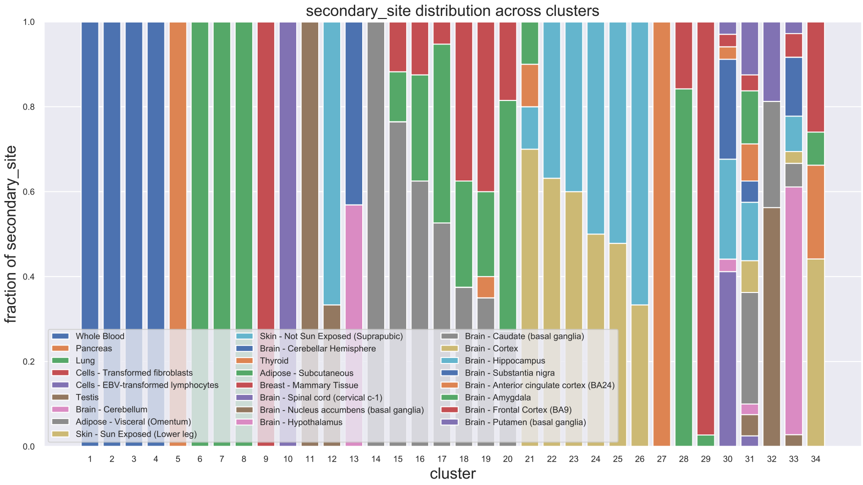


Figure 5.14: Caption

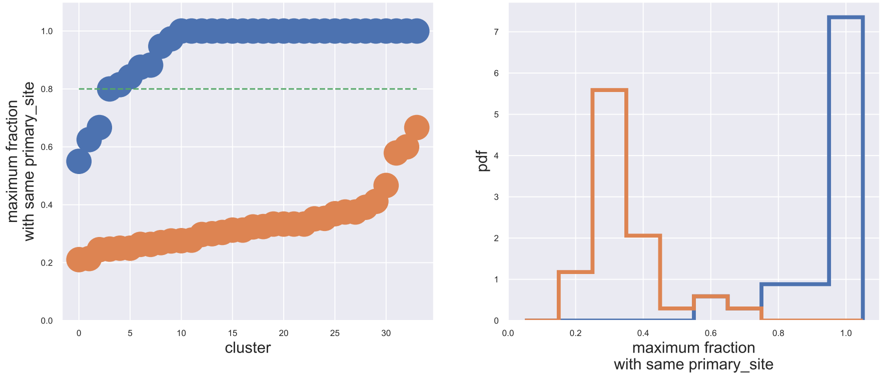
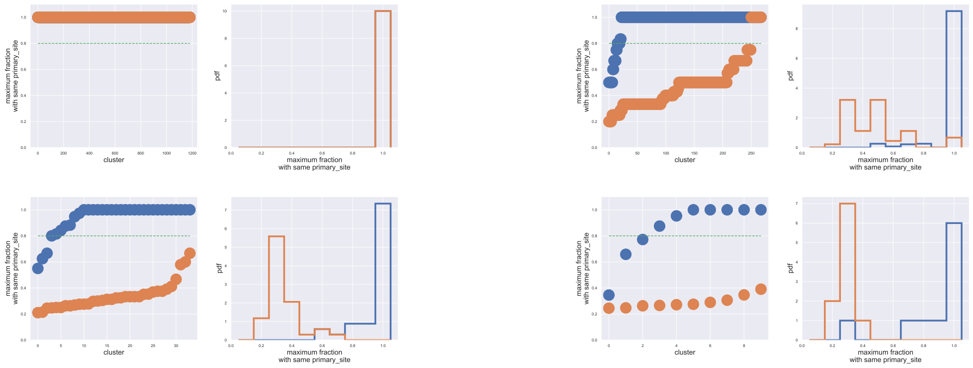


Figure 5.15: Caption



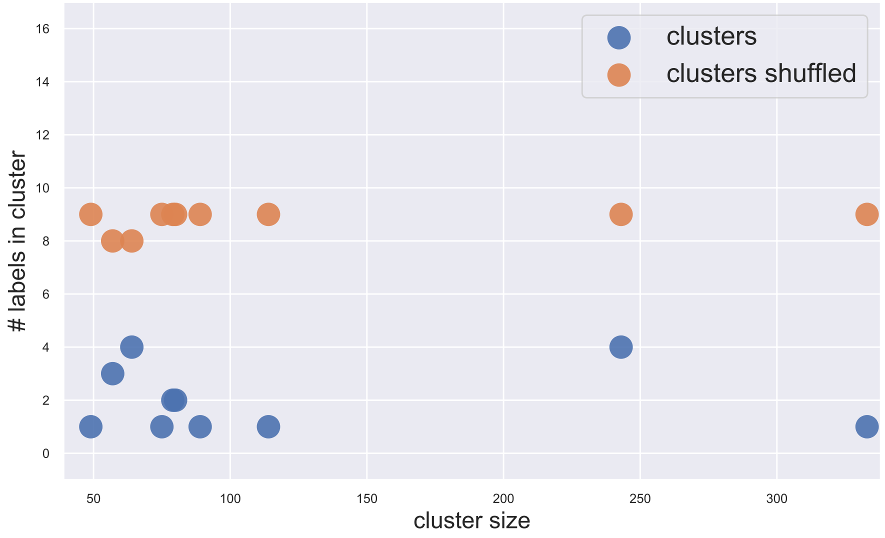
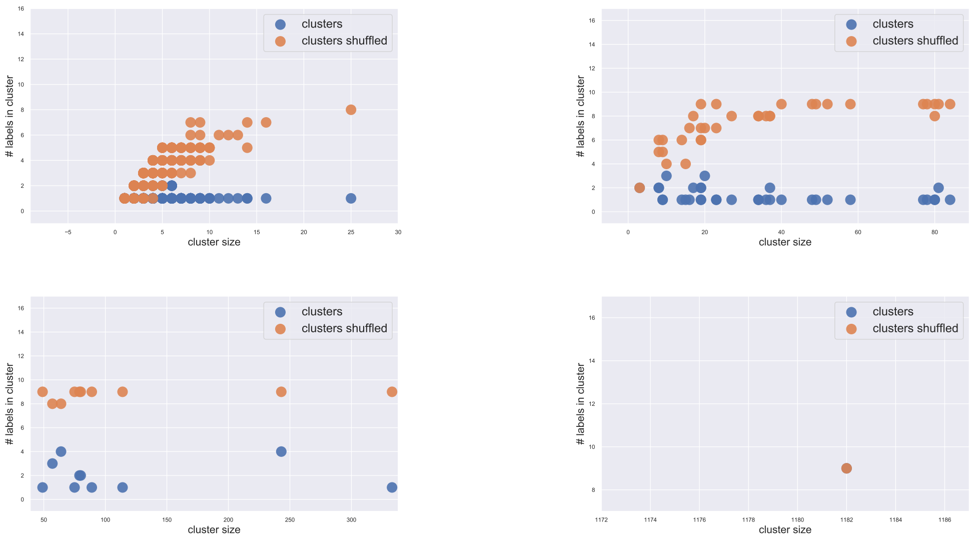


Figure 5.16: Caption



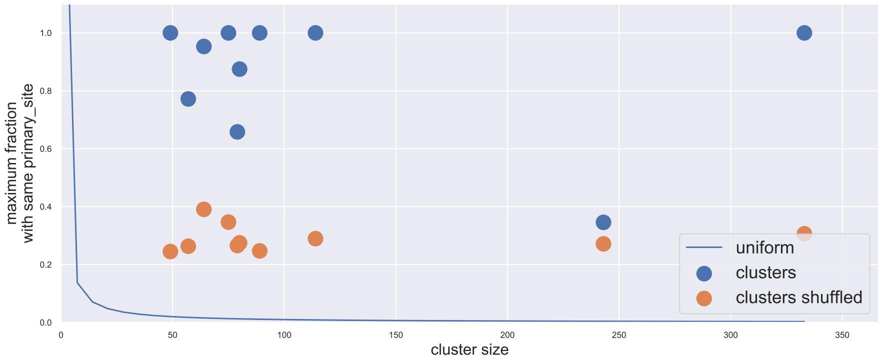
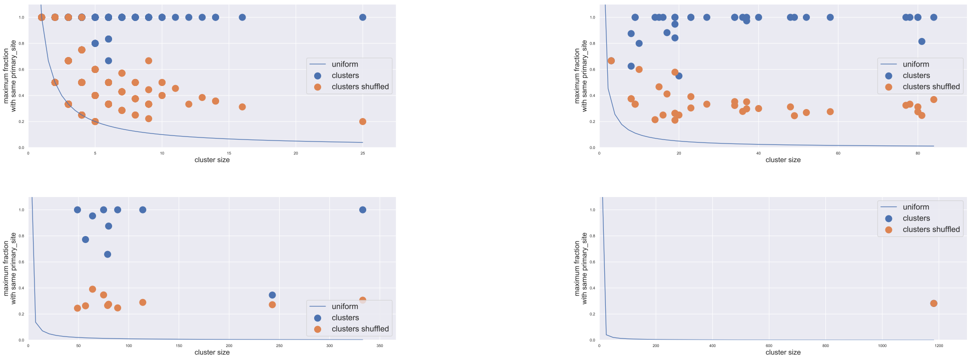


Figure 5.17: Caption



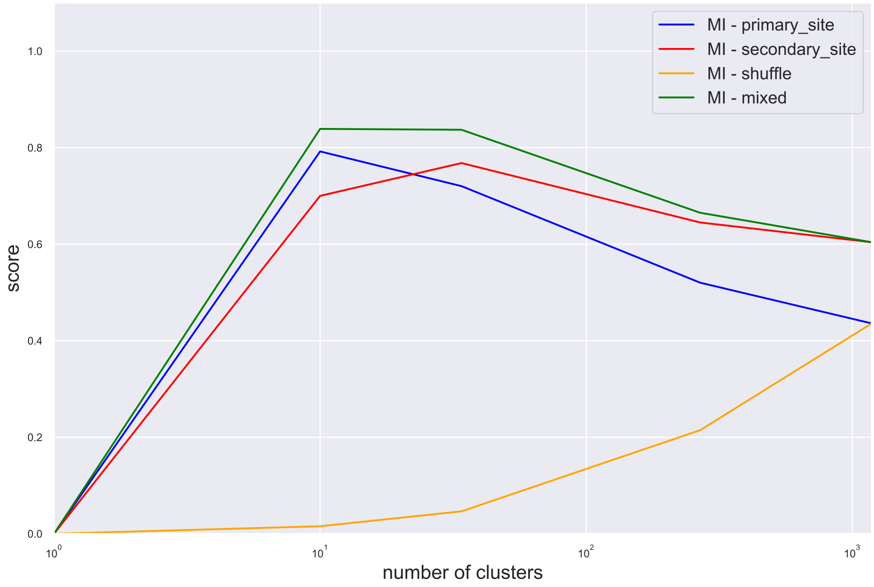


Figure 5.18: Scores across hierarchy. The scored is compared with some random labels

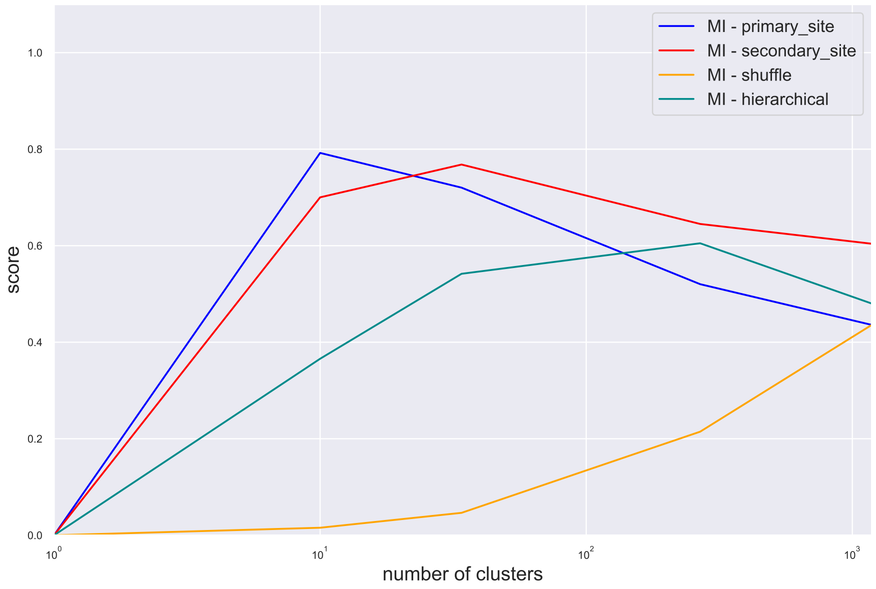


Figure 5.19: Scores across hierarchy. The scored is compared with some random labels

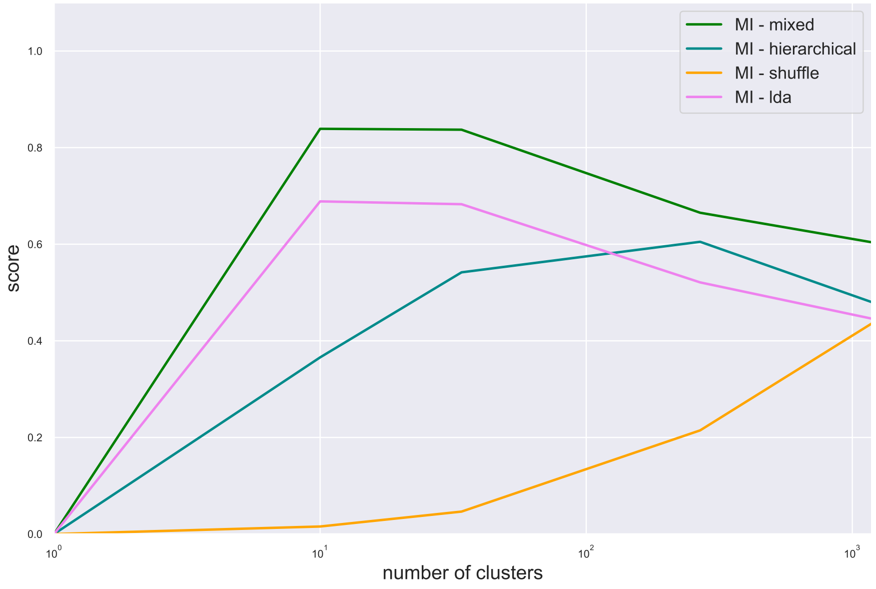


Figure 5.20: Scores across hierarchy. The scored is compared with some random labels

Chapter 6

Results

Finally this work demonstrates that RNA-Sequencing datasets can be analysed from a component systems point of view. This kind of data shows typical trends famous, for example, in linguistics, moreover some interesting biological signatures were found. RNA-Seq dataset have a great core of protein coding genes that express everywhere, this is evident looking at Us, Heaps' law. The presence of a power law distribution in the ranked abundances, the so called Zipf's law is observed and characterise 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 evidences 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 distribution encouraged to explore the possibility of using a topic model approach to reveal the hidden structure of these datasets. This approach is 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 these analysis. First of all it was developed a pipeline that begins with creating a network with useful genes and samples, this network is than processed with hierarchic stochastic block model topic model algorithm and then analysed.

One interesting result is that this method is able to reproduce the distinction between different tissues, 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 reproduce the main tissue label, but was demonstrated that running along the hierarchy of the clusters even the sub tissue specific labels were distinguished. The mutual information score confirms this behaviour of the model were tissues were separated at an higher level of the hierarchy and in next one the sub tissue were explained.

A null model realised shuffling the labels confirms that the results achieved are non trivial and represent somehow the real tissue structure of the data.

The results were compared with more standard approaches such as Latent Dirichlet Allocation and hierarchical clustering. In both cases the results of the approach presented in this work are better than the standard and obtain higher scores. Moreover topic modelling (both LDA and hSBM) is better than standard algorithms. This confirms the good quality of a topic model approach.

On the other side looking at the block of genes, the so called topics, enrichment tests confirms that the topics represent interesting group of genes. In particular some dataset-specific labels were found in GTEx.

In the end the relation between samples and topics reveals that the topics the have uncommon behaviour in the samples of a specific tissue enrich for a function specific for that tissue. The distribution of the topic abundance across samples reveals that it is possible, despite to what a LDA approach does, to describe the tissue differentiation with topic that varies slightly between samples. Biologically this means that all genes are necessary everywhere and a fine tuning of their expression differentiate by tissue.

In the end it was demonstrated that analysing data coming from merged dataset the differentiation is still evident and going forward in the hierarchy depth the separation involves also the healthy or diseased status. Th tissue separation in the firsts layer confirms that what the algorithm does is separating tissues and there is no, evident, bias between datasets.

Chapter 7

Conclusions

In conclusion topic model reveal itself as a useful approach to this kind of data.

Verified that benchmark give good results the next goal would be to

future: Loredana, Jacopo, topic on sampling, Jonathan

List of Figures

List of Tables

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},” 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," 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] P. W. Holland, K. Blackmond, and S. Leinhardt, “STOCHASTIC BLOCKMODELS: FIRST STEPS,” tech. rep., 1983.

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

[30] 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.

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

[32] 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.

[33] 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.

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

[35] 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.

Hierarchical stochastic block model

The algorithm is called hierarchic Stochastic Block Model.

The first step of hierarchical stochastic block model, as discussed in [33], 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

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

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

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

the 1 can be written as $\frac{e^{-\Sigma}}{P(G)}$, so maximising that is equivalent to minimise the description length 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.

.0.1 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.

It is useful to define a traditional entropy:

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

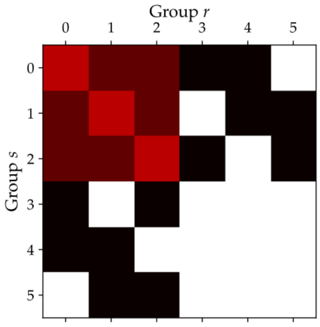


Figure 1: Example of a edge's matrix from [8]

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

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 3 can be corrected taking care of degree distribution obtaining corrected entropy S_c

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

How to change group of a node? At each step according to [33] node i can change group from r to s with a probability

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

where j is a random neighbour of i: $j \in N_i$, $t \in \{b_j\}$ its block as defined in [33]. ϵ is a parameter that according to [29] has no significant impact in the algorithm, provided it is sufficiently small.

5 can be rewritten as

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

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

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 5; note that for $\epsilon \to \infty$ 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 [29]. 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 5 became

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

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

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

How to find 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 1 is created, where edges are considered between groups of the previous layer.

The posterior probability became

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

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 7 gives the correct number of layers.

Adding a layer is done in 3 steps described in [34]:

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 $l = L$ [34]. 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 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 [35] 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 3 is $S_t = Ln\Omega$. This corresponds to an augmented graph generated via a non overlapping block model with $N' = \Sigma_r n_r > N$ nodes and the same 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. $\left(\!\binom{B}{N}\!\right)$ is the number of histograms with area N and B distinguishable bins. $B - 1$ can be used instead of B to avoid node with no group, in this case $d \in [1, B]$.

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

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

.

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

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

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

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

the global posterior as defined in [35] is

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

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

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

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

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

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.

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

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 [33] that this property is preserved and fully reflected in the final block structure.

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

$$h = 1 - \frac{H(C|K)}{H(C)} \tag{12}$$

defining the entropy

$$H(C|K) = \sum_{c \in \text{modellabels}, k \in \text{clusters}} \frac{n_{ck}}{N} Log\left(\frac{n_{ck}}{n_k}\right) \tag{13}$$

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. The completeness:

$$c = 1 - \frac{H(K|C)}{H(K)}, \tag{14}$$

$H(K|C)$ 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 [30], which is the harmonic average of the two:

$$2\frac{hc}{h+c}. \tag{15}$$

The product hc is equal to

$$\frac{(H(C) - H(C|K))(H(K) - H(K|C))}{H(K)H(C)}, \tag{16}$$

the sum $h + c$ is

$$\frac{H(K)(H(C) - H(C|K)) + H(C)(H(K) - H(K|C))}{H(K)H(C)}. \tag{17}$$

Expressing the conditional entropy

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

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 $2\frac{hc}{h+c}$ becomes

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

which is called V-measure and is actually the mutual information between $P(C)$ and $P(K)$ normalised 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.

Acknowledgements

Ringrazio bla bla bla...

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