Bioinformatics, 33, 2017, i59–i66
doi: 10.1093/bioinformatics/btx226
ISMB/ECCB 2017

OXFORD

Rectified factor networks for biclustering of omics data

Djork-Arné Clevert,[1,*] Thomas Unterthiner,[2] Gundula Povysil[2] and Sepp Hochreiter[2,*]

[1]Bioinformatics Department, Bayer AG, Berlin, Germany and [2]Institute of Bioinformatics, Johannes Kepler University Linz, Linz, Austria

*To whom correspondence should be addressed.

Abstract

Motivation: Biclustering has become a major tool for analyzing large datasets given as matrix of samples times features and has been successfully applied in life sciences and e-commerce for drug design and recommender systems, respectively. **F**actor **A**nalysis for **Bi**cluster **A**cquisition (FABIA), one of the most successful biclustering methods, is a generative model that represents each bicluster by two sparse membership vectors: one for the samples and one for the features. However, FABIA is restricted to about 20 code units because of the high computational complexity of computing the posterior. Furthermore, code units are sometimes insufficiently decorrelated and sample membership is difficult to determine. We propose to use the recently introduced unsupervised Deep Learning approach Rectified Factor Networks (RFNs) to overcome the drawbacks of existing biclustering methods. RFNs efficiently construct very sparse, non-linear, high-dimensional representations of the input via their posterior means. RFN learning is a generalized alternating minimization algorithm based on the posterior regularization method which enforces non-negative and normalized posterior means. Each code unit represents a bicluster, where samples for which the code unit is active belong to the bicluster and features that have activating weights to the code unit belong to the bicluster.

Results: On 400 benchmark datasets and on three gene expression datasets with known clusters, RFN outperformed 13 other biclustering methods including FABIA. On data of the 1000 Genomes Project, RFN could identify DNA segments which indicate, that interbreeding with other hominins starting already before ancestors of modern humans left Africa.

Availability and implementation: https://github.com/bioinf-jku/librfn

Contact: djork-arne.clevert@bayer.com or hochreit@bioinf.jku.at

1 Introduction

Biclustering is widely used in statistics (Kasim *et al.*, 2016), machine learning (O'Connor and Feizi, 2014; Kolar *et al.*, 2011; Lee *et al.*, 2015) and bioinformatics (Cheng and Church, 2000; Hochreiter, 2013; Madeira and Oliveira, 2004; Povysil and Hochreiter, 2014, 2016), e.g. for analyzing large dyadic data given in matrix form, where one dimension are the samples and the other the features. A matrix entry represents a feature value for the according sample. A *bicluster* is a pair of a sample set and a feature set for which the samples are similar to each other on the features and vice versa. Biclustering simultaneously clusters rows and columns of a matrix. It clusters row elements that are similar to each other on a subset of column elements. In contrast to standard clustering, the samples of a bicluster are only similar to each other on a subset of features. Furthermore, a sample may belong to different biclusters or to no bicluster at all. Thus, biclusters can overlap in both dimensions. For example, in drug design biclusters are compounds which activate the same gene module and thereby indicate a side effect. In this example, different chemical compounds are added to a cell line and the gene expression is measured (Verbist *et al.*, 2015). If multiple pathways are active in a sample, it belongs to different biclusters and may have different side effects.

FABIA (**F**actor **A**nalysis for **Bi**cluster **A**cquisition, Hochreiter *et al.*, 2010) evolved into one of the most successful biclustering methods. A detailed comparison has shown FABIA's superiority over existing biclustering methods both on simulated data and real-world gene expression data (Hochreiter *et al.*, 2010). FABIA outperformed nonnegative matrix factorization with sparseness constraints

and state-of-the-art biclustering methods. It has been applied to genomics, where it identified in gene expression data task-relevant biological modules (Xiong *et al.*, 2014). In the large drug design project Quantitative Structure Transcriptional Activity Relationships (QSTAR), FABIA, was used to extract biclusters from a data matrix that contains bioactivity measurements across compounds (Verbist *et al.*, 2015). FABIA has been applied to genetic data, where it has been used to identify DNA segments that are identical by descent (IBD) in different individuals because these individuals inherited the segment from a common ancestor (Hochreiter, 2013; Povysil and Hochreiter, 2014). FABIA is a generative model that enforces sparse codes (Hochreiter *et al.*, 2010) and, thereby, detects biclusters. Sparseness of code units and parameters is essential for FABIA to find biclusters, since only few samples and few features belong to a bicluster. Each FABIA bicluster is represented by two membership vectors: one for the samples and one for the features. These membership vectors are both sparse since only few samples and only few features belong to the bicluster.

However, FABIA has shortcomings, too. A disadvantage of FABIA is that it is only feasible with about 20 code units (the biclusters) because of the high computational complexity which depends cubically on the number of biclusters, i.e. the code units. If less code units would be used, only the large and common input structures would be detected, thereby occluding the small and rare ones. Another shortcoming of FABIA is that units are insufficiently decorrelated and, therefore, multiple units may encode the same event or part of it. A third shortcoming of FABIA is that the membership vectors do not have exact zero entries, i.e. the membership is continuous must be thresholded for clear membership assignment. This threshold is difficult to adjust. A forth shortcoming is that biclusters can have large positive but also large negative members of samples (i.e. positive or negative code values). In this case, it is not clear whether the positive pattern or the negative pattern has been recognized.

Rectified factor networks (RFNs; Clevert *et al.*, 2015) overcome the shortcomings of FABIA. The first shortcoming of only few code units is avoided by extending FABIA to thousands of code units in a computationally feasible way. RFNs introduce rectified units to FABIA's posterior distribution and, thereby, allow for fast computations on graphical processing units (GPUs). Even though rectification is well established in Deep Learning by rectified linear units, the RFN approach is the first method which applies rectification to the posterior distribution of factor analysis and matrix factorization. RFNs transfer the methods for rectification from the neural network field to latent variable models. Addressing the second shortcoming of FABIA, RFNs achieve decorrelation by increasing the sparsity of the code units using dropout (Srivastava *et al.*, 2014), a method used in Deep Learning to avoid coadaptation of latent variables. RFNs also address the third shortcoming of FABIA: because the rectified posterior means yield exact zero values, membership to biclusters can be readily assigned to all non-zero values. Since RFNs only have non-negative code units, the forth problem of separating the negative from the positive pattern disappears, too.

2 Identifying biclusters by RFNs

We propose to use the recently introduced RFNs (Clevert *et al.*, 2015) for biclustering to overcome the drawbacks of the FABIA model. The factor analysis model and the construction of a bicluster matrix are depicted in Figure 1. RFNs efficiently construct very sparse, non-linear, high-dimensional representations of the input. RFN models identify rare and small events in the input, have a low interference between code units, have a small reconstruction error and explain the data covariance structure.

RFN learning is a generalized alternating minimization algorithm (Gunawardana and Byrne, 2005) derived from the posterior regularization method (Ganchev *et al.*, 2010) which enforces non-negative and normalized posterior means. These posterior means are the latent code of the input data. The RFN code can be computed very efficiently. For non-Gaussian priors, the computation of the posterior mean of a new input requires either to numerically solve an integral or to iteratively update variational parameters. In contrast, for Gaussian priors the posterior mean is the product between the input and a matrix that is independent of the input. RFNs use a rectified Gaussian posterior therefore; they have the speed of Gaussian posteriors but lead to sparse codes via rectification.

The RFN model is a factor analysis model

$$\boldsymbol{v} = \mathrm{W}\mathrm{h} + \epsilon\,, \tag{1}$$

which extracts the *covariance structure* of the data. The prior $\boldsymbol{h} \sim \mathcal{N}(0, \boldsymbol{I})$ of the hidden units (factors) $\boldsymbol{h} \in R^l$ and the noise $\epsilon \sim \mathcal{N}(0, \boldsymbol{\Psi})$ of visible units (observations) $\boldsymbol{v} \in R^m$ are independent. The model parameters are the weight (factor loading) matrix $\boldsymbol{W} \in R^{m \times l}$ and the noise covariance matrix $\boldsymbol{\Psi} \in \mathrm{R}^{\mathrm{m} \times \mathrm{m}}$.

RFN model selection is done via the posterior regularization method, that introduces a *variational distribution* $Q(\boldsymbol{h}|\boldsymbol{v}) \in \mathcal{Q}$ from a family $\mathcal{Q}$, which approximates the posterior $p(\boldsymbol{h}|\boldsymbol{v})$. We choose $\mathcal{Q}$ to constrain the posterior means to be non-negative and normalized. The full model distribution $p(\boldsymbol{h}, \boldsymbol{v})$ contains all model assumptions and, thereby, defines which structures of the data are modeled. $Q(\boldsymbol{h}|\boldsymbol{v})$ contains data dependent constraints on the posterior, therefore on the code.

For data $\{\boldsymbol{v}\} = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n\}$, it maximizes the objective $\mathcal{F}$:

$$\frac{1}{n}\sum_{i=1}^{n} \log p(\boldsymbol{v}_i) - \frac{1}{n}\sum_{i=1}^{n} D_{\mathrm{KL}}(Q(\boldsymbol{h}_i|\boldsymbol{v}_i) \parallel p(\boldsymbol{h}_i|\boldsymbol{v}_i)), \tag{2}$$

where D_{KL} is the Kullback-Leibler distance. Maximizing $\mathcal{F}$ achieves two goals simultaneously: (i) extracting desired structures and information from the data as imposed by the generative model and (ii) ensuring sparse codes via Q from the set of rectified Gaussians. In the variational framework, Q is the variational distribution and $\mathcal{F}$ is called the negative *free energy* Neal and Hinton (1998). If $p(\boldsymbol{h}|\boldsymbol{v}) \in \mathcal{Q}$, then $Q(\boldsymbol{h}|\boldsymbol{v}) = p(\boldsymbol{h}|\boldsymbol{v})$ and we obtain the classical EM algorithm. The EM algorithm maximizes the lower bound $\mathcal{F}$ on the log-likelihood as seen at the first line of Equation (2) and ensures in its E-step $Q(\boldsymbol{h}|\boldsymbol{v}) = p(\boldsymbol{h}|\boldsymbol{v})$.

For Gaussian posterior distributions, and mean-centered data $\{\boldsymbol{v}\} = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n\}$, the posterior $p(\boldsymbol{h}_i|\boldsymbol{v}_i)$ is Gaussian with mean vector $\left(\boldsymbol{\mu}_p\right)_i$ and covariance matrix $\boldsymbol{\Sigma}_{\mathrm{p}}$:

$$\begin{aligned} \left(\mu_p\right)_i &= \left(\boldsymbol{I} + \boldsymbol{W}^{\mathrm{T}}\boldsymbol{\Psi}^{-1}\boldsymbol{W}\right)^{-1}\boldsymbol{W}^T\boldsymbol{\Psi}^{-1}\,\boldsymbol{v}_i, \\ \Sigma_p &= \left(\boldsymbol{I} + \boldsymbol{W}^{\mathrm{T}}\boldsymbol{\Psi}^{-1}\boldsymbol{W}\right)^{-1}. \end{aligned} \tag{3}$$

For rectified Gaussian posterior distributions, $\boldsymbol{\Sigma}_{\mathrm{p}}$ remains as in the Gaussian case, but minimizing the second D_{KL} of Equation (2) leads to the constrained optimization problem (see Clevert *et al.* 2015) for a detailed description of the RFN objective and the algorithm's correctness and convergence.)

$$\begin{aligned} &\min_{\boldsymbol{\mu}_i} \ \frac{1}{n}\sum_{i=1}^{n}\left(\boldsymbol{\mu}_i - \left(\boldsymbol{\mu}_p\right)_i\right)^{\mathrm{T}}\Sigma_p^{-1}\left(\boldsymbol{\mu}_i - \left(\mu_p\right)_i\right) \\ &\text{s.t.} \ \ \forall_i : \boldsymbol{\mu}_i \geq \boldsymbol{0}\,, \ \forall_j : \frac{1}{n}\sum_{i=1}^{n}\mu_{ij}^2 = 1\,, \end{aligned} \tag{4}$$

where '$\geq$' is component-wise. In the E-step of the *generalized alternating minimization* algorithm (Gunawardana and Byrne, 2005),

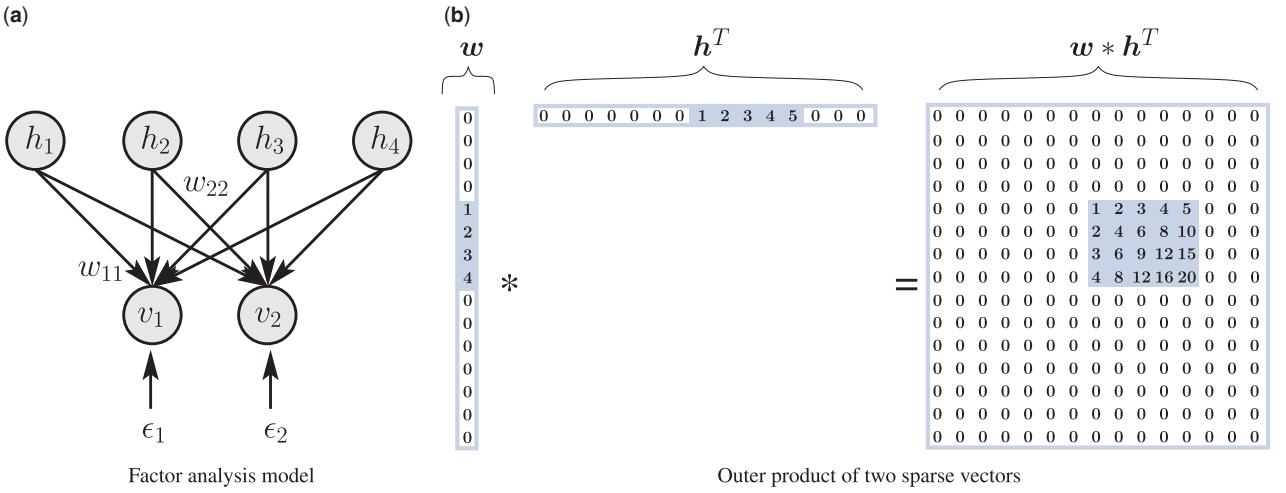

Fig. 1. Left: Factor analysis model: hidden units (factors) h, visible units v, weight matrix W, noise ϵ. Right: The outer product $w h^{\mathrm{T}}$ of two sparse vectors results in a matrix with a bicluster. Note that the non-zero entries in the vectors are adjacent to each other for visualization purposes only

which is used for RFN model selection, we only perform a step of the *gradient projection algorithm* (Bertsekas, 1976; Kelley, 1999), in particular a step of the *projected Newton method* for solving Equation (4) (Clevert *et al.*, 2015).

Therefore, RFN model selection is extremely efficient but still guarantees the correct solution. Additional speed-up is generated by implementing RFNs on GPUs.

2.1 RFN biclustering

For an RFN model, each code unit represents a bicluster, where samples for which the code unit is active, belong to the bicluster. On the other hand, features that activate the code unit belong to the bicluster, too. The vector of activations of a unit across all samples is the sample membership vector. The weight vector which activates the unit is the feature membership vector. The unconstrained posterior mean vector is computed by multiplying the input with a matrix according to Equation (3). The constrained posterior of a code unit is obtained by multiplying the input by a vector and subsequently rectifying and normalizing the code unit (Clevert *et al.*, 2015).

To keep feature membership vectors sparse, we introduce a *Laplace prior on the parameters of the original RFN model*. Therefore, only few features contribute to activating a code unit, that is, only few features belong to a bicluster. Sparse weights $\boldsymbol{W}_i$ are achieved by a component-wise independent *Laplace* prior for the weights:

$$p(\boldsymbol{W}_i) = \left(\frac{1}{\sqrt{2}}\right)^n \prod_{k=1}^{n} e^{-\sqrt{2}|W_{ki}|} \tag{5}$$

The weight update for RFN (Laplace prior on the weights) is

$$\boldsymbol{W} = \boldsymbol{W} + \eta\left(\boldsymbol{U}\,\boldsymbol{S}^{-1} - \boldsymbol{W}\right) - \alpha\,\mathrm{sign}(\boldsymbol{W}). \tag{6}$$

Whereby the sparseness of the weight matrix can be controlled by the hyper-parameter α and $\boldsymbol{U}$ and $\boldsymbol{S}$ are defined as $\boldsymbol{U} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{v}_i \boldsymbol{\mu}_i^{\mathrm{T}}$ and $\boldsymbol{S} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{\mu}_i \boldsymbol{\mu}_i^{\mathrm{T}} + \boldsymbol{\Sigma}$, respectively. To enforce more sparseness of the sample membership vectors, we introduce *dropout* of code units. Dropout means that during training some code units are set to zero at the same time as they get rectified. Dropout avoids co-adaptation of code units and reduces correlation of code units—another problem of FABIA which is solved.

RFN biclustering does not require a threshold for determining sample memberships to a bicluster since rectification sets code units to zero. Further crosstalk between biclusters via mixing up negative and positive memberships is avoided; therefore spurious biclusters appear less often.

2.2 Extraction of IBD segments from RFN biclusters

RFN biclusters that result from applying RFN to genotype data, represent individuals that are similar to each other because they share minor alleles of a subset of SNVs (single nucleotide variants). However, a bicluster does not automatically represent an IBD segment because RFN does not regard the physical location or the temporal order of the features (SNVs). Only shared minor alleles that accumulate locally constitute IBD segments as shown in Hochreiter (2013). To distinguish random minor allele matches extracted by RFN from true IBD segments, we compute a histogram of counts of the RFN model SNVs and calculate the probability of observing k or more counts by chance. Let p be the probability of a random minor allele match between t individuals. If n SNVs are in a segment of DNA, the probability of observing k or more model SNVs by chance in this segment is given by:

$$\Pr(K \geq k) = \sum_{i=k}^{n} \binom{n}{i} p^i (1-p)^{n-i}. \tag{7}$$

Towards this end, the routine implemented in HapFABIA (Hochreiter, 2013) was adjusted to extract IBD segments from RFN biclusters. The binomial test (Equation 7) is used as a first step to identify local accumulations of minor alleles that were extracted by RFN. In a second step IBD segments are disentangled and individuals reassigned. Later-on, spuriously correlated minor alleles are removed based on an exponential test on long physical distances. Finally, similar IBD segments that were separated in the first step because of their length are rejoined in the last step.

Pairwise IBD detection methods like fastIBD Browning and Browning (2011) or GERMLINE Gusev *et al.* (2009) directly look for shared continuous DNA segments and incorporate the likelihood of IBD in their original model. In contrast to that we first look for shared minor alleles in multiple individuals via biclusters and only in subsequent steps use local accumulations and likelihood computations to extract IBD segments from the biclusters.

3 Experiments

In this section, we will present numerical results on multiple synthetic and real-world datasets to verify the performance of our RFN biclustering algorithm, and compare it with various other biclustering methods.

3.1 Methods compared

To assess the performance of RFNs as unsupervised biclustering methods, we compare the following 14 biclustering methods:

1. **RFN**: rectified factor networks (Clevert *et al.*, 2015),
2. **FABIA**: factor analysis with Laplace prior on the hidden units (Hochreiter *et al.*, 2010; Hochreiter, 2013),
3. **FABIAS**: factor analysis with sparseness projection (Hochreiter *et al.*, 2010),
4. **MFSC**: matrix factorization with sparseness constraints (Hoyer, 2004),
5. **plaid**: plaid model (Chekouo *et al.*, 2015; Lazzeroni and Owen, 2002),
6. **ISA**: iterative signature algorithm (Ihmels *et al.*, 2004),
7. **OPSM**: order-preserving sub-matrices (Ben-Dor *et al.*, 2003),
8. **SAMBA**: statistical-algorithmic method for bicluster analysis (Tanay *et al.*, 2002),
9. **xMOTIF**: conserved motifs (Murali and Kasif, 2003),
10. **Bimax**: divide-and-conquer algorithm (Prelic *et al.*, 2006),
11. **CC**: Cheng-Church δ-biclusters (Cheng and Church, 2000),
12. **plaid_t**: improved plaid model (Turner *et al.*, 2003),
13. **FLOC**: flexible overlapped biclustering, a generalization of CC (Yang *et al.*, 2005) and
14. **spec**: spectral biclustering (Kluger *et al.*, 2003).

For a fair comparison, the parameters of the methods were optimized on auxiliary toy datasets. If more than one setting was close to the optimum, all near optimal parameter settings were tested. In the following, these variants are denoted as *method_ variant* (e.g. plaid_ss). For RFN we used the following parameter setting: 13 hidden units, a dropout rate of 0.1, 500 iterations with a learning rate of 0.1, and set the parameter α (controlling the sparseness on the weights) to 0.01.

3.2 Simulated datasets with known biclusters

In the following subsections, we describe the data generation process and results for synthetically generated data according to either a multiplicative or additive model structure.

3.2.1 Data with multiplicative biclusters

We assumed $n = 1000$ features and $l = 100$ samples and implanted $p = 10$ multiplicative biclusters. The bicluster datasets with p biclusters are generated by the following model:

$$\boldsymbol{X} = \sum_{i=1}^{p} \boldsymbol{\lambda}_i \boldsymbol{z}_i^{\mathrm{T}} + \Upsilon, \tag{8}$$

where $\Upsilon \in R^{n \times l}$ is additive noise; $\boldsymbol{\lambda}_i \in R^n$ and $\boldsymbol{z}_i \in R^l$ are the bicluster membership vectors for the ith bicluster. The $\boldsymbol{\lambda}_i$'s are generated by (i) randomly choosing the number N_i^{λ} of genes in bicluster i from $\{10, \ldots, 210\}$, (ii) choosing N_i^{λ} features randomly from $\{1, \ldots, 1000\}$, (iii) setting $\boldsymbol{\lambda}_i$ components not in bicluster i to $\mathcal{N}\left(0, 0.2^2\right)$ random values, and (iv) setting $\boldsymbol{\lambda}_i$ components that are in bicluster i to $\mathcal{N}(\pm 3, 1)$ random values, where the sign is chosen randomly for each gene. The $\boldsymbol{z}_i$'s are generated by (i) randomly choosing the number N_i^{z} of samples in bicluster i from $\{5, \ldots, 25\}$, (ii) choosing N_i^{z} samples randomly from $\{1, \ldots, 100\}$, (iii) setting $\boldsymbol{z}_i$ components not in bicluster i to $\mathcal{N}\left(0, 0.2^2\right)$ random values and (iv) setting $\boldsymbol{z}_i$ components that are in bicluster i to $\mathcal{N}(2, 1)$ random values. Finally, we draw the Υ entries (additive noise on all entries) according to $\mathcal{N}\left(0, 3^2\right)$ and compute the data $\boldsymbol{X}$ according to Equation (8). Using these settings, noisy biclusters of random sizes between 10×5 and 210×25 (features $\times$ samples) are generated. In all experiments, rows (features) were standardized to mean 0 and variance 1.

3.2.2 Data with additive biclusters

In this experiment, we generated biclustering data where biclusters stem from an additive two-way ANOVA model:

$$\boldsymbol{X} = \sum_{i=1}^{p} \boldsymbol{\theta}_i \odot \left(\boldsymbol{\lambda}_i \boldsymbol{z}_i^{\mathrm{T}}\right) + \Upsilon, \tag{9}$$

where $\theta_{ikj} = \mu_i + \alpha_{ik} + \beta_{ij}$ and $\odot$ is the element-wise product of matrices and both $\boldsymbol{\lambda}_i$ and $\boldsymbol{z}_i$ are binary indicator vectors which indicate the rows and columns belonging to bicluster i. The ith bicluster is described by an ANOVA model with mean μ_i, kth row effect α_{ik} (first factor of the ANOVA model), and jth column effect β_{ij} (second factor of the ANOVA model). The ANOVA model does not have interaction effects. Although the ANOVA model is described for the whole data matrix, only the effects on rows and columns belonging to the bicluster are used in data generation. Noise and bicluster sizes are generated as in previous Subsection 3.2.1. Data were generated for three different signal-to-noise ratios which are determined by the distribution from which μ_i is chosen: A1 (low signal) $\mathcal{N}\left(0, 2^2\right)$, A2 (moderate signal) $\mathcal{N}\left(\pm 2, 0.5^2\right)$ and A3 (high signal) $\mathcal{N}\left(\pm 4, 0.5^2\right)$, where the sign of the mean is randomly chosen. The row effects α_{ki} are chosen from $\mathcal{N}\left(0.5, 0.2^2\right)$ and the column effects β_{ij} are chosen from $\mathcal{N}\left(1, 0.5^2\right)$.

3.2.3 Results on simulated datasets

For method evaluation, we use the previously introduced biclustering consensus score for two sets of biclusters (Hochreiter *et al.*, 2010), which is computed as follows:

1. Compute similarities between all pairs of biclusters by the Jaccard index, where one is from the first set and the other from the second set.
2. Assign the biclusters of one set to biclusters of the other set by maximizing the assignment by the Munkres algorithm.
3. Divide the sum of similarities of the assigned biclusters by the number of biclusters of the larger set.

Step (3) penalizes different numbers of biclusters in the sets. The highest consensus score is 1 and only obtained for identical sets of biclusters.

Table 1 shows the biclustering results for these datasets. RFN significantly outperformed all other methods (t-test and McNemar test of correct elements in biclusters).

3.2.4 Runtime comparison

Our open-source implementation of RFN offers high-performance CPU and GPU versions. In a runtime comparison on synthetic data displayed in Figure 2, we can clearly see how execution times for RFN is much lower and scales much better with the number of biclusters than its main competitor FABIA. This comparison was run on an Intel i5-3470 CPU and an NVIDIA Titan X GPU.

3.3 Gene expression datasets

In this experiment, we test the biclustering methods on gene expression datasets, where the biclusters are gene modules. The genes that

Table 1. Results are the mean of 100 instances for each simulated dataset

	Mult. model	Add. model		
Method	M1	A1	A2	A3
RFN	**0.643 ± 7e-4**	**0.475 ± 9e-4**	**0.640 ± 1e-2**	**0.816 ± 6e-7**
FABIA	0.478 ± 1e-2	0.109 ± 6e-2	0.196 ± 8e-2	0.475 ± 1e-1
FABIAS	0.564 ± 3e-3	0.150 ± 7e-2	0.268 ± 7e-2	0.546 ± 1e-1
SAMBA	0.006 ± 5e-5	0.002 ± 6e-4	0.002 ± 5e-4	0.003 ± 8e-4
xMOTIF	0.002 ± 6e-5	0.002 ± 4e-4	0.002 ± 4e-4	0.001 ± 4e-4
MFSC	0.057 ± 2e-3	0.000 ± 0e-0	0.000 ± 0e-0	0.000 ± 0e-0
Bimax	0.004 ± 2e-4	0.009 ± 8e-3	0.010 ± 9e-3	0.014 ± 1e-2
plaid_ss	0.045 ± 9e-4	0.039 ± 2e-2	0.041 ± 1e-2	0.074 ± 3e-2
CC	0.001 ± 7e-6	4e-4 ± 3e-4	3e-4 ± 2e-4	1e-4 ± 1e-4
plaid_ms	0.072 ± 4e-4	0.064 ± 3e-2	0.072 ± 2e-2	0.112 ± 3e-2
plaid_t_ab	0.046 ± 5e-3	0.021 ± 2e-2	0.005 ± 6e-3	0.022 ± 2e-2
plaid_ms5	0.083 ± 6e-4	0.098 ± 4e-2	0.143 ± 4e-2	0.221 ± 5e-2
plaid_t_a	0.037 ± 4e-3	0.039 ± 3e-2	0.010 ± 9e-3	0.051 ± 4e-2
FLOC	0.006 ± 3e-5	0.005 ± 9e-4	0.005 ± 1e-3	0.003 ± 9e-4
ISA	0.333 ± 5e-2	0.039 ± 4e-2	0.033 ± 2e-2	0.140 ± 7e-2
spec	0.032 ± 5e-4	0.000 ± 0e-0	0.000 ± 0e-0	0.000 ± 0e-0
OPSM	0.012 ± 1e-4	0.007 ± 2e-3	0.007 ± 2e-3	0.008 ± 2e-3

Datasets M1 and A1–A3 were multiplicative and additive biclusters, respectively. The numbers denote average consensus scores with the true biclusters together with their standard deviations in parentheses. The best results are printed bold and the second best in italics ('better' means significantly better according to both a paired *t*-test and a McNemar test of correct elements in biclusters).

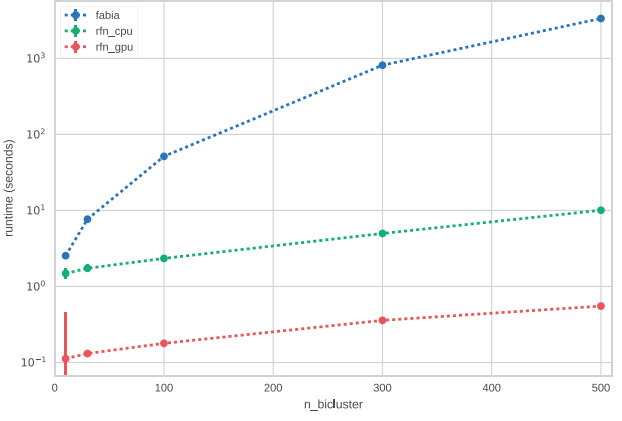

Fig. 2. Runtime comparison of FABIA and RFN for 10, 30, 100, 300 and 500 biclusters on synthetic inputs of $n = 500$ features and $l = 1000$ samples for 100 iterations each. Shown data are the median of five measurements, error bars are standard errors of the mean

are in a particular gene module belong to the according bicluster and samples for which the gene module is activated belong to the bicluster. We consider three gene expression datasets which have been provided by the Broad Institute and were previously clustered by Hoshida *et al.* (2007) using additional datasets as side information. Note, that Hoshida *et al.*'s clustering may include falsely assigned cluster memberships, which could affect the benchmark results.

1. The '*breast cancer*' *dataset* (vanta Veer *et al.*, 2002) was aimed at a predictive gene signature for the outcome of a breast cancer therapy. We removed the outlier array S54 which leads to a dataset with 97 samples and 1213 genes. In Hoshida *et al.* (2007), three biologically meaningful sub-classes were found that should be re-identified.

Table 2. Results on the (A) breast cancer, (B) multiple tissue samples, (C) DLBCL datasets measured by the consensus score

	(A) breast cancer				(B) multiple tissues				(C) DLBCL			
method	sco	#bc	#g	#s	sco	#bc	#g	#s	sco	#bc	#g	#s
RFN	**0.57**	3	73	31	**0.77**	5	75	33	*0.35*	2	59	72
FABIA	*0.52*	3	92	31	0.53	5	356	29	**0.37**	2	59	62
FABIAS	*0.52*	3	144	32	0.44	5	435	30	*0.35*	2	104	60
MFSC	0.17	5	87	24	0.31	5	431	24	0.18	5	50	42
plaid_ss	0.39	5	500	38	0.56	5	1903	35	0.30	5	339	72
plaid_ms	0.39	5	175	38	0.50	5	571	42	0.28	5	143	63
plaid_ms5	0.29	5	56	29	0.23	5	71	26	0.21	5	68	47
ISA_1	0.03	25	55	4	0.05	29	230	6	0.01	56	26	8
OPSM	0.04	12	172	8	0.04	19	643	12	0.03	6	162	4
SAMBA	0.02	38	37	7	0.03	59	53	8	0.02	38	19	15
xMOTIF	0.07	5	61	6	0.11	5	628	6	0.05	5	9	9
Bimax	0.01	1	1213	97	0.10	4	35	5	0.07	5	73	5
CC	0.11	5	12	12	nc	nc	nc	nc	0.05	5	10	10
plaid_t_ab	0.24	2	40	23	0.38	5	255	22	0.17	1	3	44
plaid_t_a	0.23	2	24	20	0.39	5	274	24	0.11	3	6	24
spec	0.12	13	198	28	0.37	5	395	20	0.05	28	133	32
FLOC	0.04	5	343	5	nc	nc	nc	nc	0.03	5	167	5

An 'nc' entry means that the method did not converge for this dataset. The best results are in bold and the second best in italics ('better' means significantly better according to a McNemar test of correct samples in clusters). The columns 'sco', '#bc', '#g', '#s' provide the consensus score, the numbers of biclusters, their average numbers of genes, and their average numbers of samples, respectively. RFN is two times the best method and once on second place.

2. The '*multiple tissue types*' *dataset* (Su *et al.*, 2002) are gene expression profiles from human cancer samples from diverse tissues and cell lines. The dataset contains 102 samples with 5565 genes. Biclustering should be able to re-identify the tissue types.

3. The '*diffuse large-B-cell lymphoma (DLBCL)*' *dataset* (Rosenwald *et al.*, 2002) was aimed at predicting the survival after chemotherapy. It contains 180 samples and 661 genes. The three classes found by Hoshida *et al.* (2007) should be re-identified.

For methods assuming a fixed number of biclusters, we chose five biclusters—slightly higher than the number of known clusters to avoid biases towards prior knowledge about the number of actual clusters. Besides the number of hidden units (biclusters) we used the same parameters as described in Section 3.1. The performance was assessed by comparing known classes of samples in the datasets with the sample sets identified by biclustering using the consensus score defined in Subsection 3.2.3—here the score is evaluated for sample clusters instead of biclusters. The biclustering results are summarized in Table 2. In two out of three datasets, RFN biclustering yielded significantly better results than all other methods and was on second place for the third dataset (significantly according to a McNemar test of correct samples in clusters).

3.4 1000 Genomes datasets

In this experiment, we used RFN for detecting DNA segments that are IBD. A DNA segment is IBD in two or more individuals, if it is identical because they have inherited it from a common ancestor, that is, the segment has the same ancestral origin in these individuals. Biclustering is well-suited to detect such IBD segments in a genotype matrix (Hochreiter, 2013; Povysil and Hochreiter, 2014, 2016), which has individuals as row elements and genomic SNVs as column elements. Entries in the genotype matrix usually count how often the minor allele of a particular SNV is present in a particular individual.

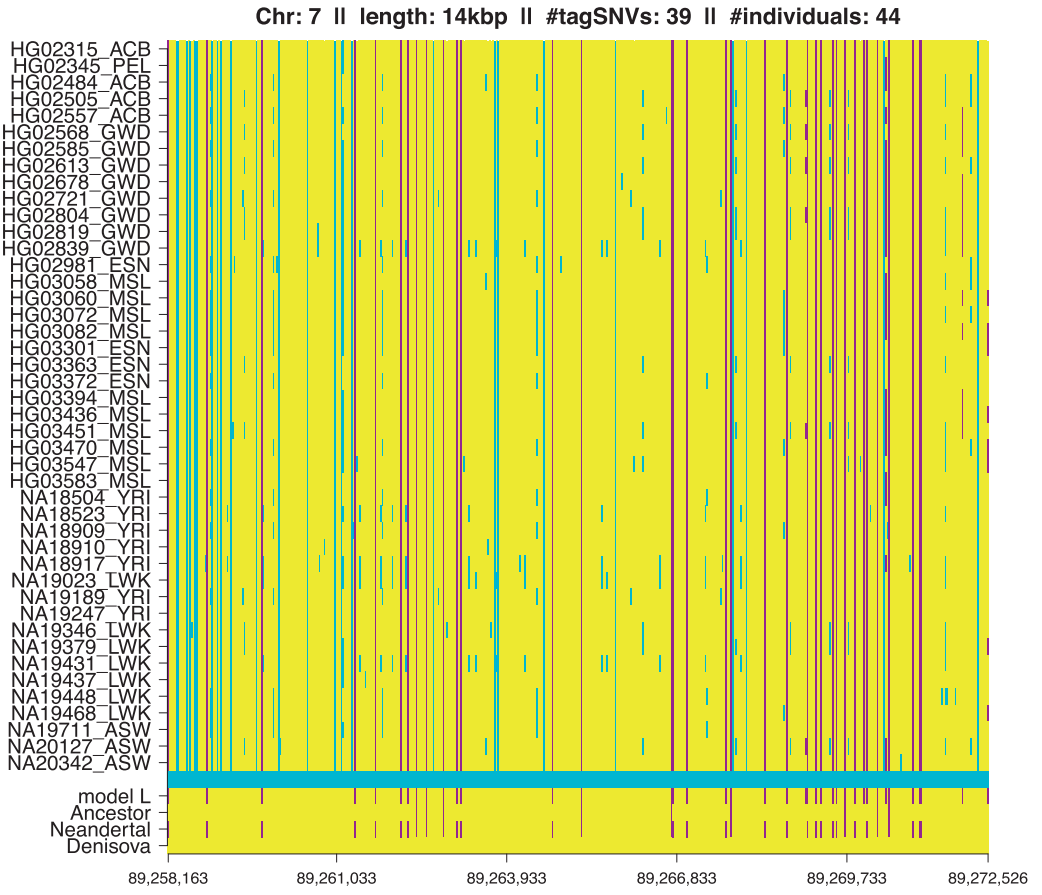

Fig. 3. Example of an IBD segment matching the Neanderthal genome shared among Africans and Admixed Americans. The rows represent all individuals that have the IBD segment, and columns represent consecutive SNVs. Major alleles are shown in yellow, minor alleles of tagSNVs in violet, and minor alleles of other SNVs in cyan. The row labeled *model L* indicates tagSNVs identified by RFN in violet. The rows *Ancestor, Neanderthal* and *Denisova* show bases of the respective genomes in violet if they match the minor allele of the tagSNVs (in yellow otherwise). For the *Ancestor genome* we used the reconstructed common ancestor sequence that was provided as part of the 1000 Genomes Project data

Individuals that share an IBD segment are similar to each other because they also share minor alleles of SNVs (tagSNVs) within the IBD segment, therefore IBD segments can be seen as biclusters.

For our IBD-analysis we used the next generation sequencing data from the 1000 Genomes Phase 3 (The 1000 Genomes Project Consortium, 2015) [ftp://ftp.1000genomes.ebi.ac.uk/Vol03325/ftp/release/20130502/ (31 October 2014, date last accessed)]. This dataset consists of low-coverage whole genome sequences from 2504 individuals of the main continental population groups (Africans, East Asians, South Asians, Europeans and Admixed Americans). Individuals that showed cryptic first degree relatedness to others were removed so that the final dataset consisted of 2493 individuals (see Povysil and Hochreiter, 2016). High-coverage genomes of the Altai Neanderthal and Denisovan were provided by the Max Planck Institute for Evolutionary Anthropology (Meyer *et al.*, 2012; Prüfer *et al.*, 2014) [http://cdna.eva.mpg.de/denisova/ (2 February 2012, date last accessed) and http://cdna.eva.mpg.de/neandertal/altai/, 23 May 2013, date last accessed]. Furthermore, we used the sequence of the reconstructed common ancestor of human, chimpanzee, gorilla, orang-utan, macaque and marmoset genomes which was part of the 1000 genomes project data.

Like Povysil and Hochreiter (2016), we restricted the analysis to SNVs and removed repeat regions and CpGs. RFN IBD detection is based on low frequency and rare variants (minor allele frequency < 0.05), therefore we removed common and private SNVs prior to the analysis. Afterwards, all chromosomes were divided into intervals of 10 000 SNVs with adjacent intervals overlapping by 5000 SNVs. RFN was applied to the unphased genotype data and IBD segments were extracted from biclusters as described in Section 2.2.

To distinguish true IBD segments from random findings we define an IBD score as the total sum of minor allele presences of individuals that share the IBD segment and tagSNVs that were extracted by RFN. True IBD segments should have an IBD score close to the number of individuals times the number of tagSNVs. To determine the significance of a finding, we calculate the empirical distribution of IBD scores based on 10E + 5 randomly sampled DNA segments of the same size as the detected segment. This allows us to calculate the *P*-value of our detected IBD segments under this H0 distribution. To get randomly sampled DNA segments of the same size, we first sample the same number of individuals from the total set of individuals and a start SNV that can be anywhere in the genome. Afterwards, we extract the genotype matrix consisting of the sampled individuals and a number of SNVs equal to the number of SNVs between the first and the last tagSNV of the IBD segment, beginning from the sampled start SNV. Finally, we sample tagSNVs from these SNVs and calculate the IBD score as described above. The depicted IBD segment in Figure 3 has a highly significant IBD score (*P*-value $<$ 1E-5).

In the data of the 1000 Genomes Project Phase 3, we found $>$1.5 million IBD segments. About 70% of the IBD segments were only

shared by Africans, while <1% were shared by individuals from all five continental populations. In contrast to HapFABIA, which was used for the analyses in Hochreiter (2013) and Povysil and Hochreiter (2016), IBD segments found with RFN require less post-processing because RFNs can extract much more biclusters and therefore IBD segments in a single run. Thus, problems caused by the iterative approach of HapFABIA can be avoided. To gain insights into the genetic relationships between humans, Neanderthals and Denisovans, we compared the detected IBD segments with the respective ancient genomes as described in Povysil and Hochreiter (2016). Furthermore, we excluded segments that were already present in the reconstructed ancestral sequence of all primates to distinguish IBD segments stemming from this ancestor from such that are due to later interbreedings. We could confirm that a surprisingly high number of IBD segments is shared between Africans and Neanderthals/Denisovans (see Fig. 3 for an example of an IBD segment that matches the Neanderthal genome). Neanderthal- and Denisova-matching IBD segments only observed in Africans are clearly shorter than IBD segments shared between non-Africans and the ancient genomes (5500 versus 12 500 bp and 5000 versus 12 000 bp, respectively for Neanderthal- and Denisova-matching segments). Since shorter segments are assumed to be older than longer ones (Povysil and Hochreiter, 2014), this is an indication of very early interbreedings within Africa that involved ancestors of Neanderthals and Denisovans, as well as ancestors of modern Africans (Povysil and Hochreiter, 2016).

4 Conclusion

We have introduced RFNs for biclustering and benchmarked it with 13 other biclustering methods on artificial and real-world datasets.

On 400 benchmark datasets with artificially implanted biclusters, RFN significantly outperformed all its competitors including FABIA. On three gene expression datasets with previously verified ground-truth, RFN biclustering yielded twice significantly better results than all other methods and was once the second best performing method. On data of the 1000 Genomes Project, RFN could identify IBD segments that previous IBD detection methods were unable to discover. Those detected segments support the hypothesis that interbreedings between ancestors of humans and other ancient hominins already have taken place in Africa.

RFN biclustering is geared to large datasets, sparse coding, many coding units and distinct membership assignment. Thereby RFN biclustering overcomes the shortcomings of FABIA and has the potential to become the new state of the art biclustering algorithm.

Acknowledgement

We thank the NVIDIA Corporation for supporting this research with several Titan X GPUs.

Funding

This work was was funded by the Institute of Bioinformatics.

Conflict of Interest: none declared.

References

Ben-Dor,A. *et al.* (2003) Discovering local structure in gene expression data: the order-preserving submatrix problem. *J. Comput. Biol.*, **10**, 373–384.

Bertsekas,D.P. (1976) On the Goldstein-Levitin-Polyak gradient projection method. *IEEE Trans. Automat. Control*, **21**, 174–184.

Browning,B.L. and Browning,S.R. (2011) A fast, powerful method for detecting identity by descent. *Am. J. Hum. Genet.*, **88**, 173–182.

Chekouo,T. *et al.* (2015) The gibbs-plaid biclustering model. *Ann. Appl. Stat.*, **9**, 1643–1670.

Cheng,Y. and Church,G.M. (2000) Biclustering of expression data. In *Proceedings of the International Conference on Intelligent Systems for Molecular Biology*, Vol. **8**, San Diego, U.S.A., pp. 93–103.

Clevert,D.-A. *et al.* Rectified factor networks. (2015) In: Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M. and Garnett, R. (eds.) *Advances in Neural Information Processing Systems 28* (NIPS), 2015, Montreal, Canada, Curran Associates, Inc.

Ganchev,K. *et al.* (2010) Posterior regularization for structured latent variable models. *J. Mach. Learn. Res.*, **11**, 2001–2049.

Gunawardana,A. and Byrne,W. (2005) Convergence theorems for generalized alternating minimization procedures. *J. Mach. Learn. Res.*, **6**, 2049–2073.

Gusev,A. *et al.* (2009) Whole population, genome-wide mapping of hidden relatedness. *Genome Res.*, **19**, 318–326.

Hochreiter,S. (2013) HapFABIA: Identification of very short segments of identity by descent characterized by rare variants in large sequencing data. *Nucleic Acids Res.*, **41**, e202.

Hochreiter,S. *et al.* (2010) FABIA: factor analysis for bicluster acquisition. *Bioinformatics*, **26**, 1520–1527.

Hoshida,Y. *et al.* (2007) Subclass mapping: Identifying common subtypes in independent disease data sets. *PLoS One*, **2**, e1195.

Hoyer,P.O. (2004) Non-negative matrix factorization with sparseness constraints. *J. Mach. Learn. Res.*, **5**, 1457–1469.

Ihmels,J. *et al.* (2004) Defining transcription modules using large-scale gene expression data. *Bioinformatics*, **20**, 1993–2003.

Kasim,A. *et al.* (2016) *Applied Biclustering Methods for Big and High-Dimensional Data Using R*. Chapman and Hall/CRC.

Kelley,C.T. (1999) *Iterative Methods for Optimization*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia.

Kluger,Y. *et al.* (2003) Spectral biclustering of microarray data: coclustering genes and conditions. *Genome Res.*, **13**, 703–716.

Kolar,M. *et al.* Minimax localization of structural information in large noisy matrices. (2011) In: Shawe-Taylor, J., Zemel, R.S., Bartlett, P.L., Pereira, F. and Weinberger, K.Q. (eds.) *Advances in Neural Information Processing Systems 24*, pp. 909–917. Curran Associates, Inc.

Lazzeroni,L. and Owen,A. (2002) Plaid models for gene expression data. *Stat. Sinica*, **12**, 61–86.

Lee,J.D. *et al.* (2015) Evaluating the statistical significance of biclusters. In: Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M. and Garnett, R. (eds.) *Advances in Neural Information Processing Systems 28* (NIPS), 2015, Montreal, Canada, pp. 1324–1332. Curran Associates, Inc.

Madeira,S.C. and Oliveira,A.L. (2004) Biclustering algorithms for biological data analysis: a survey. *IEEE ACM Trans. Comput. Biol. Bioinform.*, **1**, 24–45.

Meyer,M. *et al.* (2012) A high-coverage genome sequence from an archaic denisovan individual. *Science*, **338**, 222–226.

Murali,T.M. and Kasif,S. (2003) Extracting conserved gene expression motifs from gene expression data. In *Pacific Symposium on Biocomputing*, pp. 77ges.

Neal,R. and Hinton,G.E. (1998) A view of the EM algorithm that justifies incremental, sparse, and other variants. In: Jordan, M.I. (ed.) *Learning in Graphical Models*. MIT Press, Cambridge, MA, pp. 355–368.

O'Connor,L. and Feizi,S. (2014) Biclustering using message passing. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D. and Weinberger, K.Q. (eds.), *Advances in Neural Information Processing Systems 27* (NIPS), 2014, Montreal, Canada, Curran Associates, Inc., pp. 3617–3625.

Povysil,G. and Hochreiter,S. (2014) Sharing of Very Short IBD Segments between Humans, Neandertals, and Denisovans. *bioRxiv*. doi: 10.1101/003988.

Povysil,G. and Hochreiter,S. (2016) IBD Sharing between Africans, Neandertals, and Denisovans. *Genome Biol. Evol.*, **8**, 3406.

Prelic,A. *et al.* (2006) A systematic comparison and evaluation of biclustering methods for gene expression data. *Bioinformatics*, **22**, 1122–1129.

Prüfer,K. *et al.* (2014) The complete genome sequence of a Neanderthal from the Altai Mountains. *Nature*, **505**, 43–49.

Rosenwald,A. *et al*. (2002) The use of molecular profiling to predict survival after chemotherapy for diffuse large-B-cell lymphoma. *N. Engl. J. Med*., **346**, 1937–1947.

Srivastava,N. *et al*. (2014) Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res*., **15**, 1929–1958.

Su,A.I. *et al*. (2002) Large-scale analysis of the human and mouse transcriptomes. *Proc. Natl. Acad. Sci. USA*, **99**, 4465–4470.

Tanay,A. *et al*. (2002) Discovering statistically significant biclusters in gene expression data. *Bioinformatics*, **18**(Suppl. 1), S136–S144.

The 1000 Genomes Project Consortium (2015) A global reference for human genetic variation. *Nature*, **526**, 68–74. ISSN 0028-0836.

Turner,H. *et al*. (2003) Improved biclustering of microarray data demonstrated through systematic performance tests. *Comput. Stat. Data Anal*., **48**: 235–254.

van't Veer,L.J. *et al*. (2002) Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, **415**, 530–536.

Verbist,B. *et al*. (2015) Using transcriptomics to guide lead optimization in drug discovery projects: lessons learned from the QSTAR project. *Drug Discov. Today*, **20**, 505–513. ISSN 1359-6446.

Xiong,M. *et al*. (2014) Identification of transcription factors for drug-associated gene modules and biomedical implications. *Bioinformatics*, **30**, 305–309.

Yang,J. *et al*. (2005) An improved biclustering method for analyzing gene expression profiles. *Int. J. Artif. Intell. Tools*, **14**, 771–790.