

Improving Entropy Estimation and the Inference of Genetic Regulatory Networks

Jean Hausser

August 2006



Dépt Biosciences
Bâtiment Louis Pasteur
11, avenue Jean Capelle
F-69621 Villeurbanne Cedex

In candidacy for the degree of
Master of Engineering in Bioinformatics and Modelling
of the National Institute of Applied Sciences Lyon

Written under the supervision of :

Dr Korbinian Strimmer
Department of Statistics
University of Munich
Ludwigstrasse 33, D-80539 Munich, Germany

Abstract

This paper explores how entropy and other information theoretic quantities may be used to reverse-engineer genetic regulatory networks from repeated microarray data. The problem of differentiating genes that undergo direct coregulation from genes whose expression is similar because they belong to the same regulatory pathway is studied from a graphical modeling viewpoint. This leads to the criteria of conditional independence which can be evaluated by computing the conditional mutual information. The latter is completely characterized by the sum of the entropies of joint variables, underlining the need for an entropy estimator that is accurate even in low sampling conditions.

We introduce a new plug-in entropy estimator obtained from shrinking maximum likelihood multinomial proportions estimates to the maximum entropy target. We derive the closely related ZIPshrink and ZINBshrink entropy estimators which enhance the shrinkage estimator by first adjusting the shrinkage target depending on the fraction of structural zeros in the multinomial model. The fraction of structural zeros is estimated using a Zero-Inflated Poisson or Zero-Inflated Negative Binomial distribution to model the histogram of bin counts.

We compare these three new estimators to state of the art estimators. We show that they give acceptable estimates even in the low sampling regime and are as accurate as the best estimator available today while being 100 faster, making it more suitable for large scale computations. We then compare existing approximations of conditional independence networks such as 0-1 networks and a data processing inequality based approach. As a conclusion, we briefly consider limitations of the method as well as issues related to unobserved variables, causal inference and time series as opposed to steady state experiments.

Part I serves both as an introduction and a motivation. It presents the notions of conditional independence and explains why entropy estimation is critical to genetic regulatory network inference. Part II has the core results of this report : it reviews existing entropy estimators for the discrete case, introduces a new entropy estimator based on the statistical notion of shrinkage and compares their performance. Finally, part III compares data processing inequality based approach to genetic regulatory networks reverse-engineering with the so-called 0-1 networks approach. It also has considerations about limitations, pitfalls and possible extensions of the method.

Acknowledgments

Thanks go to Korbinian Strimmer for welcoming me in his group, for proposing me to work on a very educative, exciting topic, for introducing me to the shrinkage technique, for having been a caring mentor, for the pleasant work atmosphere and for his much appreciated financial support.

Merci à Samuel Soubeyrand de l'INRA Avignon pour avoir volontiers partagé ses connaissances sur les statistiques en m'en expliquant les notions qui m'étaient obscures. Merci aussi pour son aide à naviguer dans les méandres de LaTeX, natbib et autres BibTeX ainsi que pour les bananes au chocolat dont mes papilles se souviennent encore.

Thanks to Maria Persico, for the daily Italian lessons and the hours of deep, passionate debate related to soccer during the world cup.

I'd like to thank Vincent Suc, Susana & Andrés Eyheramendy, Stefan Pilz, Rainer Opgen-Rhein, Helen Gunnesch, Ole Daman and the others for all the good time at lunch, the concerts, the pleasure to speak Spanish once in a while, the outings in Germany and outside Germany, the (numerous) bbqs and other cheese related experiments.

Finally, thanks go to Hubert Charles for his interest in the present topic. Knowing that a biologist was interested in my work definitely helped me to keep the motivation all along my stay here.

About the Strimmer lab

This report was written during a 4 months stay with the statistical bioinformatics group of Korbinian Strimmer, at the Department of Statistics of the Ludwig-Maximilian University (Munich), from April, 24th to August, 31st 2006

Institut für Statistik The Department of Statistics has three chairs : the Seminar for Applied Stochastic Processes, the Seminar for Econometrics, Financial Econometrics and Statistics, and the Chair of statistics and applied statistics in economics, business administration and the social sciences. In addition to these chairs, there are currently three workgroups : the “statistical methods in sociology” group, the biostatistics unit, and the statistical bioinformatics group.

75 people work at the institute, including administrative staff. The main source of funding of the institute comes from the Sonderforschungsbereich SFB386, which is a federal project of the German government concerned with statistical analysis of discrete structures. In addition to that, the institute provides consulting services for companies and administrations through the StaBLab (Statistisches Beratungslabor) structure.

The Statistical Bioinformatics group The group was founded in January 2002 and is currently composed of one principal investigator, one post-doctoral fellow, one Ph. D. student, one visiting Ph. D. student and one M. Eng. student (me). Its research may be broadly summarized under the following three headings :

- Modeling-oriented bioinformatics and computational biology,
- Information theory and statistical inference, and
- Modeling, simulation, and reverse-engineering of complex networked systems.

Currently, the group devotes most of its energy to study models for functional genomics, proteomics, and systems biology. It is funded by the Emmy Noether excellence program of the Deutsche Forschungsgemeinschaft (DFG).

Contents

I	Motivation	6
II	Entropy estimation	11
1	Introduction	11
2	Estimation of the Shannon Entropy	11
2.1	Maximum likelihood multinomial proportions	12
2.2	Bayesian multinomial proportion estimators (Dirichlet prior)	12
2.3	The Nemenman-Shafee-Bialek (NSB) prior	14
2.4	Shrinkage estimate of multinomial proportions	15
2.4.1	Choosing the optimal shrinkage intensity	15
2.4.2	Shrinkage Estimator of Cell Probabilities	16
2.4.3	ZIPshrink and ZINBshrink	16
3	Estimating mutual information	19
3.1	Conditional mutual information	20
4	Results	20
4.1	Simulations setup	20
4.2	Entropy	22
5	Discussion	22
6	Future work	25
III	Conclusion	26

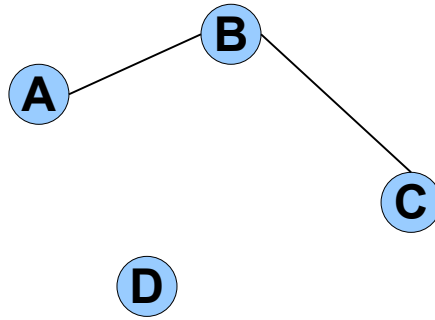


Figure 1: A simple graph made of 4 vertices A, B, C and D and 2 edges (A, B) and (B, C) .

Part I

Motivation

The microarray technology that was invented a decade ago opened new possibilities in genetics. This technology makes it possible to monitor the expression level of thousands of genes simultaneously (Schena et al., 1995). Therefore, it has become an essential technique in genetics and molecular biology.

As the microarray technique generates a large amount of data, computational and statistical techniques had to be developed to synthesize datasets and assist biologists at interpreting the experimental results. State of the art methods include expression patterns clustering (Eisen et al., 1998) which attempts to find clusters of functionally related genes by grouping together genes that have a similar expression profile. Other methods such as the Significance Analysis of Microarrays (SAM) described in Tusher et al. (2001) and MAANOVA (Wu et al., 2003) aim at telling which genes are significantly differently expressed. Within the Strimmer lab, previous research focused on subjects such as identifying periodically expressed genes (Wichert et al., 2004) and selecting emerging patterns of genes that allow improved classification accuracy in a supervised learning context (Boulesteix et al., 2003). Over the years, the microarray technology improved in reliability. And with costs falling due to the technique spreading out quickly, it makes more and more sense to consider inferring the genetic regulatory network from gene expression data. By inferring — or “reverse-engineering” as some authors would call it — the genetic regulatory network, we mean that for each pair of genes, we want to decide whether the data accounts for one of the two genes to regulate the other one.

So far, graphs have been the standard way of representing genetic regulatory networks. Graphs are made of vertices which represent genes, and edges that represent coregulation. Connecting two vertices with an edge means that one of the two corresponding genes regulates the other one.

Figure 1 shows a sample graph made of 4 vertices. Such a graph is called “undirected” because edges or not oriented, *i.e.* vertices are connected with lines instead of arrows. As vertices represent genes and edges symbolize regulation, just from looking at the graph, we can’t tell whether gene A regulates gene B or, on the opposite, if gene B regulates gene A . Ideally, we would like to replace edges with arrows in this graph, since it may be important for the biologist to know whether A regulates B or the other way around. But choosing a direction for the edges is a difficult problem which is related to causal inference and at this point, we will just ignore it. But we will briefly come back to it in section III and explain what makes it difficult. From now on, we will focus on inferring the *structure* of the network, without trying to determine the directions of the edges. This is a simpler problem than the previous one. Furthermore, if we had the structure of the network, we would be in a good position to start investigating the directions of the edges.

At this point, we need to define clearly what we mean by “gene regulation”. Looking at figure 1, we see that there is an edge between A and B . In our conventions, this means that either gene A regulates gene B , or that gene B regulates gene A . Let’s assume that gene A codes for a transcription factor of B , so

that A regulates B . Moreover, the graph has the edge (B, C) and we will assume that B codes for a protein that represses C by binding to its promoter, hence preventing the DNA Polymerase from transcribing C 's sequence. As far as the network structure — or equivalently, its topology, or connectivity — is concerned, we can sum up this situation as :

1. A regulates B (direct regulation through protein - protein interaction)
2. B regulates C (direct regulation through protein - DNA interaction)
3. therefore, A controls the expression of C through B (indirectly)

Here, we have to decide whether to include indirect control as edges in the graph — the (A, C) edge for instance — or only draw edges when there is a direct interaction — (A, B) and (B, C) in the previous example. In this work, we decide *not to represent indirect interaction* in the graph. Indeed, shall we represent them, the number of edges would be likely to be huge, making the resulting graph unreadable. Furthermore, since we're studying the gene regulatory network as a whole, it wouldn't be surprising that everything is connected to virtually everything by proxy. For instance, all genes depend on a DNA Polymerase to be expressed, so if we drew edges corresponding to indirect interactions, we would end up with a graph where each vertex is connected to all the other vertices. Such a graph doesn't carry any useful information to the experimentalist. Therefore, we decide that edges represent "regulation" in the strong, direct sense.

Problem statement and previous work The direct vs. indirect regulation discussion naturally brings us to the central problem of genetic regulatory network inference : we need a criteria to discriminate pairs of genes that are in direct interaction from pairs whose expression profiles are similar because they belong to the same regulatory pathway and therefore undergo indirect regulation. The direct interaction we are interested in might take the form of a protein - DNA binding, of a miRNA-mediated regulation, or any other form of interaction. The point is that we need to be able to detect direct regulation specifically.

In this work, we follow an information theoretic approach to the problem of direct regulation discrimination. The mathematical background is standard graphical modelling theory and has been developed by Whittaker (1990), Lauritzen, Edwards and other authors. The underlying idea is the following. Consider the regulatory network represented by figure 1. Let's say we design a microarray to measure the expression level of the 4 genes A, B, C and D under various experimental conditions, e.g. nutritional stress, heat shock, UV exposure, *etc.* We further make the same assumption as before, that is that A is an activator for B , which in turns represses C . Because, A controls C through B rather than directly, we would expect the expression profiles of A and B to be more similar than those of A and C . Similarly, the expression profiles of A and C should differ more than those of B and C . At this point, Eisen et al. (1998) steps in and suggests to build a dendrogram that uses correlation as a metric and pairs together genes whose expression profiles are the closest. But instead of thinking in terms of distance, we could think in terms of information flow. In this case, assuming the network of figure 1, we would expect the expression profile of A to provide us with more information about the expression profile of B than about C 's profile. In other words, given the profile of A , it is harder to predict C 's profile than B 's profile. The ARACNE algorithm (Margolin et al., 2006) uses just this idea to remove indirect interactions from a preexisting network : given expression profiles for n genes, it considers all possible gene triplets A, B, C and removes the edge between the least predictable pair of genes among the pairs $(A, B), (B, C)$ and (A, C) .

ARACNE can only guarantee to recover the underlying regulatory network without error if both of the two conditions following conditions are satisfied :

1. the network has the topology of a tree
2. there are only pairwise genetic interactions in the network. That is, N-way interactions with $N > 2$ are negligible.

In contrast to that, Whittaker's graphical model approach needs only the second assumption to be satisfied. Another condition with both methods is that we are able to determine "predictability" in a reliable fashion, but we'll come back to it later.

The network having the topology of a tree means that if we choose two genes arbitrarily in the network, there is only one possible path from one gene to the other. That is, there are no loops in the network. Given what is known about genetic regulatory networks, this has to be wrong. As for N-way interactions, these occur in transcription complexes for instance, when N regulatory proteins bind together with the DNA polymerase and the DNA, thereby regulating gene transcription. We therefore expect these N-way interactions to be fundamental in genetic regulatory networks, but again, we will let the question aside until section III and assume that there are only pairwise interactions. In practise though, ARACNE seems to work better than other information theory-based approaches to genetic regulatory networks reverse-engineering such as Butte and Kohane (2000) and general purpose Bayesian networks.

Graphical modelling and mutual independence We now turn to graphical modeling in order to answer the following question : how can information theoretic quantities help us to infer the structure of the regulatory network given microarray expression profiles ?

Let X, Y and Z be three variables taking real values x, y and z with probability density distribution $f_X(x), f_Y(y), f_Z(z)$. Let further f_{XY} be the probability density function (pdf) of the joint variable XY . Information theory (MacKay, 2003) defines *mutual information* $I(X; Y)$ between X and Y as

$$I(X; Y) = \int_{\mathbb{R}^2} f_{XY}(x, y) \log \frac{f_{XY}(x, y)}{f_X(x)f_Y(y)} dx dy \quad (1)$$

Mutual information quantifies the stochastic dependence — or the degree of predictability — between two variables. Note that it divides f_{XY} (in the numerator) by $f_X f_Y$ (in the denominator). In other words, it computes the ratio between the joint distribution f_{XY} of (X, Y) and the product of the marginals $f_X f_Y$. Now, when the joint distribution equals the marginals, that is when $\forall x, y : f_{XY}(x, y) = f_X(x)f_Y(y)$, we have stochastic independence between X and Y . It can be shown using Jensen's inequality that mutual information is positive definite, *i.e.* $I(X; Y) \geq 0$. Furthermore, it is possible to prove that

$$I(X; Y) = 0 \Leftrightarrow f_{XY}(x, y) = f_X(x)f_Y(y)$$

In plain English, mutual information is 0 if and only if $f_{XY}(x, y) = f_X(x)f_Y(y)$, that is if X and Y are stochastically independent. The bigger $I(X; Y)$, the more dependent X and Y . Moreover, it can be shown that $I(X; Y)$ generalizes Pearson's χ^2 test for independence in two-way contingency tables to all distributions.

Butte and Kohane (2000) first used mutual information for genetic regulatory networks inference in the following way : for all pairs of genes (G_i, G_j) , they computed $I(G_i, G_j)$. Whenever $I(G_i, G_j)$ was over a given threshold, they would connect the corresponding vertices G_i and G_j with an edge. However, this approach doesn't take into account the direct vs. indirect regulation problem ! Being part of the same system, many pairs of genes tend to have information in common. Therefore, the network resulting from this approach has many "false positives" in the sense that it draws many edges that actually correspond to indirect interactions.

Fortunately, graphical modelling provides an answer to this problem. Going back to figure 1, it wouldn't be surprising that $I(A; B)$, $I(B; C)$ and $I(A; C)$ are all > 0 since A, B and C belong to the same regulatory pathway and hence have to be stochastically related in some way. But let's see what happens if we compute $I(A; C|B)$, that is the mutual information of A and C conditioned on B defined as

$$I(X; Y|Z) = \int_{\mathbb{R}^3} f_{XY|Z}(x, y, z) \log \frac{f_{XY|Z}(x, y, z)}{f_{X|Z}(x, z)f_{Y|Z}(y, z)} dx dy dz \quad (2)$$

$$= \int_{\mathbb{R}^3} f_{XY|Z}(x, y, z) \log \frac{f_{XYZ}(x, y, z)}{f_{X|Z}(x, z)f_{Y|Z}(y, z)f_Z(z)} dx dy dz \quad (3)$$

where $f_{X|Z}$ and $f_{Y|Z}$ are the probability density functions of $X|Z$ and $Y|Z$ respectively. When we compute $I(X; Y|Z)$, we actually compare f_{XYZ} to $f_{X|Z}f_{Y|Z}f_Z$. Figure 2 shows the possible interactions captured by f_{XYZ} and $f_{X|Z}f_{Y|Z}f_Z$.

Now, let us compute $I(A; C|B)$. If we replace X, Y and Z with A, C and B , figure 2 tells us that when we compute $I(A; C|B)$, we actually compare the f_{ABC} and $f_{A|B}f_{C|B}f_B$ probability density functions. The

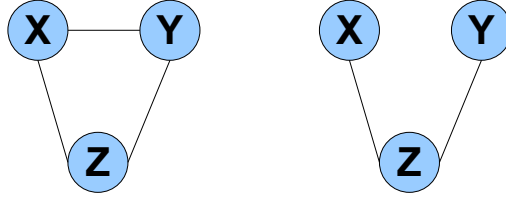


Figure 2: Comparing f_{XYZ} and $f_{X|Z}f_{Y|Z}f_Z$. Left: possible pairwise interaction captured by f_{XYZ} . Each node is connected to all the other nodes. f_{XYZ} actually also accounts for a 3-way interaction not shown on the picture since we earlier hypothesized that there were no 3-way interactions. Right: possible 2-way interactions accounted for by $f_{X|Z}f_{Y|Z}f_Z$. X is connected to Z which in turns is connected to Y . But there is no edge connecting X and Y . Unlike f_{XYZ} , $f_{X|Z}f_{Y|Z}f_Z$ can't account for any three-way interaction, so the picture faithfully represents all the possible interactions captured by the distribution.

key insight here is that the two distributions can only be equal — or equivalently $I(A; C|B) = 0$ — if there is no edge between A and C , *i.e.* there are no direct interactions between A and C . When this happens, we say that A and C are *conditionally independent* given B .

Hence, conditional independence enables us to differentiate direct interactions from indirect interactions, which is just the problem we had to solve in order to be able to reverse-engineer genetic regulatory networks !

However, we are not quite done yet because the conditional independence criteria may work well for three variables, but what happens when we have n variables (genes), with n possibly being several thousands ? Let us first look at figure 1 again and ask what happens if we compute $I(A; C|B, D)$. From figure 1, we see that D is not connected to any other gene of the network and is thus unrelated to A , B and C . So, provided A and C are conditional independent given B , we would intuitively expect them to be independent given B and D , *i.e.* $I(A; C|B, D) = 0$. Whittaker (1990) proves that this is actually the case. Furthermore, it provides the theory to generalize conditional independence to an arbitrary number of variables n . The outcome of the theory is that, provided we observe all variables X_1, X_2, \dots, X_n in the system, we can test for direct interaction between X_1 and X_2 by computing $I(X_1; X_2|X_3, \dots, X_n)$. In other words, one computes the mutual information between the two variables of interest given all the others. When this quantity is zero, there are no direct interactions but only indirect ones through other variables, that is control by proxy. Note that testing whether $I(X_1; X_2|\text{rest}) = 0$ is the graphical theoretic equivalent to looking at the F-statistic of a two-way interaction effect when selecting a linear regression model.

So what's the catch ? The mathematical theory of graphical modeling is well understood. But in practice, estimating the conditional mutual information is not a trivial problem. For a pair of genes (i, j), the critical quantity is $I(X_i; X_j|\text{rest})$ as exposed in the previous paragraph. But how do we actually compute it ? One could use equation 2 or 3 directly, but in that case, we need to estimate the various probability density functions since these are usually unknown.

Another way of computing $I(X_i; X_j|\text{rest})$ is from the entropies of the underlying variables. The entropy H of a variable X with pdf f_X is defined as

$$H(X) = \int f_X(x) \log \frac{1}{f_X(x)} dx$$

and is related to mutual information through the following equality :

$$I(X; Y) = H(X) + H(Y) - H(X, Y)$$

Finally, the conditional mutual information of X and Y given Z_1, Z_2, \dots, Z_n can be written as

$$I(X; Y|Z_1, \dots, Z_n) = H(X, Z_1, \dots, Z_n) + H(Y, Z_1, \dots, Z_n) - H(X, Y, Z_1, \dots, Z_n) - H(Z_1, \dots, Z_n) \quad (4)$$

The calculation leading to this result is trivial and is given in section 3.1. Our goal here is not to discuss the meaning of the equation but just to make clear that $I(X_i; X_j | \text{rest})$ can be computed from the entropies of the underlying variables. That is why it is critical to be able to estimate entropy in a reliable way.

Unfortunately, this is not an easy problem, mainly because the pdf f is not known and has to be estimated. Kernel based density estimation methods are expected to perform poorly in this case because we usually have little data compared to the number of genes. One way out is to choose the best-fitting f from a narrow class of distributions, *e.g.* Gaussians. We then win nothing from choosing to use mutual information over the correlation coefficient, but the problem becomes simpler and is known as Gaussian Graphical Modeling (GGM). Schäfer and Strimmer (2005) propose a method to infer the genetic regulatory network by modeling expression levels with a multivariate Gaussian distribution and which works in low sampling conditions. However, modeling gene expression with a Gaussian distribution doesn't allow non-linear dependencies such as saturation effects to be taken into account. Alternatively, we could also start with a larger class of distributions but it then becomes difficult to fit a distribution because the data is highly dimensional and we may hence have to estimate a lot of parameters with only little data. In both situations, we may then end up having to make some strong assumption about the probability density function.

The reader may be surprised to read that we mention a *lack* of data, especially after we said in the first paragraph that microarray experiments tended to generate a lot of data : aren't there several thousands of genes per microarray ? Actually, the problem comes from measuring too many genes compared to the number of experimental conditions. Indeed, equation 4 tells us that in order to compute conditional mutual information, we need to estimate the entropy of the joint distribution of all genes at some point. The joint distribution of all n genes is supported by a n -dimensional space. n being the number of genes we include in the study, we would like it to be as large as possible, leading to a pathological under-sampling situation due to the so-called "Curse of Dimensionality". To fit such a high-dimensional function, we would normally need much more than n points, which in the case of genetic regulatory networks inference are experimental conditions !

To address this problem, instead of trying to estimate the joint pdf, we focus our attention on obtaining an estimator for its entropy that works even if we are in low sampling conditions, *i.e.* when the number of microarray experiments is much smaller than the number of genes. To further reduce the complexity of the estimation problem, we discretize expression levels in p levels — or bins in the multinomial distribution terminology. This leads us to a multinomial model, where we only have to determine multinomial proportions. Section II introduces a new entropy estimator for that case and compares it to other entropy estimators.

To conclude this introduction, we will just summarize under what assumptions the conditional mutual information based graphical modeling approach will be guaranteed to fully recover the genetic regulatory network without errors :

1. there are only pairwise interactions between genes, that is N -way genes interactions with $N > 2$ are negligible
2. we are able to estimate the conditional mutual information without error
3. we observe all the genes from the cell on the microarrays, *i.e.* there are no missing data

If all three conditions are satisfied, the inferred network is guaranteed to reflect the genetic regulatory interactions that occur under the experimental conditions.

Part II

Entropy estimation

1 Introduction

Information theory has proved to be very useful in many fields : neuroscience (Makeig et al., 1997; Strong et al., 1998), bioinformatics (Beerenwinkel et al., 2002), statistics (Whittaker, 1990), medical image analysis (Meyer et al., 1997) and so on. One of the reasons that accounts for its generalized use is that it provides a framework to quantify information in a very general sense, through quantities such as the Shannon entropy or the mutual information (MacKay, 2003). The latter quantifies stochastic independence, in contrast to correlation which only quantifies orthogonality, that is linear independence. Stochastic independence and uncorrelation are equivalent only with multivariate Gaussian distributions, while mutual information is generalized measure of stochastic dependence and is valid with any distribution.

In cases where one has many data points, it is quite easy to estimate entropy reliably. However, in under-sampled problems, the estimation problem becomes quite challenging. Low sampling conditions can't always be avoided and may be intrinsic to the scientific question or to the experimental method. For instance, let's say we want to compute the entropy of a language from a book written in that language. The language is necessarily made of many more words than a book may contain, which leads to under-sampling. Or, if one is interested in computing the entropy of high-dimensional datasets, under-sampling may occur rapidly through the curse of dimensionality.

The entropy and mutual information estimation problems have a long history. Beirlant et al. (1997) reviews non-parametric entropy estimation methods. Despite the theory being more than 50 years old, the estimation problem in low sampling conditions is still an open problem. Here we present a new type of estimator that is obtained by shrinking the maximum likelihood estimates towards the maximal entropy target. This improves the precision of the maximum likelihood estimator by reducing its bias. The shrinkage intensity is analytically chosen so as to minimize the mean square error (MSE) of the resulting estimator.

2 Estimation of the Shannon Entropy

Let's consider an experiment with p possible outcomes B_1, B_2, \dots, B_p . We define θ_i as the probability for B_i to occur, for $i = 1, 2, \dots, p$. Taken together, the θ_i form a vector $\theta = (\theta_1, \theta_2, \dots, \theta_p)$. Since B_1, B_2, \dots, B_p are the only the possible outcomes and that one of them has to occur every time we perform the experiment, we require $\sum_{i=1}^p \theta_i = 1$.

The Shannon entropy of the resulting discrete distribution is defined as

$$H(\theta) = - \sum_{i=1}^p \theta_i \log(\theta_i)$$

If we renew this experiment n times, we will observe that B_i occurred y_i times. Equivalently, we can look at it as an experiment where n objects are spread on p bins, the probability for each object to land in bin B_i being θ_i . At the end, we count how many objects y_i landed in each bin B_i , which gives us the count vector $y = (y_1, y_2, \dots, y_p)$.

In this case, counts y can be modeled using a multinomial distribution with parameters $\theta_1, \theta_2, \dots, \theta_p$ and $\sum_{i=1}^p \theta_i = 1$. The probability of observing a given vector of counts y given the bin probability vector θ is thus

$$\text{Prob}(y; \theta) = \frac{n!}{\prod_{i=1}^p y_i!} \prod_{i=1}^p \theta_i^{y_i} \quad (5)$$

provided $\sum_{i=1}^p y_i = n$.

As far as parameter estimation is concerned, fitting a multinomial(θ) distribution to the observed counts y consists in estimating p parameters from n observation.

2.1 Maximum likelihood multinomial proportions

At present time, entropy estimation from maximum likelihood multinomial proportions estimates seems to be the most popular method in the literature (see Meyer et al., 1997; Ong and Chen, 1999; Beerenwinkel et al., 2002, for instance). It is sometimes to be found under the name “histogram technique” (Butte and Kohane, 2000). It is the most intuitive as well. One accumulates bin counts and then, for each bin, one determines an estimate of the multinomial proportion by dividing the number of counts that fell into the bin by the total number of counts :

$$\hat{\theta}_i^{\text{ML}} = \frac{y_i}{n}$$

It can be shown that this corresponds to the maximum likelihood (ML) estimate of the multinomial proportions. In other words, the empirical frequencies are the ML estimate. The variance of the ML estimator of multinomial proportions is $\text{Var}(\hat{\theta}_i^{\text{ML}}) = \frac{\theta_i(1-\theta_i)}{n}$. Furthermore, the estimator can be shown to be unbiased : $\text{Bias}(\hat{\theta}_i^{\text{ML}}) = 0$ as $E(\hat{\theta}_i^{\text{ML}}) = \theta_i$.

For large sample sizes this estimator enjoys many optimality properties because ML is the optimal method to summarize the data and it is also a good method for inferring the true model (e.g. Efron, 1982), but only for large n/p . Likelihood inference is only an asymptotic theory : for moderate to small ratios n/p , the ML estimator is suboptimal and in the case of multinomial proportions estimation, it may not deal well with zero counts. One can then obtain improved estimators through regularization and by appealing to the Bayesian framework for instance. Another strategy for obtaining a better estimator is to reduce the Mean Squared Error (MSE). In the case of the multinomial proportions estimation problem, ML estimates have no bias but large variance. As a result, the MSE is entirely due to the variance. Thus, the overall accuracy can be greatly improved by deliberately introducing some bias in the ML estimator (Stein, 1956; Efron, 1975).

ML entropy estimate \hat{H}^{ML} is obtained by plugging the ML multinomial proportions estimates in the definition of the Shannon entropy :

$$\hat{H}^{\text{ML}}(y) = - \sum_{i=1}^p \hat{\theta}_i^{\text{ML}} \log \hat{\theta}_i^{\text{ML}}$$

It is well known that ML estimates have a strong negative bias when probabilities are evenly spread out, that is it severely underestimates the true entropy (Paninski, 2003).

This is straightforward to understand : ML sets cell probability θ_i to 0 if no sample land in the B_i bin ($y_i = 0$). In the case of under-sampled multinomial distributions where cell probabilities θ are spread out evenly enough, this results in setting cell probabilities to 0 while they are not. As a result, the estimated distribution looks less informative than it really is. The large MSE of $\hat{\theta}_i^{\text{ML}}$ then propagates to the estimate of H . But this time, the error is mostly in the bias (note that if the estimator for θ_i is unbiased, then $\theta_i \log \theta_i$ cannot be unbiased).

To tackle this problem, a posteriori bias corrections for the entropy computed from ML count probabilities have been suggested (for instance Miller, 1955; Paninski, 2003). An alternative route is to try to estimate the underlying distribution as precisely as possible, and only then apply the above definition of H . Using a biased estimator of θ with minimum MSE would lead to a plug-in entropy estimator that also has to be *less* biased.

2.2 Bayesian multinomial proportion estimators (Dirichlet prior)

Obtaining improved regularized estimates with better MSE properties than the ML estimator is standard Bayesian folklore. The usual approach (for instance, see Gelman et al., 2004) is to assume a Dirichlet

prior with parameters a_1, a_2, \dots, a_p :

$$f(\theta; a_1, \dots, a_p) = \frac{\Gamma\left(\sum_{i=1}^p a_i\right)}{\prod_{i=1}^p \Gamma(a_i)} \prod_{i=1}^p \theta_i^{a_i-1} \delta\left(1 - \sum_{i=1}^p \theta_i\right) \quad (6)$$

where δ , defined as

$$\delta(x) = \begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{otherwise} \end{cases}$$

ensures that θ s sum up to 1.

The Dirichlet distribution is the conjugate prior to the multinomial distribution, so the posterior is also Dirichlet-distributed with mean

$$\hat{\theta}_i^{\text{Bayes}} = \frac{y_i + a_i}{n + m} \quad (7)$$

where $m = \sum_{i=1}^p a_i$. Note that using a Dirichlet prior with parameters a_i is equivalent to adding $a_i \geq 0$ “pseudo-counts” to each of the p cells for regularization. With the prior, we actually provide the estimator with the information that a_i counts have been observed in previous experiments. From that viewpoint, m becomes the *a priori* sample size.

Common choices for a_i include $a_i = \frac{1}{p}$, $a_i = \frac{1}{2}$, $a_i = 1$, and $a_i = \frac{\sqrt{n}}{p}$ for $i = 1, \dots, p$ (Fienberg and Holland, 1973; Geisser, 1984; Santner and Duffy, 1989). The above assignments of a_i are all plausible as non-informative priors. For instance, setting $a_i = 1$ for $i = 1, \dots, p$ corresponds to assuming a prior distribution of multinomial proportions which peaks when these proportions are uniform — the classic Laplace solution — and $a_i = 1/2$ is equivalent to using Jeffreys’s prior (Jeffreys, 1946). $a_i = 0$ gives the ML solution, and $a_i = \frac{\sqrt{n}}{p}$ provides the minimax estimator. Most of them have been used to estimate entropy : $a_i = 0$ in Meyer et al. (1997); Ong and Chen (1999); Beerenwinkel et al. (2002), $a_i = 1$ in Chiu and Kolodziejczak (1991), $a_i = \frac{1}{2}$ in Krichevsky and Trofimov (1981), and $\frac{1}{p}$ in Schürmann and Grassberger (1996).

Note that the binomial case ($p = 2$) is included in the multinomial-Dirichlet framework : if we take the multinomial-Dirichlet model and set $p = 2$, we obtain the common binomial-beta model.

Since we have no *a priori* reason to use different $a_i, i = 1, \dots, p$, from now on, we will assume $a_1 = \dots = a_p = a$ and refer to the corresponding Dirichlet distribution as to “Dirichlet(a)”. Under this assumption, the multinomial proportions estimator can be written

$$\hat{\theta}_i^{\text{Bayes}} = \frac{y_i + a}{n + pa}$$

which leads to the following entropy estimator :

$$\hat{H}^{\text{Bayes}} = - \sum_{i=1}^p \hat{\theta}_i^{\text{Bayes}} \log(\hat{\theta}_i^{\text{Bayes}})$$

We call this class of estimators Dirichlet prior based entropy estimators. Unfortunately, if we try to see how well they perform, we get very unpleasant results.

Figure 3 provides a striking picture : Dirichlet prior based entropy estimators perform very well in some cases but can also completely break down depending on the characteristics of the multinomial distribution we sample from. The cause for this behavior was clearly pointed out by Nemenman et al. (2002) : multinomials whose bins probabilities are sampled from a Dirichlet(a) tend to have very closely distributed entropies, with variance vanishing with growing p . In other words, given a , the distribution of the entropy of multinomial whose cell probabilities are sampled from a Dirichlet(a) distribution is very spiky (see figure 5). Furthermore, the authors noticed that the average entropy of multinomials with cell probabilities sampled from Dirichlet(a) distributions moves from 0 to the maximum $\log_2(p)$ as a moves from 0 to ∞ . Finally, their work showed that the prior tended to dominate entropy estimates even after the data had been observed. This behavior can be observed in figure 3 : for the Dirichlet(15) case, a poor choice of a such as $a = \frac{1}{p}$ causes a strong and persisting bias in the estimated entropy until $n/p \geq 5$.

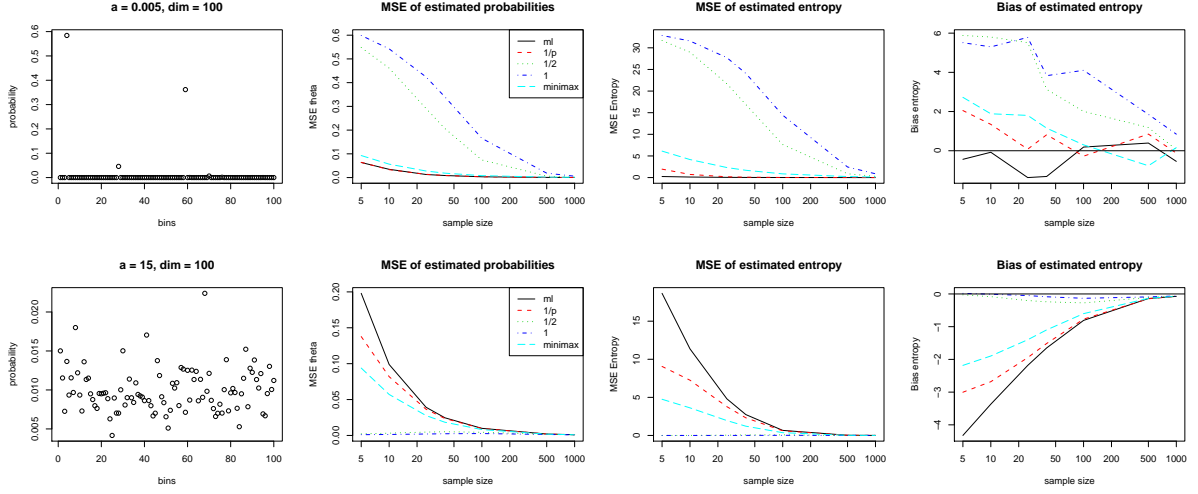


Figure 3: Comparing the performance of the maximum likelihood ($a = 0$), Schürmann & Grassberger ($a = \frac{1}{p}$), Jeffrey ($a = \frac{1}{2}$), Laplace ($a = 1$) and minimax ($a = \frac{\sqrt{n}}{p}$) Dirichlet prior based multinomial proportion estimators for the 100 bins case, as a function of the number of samples we take from the distribution. Note that for the entropy H , we have $0 \leq H \leq \log_2(p)$, with $\log_2(100) \simeq 6.64$. 1500 repetitions were performed for each sampling condition and each estimator. First row : 100-dimensional cell probability vectors were sampled from a Dirichlet(a) distribution with $a = .005$. A typical multinomial probability distribution is shown in the first column. The Mean Square Error (MSE) of estimated multinomial proportions and entropy as well as the bias of each of the 5 estimators are plotted in the second, third and fourth column. Second row : same as the first row, except 100-dimensional cell probability vectors were sampled from a Dirichlet(a) distribution with $a = 15$ resulting in multinomials whose typical proportions are shown in the first column. Note how the maximum likelihood, the $1/p$ and the minimax estimators perform well when $a = .005$ (first row) but break down when $a = 15$ (second row). Swapping the a s, the same can be said about the Laplace and $1/2$ estimators. More details about the simulations are given in section 4.

2.3 The Nemenman-Shafee-Bialek (NSB) prior

This suggests a way of removing the bias by building a prior on multinomial proportions θ — in other words, a probability density function $f_{\text{NSB}}(\theta)$ — that is suitable for the problem of entropy estimation.

Remember we are interested in the entropy of a multinomial distribution with p cells and cell proportions $\theta = (\theta_1, \theta_2, \dots, \theta_p)$. With the entropy estimator based on Bayesian estimation of multinomial proportion we had

$$\hat{H}^{\text{Bayes}} = - \sum_{i=1}^p \hat{\theta}_i^{\text{Bayes}} \log(\hat{\theta}_i^{\text{Bayes}})$$

where $\hat{\theta}_i^{\text{Bayes}}$ are obtained by computing the maximum a posteriori (MAP) estimate of

$$\text{Prob}(\theta | y) = \frac{\text{Prob}(y | \theta) \text{Prob}(\theta)}{\text{Prob}(y)}$$

having

$$\text{Prob}(y|\theta) = \text{multinom}(y; \theta)$$

$$\text{Prob}(\theta) = \text{Dirichlet}(\theta; a)$$

The problem with this model is that if we have $\theta \sim \text{dirichlet}(a)$, then $H(\theta) \simeq g(a)$ with g some function defined on positive reals (see figure 5). Moreover, if we additionally have $y \sim \text{mutlinom}(\theta)$, in low-sampling conditions of y (small n), it remains true that $\hat{H}^{\text{Bayes}}(y) \simeq g(a)$. To put it in a nutshell, choosing an a almost completely determines the value of \hat{H}^{Bayes} .

To solve the problem, the NSB approach replaces the troublesome Dirichlet prior with a different

prior $f_{\text{NSB}}(\theta)$ that is constructed so that the distribution of $H(\theta)$ is close to being uniform. The $f_{\text{NSB}}(\theta)$ prior attempts to spread the probability density of $H(\theta)$ on the whole $[0, \log_2(p)]$ interval with near uniformity instead of the distribution being spiky like in the Dirichlet case. Details about the NSB method can be found in Nemenman et al. (2002) and Nemenman et al. (2004).

According to the authors, the resulting estimator beats by far the entropy estimators which are based on Dirichlet priors. This was confirmed by our simulations (see section 4). However, the method suffers two drawbacks. First of all, it is suboptimal when data come from a smooth distributions, such as a Dirichlet(a) with $a > 10$. Second, using the default precision setting, it is time-consuming to compute in spite of being the only method implemented in C++ among all the methods benchmarked in this paper (see figure 5). Being slow can be harmful if one wants to make large-scale entropy estimations. In a machine learning context, it also makes it more difficult to run large-scale simulations and hence assess the performance of an algorithm that uses this method.

2.4 Shrinkage estimate of multinomial proportions

Another way of removing the bias is to make a wise decision when picking a , which is the underlying idea of the shrinkage estimator. Consider the following equation :

$$\hat{\theta}_i^{\text{shrink}} = \lambda t_i + (1 - \lambda) u_i = \lambda \frac{a_i}{m} + (1 - \lambda) \frac{y_i}{n} \quad (8)$$

where $i \in \{1, \dots, p\}$ and $\lambda \in [0, 1]$ is the shrinkage intensity, where $\lambda = 0$ means no shrinkage and $\lambda = 1$ indicates full shrinkage. u_i is the unregularized ML estimate, $t_i = \frac{a_i}{m}$ is the shrinkage target and $m = \sum_{i=1}^p a_i$. For equal a_i ($a = a_1 = a_2 = \dots = a_p$), the target reduces to $t_i = \frac{1}{p}$:

$$\hat{\theta}_i^{\text{shrink}} = \lambda \frac{1}{p} + (1 - \lambda) \frac{y_i}{n} \quad (9)$$

Equation 8 combines two estimators t_i and u_i in a weighted average. The target t has no variance but high bias, whereas the unregularized estimate u has large variance but is unbiased. The advantage of shrinking u towards t is that the resulting estimator outperforms either of the individual estimates both in terms of accuracy and and statistical efficiency.

There is an intrinsic connection with the Dirichlet prior based estimator : if we set $\lambda = \frac{m}{n+m}$ then $\hat{\theta}_i^{\text{shrink}} = \hat{\theta}_i^{\text{Bayes}}$. Hence, there is a one-to-one correspondence between λ and a . For any choice of pseudo-counts a in the Dirichlet(a) prior, there is an associated shrinkage intensity, and conversely :

$$a = \frac{n}{p} \left(\frac{1 - \lambda}{\lambda} \right) \quad (10)$$

2.4.1 Choosing the optimal shrinkage intensity

The primary drawback of the Dirichlet prior based approach is that there is no obvious way to select a reasonable prior, may it be formalized in terms of pseudo-counts (see equation 6) or in terms of shrinkage intensity λ (see equation 8). One way out is to avoid making a specific choice of pseudo-counts a by considering a weighted average such as

$$\hat{\theta} = \int_{b>0} \pi(b) \frac{y_i + b}{n + pb} db$$

where $\pi(b)$ is some suitable distribution for the common pseudo-count b . In the Bayesian interpretation, this corresponds to using a mixture of Dirichlets as prior. Leonhard (1977) recommends using $\pi(b) = \frac{n-1}{\log(n)(b+1)(b+n)}$.

Another idea is to estimate the shrinkage intensity analytically from the data. Here we suggest an approach that makes no assumption about the prior distribution, only about the existence of the first two moments of the sampling distribution of the unconstrained estimator.

A key question in this procedure is how to select an optimal value for the shrinkage parameter. Let

us consider explicitly minimizing a risk function $R(\lambda)$

$$R(\lambda) = E \{L(\lambda)\} = E \left\{ \sum_{i=1}^p (\hat{\theta}_i - \theta_i)^2 \right\} \quad (11)$$

The key insight is that this can be done analytically. It can be shown (Schäfer and Strimmer, 2005; Gruber, 1998) that the value of λ which minimizes the risk is

$$\lambda^* = \frac{\sum_{i=1}^p \text{Var}(u_i) - \text{Cov}(u_i, t_i) + \text{Bias}(u_i) E(u_i - t_i)}{\sum_{i=1}^p E \{(u_i - t_i)^2\}} \quad (12)$$

Note that, given λ^* , appealing to equation 10, we could compute the equivalent a^* parameter for the Dirichlet prior approach.

2.4.2 Shrinkage Estimator of Cell Probabilities

We can now directly apply the previous ideas to the problem of estimating bin probabilities θ . Let $u_i = \frac{y_i}{n}$ be our unconstrained estimate (the ML estimate in the present case). Let further $t_i = \frac{1}{p}$ be our target, which corresponds to the θ for which the entropy is maximal.

As $\text{Bias}(u_i) = 0$ and plugging-in the unbiased estimator $\widehat{\text{Var}}(u_i) = \frac{u_i(1-u_i)}{n-1}$ in equation 12, we obtain the following shrinkage estimator :

$$\hat{\theta}_i^{\text{shrink}} = \begin{cases} \hat{\lambda}^* \frac{1}{p} + (1 - \hat{\lambda}^*) \frac{y_i}{p} & \text{if } \hat{\lambda}^* < 1 \\ \frac{1}{p} & \text{if } \hat{\lambda}^* \geq 1 \end{cases}$$

with

$$\hat{\lambda}^* = \frac{\sum_{i=1}^p \widehat{\text{Var}}(u_i)}{\sum_{i=1}^p (t_i - u_i)^2} = \frac{p(n^2 - w)}{(n-1)(pw - n^2)}. \quad (13)$$

and $w = \sum_{i=1}^p y_i^2$. For $p = 2$ (binomial case) the above equations reduce to the following:

$$\hat{\theta}_i^{\text{shrink}} = \begin{cases} \frac{y_i}{n-1} \frac{2y_i - n - 1}{2y_i - n} & \text{if } \hat{\lambda}^* < 1 \\ \frac{1}{2} & \text{if } \hat{\lambda}^* \geq 1 \end{cases}$$

with

$$\hat{\lambda}^* = \frac{4y_1 y_2}{(n-1)(y_1 - y_2)^2}$$

This estimator is consistent, that is the more data is available, the closer the estimator gets to the true value. Furthermore, for finite sample size, each estimate $\hat{\theta}_i^{\text{shrink}} > 0$. As we will see below, it is more efficient than the competing estimators in the sense that the same sample size its MSE is smallest. Finally, note that in the proposed analytic shrinkage estimate, the underlying distribution is moved towards the distribution of maximal entropy, which should specifically removes the bias in H.

2.4.3 ZIPshrink and ZINBshrink

As discussed in the previous section, using the shrinked estimator of multinomial proportion is expected to remove the bias on the resulting entropy estimator even when the distribution is under-sampled. However, simulations tend to indicate that the shrinkage fails when many cells have a small or a 0 probability.

The ZIPshrink and ZINBshrink estimators attempt to fix this undesired behavior by first estimating what is the fraction of the cells associated with a near zero probability. ZIPshrink and ZINBshrink take into account the fact that some cells may have no counts because we under-sampled the distribution while others may have no counts because their probability is close to 0. In other words, ZIPshrink and ZINBshrink attempt to decide what fraction q of the cells have 0 counts due to their underlying probability being too close to 0, not because of the low-sampling conditions. To do that, ZIPshrink models the number of cells having k counts using a Zero-Inflated Poisson (ZIP) distribution.

The Zero-Inflated Poisson distribution can be understood as a mixture of a zero-valued distribution and a Poisson distribution. The Poisson distribution is a standard model when it comes to analyzing count data. So ZIP models are to be found in cases where the data exhibits a Poisson behavior but also features an excessive amount of zero counts. For instance, Lambert (1992) applies ZIP-regression to the quality-control field, modelling the number of defective items from a manufacturing equipment that could either be in a state where it would produce items with no defects at all, or in a state where the number of these defects would be Poisson-distributed.

Let X be a ZIP-distributed random variable, that is,

$$\begin{cases} X = 0 & \text{with probability } q \\ X \sim \text{Poisson}(\lambda) & \text{with probability } 1 - q \end{cases}$$

If we call $f_X(k; q, \lambda)$ the probability density function of X , we have

$$f_X(k; q, \lambda) = \begin{cases} q + (1 - q)e^{-\lambda} & \text{if } k = 0 \\ (1 - q)\frac{e^{-\lambda}\lambda^k}{k!} & \text{if } k > 0 \end{cases}$$

Furthermore, if x_1, \dots, x_n are independent and identically-distributed (iid) realizations of X , the log-likelihood of the data is

$$\mathcal{L}(q, \lambda; x_1, \dots, x_n) = c_0 \log(q + (1 - q)e^{-\lambda}) + c_1 \log(1 - q) - c_1 \lambda + \sum_i x_i \log(\lambda) - \sum_i \log \Gamma(x_i + 1)$$

where c_0 is the number of x_i that are 0 and c_1 is the number of x_i with $x_i > 0$.

The model is the following. Let's assume that a fraction q of the p cells have a 0 probability, that is $\theta_i = 0$. After we sample n times from the multinomial(θ) distribution, there will be a number n_0 of cells that end up with 0 counts. If we consider one of these cells y_i , there can be two reasons for it to be empty. Either $\theta_i = 0$, that is the distribution parameters prevent any count from landing in bin i . Or there were too few samples taken from the distribution, and bin i was just not "lucky" enough to get one of the counts. The first possibility is accounted for by the 0 part of the ZIP distribution while the Poisson part accounts for the second case.

We compute the maximum-likelihood estimates of q and λ given the cell counts. Figure 4 shows a typical fitted ZIP distribution from our simulations. Then, according to our model, q is the fraction of structural 0 in our data. Therefore, we expect $qp \theta_i$ s to be 0s, where p is the number of bins. ZIPshrink just takes these qp zeroes out and shrinks the maximum-likelihood estimates of the remaining $(1 - q)p$ fraction of bins towards $\frac{1}{(1-q)p}$ instead of $\frac{1}{p}$ as was done in section 2.4.2. In other words, we adapt the target of the shrinkage estimator and choose which cells are to be shrunk. Note that ZIPshrink reduces to the shrinkage method when $q = 0$.

ZINBshrink follows a similar principle, but models the number of cells having k counts with a Zero-Inflated Negative Binomial (ZINB) distribution instead of a ZIP. Just like Poissons, Negative-Binomial distributions are supported by non-negative integers and are commonly used in a statistical analysis of count data. But, while Poisson(λ) distributions have mean and variance equal to λ , Negative-Binomials on the other hand allow to model so-called over-dispersed variables where the variance exceeds the mean. When it comes to estimating the fraction of structural zeroes in a multinomial distribution, using a Negative-Binomial distribution instead of a Poisson distribution should give more flexibility to our estimator and allows it to better model count data. Finally, note that because a Negative Binomial distribution allows variance to exceed the mean, the estimated fraction q of structural zeros tends to be higher for ZINB than ZIP. We thus expect the ZINBshrink entropy estimates to be lower than the ZIPshrink estimate (see figure 4 for a visual intuition of why this is true).

The probability density function of Negative-Binomial-distributed variable Z is

$$f_Z(k; s, r) = \frac{(r + k - 1)!}{k!(r - 1)!} s^r (1 - s)^k$$

and models the number of failures k until r successes in a series of Bernoulli experiments with individual

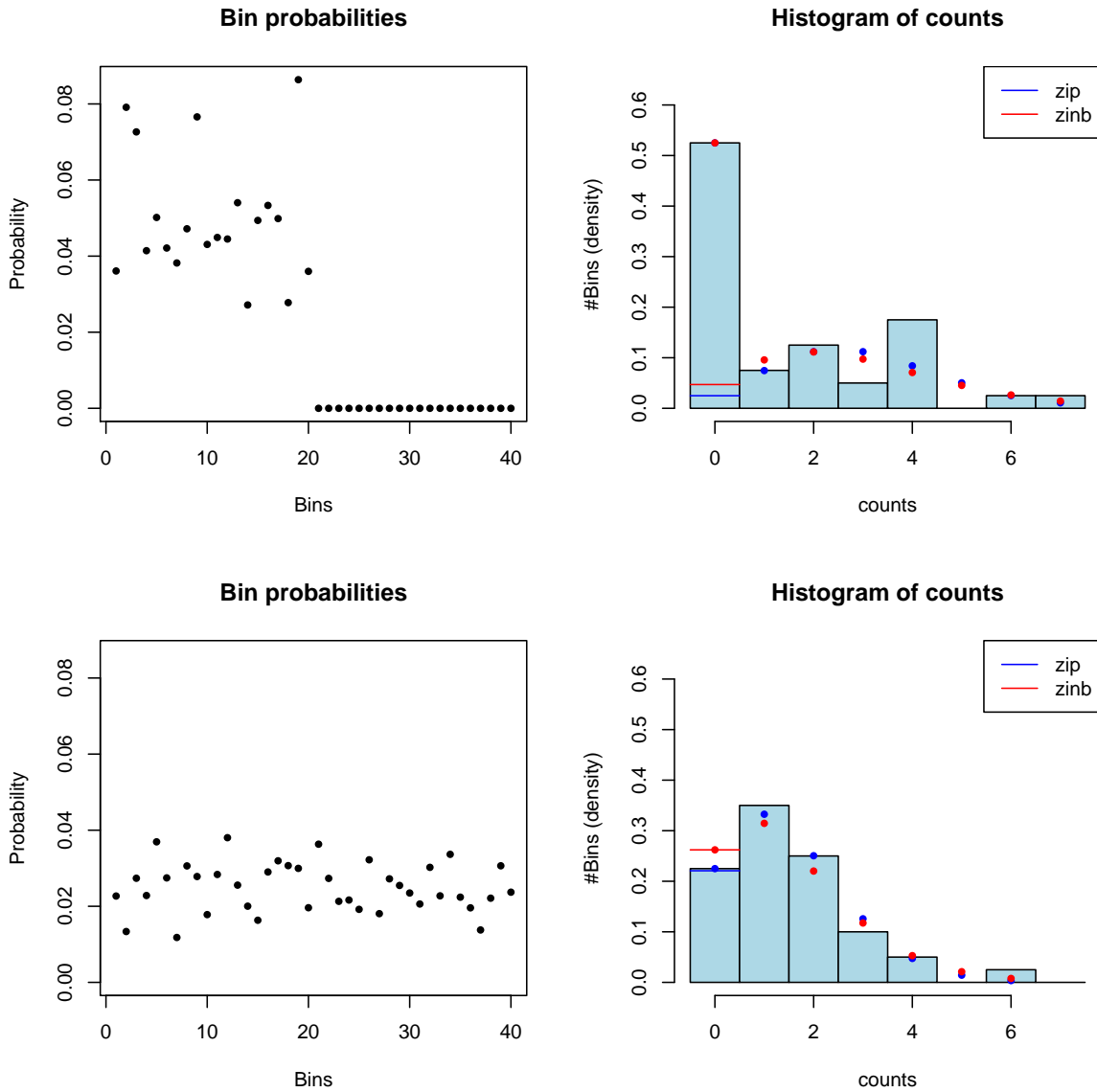


Figure 4: Fitting Zero-Inflated Poisson and Zero-Inflated Negative Binomial distributions. Top row: Estimation of the fraction of structural zeros in a Half-Dirichlet(10) distribution obtained by setting half the multinomial proportions to 0 and sampling the remaining half from a Dirichlet(10). A θ vector of multinomial proportions from a Half-Dirichlet(10) is shown on the left, with bins on the x -axis and cell probabilities on the y -axis. Taking $n = 60$ samples from that distribution, we obtain the histogram on the right. This histogram tells us what fraction of the $p = 40$ bins have $y_i = 0$ counts after sampling, what other fraction of them has 1 count, 2 counts, and so on up to 7 counts. In the present case, it is observed that almost 20% of the 40 bins have 4 counts while less than 10% have 1 count. The blue dots show the maximum-likelihood fit of a Zero-Inflated Poisson (ZIP) distribution while the red dots show the corresponding Zero-Inflated Negative Binomial (ZINB). Finally, the horizontal line in the 0 counts bar represents the number of zero counts expected under an ordinary Poisson or Negative Binomial distribution. In other words, everything between this horizontal line and the dot on top of the bar is considered to be structural zeros by the ZIP/ZINB distribution. Note that in this case, the estimated fraction of structural zeros q is close to the real value 0.5 for both the ZIP and the ZINB. Bottom row: Same as the top row, except we sample multinomial proportions θ from a Dirichlet(10), that is, there are no structural zeros. As can be seen, the dot and the line in the zero-count bar overlap both for the fitted ZIP and ZINB distributions. This means that the estimated fraction q of structural zeros in the underlying multinomial is 0, consistent with the assumptions of the simulation.

probability of success s .

Let X be a ZINB-distributed random variable, that is,

$$\begin{cases} X = 0 & \text{with probability } q \\ X \sim \text{NegBin}(s, r) & \text{with probability } 1 - q \end{cases}$$

Thus, if we call $f_X(k; q, s, r)$ the probability density function of X , we have

$$f_X(k; q, s, r) = \begin{cases} q + (1 - q)s^r & \text{if } k = 0 \\ (1 - q) \frac{(r+k-1)!}{k!(r-1)!} s^r (1 - s)^k & \text{if } k > 0 \end{cases}$$

Furthermore, if x_1, \dots, x_n are independent and identically distributed (iid) realizations of X , the log-likelihood of the data is

$$\begin{aligned} \mathcal{L}(q, s, r; x_1, \dots, x_n) = & c_0 \log(q + (1 - q)s^r) + c_1 \log(1 - q) + \sum_{i|x_i>0} \log \Gamma(r + x_i) \\ & + c_1 r \log(s) + \sum_{i|x_i>0} x_i \log(1 - s) - \sum_{i|x_i>0} \log(x_i!) - c_1 \log \Gamma(r) \end{aligned}$$

where c_0 is the number of x_i that are 0 and c_1 is the number of x_i with $x_i > 0$.

Apart from using a different underlying model, ZINBshrink proceeds just like ZIPshrink. It first fits a ZINB distribution to the number of bins having k to estimate the fraction q of structural zeroes in the data. Then, ZINBshrink trims qp zeroes from the data and computes shrunked multinomial proportions on the rest.

Provided enough samples are provided for the ZIP or ZINB distribution to be fitted, one would expect that removing structural zeroes from the data before shrinking the multinomial proportions helps removing the positive bias on the resulting entropy estimator. Whereas in the case where no cell has a quasi-zero probability, ZIPshrink and ZINBshrink are theoretically equivalent to the shrinkage method.

3 Estimating mutual information

The mutual information between two random variables X and Y is defined as :

$$I(X; Y) = H(X) - H(X|Y) \quad (14)$$

$H(Y|X)$, the conditional entropy of Y given X can be written as

$$H(Y|X) = H(X, Y) - H(X) \quad (15)$$

Substituting in the definition of the mutual information, we obtain

$$I(X; Y) = H(X) + H(Y) - H(X, Y) \quad (16)$$

which shows that mutual information is the difference between the information provided by X and Y taken separately and the information provided by X and Y together. Mutual information thus represents the redundancy between X and Y . Hence, if the mutual information is 0, X and Y have no redundancy and are independent.

As opposed to the correlation coefficient which is another measures of association between random variables, the mutual information is positive definite for discrete variables. Let ρ_{XY} be the correlation coefficient between X and Y ,

$$\rho_{XY} = \frac{\text{cov}(X, Y)}{\sqrt{\text{var}(X)\text{var}(Y)}}$$

If X and Y follow a bivariate Gaussian distribution, we have

$$I(X; Y) = -\frac{1}{2} \log(1 - \rho_{XY}^2)$$

In expression 16, $H(X)$ and $H(Y)$ can be estimated as explained in the previous subsection. As for $H(X, Y)$, let Z be the joint random variable (X, Y) , and suppose $\dim(X) = \dim(Y) = p$. Then,

$$\hat{H}(X, Y) = \hat{H}(Z) = - \sum_{i,j=1}^p \hat{\theta}(z_{ij}) \log \hat{\theta}(z_{ij})$$

where z_{ij} are the bin counts from the joint random variables. The parameters of the $\hat{\theta}$ estimator have parameters have to be updated in order to take into account that instead of p bins and n samples, we now have p^2 bins and n samples. Hence, if we follow this approach, we have to keep in mind that mutual information is much harder to estimate than entropy.

3.1 Conditional mutual information

The previous computations can be extended to conditional mutual information, which quantifies conditional independence. The conditional information between random variables X and Y given Z can be defined as

$$\begin{aligned} I(X; Y|Z) &= H(X|Z) - H(X|Y, Z) \\ &= H(X|Z) + H(Y|Z) - H(X, Y|Z) \\ &= H(X, Z) + H(Y, Z) - H(X, Y, Z) - H(Z) \end{aligned}$$

In the theory of graphical models (Whittaker, 1990), one is generally interested in conditional independence between two variables X, Y given all the others Z_1, Z_2, \dots, Z_n . As far as conditional mutual information is concerned, this transposes to

$$\begin{aligned} I(X; Y|Z_1, \dots, Z_n) &= H(X|Z_1, \dots, Z_n) - H(X|Y, Z_1, \dots, Z_n) \\ &= H(X|Z_1, \dots, Z_n) + H(Y|Z_1, \dots, Z_n) - H(X, Y|Z_1, \dots, Z_n) \\ &= H(X, Z_1, \dots, Z_n) + H(Y, Z_1, \dots, Z_n) - H(X, Y, Z_1, \dots, Z_n) - H(Z_1, \dots, Z_n) \end{aligned} \quad (17)$$

4 Results

To assess how efficient estimators are, we did simulations on synthesized datasets.

4.1 Simulations setup

Simulations ran on a Dual-CPU AMD Opteron 2.4 GHz machine with 7 GB working memory, although implementations of shrinkage based estimators do not currently use SMP features and neither does NSB to our knowledge. We used nsb-entropy 1.1 C++ implementation of the NSB method and implemented the shrink, ZIPshrink, ZINBshrink estimators in R 2.3.1 (R Development Core Team, 2006). Figure 5 (left plot) compares running time for various estimators.

Paninski wrote a series of articles with interesting analytical results on the existence of a consistent entropy estimator for the case where n/p is bounded (Paninski, 2004) and deriving the so-called ‘‘Best Upper Bound (BUB)’’ entropy estimator (Paninski, 2003) which is a significative improvement over Miller’s formula. However, BUB tended to perform bad in our simulations so we don’t plot it here.

When it comes to choosing a number of bins p , what matters is not as much the absolute size p as the samples per bin ratio n/p (Miller, 1955). We chose to fix $p = 100$ and let n gradually vary from very under-sampled regimes ($n = 5$, that is an average of .05 samples per bin) to oversampled regimes ($n = 1000$, or 10 expected samples per bin). Doing so enables us to check the convergence rate as well as the consistency of the benchmarked estimators. We made $R = 500$ repetitions resulting in R entropy estimates \hat{H}_k for each combination of sampling regime, estimator and underlying cell proportion distribution. From these estimates, we estimated the MSE with $\frac{1}{R} \sum_{k=1}^R (\hat{H}_k - H)^2$ and the bias with $\frac{1}{R} \sum_{k=1}^R \hat{H}_k - H$.

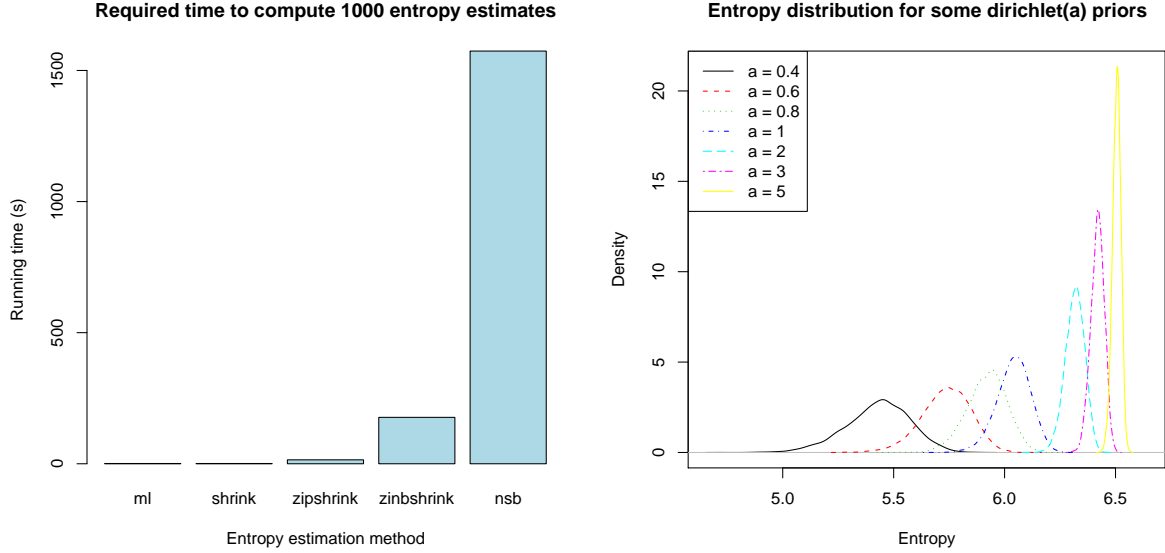


Figure 5: Left: Running time for different entropy estimators. Simulations ran while no other user was running jobs on the node. Multinomial proportions were sampled from a Dirichlet(.02) distribution, with $p = 100$ bins. 250 repetitions were performed with 5, 35, 100 and 1000 samples from multinomial distributions resulting in computing a total of 1000 entropy estimates. ML is included as a reference (total runtime is 1 s). Shrink is just as fast as ML as the total runtime is 1s too. ZIPshrink begins with a two-dimensional grid search for the maximum likelihood, then optimizes it with L-BFGS-B (Byrd et al., 1995) and so does ZINBshrink. However, ZINBshrink is about 10 times slower than ZIPshrink because the Negative Binomial distribution has two parameters instead of one for the Poisson, which slows the preliminary grid search step down. Finally, the C++ implementation of NSB is 100 times slower than ZIPshrink and 9 times slower than ZINBshrink. Right: Entropy distribution of multinomials whose proportions are sampled from different Dirichlet(a) priors. The curves were obtained for multinomials with $p = 100$ bins, so the infimum for the entropy is 0 while the supremum is ≈ 6.64 . Entropy densities were estimated from 10000 repetitions using a Gaussian kernel. Note that for $a = 5$, the entropy distribution is close to being a δ function : the prior determines the entropy. The effect attenuates as $a \rightarrow 0$ but it still is very strong for $a = .4$, as the entropy of the resulting multinomials is comprised between 5 and 5.8 — only 12% of the possible range.

In the present simulations, the θ multinomial proportions vector is sampled from three different distribution types. These are :

Dirichlet Multinomial proportions $\theta_i, i = 1, \dots, p$ are sampled from a Dirichlet(a) distribution (see equation 6).

$$\theta = (\theta_1, \dots, \theta_p) \sim \text{Dirichlet}(a)$$

Then, we generate counts by drawing n samples from the resulting multinomial(θ) distribution :

$$y = (y_1, \dots, y_p) \sim \text{multinomial}(\theta)$$

and hence, we have $\sum_{i=1}^p y_i = n$. Finally, entropy is estimated from the counts y , and the whole cycle starts again until the required number of repetitions has been performed. We then turn on to the next sampling condition, perform the required number of repetitions, and so on.

For the parameter a , we choose $a = 0.005$, $a = 1$ and $a = 10$ because very different θ random vectors occur under these values. Typical θ vectors sampled from Dirichlet(a) are plotted on figure 6 for different a s. When $a = .005$, the Dirichlet distribution privileges θ vectors for which the probability is concentrated in a few bins while the remaining θ_i s are near-0. Empirically, smaller values of a tend to give θ s which are so close to 0 that they break the floating-point number

representativity limit. As a result, R turns these multinomial proportions into NAs, which in turns makes it impossible to use the resulting θ vector as probability vector for subsequent multinomial sampling. Under Dirichlet(1), all θ s are equally likely as can be seen from the definition of the Dirichlet distribution (see equation 6). The distribution is then “neutral” in the sense that it does not privilege any θ in particular. Note that this doesn’t imply that $\theta_1 = \dots = \theta_p$! For $a = 10$ finally, it becomes very unlikely to have some i with θ_i close to 0. Dirichlet(10) tends to prefer θ vectors which have θ_i s close to one another. The bigger a , the closer Dirichlet(a)-sampled θ vectors get to the uniform θ ($\theta_i = \frac{1}{p}, i = 1, \dots, p$).

Half-Dirichlet We call “half-Dirichlet” an equal mixture of two distributions, where half the bin probabilities θ are set to 0 and the rest is Dirichlet(a) distributed. We set $a = 10$ to ensure $\theta_i = 0$ for half the cells and $\theta_i \gg 0$ for the rest. We then proceed like in the Dirichlet case. This distribution enables us to test how estimators deal with structural zeros.

Zipf Multinomial proportions θ are set to

$$\theta_i = \frac{1}{i \sum_{k=1}^p \frac{1}{k}}$$

for $i = 1, \dots, p$, resulting in the i^{th} multinomial proportion to be inversely proportional to i . Empirically, the Zipf distribution occurs in various disciplines such as linguistics (the most frequently used word is used nearly twice as much as the second most used word, almost three times as much as the third most used word, *etc.*) or economics (wealth repartition, where the richest person is about twice as rich than the second richest, roughly three times as rich as the third richest person, and so on). Because it tends to be common in “natural” phenomena, it is interesting to study how entropy estimators perform when we sample from this distribution.

For simulations, we proceeded just like in the Dirichlet case, except the θ parameter to the multinomial(θ) was the same for all the repetitions instead of changing from repetition to repetition.

In total, we simulated multinomial proportions from 5 distributions : Dirichlet(0.005), Dirichlet(1), Dirichlet(10), Half-Dirichlet(10) and Zipf.

4.2 Entropy

Figure 6 summarizes the results for the three Dirichlet distributions. Figure 7 on the other hand shows how estimators perform in the Half-Dirichlet and Zipf cases.

Remember that the MSE of Dirichlet prior based entropy estimators introduced in section 2.2 was commonly oscillating between 5 and 30 (see fig. 3). Plotting these estimators together with shrink, ZIPshrink, ZINBshrink and NSB would make the latter look like a 0 horizontal flat-line since these estimators have an MSE of less than 4 in the worse case. That is why we don’t plot Dirichlet prior based estimators any further in these simulations.

5 Discussion

First of all, simulations showed that the usual Bayesian priors are no good choice for under-sampled data because they are too picky about the prior. Choosing a bad prior on multinomial proportions may result in very bad multinomial proportions estimates and, consequently, to erroneous entropy estimates.

We have proposed a simple but statistically highly efficient shrinkage estimator of count probabilities and entropy. The resulting entropy estimator performs well for both small n (under-sampling) and large n . With this method, no parameters need to be specified and no specification of prior distribution is required either. At the same time, estimates remain very simple to compute. However, the estimator tends to perform poorly on distributions which are not smooth because probability is concentrated in just a few bins.

We then proposed a way of generalizing the estimator to distributions on which the shrinkage estimator fails. We achieve this by estimating the fraction of structural zeros in the data with a Zero-Inflated

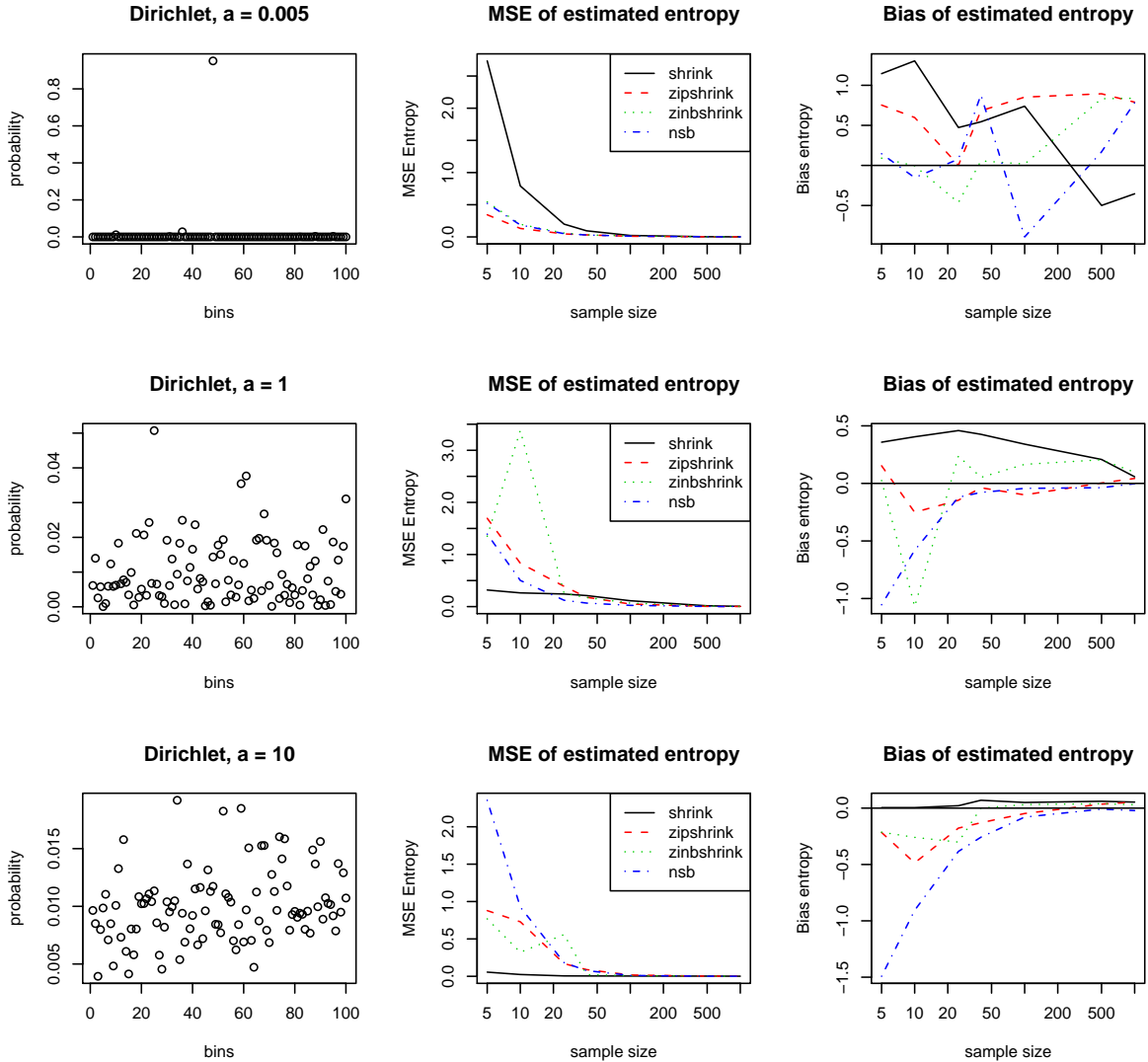


Figure 6: Assessing the performance of entropy estimators on multinomials with various Dirichlet(a) distributed cell probabilities. The left column shows typical cell probabilities for $a = .005, 1$ and 10 . Note how cell probabilities get closer to one another as a grows. The middle column shows the MSE and the right column has the bias. For $a \leq 1$, the entropy varies too much from repetition to repetition for the asymptotic regime to be reached in the bias with just 500 repetitions. NSB is good when a is small but deteriorates as a grows because it has a negative bias. On the other hand, the Shrinkage proportions estimator betters as a grows but has a persisting positive bias. ZIPshrink which first trims structural zeros before shrinking the proportions allows the MSE to stay under 2.0 even in the most under-sampled regime and helps getting rid of the bias. Finally, ZINBshrink looks good, but only if we have a reasonable amount of samples ($n \geq 20$).

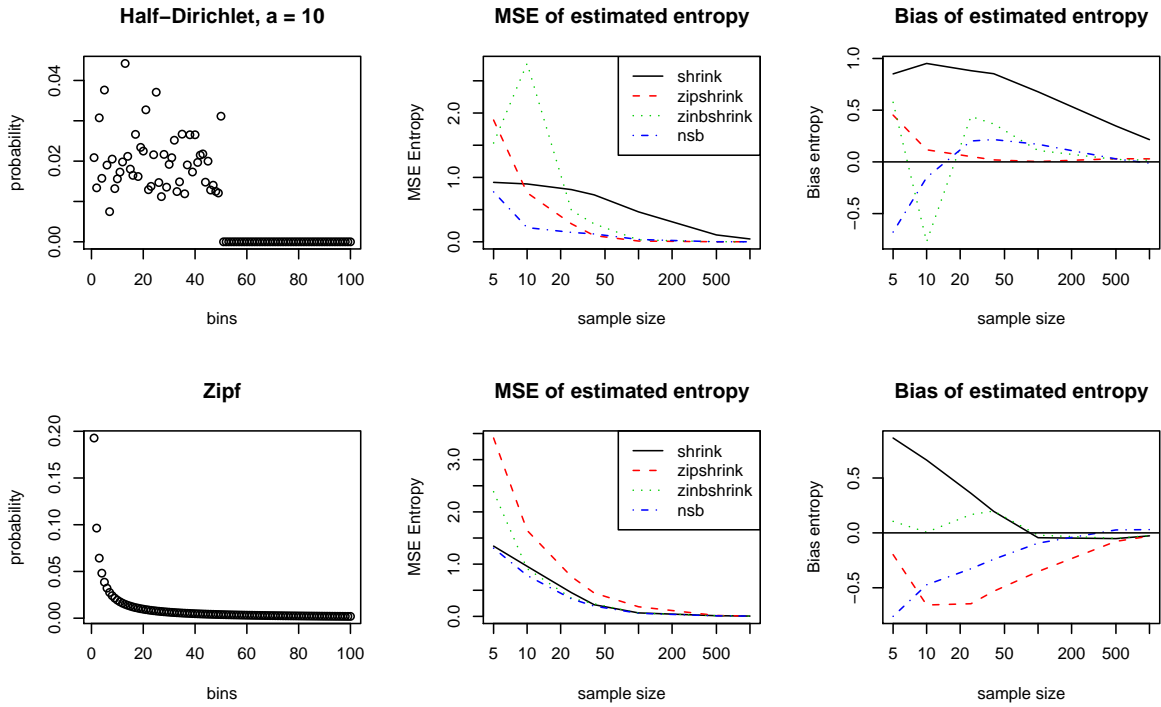


Figure 7: Assessing entropy estimators efficiency on multinomials with half-Dirichlet(a) and Zipf distributed cell probabilities. The plots in the three columns were made as in figure 6. NSB performs best on the half-Dirichlet distribution, though ZIPshrink is just as good for $n \geq 30$ according to the MSE and has no bias. ZINBshrink seems to become reliable when $n \geq 30$, but is not as good as ZIPshrink with shrink being the worst of the 4 estimators in this case. On the other hand, shrink and NSB are the best on the Zipf distribution, with ZINBshrink being just as good for $n \geq 10$ and having no bias and ZIPshrink being the worst in this case.

Poisson or a Zero-Inflated Negative Binomial distribution. The resulting estimators perform much better than the sole shrinkage on distributions for which shrinkage fails while doing reasonably well on distributions on which shrinkage performs well.

Furthermore, we were able to reproduce the good behavior of the NSB estimator compared the various flavors of Dirichlet prior based estimators.

However, our simulations tended to show that none of the benchmarked estimator was significantly better than the others overall. For instance, the shrinkage estimator is by far the best (and fastest) on smooth distributions but it breaks down when probability is concentrated in a few bins, that is in the Dirichlet(a) case with $a < 1$. NSB works on the other hand, does very well in this case but takes long to compute and is suboptimal on smooth distributions. Finally, ZIPshrink and ZINBshrink adapt to a wide class of distributions but need a reasonable amount of data for the ZIP and ZINB to be fitted. In the case where we have $p = 100$ bins, ensuring $n \geq 20$ seems to be a safe requirement, that is an average of 0.2 sample per bin.

Hence, in the end, the choice of a particular estimator depends on our *a priori* knowledge/assumption of the underlying distribution and considerations related to computational cost. We would recommend to

- use ZIPshrink if $n/p \geq .1$ or ZINBshrink if $n/p \geq .3$
- use NSB if you have time and/or CPU power and you know the distribution not to be smooth
- use shrink if you know the underlying distribution to be smooth

As a final word of advice, note that the entropy estimators we compare in the present paper make very little assumptions on the probability distribution of which they try to estimate the entropy. For instance,

we assume that bin B_i and bin B_{i+1} being neighbors tells us nothing about how close the multinomial proportions θ_i and θ_{i+1} might be. Depending on the field of application, this can be a very desirable assumption or a failure to incorporate part of our knowledge in the estimator.

For instance, let's say we want to estimate the entropy of a car color sequence : looking through the window, we note the color of each car passing by and want to estimate the entropy of the resulting sequence. We can model the experiment as successively drawing iid samples from a multinomial distribution. Each bin in the multinomial corresponds to one possible car color. In this case, we don't look at the colors as if we were doing optics or light physics but as the mark of individual tastes. Tastes are well-known to be hard to "compare" or "measure", especially in the present case : it sure doesn't make much sense to claim that "green" is closer to "red" than "blue". Therefore, it doesn't make sense to have a metric and the model we use in the present paper is adapted.

On the other hand, let's consider estimating the entropy of a quantitative variable such as the expression profile of a single gene. We measure the expression of that gene under n experimental conditions, resulting in n measurements x_1, x_2, \dots, x_n . In this case, expression levels x_1, \dots, x_n are just integers such as $x_1 = 253, x_2 = 192, x_3 = 402$. It does make sense to say that x_1 is closer to x_2 than x_3 . We can actually measure how close two expression levels are by computing their difference for instance. In this case, we have a metric (for instance $d_{ij} = |x_i - x_j|$). In the case where we have a metric, it may be desirable to use it for the estimation problem as putting more knowledge in our model should help us to make better estimates. For example, the Kozachenko-Leonenko entropy estimator reviewed in Kraskov et al. (2004) is based on the assumption that the the probability density function is close to being constant locally, where locality is defined in the sense of some metric.

6 Future work

In the future, we plan to use our entropy estimator with the ARACNE algorithm (Margolin et al., 2006). ARACNE attempts to reverse-engineer genetic regulatory networks from microarray gene expression data using an information theoretic measure of dependence. We hope that using one of our estimates instead of the one currently used by ARACNE will result in an increased reliability and accuracy.

Part III

Conclusion

Once that we have an entropy estimator we are satisfied with, how do we infer the network ? In theory, we just have to apply the conditional independence criteria. But given the difficulty to estimate conditional mutual information on high-dimensional spaces, some methods have been proposed to learn the conditional independence graph while keeping the estimation problem tractable. Two of them (ARACNE and 0-1 networks) are studied in the first section. We then turn to the problem of missing data and conclude on considerations on time-series and directionality in the graph.

ARACNE and 0-1 networks

de Campos and Huete (2000) introduces the so-called “0-1 networks” which only rely on low-order conditional independence tests. On the other hand, ARACNE (Margolin et al., 2006) uses the data processing inequality to get rid of edges corresponding to indirect interactions.

Remember that in conditional independence graphs, we draw an edge between two variables if their conditional mutual information given all the others variables is 0. Calling X and Y the two variables of interest and Z_1, \dots, Z_N all the others, we draw an edge between X and Y if

$$I(X; Y|Z_1, \dots, Z_N) > 0$$

In 0-1 networks, we draw an edge between X and Y if

$$I(X; Y) > 0 \wedge \forall i \in \{1, \dots, N\}, I(X; Y|Z_i) > 0 \quad (18)$$

Finally, omitting details related to mutual information estimation issues, ARACNE connects X and Y if

$$I(X; Y) > 0 \wedge \forall i \in \{1, \dots, N\}, I(X; Y) > \min \{I(X; Z_i), I(Y; Z_i)\} \quad (19)$$

that is if the so-called “data processing inequality” (DPI) is satisfied.

Both the use of the DPI and the 0-1 network testing strategy result in a simpler entropy estimation problem because they make it possible to only deal with low dimensional spaces : 2 dimensional spaces if we use the DPI, or 3 dimensional with 0-1 networks.

One can prove that equation 19 implies equation 18. One way of doing could be as follows. Consider the $I(X; Y|Z_i)$ term in equation 18 :

$$\begin{aligned} I(X; Y|Z_i) &= H(X|Z_i) + H(Y|Z_i) - H(X, Y|Z_i) \\ &= H(X|Z_i) + H(Y, Z_i) - H(X, Y, Z_i) \end{aligned}$$

Therefore,

$$\begin{aligned} I(X; Y|Z_i) > 0 &\Leftrightarrow H(X) + H(Y, Z_i) - H(X, Y, Z_i) > H(X) - H(X|Z_i) \\ &\Leftrightarrow H(X) - H(X|Y, Z_i) > I(X; Z_i) \\ &\Leftrightarrow I(X, (Y, Z_i)) > I(X; Z_i) \end{aligned} \quad (20)$$

Because the mutual information is symmetric in X and Y , we also have

$$I(X; Y|Z_i) > 0 \Leftrightarrow I(Y; (X, Z_i)) > I(Y; Z_i) \quad (21)$$

But note that, by definition,

$$\begin{aligned}
I(X; Z_i|Y) \geq 0 &\Leftrightarrow H(X|Y) - H(X|X|Y, Z_i) \geq 0 \\
&\Leftrightarrow H(X|Y) \geq H(X|Y, Z_i) \\
&\Leftrightarrow H(X) - H(X|Y) \leq H(X) - H(X|Y, Z_i) \\
&\Leftrightarrow I(X; Y) \leq I(X, (Y, Z_i))
\end{aligned} \tag{22}$$

On the other hand, the DPI formula (see equation 19) requires

$$\begin{aligned}
I(X; Y) &> \min\{I(X; Z_i), I(Y; Z_i)\} \\
&\Leftrightarrow I(X; Y) > I(X; Z_i) \vee I(X; Y) > I(Y; Z_i) \\
&\Rightarrow I(X; (Y, Z_i)) > I(X; Z_i) \vee I(Y; (X, Z_i)) > I(Y; Z_i) && \text{using inequality 22} \\
&\Leftrightarrow I(X; Y|Z_i) > 0 && \text{using equations 20 and 21}
\end{aligned} \tag{23}$$

Therefore, whenever the DPI condition is satisfied, at least one of the left hand and right hand side expressions in 23 has to be true. But the left hand side expression is just equation 20 while the right hand side expression is equation 21, both of them characterize 0-1 networks ! In other words, whenever ARACNE connects two vertices, 0-1 networks would draw an edge as well. But is the converse true ? That is, are there edges that 0-1 networks would correctly identify but that ARACNE would miss ? Actually, one can show that this is the case using the following example.

Let us say we have N variables grouped in a random vector $X = (X_1, \dots, X_N)$. Let X be multivariate Gaussian distributed, that is, if f_X is the pdf of X , we have :

$$f_X(x_1, \dots, x_N) = \frac{1}{\sqrt{(2\pi)^N |\Sigma|}} \exp\left(-\frac{1}{2}(x - \mu)^T \Sigma^{-1}(x - \mu)\right) \tag{24}$$

Let further $\Sigma^{-1} = ((\omega_{ij}))$ and $\Sigma = ((\sigma_{ij}))$. Σ is symmetric by definition, so Σ^{-1} is symmetric too. Note that the pdf formula 24 makes it clear that whenever $\omega_{ij} = 0$, there can be no direct interactions between variable X_i and variable X_j , though they could be correlated if they happen to be correlated to a common variable. Let us consider the case where we have $N = 3$ multivariate Gaussian distributed variables, with mean $\mu = 0$ and covariance matrix

$$\Sigma = \begin{pmatrix} 1.0 & 0.2 & -0.4 \\ 0.2 & 1.0 & -0.8 \\ -0.4 & -0.8 & 1.0 \end{pmatrix}$$

which is positive definite and symmetric. Inverting it gives

$$\Sigma^{-1} = \begin{pmatrix} 1.25 & 0.42 & 0.83 \\ 0.42 & 2.92 & 2.50 \\ 0.83 & 2.50 & 3.33 \end{pmatrix}$$

where no entry is 0, so each variable is *conditionally dependent* on all the others. As a result, the conditional independence graph has three vertices and connects each vertex to the two others in a similar way to the left picture on figure 2.

In the $N = 3$ variables case, the 0-1 network is just the conditional independence graph.

What about the DPI based network ?

The mutual information between two Gaussian distributed random variables X_i and X_j with correlation coefficient ρ_{ij} is

$$I(X_i, X_j) = -\frac{1}{2} \log(1 - \rho_{ij}^2)$$

In our case, $\rho_{ij} = \sigma_{ij}$. It follows that $I(X_1, X_2) = .02$, $I(X_1, X_3) = .08$, $I(X_2, X_3) = .51$ and hence $I(X_1, X_2) < \min\{I(X_1, X_3), I(X_2, X_3)\}$ while all $I(X_i, X_j)$ are strictly positive. The resulting graph connects X_1 to X_3 and X_3 to X_2 while drawing no edge between X_1 and X_2 , resulting in a network similar to the

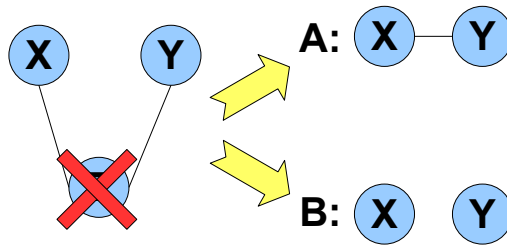


Figure 8: The effect of missing data on the inferred conditional independence graph. Left: True network made of three variables X , Y and Z . X is connected to Z which in turns connects to Y , but there is no direct interaction between X and Y . We ask what happens if we don't measure variable Z . Right: The upper graph represents outcome "A" where the X connects to Y as a consequence to not observing Z . The lower graph pictures possibility "B" where no edge is drawn between X and Y .

right picture on figure 2. In a word, the DPI approach approximates 0-1 networks but it may remove edges corresponding to direct interactions that 0-1 networks are able to spot, for instance triangular interactions such as the owns shown on figure 2 (left picture). However, the DPI approach has the advantage that only requires entropy to be estimated on a two dimensional space instead of three for the 0-1 networks. Hence, DPI trades inference accuracy for lower dimensionality. The question of wether it is worth sacrificing inference accury in order to reduce the dimensionality from 3 to 2 depends on the estimator we use. If our estimator is reliable enough on a 3 dimensional space, the 0-1 networks strategy should be preferred. If not, the DPI approach may be more suitable. Finally, Wille and Bühlmann (2006) investigates under which conditions the 0-1 graph coincides with the conditional independence graph and what happens when this is not the case.

Until now, we assumed that there were only pairwise interactions in the network. But in the past years, some authors started investigating the possibility to infer N -way interactions as well, with $N > 3$. N -way interactions are important and are very likely to be a requirement in order to model transcription complexes mediated regulation for instance, when N regulatory proteins bind together with the DNA polymerase and the DNA, thereby regulating gene transcription. A theoretical framework is developed in Schneidman et al. (2003) and Nemenman (2004). Currently though, the bottleneck seems to be the estimation problems in high dimensional spaces.

Missing data

Another question relates to the exhaustivity of microarray experiments. Until now, we assumed that we measured all the existing genes in the cell. But what happens if this is not the case ? This issue needs to be addressed with care since serious interpretation errors may result from attempting to do causal inference while not properly considering the problem of missing data (*e.g.* Simpson's paradox).

Obviously, we can only infer edges between genes we actually measured. But when it comes to edges representing direct vs. indirect interactions, what would be a desirable behavior of the genetic regulatory network inference method in the case of missing data ? Figure 8 suggests two possibilities. Let us consider the network on the left hand side picture where two genes X and Y are connected through Z . If we didn't observe Z , we argue for preserving the (indirect) interaction between X and Y , connecting together all the vertices that bind to Z (case "A" on the right hand side picture of figure 8). Alternatively, one could strictly stick to the "direct interaction" policy discussed in section I and draw no edges between the two (case "B" on the right hand side picture of figure 8).

Assuming conditional mutual information is known without error, we believe that applying the conditional independence criteria should result in solution "A" to be chosen over solution "B", that is connectivity is preserved rather than enforcing the "direct interaction" policy. In practice however, we first have to estimate conditional mutual information and then decide if it is significantly different from 0. Under such circumstances, we expect mutual information estimation errors to make the "B" behavior

possible too. But we have no certitude about this question yet and need to look more carefully into it.

Finally, to address the problem of missing data, some authors suggest using more complex, hidden space models (for example Beal et al., 2005).

Edge directionality & causality

So far, we have concentrated on undirected graphs and left aside the question of how to replace edges with arrows. We argued that we would focus on determining the structure of the network because knowing the structure may be a good starting point for further analysis such as inferring edge directionality or estimating the dynamics in the network.

When it comes to replacing genes with arrows, we have to ask ourselves what information we would like to be carried by edge directionality. Ideally, since we are trying to infer genetic regulatory networks, we would like an arrow going from gene “A” to gene “B” to represent regulation in a causal sense. But causal inference is a delicate problem and constitutes a research topic in itself. For instance, problems that occur when determining directionality in a graph include :

Computational intractability To do maximum likelihood based parameter estimation on all possible directed graphs having a known undirected skeleton and if we assume there are no loops with just two genes, one has to do optimize a likelihood function 2^E times, where E is the number of undirected edge in the skeleton. For a network with just 9 genes, this number may be as big 6.87e10 already.

Loops When trying to model a genetic regulatory network with a Bayesian network, one has the problem that Bayesian networks can’t model loops in the graph because in that case, the joint pdf can’t be factorized in conditional probabilities. That is, the network has to be a directed acyclic graph (DAG), *i.e.* it must have the topology of a tree.

Causal inference Even if we omit the computational aspects, inferring edge directionality using the maximum likelihood principle will select a model that best fits the data in a logical sense. But the direction of logical deductibility *may or may not* be the same as the direction of physical causality¹.

To address both the first and the third issues, learning algorithms usually start with a causal skeleton, or at least a variable ordering that has to be specified by the scientist or by a field expert, possibly using other (experimental) evidence.

For instance, to specify directionality, the scientist may use the intervention principle, *i.e.* change something in the system of interest and study what happens as a result of that change. A biologist who knows that gene A and gene B interact directly and wants to know whether gene A regulates B or the reverse may knock out gene A .

Alternatively, the scientist may use the principle that the cause has to occur before the consequence in time. He may thus be interested in gathering gene expression time series which we briefly discuss in the next and last section.

Time-series vs. steady-state

When designing his experiments, the biologist may wonder whether he has to make time series of microarray experiments in order to reverse-engineer a genetic network or if he may just as well set up experimental conditions of interest (heat shock, UV exposure, nutritional stress, exponential grow, an so on) and do the microarray measurement at steady state.

¹To illustrate this point, let us consider a man at home, on a winter night. The man is about to open the door and enter a room which we know to be well isolated and properly heated. Opening the window (noted W) of that room causes the room to get cold (noted C) as air from outside mixes with warm air from the room. Hence, W implies C , in the physical, causal sense. But if the man enters a cold room (C) on that cold night and notices that the room is freezing, he would automatically deduce that the window was left opened (W). Hence, in contrast to the direction of physical causality, C implies W , in the logical sense.

From the viewpoint of genetic regulatory network inference, the main question is : do we need to determine edge directionality, e.g. do we want to do causal inference ? If so, as mentioned in the previous section, time series may help us to do it. On the other hand, time series may be more difficult (and possibly more expensive) to gather. Therefore, in the case when one is just interested in inferring what gene is associated with what other gene but one doesn't care whether association is of type "regulates" or "is regulated by", one may just as well perform repeated steady states microarray measurements. One could then apply the conditional independence criteria to reverse-engineer the genetic regulatory network.

Finally, in theory, it should be possible to feed time series to the conditional independence graphical model even if it wasn't specifically designed to account for temporal dynamics. Beal et al. (2005) introduces a Bayesian version of the Kalman filter which models temporal dynamics and missing data. The authors analyze time series of microarray data made of 10 time points but restricting the number of genes to 40. Schäfer and Strimmer (2005) uses a Gaussian graphical model (GGM) approach which models gene expression with a multivariate Gaussian while doing nothing specifically to account for temporal dynamics. However, with their method — available as an R package — they analyze time series of 102 genes and 8 time points and recover a biological plausible network while doing better than other methods.

Future work

We want to use our estimator together with the DPI based approach to genetic regulatory network reverse-engineering and run simulations on synthetic data to see if it improves the reliability of the algorithm. Furthermore, since 0-1 networks are better approximations to conditional independence networks than DPI based approaches, we plan to implement an algorithm based on 0-1 networks too and compare their respective outcomes.

References

- Beal, M. J., Falciani, F., Ghahramani, Z., Rangel, C., and Wild, D. L. (2005). “A Bayesian approach to reconstructing genetic regulation networks with hidden factors.” *Bioinformatics*, 21, 3, 349–356.
- Beerenwinkel, N., Schmidt, B., Walter, H., Kaiser, R., Lengauer, T., Hoffmann, D., Korn, K., and Selbig, J. (2002). “Diversity and complexity of HIV-1 drug resistance: A bioinformatics approach to predicting phenotype from genotype.” *PNAS*, 99, 12, 8271–8276.
- Beirlant, J., Dudewicz, E. J., Györfi, L., and van der Meulen, E. C. (1997). “Nonparametric entropy estimation: an overview.” *International Journal of Math. Stat. Sci.*, 6, 1, 17–39.
- Boulesteix, A.-L., Tutz, G., and Strimmer, K. (2003). “A CART-based approach to discover emerging patterns in microarray data.” *Bioinformatics*, 19, 18, 2465–2472.
- Butte, A. J. and Kohane, I. S. (2000). “Mutual information relevance networks: Functional genomic clustering using pairwise entropy measurements.” In *Proceedings of the 5th Pacific Symposium on Biocomputing*, 415–426.
- Byrd, R. H., Lu, P., Nocedal, J., and Zhu, C. (1995). “A Limited Memory Algorithm for Bound Constrained Optimization.” *SIAM Journal on Scientific and Statistical Computing*, 16, 5, 1190–1208.
- Chiu, D. K. Y. and Kolodziejczak, T. (1991). “Inferring consensus structure from nucleic acid sequences.” *Bioinformatics*, 7, 3, 349.
- de Campos, L. M. and Huete, J. F. (2000). “A new approach for learning belief networks using independence criteria.” *International Journal of Approximate Reasoning*, 24, 11–37.
- Efron, B. (1975). “Biased versus unbiased estimation.” *Adv. Math.*, 16, 259–277.
- (1982). “Maximum likelihood and decision theory.” *Ann. Statist.*, 10, 340–356.
- Eisen, M. B., Spellman, P. T., Brown, P. O., and Botstein, D. (1998). “Cluster analysis and display of genome-wide expression patterns.” *PNAS*, 95, 25, 14863–14868.
- Fienberg, S. E. and Holland, P. W. (1973). “Simultaneous estimation of multinomial cell probabilities.” *J. Amer. Statist. Assoc.*, 68, 683–691.
- Geisser, S. (1984). “On prior distributions for binary trials.” *The American Statistician*, 38, 244–251.
- Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (2004). *Bayesian Data Analysis*. 2nd ed. Boca Raton: Chapman & Hall/CRC.
- Gruber, M. H. J. (1998). *Improving Efficiency by Shrinkage*. New York: Marcel Dekker. ISBN 0-8247-0156-9.
- Jeffreys, H. (1946). “An invariant form for the prior probability in estimation problems.” *Proc. Roc. Soc. (Lond.) A*, 186, 453–461.
- Kraskov, A., Stögbauer, H., and Grassberger, P. (2004). “Estimating Mutual Information.” *Physical Review E*, 69.
- Krichevsky, R. E. and Trofimov, V. K. (1981). “The performance of universal encoding.” *IEEE Trans. Inf. Theory*, 27, 199–207.
- Lambert, D. (1992). “Zero-Inflated Poisson Regression, with an Application to Defects in Manufacturing.” *Technometrics*, 34, 1, 1–14.
- Leonhard, T. (1977). “A Bayesian approach to some multinomial estimation and pretesting problems.” *J. Amer. Statist. Assoc.*, 72, 869–874.

- MacKay, D. J. C. (2003). *Information Theory, Inference, and Learning Algorithms*. Cambridge, United Kingdom: Cambridge University Press.
- Makeig, S., Jung, T.-P., Bell, A. J., Ghahremani, D., and Sejnowski, T. J. (1997). “Blind separation of auditory event-related brain responses into independent components.” *PNAS*, 94, 10979–10984.
- Margolin, A. A., Nemenman, I., Basso, K., Klein, U., Wiggins, C., Stolovitzky, G., Favera, R. D., and Califano, A. (2006). “ARACNE: An Algorithm for the Reconstruction of Gene Regulatory Networks in a Mammalian Cellular Context.” *BMC Bioinformatics*, 7, S7.
- Meyer, C. R., Boes, J. L., Kim, B., Bland, P., Zasadny, K. R., Kison, P. V., Koral, K., Frey, K. A., and Wahl, R. L. (1997). “Demonstration of accuracy and clinical versality of mutual information for automatic multimodality image fusion using affine and thin-plate spline warped geometric deformations.” *Medical Image Analysis*, 1, 3, 197.
- Miller, G. A. (1955). *Information Theory in Psychology; Problems and Methods*, 95–100. Glencoe, Illinois: Free Press.
- Nemenman, I. (2004). “Information theory, multivariate dependence, and genetic network inference.” Tech. rep., Kavli Institute for Theoretical Physics, University of California Santa Barbara.
- Nemenman, I., Bialek, W., and van Steveninck, R. R. d. R. (2004). “Entropy and information in neural spike trains: Progress on the sampling problem.” *Physical Review E*, 69, 5.
- Nemenman, I., Shafee, F., and Bialek, W. (2002). “Entropy and inference, revisited.” In *Advances in Neural Information Processing Systems 14*, eds. T. G. Dietterich, S. Becker, and Z. Ghahramani, 471–478. Cambridge: MIT Press.
- Ong, T. and Chen, H. (1999). “Updateable PAT-Tree Approach to Chinese Key Phrase Extraction using Mutual Information: A Linguistic Foundation for Knowledge Management.” *Proceedings of the Second Asian Digital Library Conference*, 63–84.
- Paninski, L. (2003). “Estimation of Entropy and Mutual Information.” *Neural Computation*, 15, 1191–1253.
- (2004). “Estimating Entropy on m Bins Given Fewer Than m samples.” *IEEE Transactions on Information Theory*, 50, 9, 2200–2203.
- R Development Core Team (2006). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.
- Santner, T. J. and Duffy, D. E. (1989). *The Statistical Analysis of Discrete Data*. New York: Springer.
- Schäfer, J. and Strimmer, K. (2005). “A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics.” *Statist. Appl. Genet. Mol. Biol.*, 4, 32.
- Schena, M., Shalon, D., Davis, R. W., and Brown, P. O. (1995). “Quantitative Monitoring of Gene Expression Patterns with a Complementary DNA Microarray.” *Science*, 270, 5235, 467–470.
- Schneidman, E., Still, S., Berry, M. J., and Bialek, W. (2003). “Network information and connected correlations.” *Physical Review Letters*, 91, 23, 238701.
- Schürmann, T. and Grassberger, P. (1996). “Entropy estimation of symbol sequences.” *Chaos*, 6, 414–427.
- Stein, C. (1956). “Inadmissibility of the usual estimator for the mean of a multivariate normal distribution.” In *Proc. Third Berkeley Symp. Math. Statist. Probab.*, ed. J. Neyman, vol. 1, 197–206. Berkeley: Univ. California Press.

- Strong, S. P., Koberle, R., van Steveninck, R. R. d. R., and Bialek, W. (1998). "Entropy and Information in Neural Spike Trains." *Physical Review Letters*, 80, 1, 197–200.
- Tusher, V. G., Tibshirani, R., and Chu, G. (2001). "Significance analysis of microarrays applied to the ionizing radiation response." *PNAS*, 98, 9, 5116–5121.
- Whittaker, J. (1990). *Graphical Models In Applied Mathematical Multivariate Statistics*, chap. 3 and 4. Wiley Series in Probability and Mathematical Statistics. Baffins Lane, Chichester: John Wiley & Sons.
- Wichert, S., Fokianos, K., and Strimmer, K. (2004). "Identifying periodically expressed transcripts in microarray time series data." *Bioinformatics*, 20, 1, 5–20.
- Wille, A. and Bühlmann, P. (2006). "Low-Order Conditional Independence Graphs for Inferring Genetic Networks." *Statistical Applications in Genetics and Molecular Biology*, 5, 1.
- Wu, H., Kerr, M., Cui, X., and Churchill, G. (2003). *The Analysis of Gene Expression Data: Methods and Software*, chap. MAANOVA: A software package for the analysis of spotted cDNA microarray experiments. Heidelberg: Springer.