BMC Bioinformatics

BioMed Central

Penalized likelihood for sparse contingency tables with an application to full-length cDNA libraries

BMC Bioinformatics 2007, **8**:476 doi:10.1186/1471-2105-8-476

Corinne Dahinden (dahinden@stat.math.ethz.ch)
Giovanni Parmigiani (gp@jhu.edu)
Mark C Emerick (memeri@jhmi.edu)
Peter Buhlmann (buhlmann@stat.math.ethz.ch)

ISSN 1471-2105

Article type Methodology article

Submission date 16 March 2007

Acceptance date 11 December 2007

Publication date 11 December 2007

Article URL http://www.biomedcentral.com/1471-2105/8/476

Penalized likelihood for sparse contingency tables with an application to full-length cDNA libraries

Corinne Dahinden*[1,2], Giovanni Parmigiani[3], Mark C Emerick[4]and Peter Bühlmann[1,2]

[1] Seminar für Statistik, ETH Zürich, CH-8092 Zürich, Switzerland
[2] Competence Center for Systems Physiology and Metabolic Diseases, ETH Zürich, CH-8093 Zürich, Switzerland
[3] Departments of Oncology and Biostatistics, Johns Hopkins Schools of Medicine and Public Health, Baltimore, MD, USA
[4] Department of Physiology, Johns Hopkins School of Medicine, Baltimore, MD, USA

Email: Corinne Dahinden *- dahinden@stat.math.ethz.ch; Giovanni Parmigiani - gp@jhu.edu; Mark C Emerick - memeri@jhmi.edu; Peter Bühlmann - buhlmann@stat.math.ethz.ch;

*Corresponding author

Abstract

Background: The joint analysis of several categorical variables is a common task in many areas of biology, and is becoming central to systems biology investigations whose goal is to identify potentially complex interaction among variables belonging to a network. Interactions of arbitrary complexity are traditionally modeled in statistics by log-linear models. It is challenging to extend these to the high dimensional and potentially sparse data arising in computational biology. An important example, which provides the motivation for this article, is the analysis of so-called full-length cDNA libraries of alternatively spliced genes, where we investigate relationships among the presence of various exons in transcript species.

Results: We develop methods to perform model selection and parameter estimation in log-linear models for the analysis of sparse contingency tables, to study the interaction of two or more factors. Maximum Likelihood estimation of log-linear model coefficients might not be appropriate because of the presence of zeros in the table's cells, and new methods are required. We propose a computationally efficient ℓ_1- penalization approach extending the Lasso algorithm to this context, and compare it to other procedures in a simulation study. We then illustrate these algorithms on contingency tables arising from full-length cDNA libraries.

Conclusions: We propose regularization methods that can be used successfully to detect complex interaction patterns among categorical variables in a broad range of biological problems involving categorical variables.

Background

One of the most striking discoveries of the genomic era is the unexpectedly small number of genes in the human genome. This amount has decreased from more than 100000 [1] to an estimated number of roughly between 20000 and 25000 ([2, 3]), tens of thousands less than initially expected and essentially the same number as found in phenotypically much simpler organisms. A question of overriding biological significance is, how complex phenotypes of higher organisms arise from limited genomes. Part of the explanation may be that many genes undergo a process called alternative RNA splicing, which can generate many distinct proteins from a single gene.

RNA splicing is a post-transcriptional process that occurs prior to mRNA translation. After the gene has been transcribed into a pre-messenger RNA (pre-mRNA), it consists of intronic regions destined to be removed during pre-mRNA processing (RNA splicing), as well as exonic sequences that are retained within the mature mRNA. After transcription occurs the actual splicing process, where it is decided which exons are retained in the mature message and which are targets for removal. In general, exons and introns are retained and deleted in different combinations to create a diverse array of mRNAs from a common coding sequence. This process is known as alternative RNA splicing. Depending on the source, the percentage of alternatively spliced genes lies between 35% and 60% ([4–10]). By screening many full-length cDNAs it is possible to record the complete cDNA from a mature RNA for the same gene again and again and a full-length cDNA library, also known as single-gene library (SGL), builds up. The library contains detailed information about how specific exon combinations go together. This information is directly related to the functional regions of the proteins as they are grouped in domains which in many cases correspond to a single exon which encodes these domains. For example a transcription factor consists of a DNA binding domain and a regulatory domain. Thus the alteration of the exon structure corresponds to an alteration in the function of this particular domain. The central premise is that a dependency in the domains points to a functional association. If domains interact functionally then their splicing should be co-regulated. And this co-regulation has direct biological significance because it shows us which variable components also interact in the expressed protein. Because the polypeptide is intricately folded and tightly packed, segments that are separated by dozens of introns in the primary transcript may encode domains that interact functionally within the protein. These domains need not be structural neighbors even in the folded protein, but may interact through electrical or van der Waals forces, effects of global conformational changes, or even associations with other proteins. Because of these intricacies, there are no inherent distance restrictions, or limits on the number of interacting sites, and separate domains may combine their

functional effect in unpredictable ways.

Due to the large number of potential combinations in highly alternatively spliced genes, any library will only comprise a small portion of the total theoretically possible inventory of combinations. Statistically, this leads to sparse contingency tables in which dimensions represent exons and cells represent variants. The investigation of interactions among categorical variables where not all possible combinations are observed, means addressing a model selection problem that is challenging both inferentially and computationally.

As far as alternative splicing is concerned, there is an important reason to determine this interaction structure: searching for intrapeptide interactions in functional assays is a very difficult, open-ended problem, where statistical analysis of the splicing interaction structure in the transcriptome can simplify this task enormously by identifying the sets of interacting domains. And as more investigators become interested in this type of information, and large-scale single-gene libraries become available, there is a strong need for reliable statistical methods for analyzing the resulting datasets.

We develop different statistical methods to analyse sparse contingency tables in order to determine the underlying interaction pattern and we use graphical models to visualize these patterns. The methods are compared in a simulation study and illustrated on full-length cDNA libraries.

Results

Algorithm

General introduction to contingency tables and Log-linear Models

In this section we provide general definitions and notations.

Assume we have q categorical random variables or factors, $C = \{C_1, \ldots, C_q\}$, where each C_j can take on a finite number g_j of possible values, called levels. The vector $(c_1, \ldots, c_q)$ represents a particular combination of levels of the joint random variable $C = \{C_1, \ldots, C_q\}$. The total cardinality of C is $m = \prod_{j=1}^{q} g_j$, which corresponds to the m different combinations of levels ($m = 2^q$ when all C_j are dichotomous, as in our splicing example).

We simplify the notation by mapping each configuration of C to a unique natural number $i \in \{1, ..., m\}$ with a (bijective) function f:

$$f : (c_1, \ldots, c_q) \leftrightarrow i \in \{1, \ldots, m\},$$

so we may write $\mathbf{c}_i = (c_1, \ldots, c_q)$. For n observations of C, the corresponding q-way contingency table has

m cells, each listing the frequency of a particular configuration $\mathbf{c}_i$:

$$n_{c_1,\ldots,c_q} = n_i, \quad \sum_{i=1}^{m} n_i = n.$$

A general introduction to contingency tables can be found in [11].

If the observations are independent, with p_i the probability of sampling configuration $\mathbf{c}_i$, the distribution of the cell counts $(n_1, \ldots, n_q)^t$ is multinomial with probability $\mathbf{p} = (p_1, \ldots, p_q)$.

In the splicing example, we may consider the C_j as dichotomous random variables representing q sites of alternative splicing, each with two levels, denoted by $c_j \in \{1, -1\}$, corresponding to the presence or absence of exon j in a transcript. The contingency table therefore has $m = 2^q$ cells, with each cell represented by the q-dimensional binary vector $\mathbf{c}_i = (c_1, \ldots, c_q)$. A log-linear model for the cell probabilities can be written the following way:

$$\log p_i = \beta_\emptyset + \sum_{l \in \{1,\ldots,q\}} \beta_l c_l + \sum_{\substack{j,k \\ j<k \in \{1,\ldots,q\}}} \beta_{jk} c_j c_k + \ldots + \beta_{12\ldots q} c_1 c_2 \cdots c_q. \tag{1}$$

A general log-linear model represents $\mathbf{p}$ as:

$$\log(\mathbf{p}) = \mathbf{X}\boldsymbol{\beta}, \tag{2}$$

where $\boldsymbol{\beta}$ is a vector of unknown coefficients and $\mathbf{X}$ a suitable design matrix as indicated below.

Let's assume that the cell probabilities are expressed in the following way:

$$\log p_{c_1,\ldots,c_q} = \delta_\emptyset + \delta^{C_1}_{c_1} + \ldots + \delta^{C_q}_{c_q} + \delta^{C_1,C_2}_{c_1,c_2} + \ldots + \delta^{C_1,\ldots,C_q}_{c_1,\ldots,c_q}, \tag{3}$$

where $\delta_\emptyset$ is the global mean, $\delta^{C_1}_{c_1}$ is the main effect of the first variable and only depends on the distribution of C_1. Similarly $\delta^{C_1,C_2}_{c_1,c_2}$ is the first order interaction between the first two variables and its value only depends on the joint distribution of these two variables.

We now look for a suitable parametrization $\tilde{X}^{C_i}$ of the vector spaces spanned by the main effects δ^{C_i}, a parametrization $\tilde{X}^{C_i,C_j}$ for the vector spaces spanned by the first order interactions δ^{C_i,C_j} and so on. To ensure identifiability, we impose constraints on these matrices and denote the resulting matrices by X^{C_i}, X^{C_i,C_j} and so on. The design matrix $\mathbf{X}$ finally consists of these submatrices. The constitution of the design matrix $\mathbf{X}$ for factors with two levels can directly be derived from (1). The derivation of the design matrix $\mathbf{X}$ from (3) in the case of more than two levels per factor is basically an analysis of variance (ANOVA) parametrization with poly-contrasts. Details can be found in the *Additional file* Section 1.

Sometimes we may assume a smaller model without some of the interaction terms. It is of the form as in (2) with some columns removed from the design matrix $\mathbf{X}$. We denote matrices of the form $X^{C_{j_1},\ldots,C_{j_k}}$ by X_a, with $a = \{C_{j_1}, \ldots, C_{j_k}\} \subseteq C$. The corresponding subvector of $\boldsymbol{\beta}$ is denoted by β_a.

Graphical Models

A powerful way for visualizing conditional dependencies among variables is given by a graph. A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consists of a finite set $\mathcal{V}$ of vertices and a finite set $\mathcal{E}$ of edges between these vertices. In our context, the vertices correspond to the different discrete random variables. We form the so-called *Conditional Independence Graph* by connecting all pairs of vertices that appear in the same generator, that is the maximal terms $a \subseteq C$ which are present in the model. To translate a vector $\boldsymbol{\beta}$ into a graphical model we look for $\beta_a \neq 0$ with $\beta_b = 0\ \forall a \subset b$ (where b is a strict super-set of a and $|a| > 1$) and we draw edges between all vertices corresponding to a. From this graph we can directly read off all marginal and conditional independences by the global Markov property for undirected graphs which states: if two sets of variables a and b are separated by a third set of variables c then a and b are conditionally independent given c ($a \perp\!\!\!\perp b|c$), where for three subsets a, b and c of $\mathcal{V}$, we say c separates a and b if all paths from a to b intersect c. For details, see [12].

Model selection - Non-Hierarchical versus hierarchical models

In the following subsections we introduce different model selection strategies for log-linear models. We first develop an ℓ_1-regularization model selection approach, which is then expanded to the new so-called *level-*ℓ_1-regularization approach. In addition, different Bayesian model selection strategies, which we use for comparisons, are explained in the *Additional file 2* Section. Hierarchical models are a subclass of models such that if an interaction term β_a is zero, then all higher order interaction terms β_b for $b \supseteq a$ are also zero. If we consider the example above with 2 levels, this means for example that if the first order interaction coefficient $\beta_{ij} = 0$ then all higher order interaction coefficients including i and j are also zero, i.e. $\beta_{ijk} = 0, \forall k$. While it is possible that the true underlying interaction model may not be hierarchical from a biological standpoint, a difficulty in the use of non-hierarchical models arises from the fact that they are not invariant under reparametrization. We have chosen the design matrix $\boldsymbol{X}$ with some constraints to ensure identifiability, and we used a specific, namely an orthonormal basis. In terms of ANOVA, this choice is equivalent to choosing a poly-contrast. We could have imposed different constraints or have chosen a different basis, and this would have resulted in a different design matrix $\boldsymbol{X}$ or in terms of ANOVA, a

different choice of contrast. Suppose we have found an interaction vector $\boldsymbol{\beta}$ for one parametrization of the log-linear model and that this vector corresponds to a non-hierarchical model, meaning there is at least one lower order interaction term β_a equal to zero, while $\beta_b \neq 0$ for at least one $b \supseteq a$. If we reparametrize the model, using a different design matrix, the coefficient for the model term a may no longer be zero. On the other hand, by reparametrizing a hierarchical model, all zero terms remain zero after reparametrization. Therefore, hierarchicity is preserved after reparametrization while non-hierarchicity depends on the parametrization. This is a distinct advantage of working within the hierarchical class. In a hierarchical model, all zero coefficients can directly be interpreted in terms of conditional independence, while this is not true for non-hierarchical models.

ℓ_1-Regularized model selection

The Lasso, originally proposed by [13] for linear regression, performs regularized parameter estimation and variable selection at the same time. The Lasso estimate is defined as follows:

$$\widehat{\boldsymbol{\beta}}^{\lambda} = \arg\min_{\boldsymbol{\beta}} \left[\sum_i (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})_i^2 + \lambda \sum_j |\beta_j| \right],$$

where $\mathbf{Y} = (Y_1, \ldots, Y_n)$ is the response vector. This can also be viewed as a penalized Maximum Likelihood estimator, as $\sum_i (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})_i^2$ is proportional to the negative log-likelihood function for Gaussian linear regression. While the MLE for the general regression model is no longer uniquely defined and very poor in the case of more variables than observations, the Lasso estimator is still reasonable as long as $\lambda > 0$. For our analysis, we have a similar problem, namely that the MLE does not exist in case of zero counts in the contingency table: a detailed description of the existence of the MLE in general log-linear interaction models is given in [14]. Inspired by the Lasso, we estimate our parameter vector $\boldsymbol{\beta}$ by the following expression:

$$\widehat{\boldsymbol{\beta}}^{\lambda} = \arg\min_{\boldsymbol{\beta}} \left[-l(\boldsymbol{\beta}) + \lambda \sum_j \mid \beta_j \mid \right], \tag{4}$$

where $l(\boldsymbol{\beta})$ is the log-likelihood function $l(\boldsymbol{\beta}) = \log \mathbb{P}_{\boldsymbol{\beta}}[\mathbf{n}] \propto \sum_{i=1}^{m} \frac{n_n}{n} (\mathbf{X}\boldsymbol{\beta})_n$. This minimization has to be calculated under the additional constraint that the cell probabilities add to 1:

$$\sum_{i=1}^{m} \exp\{(\mathbf{X}\boldsymbol{\beta})_i\} = 1. \tag{5}$$

A problem of the optimization (4) is that the solution is no longer independent of the choice of the orthogonal subspaces X_a. That is, if any set of orthogonal columns X_a of $\mathbf{X}$ is reparametrized by a

different orthogonal set, we get a different solution. To avoid this undesirable outcome we use a penalty that is intermediate between the ℓ_1- and the ℓ_2-penalty. This penalty, called group-ℓ_1-penalty, has the following form:

$$\sum_{a \subseteq C} \|\beta_a\|_{\ell_2}, \text{ where } \|\beta_a\|_{\ell_2}^2 = \sum_j (\beta_a)_j^2$$

Originally, this has been proposed by [15] for the linear regression problem with factor variables. The estimator of $\boldsymbol{\beta}$ then becomes

$$\widehat{\boldsymbol{\beta}}^{\lambda} = \arg\min_{\boldsymbol{\beta}} \left[-l(\boldsymbol{\beta}) + \lambda \sum_{\substack{a \subseteq C \\ a \neq \emptyset}} \|\beta_a\|_{\ell_2} \right], \tag{6}$$

subject to the constraint in (5). By imposing a penalty function on the coefficients of the log-linear interaction terms, overfitting as it might occur by using MLE is reduced. Furthermore, the ℓ_1-penalty encourages sparse solutions for the single components of $\boldsymbol{\beta}$, the group ℓ_1-penalty encourages sparsity at the interaction level, meaning that the vector β_a, which corresponds to the interaction term a is either present or absent in the model as a whole. In case of factors with only 2 levels, the group ℓ_1-penalty and the ℓ_1-penalty are equivalent.

For both the ℓ_1-, and the group ℓ_1-regularization, the parameter λ can be assessed by cross-validation: we divide the individual counts into a number of equal parts and in turn leave out one part for the rest to form a training contingency table with cell counts $\boldsymbol{n}_{train}$. The solution for an array of values for λ, the so-called solution path, is calculated according to an algorithm described in the following *Implementation* section. The corresponding vectors of cell probabilities are denoted by $p\left(\widehat{\boldsymbol{\beta}}^{\lambda}\right)$. We then use the remainder of the cell counts $\mathbf{n}_{test}$ to calculate the predictive negative log-likelihood score

$$\frac{-\sum_{i=1}^{m} \mathbf{n}_{test,i} \cdot \log\left(p_i(\widehat{\boldsymbol{\beta}}^{\lambda})\right)}{\sum_{i=1}^{m} \mathbf{n}_{test,i}}, \tag{7}$$

which is proportional to the out-of-sample negative log-likelihood. This score is on the same scale when varying the number of observations and may therefore be used to compare contingency tables of the same dimension but with different numbers of cell entries. The parameter λ is chosen as the value which minimizes the cross-validated score in (7). We use a ten-fold cross-validation in our example.

The resulting model does not necessarily have to be hierarchical and if we consider the hierarchical model induced by this procedure, it might happen that the final model is large for example if a single high order interaction is estimated to be active. To address this, we set up an algorithm described in the next Section.

Level-ℓ_1-regularized model selection

In order to prevent the procedure from choosing single high-order interactions, we alter the ℓ_1-regularized algorithm described in the previous Section: we do not exclusively apply it to the fully saturated model but also to submodels with lower order interactions. Precisely, a model is fitted with main effects only, and the predictive negative log-likelihood score (7) is calculated for the best main effects model (level 1). The same is done for the model including all main effects and first order interactions (level 2). Proceeding accordingly, we get $|C|$ log-likelihood scores corresponding to the $|C|$ levels. The level with minimal score (7) is then chosen (and within this selected level, we have an ℓ_1-regularized estimate).
With this procedure the tendency of including a single high-order interaction while most of its lower order interactions are absent is decreased, and the inclusion is only forced if the predictive negative log-likelihood score strongly speaks in favour of the inclusion. Therefore we tend to select sparser models which can be better hierarchized and interpreted in terms of conditional independence, in contrast to the ordinary ℓ_1-model selection procedure.

Algorithm for ℓ_1-regularization for factors with two levels

For the regularization approaches we calculate $\widehat{\boldsymbol{\beta}}^{\lambda}$ over a large number of values of λ in order to do some cross-validation using (7). For this purpose, an efficient algorithm is required. As one can easily verify by introducing Lagrange multipliers, finding the solution to (6) under the constraint (5) is equivalent to minimizing an unconstrained function $g(\boldsymbol{\beta})$:

$$g(\boldsymbol{\beta}) = -l(\boldsymbol{\beta}) + \sum_{i=1}^{m} \exp(\mu_i) + \lambda \sum_{\substack{a \subseteq C \\ a \neq \emptyset}} \|\beta_a\|_{\ell_2}, \tag{8}$$

with $\boldsymbol{\mu} = \mathbf{X}\boldsymbol{\beta}$ and $l(\boldsymbol{\beta}) \propto \sum_i \frac{n_i}{n}(\mathbf{X}\boldsymbol{\beta})_i$. Here, g is a convex function. If each factor has two levels only, as in our application with single-gene libraries, we can set up an algorithm, which efficiently yields the estimates for a whole sequence of parameters λ. Let $\mathcal{A}$ denote the set of active interaction terms, which means for $a \in \mathcal{A}$ it holds that $\beta_a \neq 0$; $\mathbf{X}_{\mathcal{A}}$ is the corresponding sub-matrix of $\mathbf{X}$, $\boldsymbol{\beta}_{\mathcal{A}}$ the corresponding sub-vector of $\boldsymbol{\beta}$ and $g_{\mathcal{A}}$ is g restricted to the subspace $\boldsymbol{\beta}_{\mathcal{A}}$. We restrict ourselves to the currently active set $\mathcal{A}$, where $\nabla g_{\mathcal{A}}$ and $\nabla^2 g_{\mathcal{A}}$ are well-defined:

$$\begin{aligned} \nabla g_{\mathcal{A}}(\boldsymbol{\beta}_{\mathcal{A}}, \lambda) &= -\mathbf{X}_{\mathcal{A}}^t \{\frac{\mathbf{n}}{n} - \cdot \exp(\mathbf{X}_{\mathcal{A}}\boldsymbol{\beta}_{\mathcal{A}})\} + \lambda(0, sign(\boldsymbol{\beta}_{\mathcal{A}}))^t \\ \nabla^2 g_{\mathcal{A}}(\boldsymbol{\beta}_{\mathcal{A}}, \lambda) &= \mathbf{X}_{\mathcal{A}}^t diag\{\exp(\mathbf{X}\boldsymbol{\beta})\}\mathbf{X}_{\mathcal{A}}. \end{aligned}$$

The algorithm, which is an adaption of the path following algorithm proposed by [16], is set up as follows:

(1) Start with $\widehat{\boldsymbol{\beta}} = (-\log(m), 0, \ldots, 0)$

(2) Set: $\lambda_0 = 1, \mathcal{A} = \{\emptyset\}$ and $t = 0$.

(3) While $(\lambda_t > \lambda_{min})$

(3.1) $\lambda_{t+1} = \lambda_t - \epsilon$

(3.2) $\mathcal{A} = \mathcal{A} \cup \{j \notin \mathcal{A} : |[\mathbf{X}^t \cdot \frac{\mathbf{n}}{n} - \exp\left(\mathbf{X}\widehat{\boldsymbol{\beta}}\right)]_j| > \lambda_{t+1}\}$

(3.3) $\widehat{\boldsymbol{\beta}}$ is updated as $\widehat{\boldsymbol{\beta}}_{t+1} = \widehat{\boldsymbol{\beta}}_t - \nabla^2 g_{\mathcal{A}}(\widehat{\boldsymbol{\beta}}_t, \lambda_{t+1})^{-1} \cdot \nabla g_{\mathcal{A}}(\widehat{\boldsymbol{\beta}}_t, \lambda_{t+1})$.

(3.4) $\mathcal{A} = \mathcal{A} \setminus \{j \in \mathcal{A} : |\widehat{\boldsymbol{\beta}}_{t+1,j}| < \delta\}$

(3.5) $t = t + 1$

The pairs $(\widehat{\boldsymbol{\beta}}_t, \lambda_t)$, obtained from the algorithm above, represent the estimates from (6) under the constraint (5) for a range of penalty parameters λ_t e.g. $(t = \epsilon, 2\epsilon, \ldots)$. The choice of the step length ϵ represents the tradeoff between computational complexity and accuracy. To increase accuracy, one can perform more than one Newton step (3.3) if the gradient starts deviating from zero. The coefficient δ is also flexible. Typically it is chosen in the order of ϵ. The lowest λ for which one wants the solution to be calculated is denoted by λ_{min}.

Technical details concerning the algorithm can be found in the Appendix.

Testing

Data

We choose the true underlying interaction vector $\boldsymbol{\beta}$ consisting of 5 factors of 2 levels. By enumerating the factors from 1 to 5, the generators of the model are $345 + 235 + 234 + 135 + 123 + 14$, which means that all third and fourth order interactions are absent, only five of ten second order interactions and all first order interactions are present. The corresponding coefficients of $\boldsymbol{\beta}$ are independently simulated using a normal distribution with mean zero and variance one.

Then, 250 draws from a multinomial distribution with probability vector $\mathbf{p}$ where $\log(\mathbf{p}) = \mathbf{X}\boldsymbol{\beta}$, are taken. This corresponds to a reasonable number of cDNAs in a single-gene library. This is then repeated 10 times. With our choice of $\boldsymbol{\beta}$, the resulting contingency tables are sparse. With the simulated cell counts, $\widehat{\boldsymbol{\beta}}$ is estimated with different methods described in the previous sections and these methods are then compared as follows:

Criteria

As a model selection score (MSS), we consider the fraction of correctly assigned model terms:

$$\text{MSS} = 1 - \frac{1}{m}\sum_{i=1}^{m} |1_{\{\boldsymbol{\beta_i}\neq 0\}} - 1_{\{\widehat{\boldsymbol{\beta}}_{\boldsymbol{i}}\neq 0\}}|.$$

Moreover, we consider the root mean squared error for the interaction coefficients,

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(\widehat{\beta}_i - \beta_i)^2}.$$

For assessing how much the estimation of $\boldsymbol{\beta}$ varies over multiple datasets, we calculate for every coefficient $\widehat{\beta}_i$ the estimated standard deviation $\widehat{\sigma}_i$. The means of these standard deviations are reported as

$$\text{SPREAD} = \frac{1}{m}\sum_{i=1}^{m}\widehat{\sigma}_i,$$

a measure of variability.

To compare the different procedures for estimation of probabilities $\mathbf{p} = \exp(\mathbf{X}\boldsymbol{\beta})$, we calculate the negative log-likelihood score (NLS) similar to the score in (7):

$$\text{NLS}(\widehat{\boldsymbol{\beta}}) = -\sum_{i=1}^{m} p_i \cdot \log\left(p_i(\widehat{\boldsymbol{\beta}})\right).$$

Results of simulation study

The results of the simulation study are summarized in Table 1, where we also include the MAP estimators of the Bayesian approaches described in the *Additional file* Section 2. We notice that the penalty-based regularization approaches proposed in this article leads to comparable or better results than the Bayesian approaches with respect to the NLS-score, RMSE and the variation (SPREAD), though the results of Bayesian approaches vary with the prior and the set of possible priors has not been extensively explored. The level- ℓ_1-regularization and the relaxed ℓ_1-regularization (see below) are both competitive and can be better than MCMC for model selection.

The results of the MCMC procedures are sensitive to the choice of the prior value or the prior distribution for σ^2. A flat prior for α_a ($\sigma^2 = 2$) results in worse performance than that of a prior that shrinks the coefficients more towards zero ($\sigma^2 = 1/2$). This suggests that specification of this prior hyperparameter may be difficult in practice, while we can easily optimize λ in the regularization approach by cross-validation.

The MCMC approaches without model selection perform poorly, as should be expected from data generated by a sparse model. MCMC methods based on a non-hierarchical model selection are also clearly inferior to the hierarchical counterpart. This is not surprising, as we have simulated data from a hierarchical model. In Table 1 we have also added an additional approach, denoted by ℓ_2, the equivalent to the ℓ_1-regularization but using an ℓ_2-penalty instead of an ℓ_1-penalty on the coefficients of the log-linear model. This method is equivalent to the MAP estimator with Gaussian priors on β_a, with the parameter of the distribution optimized by cross-validation. This Ridge-type method does not perform variable selection, but it is competitive for all other criteria that we assessed.

In addition we consider the *relaxed* ℓ_1-regularization approach. Rather than using a single penalty parameter λ, the idea of this method is to control variable selection and parameter estimation by incorporating two penalty parameters. For linear regression it has been proven theoretically as well as empirically [17] that under suitable conditions the relaxed ℓ_1-regularization is better than Lasso.

Overall, the level-ℓ_1-regularization has good model selection performance (high MSS score) in combination with low negative log-likelihood score (NLS) and a low mean squared error for the true $\boldsymbol{\beta}$ (RMSE). In addition, it is feasible to optimize the tuning parameter λ by cross-validation as the computational cost is very low compared to the MCMC approaches. On the other hand, posterior distributions of estimates from MCMC methods provide additional information about uncertainty in the model space, compared to point estimates from ℓ_1- or ℓ_2- regularization.

Implementation

Dataset

We estimate the splicing interaction pattern for a dataset corresponding to the *itpr1* gene, one of three mammalian genes encoding receptors for the second messenger inositol 1,4,5-trisphosphate (InsP_3). This gene is subject to alternative RNA splicing, with seven sites of transcript variation, 6 of these within the ORF and among these, $q = 5$ were completely assessed in the single-gene libraries. Five single-gene libraries were built, one for adult rat cerebrum as well as four for different stages of postnatal cerebellar development, namely on days 6, 12, 22 and 90, the latter being considered as adult. Each library consists of between 179 and 277 transcripts which were assessed, i.e. $\sum_{j=1}^{m} n_j \in [179, 277]$. This gene is 89% identical at the cDNA level and 95% identical at the amino acid level with the human receptor gene. The complete dataset can be found in [18].

Results of application to Single-Gene Libraries

Unless stated differently, we report the results using the level ℓ_1-penalization method. We display the interaction vector $\widehat{\boldsymbol{\beta}}$ graphically by plotting the components $\widehat{\beta}_j$ for the different tissue and development stages in Figure 1. Our results suggest that the exons interact mainly in pairs and there is no reliably estimated higher order interaction in the splicing interaction pattern of rat cerebellum. We further notice that the main interaction pattern is very well conserved over different developmental stages. A strong mutual interaction between exons number three, four and five can be observed in all development stages of rat cerebellum as well as in the cerebral tissue. The biggest changes in the interaction pattern during development of rat cerebellum occur from postnatal day six to day 12. This can be seen at position number 10 on the x-axis in Figure 1, and it corresponds to the first order interaction between exons two and three, and from day 12 to day 16, the first main effect changes in sign and magnitude. The first main effect decreases progressively from day 6 to adult, reversing in sign between day 12 and 22. Between day 22 and 90, the interaction pattern is strongly conserved. Comparing the splicing interaction patterns between cerebellum and cerebrum in the adult rat, we see a much more complex pattern in the cerebrum, involving several second order interactions, and therefore a clear distinction from that of the cerebellum.

The conditional independence graphs for the estimated log-linear models are drawn in Figure 2, where the thickness of the edges are proportional to the corresponding coefficient of the interaction vector $\widehat{\boldsymbol{\beta}}$ (the largest, if there are several giving rise to the same edge) and the radius of the vertices are chosen proportional to the corresponding main effect coefficient. Figure 2 graphically exploits the strongly conserved interactions between exons three, four and five. Except for a rather strong interaction between exon two and three on day six, all other interactions appear to be rather small. The graphical representation of the interaction pattern of adult rat cerebrum reveals a more complex interaction pattern with no conditional independences.

The approaches and results presented here can provide valuable insight into the underlying processes in alternative splicing in general, and specifically in the brain development experiments considered here. Most striking is the strong conservation over developmental stages at day 12, 22 and 90 (adult); some differences are showing between postnatal day six and day 12. Also, the conservation between the cerebellum and cerebrum is less pronounced than over developmental stages. Finally, second- or higher-order interaction terms seem to be of minor relevance, suggesting that in this gene/tissue combination, direct interaction mainly happens between pairs of exons, but not combinations of three or more exons.

We have also estimated $\boldsymbol{\beta}$ with the hierarchical Bayesian approach using MCMC. For the choice of $\sigma^2 = 1$

this resulted in very similar interaction patterns as for the level ℓ_1-penalization method. For $\sigma^2 = 2$ it led to remarkably different results. In addition to this, a further dataset was analyzed where the details can be found in the *Additional file* Section 3.

Conclusions

We have developed an efficient method for identifying interaction patterns of categorical variables. This can be used to fit a graphical model which is a valuable tool to visualize the conditional dependence structure among the random variables. In a simulation study, the results of the new level-ℓ_1-regularization method are superior in comparison to ℓ_1-regularization and slightly better than the MAP estimator from some of the MCMC methods we considered. With real data, the level ℓ_1-regularization and hierarchical Bayesian approach led to similar results, subject to a specific choice of priors for the Bayesian method. An important computational advantage of the level-ℓ_1-method in comparison to MCMC, is that cross-validation becomes feasible which in turn allows for an empirical choice of the tuning parameter. While the methodology described in this article is motivated by the study of exon splicing interactions in single-gene transcriptomes, it provides a general and flexible toolbox for regularization analysis in relatively high dimensional, sparse contingency tables. Model selection in high dimensional contingency tables has been a traditionally challenging area, and we hope that our generalization of regularization methodologies to this context will prove useful in a variety of areas of computational biology and biostatistics. Several technologies generate categorical data: these include SNP chips that provide genotype and copy number information at the DNA level, sequencing technologies, assays that study binding properties of proteins and binding of RNA to DNA, a variety of disease phenotypes, and more. In most of these contexts the interactions among the variables are critical features in systems biology investigations that aim at studying how the components of complex systems work together in influencing biological outcomes. For example, the log-linear models described here provide a natural approach for fitting very general classes of networks to discrete data. The level-ℓ_1-regularization is a general tool which can be applied to a wide variety of problems involving sparse contingency tables.

An R package called *logilasso* will be available for download on the Comprehensive R Archive Network (CRAN).

Authors' contributions

CD derived the mathematical details, implemented and tested the algorithm. GP initiated the project, suggested ideas and edited the manuscript. ME provided the datasets and the biological interpretation. PB supervised the project and suggested some of the main ideas. All authors read and approved the final manuscript.

Appendix

We note that if $\boldsymbol{\beta}$ is a minimum of g, then $\boldsymbol{\beta}_{\mathcal{A}}$ is a minimum of $g_{\mathcal{A}}$.

In our application with single-gene libraries, all factors have two levels only, which allows to construct an efficient algorithm. Since the gradient

$$\nabla \left[-l(\boldsymbol{\beta}) + \sum_{j=1}^{m} \exp\left(\mu_j\right) \right] = -\mathbf{X}^t \cdot \left(\frac{\mathbf{n}}{n} - \exp\left(\mathbf{X}\boldsymbol{\beta}\right)\right),$$

where $\exp(\mathbf{X}\boldsymbol{\beta})$ is understood as the componentwise exponential function, it follows that for a minimum $\boldsymbol{\beta}_{\mathcal{A}}$ of $g_{\mathcal{A}}$, the following equation holds:

$$\nabla g_{\mathcal{A}}(\boldsymbol{\beta}_{\mathcal{A}}) = -\mathbf{X}_{\mathcal{A}}^t \cdot \left(\frac{\mathbf{n}}{n} - \exp\left(\mathbf{X}_{\mathcal{A}}\boldsymbol{\beta}\right)\right) + (0, sign(\boldsymbol{\beta}_{\mathcal{A}}))^t \cdot \lambda = 0 \tag{9}$$

Without loss of generality, we can restrict ourselves to the subspace $\boldsymbol{\beta} \in \mathbb{R}^- \times \mathbb{R}^{m-1}$, because the constraint (5) can only be satisfied for $\beta_\emptyset < 0$ as is proved in the following Lemma 1. Therefore $\beta_\emptyset \in \mathcal{A}$.

Lemma 1. *$\beta_\emptyset < 0$ for a minimum of $g(\boldsymbol{\beta})$ for all $\lambda \in \mathbb{R}^+$.*

Proof.

$$\log\left(\mathbf{p}\right) = \mathbf{X}\boldsymbol{\beta} < 0 \text{ which yields } (1, \ldots, 1)\mathbf{X}\boldsymbol{\beta} = m\beta_\emptyset < 0 \text{ this implies } \beta_\emptyset < 0.$$

This holds because $(1, \ldots, 1)$ is orthogonal to all columns of $\mathbf{X}$ except for the first one. □

Additionally for $\boldsymbol{\beta}$ being a minimum, a necessary condition is:

$$|(\mathbf{X}^t \cdot \left(\frac{\mathbf{n}}{n} - \exp\left(\mathbf{X}\boldsymbol{\beta}\right)\right))_j| < \lambda, \forall j \notin \mathcal{A}. \tag{10}$$

Conditions (9) and (10) are sufficient for $\boldsymbol{\beta}$ being a minimum of (8). To find the $\boldsymbol{\beta}$'s that solve these equations for an array of values for λ, we set up a so-called path following algorithm. The idea is to start from an optimal solution $\boldsymbol{\beta}^{\lambda_0}$ for λ_0, and follow the path for decreasing λ, using a second-order

approximation for $\boldsymbol{\beta}_{\mathcal{A}}$. In the following, we restrict ourselves to the currently active set $\mathcal{A}$, omitting the index $\mathcal{A}$. It then holds:

$$\begin{aligned} \nabla g(\boldsymbol{\beta}_{t+1}, \lambda_{t+1}) &= 0 \approx \nabla g(\boldsymbol{\beta}_t, \lambda_{t+1}) + \nabla^2 g(\boldsymbol{\beta}_t, \lambda_{t+1}) \delta\boldsymbol{\beta}. \text{ This implies} \\ \delta\boldsymbol{\beta} &= -\nabla^2 g(\boldsymbol{\beta}_t, \lambda_{t+1})^{-1} \nabla g(\boldsymbol{\beta}_t, \lambda_{t+1}). \end{aligned} \tag{11}$$

The algorithm tries to follow the optimal path as close as possible. At each step, it aims to meet the conditions (9) and (10). In step (3.2), the active set $\mathcal{A}$ is identified, which forces $\widehat{\boldsymbol{\beta}}$ to meet the condition (10). In step (3.3), a Newton step as described in (11) is performed. Starting from a solution which meets condition (9), the new $\widehat{\boldsymbol{\beta}}^{\lambda}$ approximately meets (9) again.

Acknowledgements

CD was partially supported by the Swiss National Science Foundation grant number and by a PhD scholarship from the CC-SPMD. GP was partly supported by NSF grant DMS034211.

References

1. Liang F, Holt I, Pertea G, Karamycheva S, Salzberg S, Quackenbush J: **Gene index analysis of the human genome estimates approximately 120000 genes**. *Nature Genetics* 2000, **25**:239–240.
2. International Human Genome Sequencing Consortium: **Finishing the euchromatic sequence of the human genome**. *Nature* 2004, **431**:931–945.
3. Southan C: **Has the yo-yo stopped? An assessment of human protein-coding gene number**. *Proteomics* 2004, **4**:1712–1726.
4. Mironov A, Fickett J, Gelfand M: **Frequent alternative splicing of human genes**. *Genome Research* 1999, **9**:1288–1293.
5. Brett D, Hanke J, Lehmann G, Haase S, Delbruck S, Krueger SR, J Bork P: **EST comparison indicates 38% of human mRNAs contain possible alternative splice forms**. *FEBS Lett.* 2000, **474**:83–86.
6. International Human Genome Sequencing Consortium: **Initial sequencing and analysis of the human genome**. *Nature* 2001, **409**:860–921.
7. Brett D, Pospisil H, Valcarcel J, Reich J, Bork P: **Alternative splicing and genome complexity**. *Nature Genetics* 2002, **30**:29–30.
8. The FANTOM Consortium: **The transcriptional landscape of the mammalian genome**. *Science* 2005, **309**(5740):1559–1563.
9. Zavolan M, van Nimwegen E, Gaasterland T: **Splice variation in mouse full-length cDNAs identified by mapping to the mouse genome**. *Genome Research* 2003, **12**:1377–1385.
10. Imanishi T, Itoh T, Suzuki Y, O'Donovan C, Fukuchi S, Koyanagi m, Barrero m, Tamura T, Yamaguchi-Kabata Y, Tanino M: **Integrative annotation of 21037 human genes validated by full-length cDNA clones**. *PloS Biology* 2004, **2**:1–20.
11. Everitt BS: *The Analysis of Contingency Tables*. Monographs on Statistics and Applied Probability 45, Chapman and Hall, 2 edition 1992.
12. Lauritzen SL: *Graphical Models*. Oxford Statistical Science Series, 17, Oxford Clarendon Press 1996.
13. Tibshirani R: **Regression shrinkage and selection via the lasso**. *Journal of the Royal Statistical Society* 1996, **58**:267–288.

14. Christensen R: *Linear Models for Multivariate Time Series, and Spatial Data*. Springer-Verlag 1991.

15. Yuan M, Lin Y: **Model selection and estimation in regression with grouped variables**. *Journal of the Royal Statistical Society* 2006, **68**:49–67.

16. Rosset S: **Following Curved Regularized Optimization Solution Paths**. In *Advances in Neural Information Processing Systems 17*. Edited by Saul LK, Weiss Y, Bottou L, Cambridge, MA: MIT Press 2005:1153–1160.

17. Meinshausen N: **Lasso with relaxation**. *Computational Statistics & Data Analysis,* in press.

18. Regan MR, Lin DDM, Emerick MC, Agnew WS: **The effect of higher order RNA processes on changing patterns of protein domain selection: A developmentally regulated transcriptome of type 1 inositol 1,4,5-trisphosphate**. *Proteins: Structure, Function and Bioinformatics* 2005, **59**:312–331.

Figure legends

Figure 1 - Graphical display of interaction vector

The upper panel shows the estimated splicing interaction vectors $\widehat{\boldsymbol{\beta}}$ of rat cerebellum tissues at postnatal days six, 12 and 22. The lower panel shows the splicing interaction vector $\widehat{\boldsymbol{\beta}}$ of rat cerebellum tissues at the age of 90 days, which is considered adult, as well as the splicing interaction vector $\widehat{\boldsymbol{\beta}}$ of rat cerebral tissue at the age of 90 days. Within an interaction degree, the sequence of coefficients is ordered from left to right as follows: e.g. for 2nd order interactions, $123, 124, 125, \ldots, 345$, where $1, \ldots, 5$ represent exons 12, 23B, 40, 41, and 42 in the rip3r1 gene, as described in [18].

Figure 2 - Conditional Independence Graphs

Conditional independence graphs for the estimated log-linear models for the *itpr1* gene. For each graph, the predictive probability score (7) is reported as a goodness of fit measure. Note the strong mutual interaction between exons three, four and five.

Tables

Table 1 - Performance of different algorithms

Table 1: Performance of different algorithms

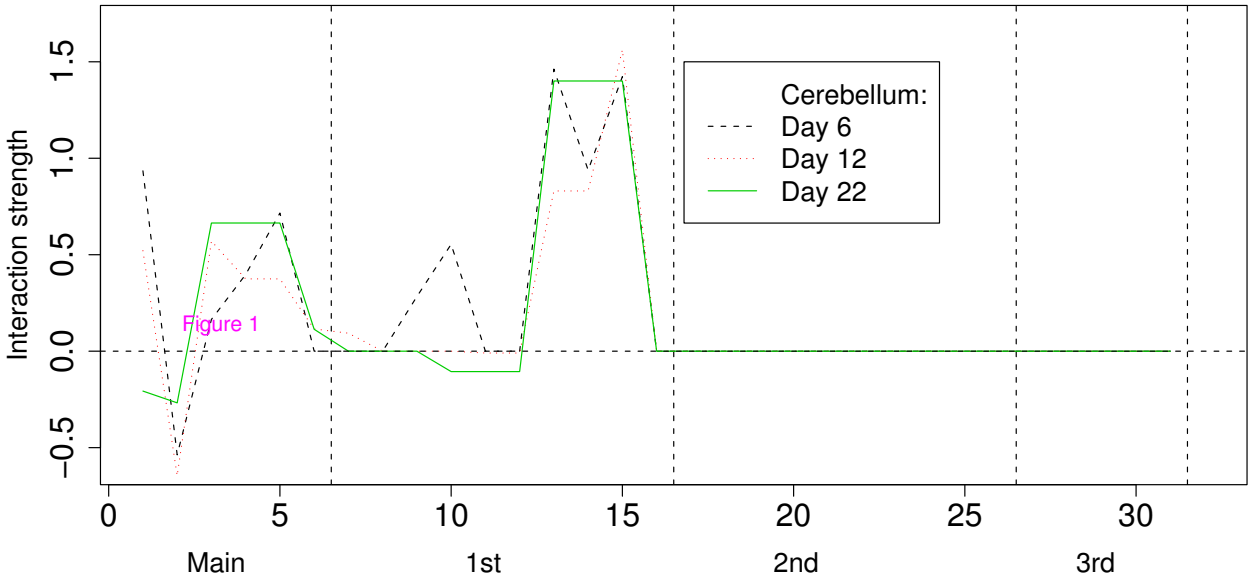
	MSS	NLS	RMSE	SPREAD
Penalty-based regularization methods:				
ℓ_1-regularization	69.7%	2.20	0.228	0.144
Level-ℓ_1-regularization	89.7%	2.22	0.237	0.179
Relaxed ℓ_1-regularization	82.2%	2.22	0.233	0.154
ℓ_2-regularization	-	2.20	0.238	0.130
MCMC without model selection:				
$\sigma^2 = 2$	-	2.32	0.747	0.401
$\sigma^2 = 1$	-	2.27	0.467	0.287
$\sigma^2 = 1/2$	-	2.24	0.294	0.201
MCMC with model selection:				
$\sigma^2 \sim \Gamma^{-1}(2, 3)$	81.5%	2.23	0.294	0.231
$\sigma^2 = 2$	76.6%	2.25	0.431	0.342
$\sigma^2 = 1$	78.4%	2.24	0.331	0.265
$\sigma^2 = 1/2$	76.6%	2.23	0.281	0.225
MCMC with hierarchical model selection:				
$\sigma^2 \sim \Gamma^{-1}(2, 3)$	84.1%	2.22	0.255	0.180
$\sigma^2 = 2$	80.6%	2.29	0.415	0.284
$\sigma^2 = 1$	83.4%	2.26	0.308	0.221
$\sigma^2 = 1/2$	83.4%	2.24	0.247	0.178
$\sigma^2 = 1/10$	86.3%	2.20	0.236	0.097
$\sigma^2 = 1/100$	69.7%	2.28	0.420	0.033

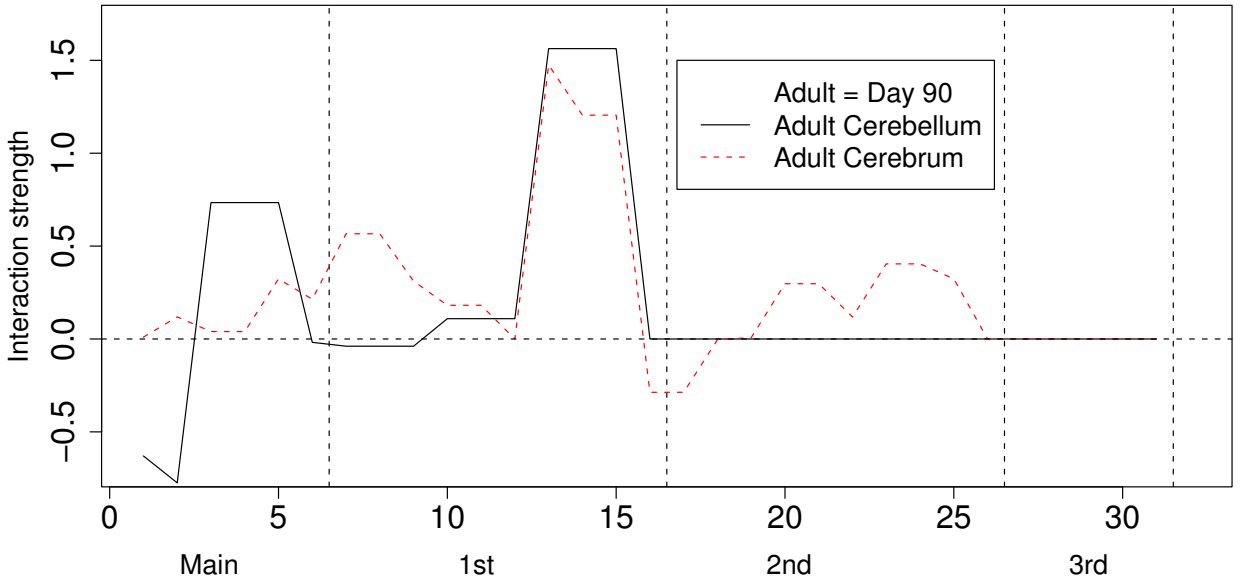
Comparison of different methods to estimate the interaction strength vector $\boldsymbol{\beta}$. MSS, NLS, RMSE and SPREAD are described in the *Implementation* section. The additional methods relaxed ℓ_1-regularization and ℓ_2-regularization listed in the Table are explained in the Results Section.

Description of Additional files

Additional files 1 to 3

The *AdditionalFiles.pdf* file consists of 3 sections. Section 1 - *Additional file 1* contains details concerning the parametrization of the log-linear model. The *Additional file 2* section describes some Bayesian model selection approaches, which were used for comparison with our algorithm. In the *Additional file 3* section a further dataset on which we tested our algorithm is introduced and the results are given on that dataset.


Interaction strength
1.5
1.0
0.5
0.0
−0.5
Cerebellum:
Day 6
Day 12
Day 22
Figure 1
0
5
10
15
20
25
30
Main
1st
2nd
3rd


Interaction strength
1.5
1.0
0.5
0.0
−0.5
Adult = Day 90
Adult Cerebellum
Adult Cerebrum
0
5
10
15
20
25
30
Main
1st
2nd
3rd

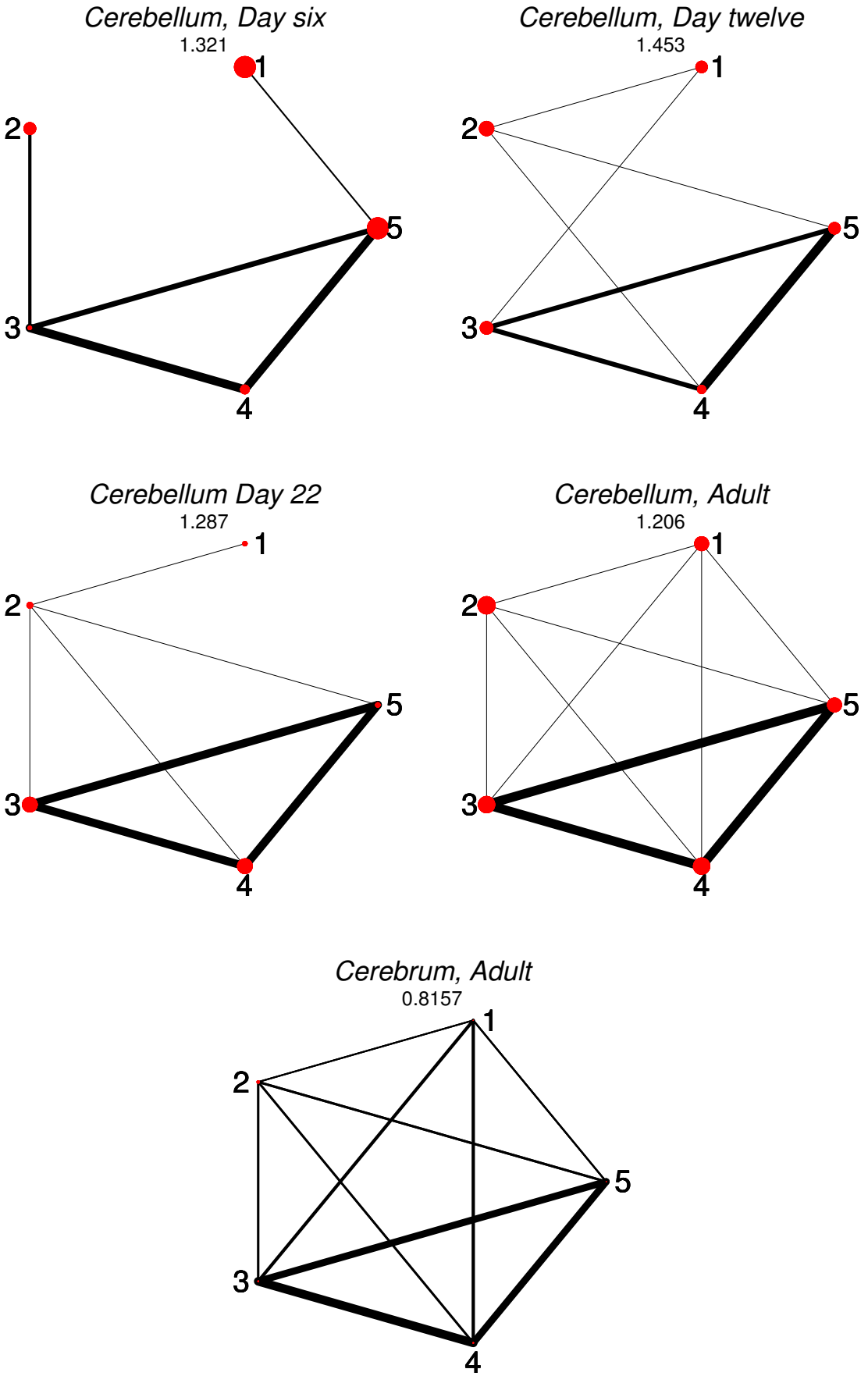

Cerebellum, Day six
1.321
1
2
3
4
5
Cerebellum, Day twelve
1.453
1
2
3
4
5
Cerebellum Day 22
1.287
1
2
3
4
5
Cerebellum, Adult
1.206
1
2
3
4
5
Cerebrum, Adult
0.8157
1
2
3
4
5

Figure 2

Additional files provided with this submission:

Additional file 1: additionalfiles.pdf, 190K
http://www.biomedcentral.com/imedia/3997621141690020/supp1.pdf