UPRIGHT PUBLICATIONS
https://upright.pub/index.php/ijrsps/

Original Contribution

Biclustering of Omics Data using Rectified Factor Networks

Mani Manavalan[1]

Keywords: Biclustering, Omics Data, Rectified Factor Networks

International Journal of Reciprocal Symmetry and Theoretical Physics

Vol. 3, Issue 1, 2016 [Pages 1-10]

Biclustering has effectively been employed in biological sciences and e-commerce for medication design and recommender systems, respectively, and has become a prominent technique for evaluating big datasets presented as matrix of samples times attributes. One of the most successful biclustering methods, Factor Analysis for Bicluster Acquisition (FABIA), is a generative model in which each bicluster is represented by two sparse membership vectors: one for the samples and one for the features. Due to the high computational complexity of computing the posterior, FABIA is limited to approximately 20 code units. Additionally, code units are not always sufficiently decorrelated, making sample membership difficult to determine. To circumvent the limitations of existing biclustering approaches, we propose using the recently introduced unsupervised Deep Learning algorithm Rectified Factor Networks (RFNs). RFNs use their posterior means to efficiently build exceedingly sparse, non-linear, high-dimensional representations of the input. RFN learning is a generalized alternating minimization approach that ensures non-negative and normalized posterior means and is based on the posterior regularization method. Each code unit represents a bicluster, consisting of samples for which the code unit is active and features for which the code unit has activating weights. RFN beat 13 other biclustering algorithms, including FABIA, on four hundred benchmark datasets and three gene expression datasets with identified clusters. RFN was able to detect DNA sequences that imply interbreeding with other hominins began before modern humans' ancestors left Africa, based on data from the 1000 Genomes Project.

INTRODUCTION

Biclustering is widely used in statistics (Kasim et al., 2016), machine learning (Kolar et al., 2011; O'Connor and Feizi, 2014; Lee et al., 2015), and bioinformatics (Cheng and Church, 2000; Hochreiter, 2013; Madeira and Oliveira, 2004; Povysil and Hochreiter, 2014, 2016; Ganapathy, 2015; Manavalan, 2014), for example, when analyzing large dyadic data given. A feature value for a given sample is represented by a matrix entry.

A bicluster is a pair of sample sets and feature sets in which the samples are similar on features but not vice versa. Biclustering groups rows and columns of a matrix at the same time. On a subset of rows, it groups row components that are comparable. Elements of a column In contrast to traditional clustering, a sample of a sample of a sample of samples of samples of samples of samples of samples of samples of only a subset of features make biclusters comparable to one another.

A sample could also belong to multiple biclusters or none at all. As a result, biclusters can encroach

[1]Technical Project Manager, Larsen & Toubro Infotech (LTI), Mumbai, **INDIA**

on each other in both dimensions. Biclusters, for example, are chemicals that activate the same gene module, indicating a side effect in medication development. Different chemical substances are given to a cell line, and gene expression is assessed in this example (Verbist et al., 2015). When numerous routes are active in a sample, it is divided into biclusters and may have diverse side effects. Factor Analysis for Bicluster Acquisition (FABIA, Hochreiter et al., 2010) has become one of the most used biclustering techniques. On both simulated and real-world gene expression data, a comprehensive comparison revealed that FABIA outperforms existing biclustering approaches (Hochreiter et al., 2010). With sparseness constraints and cutting-edge biclustering techniques, FABIA surpassed nonnegative matrix factorization (Donepudi, 2014a).

Problem Statement

In genomics, it was used to identify task-relevant biological modules in gene expression data (Xiong et al., 2014). FABIA was used to generate biclusters from a data matrix containing bioactivity measurements across substances (Verbist et al., 2015) in the major drug design project Quantitative Structure Transcriptional Activity Relationships (QSTAR). FABIA has been used to detect DNA segments that are identical by descent (IBD) in different individuals because they inherited the segment from a common ancestor in genetic data (Hochreiter, 2013; Povysil and Hochreiter, 2014). FABIA (Hochreiter et al., 2010) is a generative model that discovers biclusters by enforcing sparse coding. Because biclusters contain only a few samples and features, sparseness of code units and parameters is required for FABIA to locate them. Two membership vectors are used to represent each FABIA bicluster: one for samples and another for features. Because there are few samples and features that belong to the bicluster, these membership vectors are both sparse.

FABIA, on the other hand, has flaws. Because of the high computational complexity, which is cubically proportional to the number of biclusters, or code units, FABIA is only viable with roughly 20 code units (biclusters). Only the large and common input structures would be discovered if fewer code units were employed, occluding the small and unusual ones. Another flaw with FABIA is that the units are not adequately decorrelated, resulting in many units encoding the same event or part of it. The membership vectors in FABIA do not have exact zero entries, which means that the membership must be thresholded for clear membership assignment. It's difficult to change this threshold. A fourth flaw is that biclusters might contain substantial positive and negative sample members (i.e. positive and negative code values). It's unclear whether the positive or negative pattern was recognized in this instance.

The drawbacks of FABIA are addressed by rectified factor networks (RFNs; Clevert et al., 2015). By extending FABIA to thousands of code units in a computationally possible method, the first flaw of only a few code units is avoided. RFNs add rectified units to FABIA's posterior distribution, allowing for faster computations on GPUs (GPUs). The RFN methodology is the first to use rectification to the posterior distribution of factor analysis and matrix factorization, despite the fact that rectified linear units are widely established in Deep Learning. From the neural network field to latent variable models, RFNs transfer rectification methods (Azad et al., 2011).

To address FABIA's second flaw, RFNs achieve decorrelation by increasing the sparsity of the code units through dropout (Srivastava et al., 2014), a Deep Learning technique for avoiding latent variable coadaptation. RFNs also address FABIA's third flaw: because the rectified posterior means provide exact zero values, all non-zero values may be easily attributed to bicluster membership. Because RFNs only have non-negative code units, the fourth challenge of distinguishing between negative and positive patterns is also solved.

Objectives of the Study

This study focuses on how to circumvent the limitations of existing biclustering approaches, we propose using the recently introduced unsupervised Deep Learning algorithm, Rectified Factor Networks (RFNs). RFNs use their posterior means to efficiently build exceedingly sparse, non-linear, high-dimensional representations of the input. RFN learning is a generalized alternating minimization approach that ensures non-negative and normalized posterior means and is based on the posterior regularization method.

LITERATURE REVIEW

Detecting Biclusters by RFNs

To solve the shortcomings of the FABIA model, we propose using the recently introduced RFNs (Clevert et al., 2015) for biclustering. Figure 1 depicts the factor analysis model and bicluster

matrix formation. RFNs are capable of constructing exceedingly sparse, non-linear, high-dimensional representations of the input with ease. RFN models detect infrequent and minor events in the input, have low code unit interference, a small reconstruction error, and can explain the data covariance structure (Rouf et al., 2014).

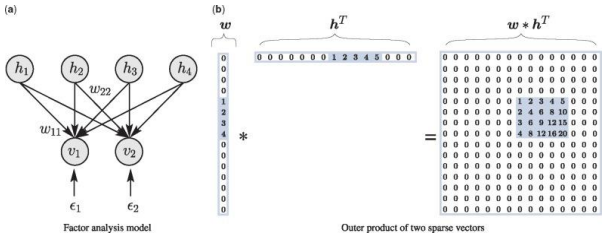

Figure 1: Left: Factor analysis model: hidden units (factors) h, visible units v, weight matrix W, noise. Right: The outer product whT of two sparse vectors results in a matrix with a bicluster. Note that the non-zero entries in the vectors are adjacent to each other for visualization purposes only

RFN learning is a simplified better cope approach (Gunawardana and Byrne, 2005) that enforces nonnegative and normalized posterior means. It is derived from the posterior regularization method (Ganchev et al., 2010). The latent code of the input data is these posterior means. It is possible to compute the RFN code in a relatively short amount of time. The estimation of the posterior mean of a new input with non-Gaussian priors necessitates either numerical integration or iterative updating of variational parameters. The posterior mean for Gaussian priors, on the other hand, is the product of the input and an independent matrix. As a result, RFNs use a rectified Gaussian posterior, which has the same speed as Gaussian posteriors but produces sparse codes due to rectification.

The RFN classical is a factor analysis model

$$v = Wh + \epsilon$$

which calculates the data's covariance structure The noise $\epsilon \sim N(0, \Psi)$ of visible units (observations) $v \epsilon R^m$ is independent of the preceding $h \sim N(0,1)$ of the hidden units (factors) $v \epsilon R^1$The $v \epsilon R^{m\,x\,1}$ weight (factor loading) and noise covariance matrices $\Psi \epsilon R^{m\,x\,m}$are the model parameters.

The posterior regularization method, which adds a variational distribution $Q\left(\frac{h}{v}\right) \epsilon Q$from a family Q to approximate the posterior phjv, is used to select RFN models. The posterior means are constrained to be non-negative and normalized using Q. All model assumptions are contained in the whole model distribution $p(h, v)$ which defines which data structures are modeled. On the posterior, and hence on the code $((Q\left(\frac{h}{v}\right))$ includes data-dependent limitations. For data $\{v\} = \{v_1, \ldots, v_n\}$, it maximizes the objective $\mathcal{F}$:

$$\frac{1}{n}\sum_{i=1}^{n} \log p\,(v_i) - \frac{1}{n}\sum\nolimits_{i=1}^{n} D_{KL}(Q(h_i|v_i)\,\|p\,(h_i|\,v_i))$$

The Kullback-Leibler distance is denoted by D_{KL}. Maximizing F accomplishes two objectives at once: (i) extracting desired structures and information from the data as dictated by the generative model, and (ii) assuring sparse codes from the set of corrected Gaussians via Q. Q is the variational distribution, and $\mathcal{F}$ is the negative free energy, according to Neal and Hinton's variational framework (1998). If $p(h|v)\,\epsilon\,Q$, $then\ Q(h|v)$, then $Q(h|v) = p(h|v)$, and the traditional EM algorithm is obtained.

For Gaussian posterior distributions, and mean-centered data $\{v\} = \{v_1, \ldots, v_n\}$, the posterior $p(h|v)$ is Gaussian with mean vector $(\mu_p)_i$ and covariance matrix $\sum_p$:

$$(\mu_p)_i = (I + W^T\Psi^{-1}W)^{-1}W^T\Psi^{-1}v_i$$
$$\sum\nolimits_p = (I + W^T\Psi^{-1}W)^{-1}$$

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

$$\min_{\mu_i} \frac{1}{n}\sum\nolimits_{i=1}^{n} (\mu_i - (\mu_p)_i))^T \sum_{p}^{-1} (\mu_i - (\mu_p)_i)$$
$$s.t. \forall_i: \mu_i \geq 0, \forall_j: \frac{1}{n}\sum\nolimits_{i=1}^{n} \mu_{ij}^2 = 1$$

where '≥' denotes a component. The generalized alternating minimization algorithm's E-step (Gunawardana and Byrne, 2005), for solving $\min_{\mu_i} \frac{1}{n} \sum_{i=1}^{n} (\mu_i - (\mu_p)_i))^T \sum_p^{-1} (\mu_i - (\mu_p)_i)$, we merely apply a stage of the gradient projection approach (Bertsekas, 1976; Kelley, 1999; Clevert et al., 2015).

As a result, RFN model selection is very efficient while yet ensuring that the correct solution is found. Implementing RFNs on GPUs provides additional speed (Ahmed & Dey, 2010).

RFN biclustering

Each code unit in an RFN model represents a bicluster, with the bicluster consisting of samples for which the code unit is active. The bicluster, on the other hand, also includes elements that activate the code unit. The sample membership vector represents the vector of a unit's activations across all samples. The feature membership vector is a weight vector that activates the unit. Equation is used to compute the unconstrained posterior mean vector by multiplying the input with a matrix (3). By multiplying the input by a vector and then rectifying and normalizing the code unit, the constrained posterior of the code unit can be derived (Clevert et al., 2015).

We apply a Laplace prior to the parameters of the original RFN model to make feature membership vectors sparse. As a result, only a few features contribute to the activation of a code unit, i.e., only a few features are bicluster-specific. A component-wise independent Laplace precondition for the weights is used to generate sparse weights W_i:

$$p(W_i) = (\frac{1}{\sqrt{2}})^n \prod_{k=1}^{n} e^{-\sqrt{2}|W_{ki}|}$$

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

$$W = W + \eta\,(US^{-1} - W) - \alpha sign(W)$$

The hyper-parameter a controls the sparseness of the weight matrix, while U and S are specified as $U = \frac{1}{n}\sum_{i=1}^{n} \mu_i \mu_i^T + \sum, and\ S = \frac{1}{n}\sum_{i=1}^{n} \mu_i \mu_i^T + \sum,$ respectively. Dropout of code units is used to impose better sparsity in the sample membership vectors. Some code units are set to zero at the same time that they are rectified during training, which is known as dropout. Dropout prevents code unit coadaptation and minimizes code unit correlation, which is another FABIA problem that is resolved. Because rectification sets code units to zero, RFN biclustering does not require a threshold for determining sample memberships to a bicluster. Further crosstalk between biclusters is prevented by mixing up negative and positive memberships, resulting in fewer bogus biclusters. Another FABIA problem that has been solved is the reduction of code unit correlation.

IBD segments were extracted from RFN biclusters

Individuals that are similar to each other because they share minor alleles of a subset of SNVs are represented by RFN biclusters, which are created by applying RFN to genotyping data (single nucleotide variants). However, because RFN does not consider the physical location or temporal order of the characteristics, a bicluster does not inevitably imply an IBD segment (SNVs). According to Hochreiter, IBD segments are made up of only shared minor alleles that accumulate locally (2013). We build a histogram of counts of the RFN model SNVs and evaluate the likelihood of witnessing k or more counts by chance to separate random minor allele matches derived by RFN from actual IBD segments. Let p be the likelihood of a minor allele match between t people at random. If n SNVs are present in a DNA segment, the probability of observing k or more model SNVs in this segment by chance is given by:

$$\Pr(\geq k) = \sum_{i=k}^{n} \binom{n}{i} p^i\,(1-p)^{n-1}$$

To do so, the HapFABIA (Hochreiter, 2013) procedure was tweaked to extract IBD segments from RFN biclusters. To identify local accumulations of minor alleles that were extracted by RFN, the binomial test is performed as a preliminary step. Individuals are reallocated after the IBD segments have been disentangled. Later, spuriously associated minor alleles are deleted using an exponential test across vast distances (Donepudi, 2014). Finally, in the last stage, identical IBD segments that were previously separated because to their length are reconnected.

Fast IBD Browning and Browning (2011) and GERMLINE Gusev et al. (2009) are pairwise IBD detection algorithms that directly look for shared continuous DNA segments and incorporate the likelihood of IBD into their original model. In contrast, we employ biclusters to find common minor alleles in several individuals, and then use local accumulations and probability computations to extract IBD segments from the biclusters in the following steps (Donepudi, 2015).

EXPERIMENTAL METHODS

The methods were compared

We compare the following 14 biclustering methods to see how well RFNs perform as unsupervised biclustering methods:

- RFN stands for rectified factor networks (Clevert et al., 2015),
- FABIA: factor analysis with Laplace prior on hidden units (Hochreiter et al., 2010; Hochreiter, 2013);
- FABIAS: factor analysis with sparseness projection (Hochreiter et al., 2010);
- FABIAS: factor analysis with sparseness projection (Hochreiter et al., 2010);
- plaid (Chekouo et al., 2015; Lazzeroni and Owen, 2002)
- Iterative Signature Process (ISA) is a six-step algorithm for generating signatures (Ihmels et al., 2004),
- Order-preserving sub-matrices (OPSM) are a type of OPSM that preserves the order of the sub-matrices (Ben-Dor et al., 2003),
- SAMBA (Statistical Algorithmic Method for Bicluster Analysis) is a statistical algorithmic method for bicluster analysis (Tanay et al., 2002; Manavalan & Bynagari, 2015),
- xMOTIF (conserved motifs) is a nine-letter acronym that stands for "conserved (Murali and Kasif, 2003),
- Divide-and-conquer algorithm (Bimax) (Prelic et al., 2006),
- Cheng-Church d-biclusters 11. CC: Cheng-Church (Cheng and Church, 2000),
- enhanced plaid model (plaid t) (Turner et al., 2003),
- FLOC is a generalization of CC and stands for flexible overlapped biclustering (Yang et al., 2005) spec: spectral biclustering and
- spec: spectral biclustering (Kluger et al., 2003).

The parameters of the techniques were optimized using auxiliary toy datasets for a fair comparison (Bynagari, 2015; Ahmed, 2012). All near-optimal parameter settings were examined if more than one setting was close to the ideal. These variants are referred to as method variants in the following (e.g. plaid ss). We used the following parameter settings for RFN: Set the parameter a (managing the sparseness on the weights) to 0.01. 13 hidden units, a dropout rate of 0.1, 500 iterations with a learning rate of 0.1, and the parameter a (controlling the sparseness on the weights) to 0.01.

Biclusters are known in simulated datasets

The data creation method and results for synthetically created data using a multiplicative or additive model structure are described in the subsections that follow.

Biclusters with multiplicative

We implanted p 14 10 multiplicative biclusters with n 14% 1000 features and l 14% 100 samples. The following model is used to construct bicluster datasets with p biclusters:

$$X = \sum_{i=1}^{p} \lambda_i z_i^T + \gamma$$

where are $\gamma \in R^{nx1}$ is additive noise, while $\lambda_i \in R^n$ and $z_i \in R^l$ are the ith bicluster's bicluster membership vectors. The λ_i are generated by

- randomly selecting N_i^{λ}genes in bicluster from 910;...; 210),
- randomly selecting N_i^{λ} features from (1;...; 1000),
- setting λ_i components not in bicluster i to N (0; 0:2^2) random values, and
- setting λ_i components in bicluster i to N(±3, 1 random values, where the sign is chosen randomly.

The z_i are created by (i) randomly selecting N_i^z samples in bicluster I from (5;...; 25), (ii) randomly selecting N_i^z samples from (1;...; 100), and (iii) setting z_i constituents not in bicluster *i* to N(0, 0.2^2) random values and (iv) setting z_i modules that are in bicluster *i* to *N*(2, 1); random values. Finally, we draw the γ entries (additive noise on all entries) according to N(0; 3^2) and compute the data X according to $X = \sum_{i=1}^{p} \lambda_i z_i^T + \gamma$. Using these settings, noisy biclusters of random sizes between 10 x 5 and 210 x 25 (features x samples) are generated. In all experiments, rows (features) were standardized to mean 0 and variance 1.

Data with additive biclusters

Biclustering data was collected in this experiment, with biclusters resulting from an additive two-way ANOVA model.

$$X = \sum_{i=1}^{p} \theta_i \odot (\lambda_i z_i^T) + \gamma$$

Where $\theta_{ikj} = \mu_i + \alpha_{ik} + \beta_{ij}\, and\, \odot$, is the element-wise product of matrices and λ_i and z_i are

binary indicator vectors indicating the rows and columns of bicluster *i*. An ANOVA model with mean μ_i, *kth* row effect α_{ik} (first component of the ANOVA model), and *jth* column effect β_{ij} (second element of the ANOVA model) describes the ith bicluster. Interaction effects are not included in the ANOVA model. Despite the fact that the ANOVA model is provided for the entire data matrix, only the effects on the bicluster's rows and columns are used in data production. Noise and bicluster sizes are created. Data was created for three different signal-to-noise ratios, each of which is dictated by the distribution from which μ_i is taken: *A1* (low signal) $N(0; 2^2)$, A2 (mid signal) $N(2; 0.5^2)$, and A3 (high signal) $N(4; 0.5^2)$, with the sign of the mean chosen at random. The row effects aki and the column effects β_{ij} are taken from $N(0.5; 0{:}2^2)$ and $N(1; 0.5^2)$, respectively.

RESULTS AND DISCUSSION

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

- Using the Jaccard index, compute similarities between all pairs of biclusters, one from the first set and the other from the second.
- Use the Munkres algorithm to maximize the assignment of biclusters from one set to biclusters from the other set.
- Multiply the sum of the allocated biclusters' similarities by the number of biclusters in the bigger set.

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

Different numbers of biclusters in the sets are penalized in step (iii). Only identical sets of biclusters get a 1 as the highest consensus score. The biclustering findings for these datasets are shown in Table 1. All other approaches (t-test and McNemar test of correct elements in biclusters) were significantly outperformed by RFN.

Runtime comparison

RFN is available in both CPU and GPU versions in our open-source implementation (Ahmed & Dey, 2009). Figure 2 shows how RFN's execution times are substantially shorter and scale far better with the number of biclusters than its major competitor FABIA, as seen in a runtime comparison on synthetic data. An Intel i5-3470 CPU and an NVIDIA Titan X GPU were used in this test.

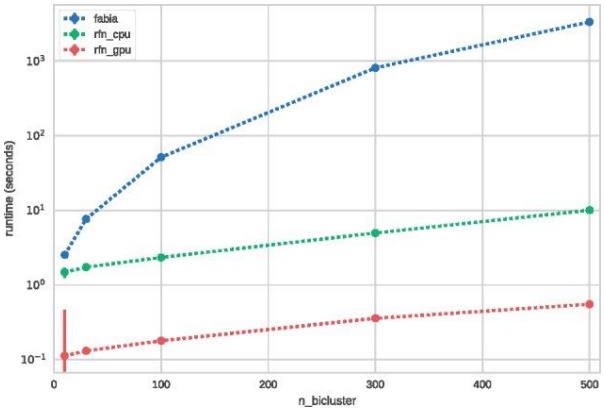

Figure 2: Runtime comparison of FABIA and RFN

Gene expression datasets

We use gene expression datasets to test biclustering algorithms, with the biclusters being gene modules. A bicluster is made up of genes that are in a specific gene module, as well as samples for which the gene module is active. Hoshida et al. (2007) used extra datasets as side information to cluster three gene expression datasets provided by the Broad Institute.

It's worth noting that Hoshida et al (2007) clustering's could contain incorrectly assigned cluster memberships, which could skew the benchmark findings.

- The 'breast cancer' dataset (vanta Veer et al., 2002) was created with the goal of identifying a gene signature that could predict the outcome of a breast cancer treatment. The outlier array S54 was eliminated, leaving a dataset with 97 samples and 1213 genes.

Three biologically significant sub-classes were discovered by Hoshida et al. (2007), and they should be renamed.

- Gene expression profiles from human cancer samples from various tissues and cell lines are included in the 'multiple tissue types' dataset (Su et al., 2002). There are 102 samples in the collection, totaling 5565 genes.
- The tissue types should be able to be re-identified using biclustering. The 'diffuse large-B-cell lymphoma (DLBCL)' dataset (Rosenwald et al., 2002) was created with the goal of predicting post-chemotherapy survival. There are 180 samples and 661 genes in it. Hoshida et al. (2007) identified three classes that should be renamed.

To prevent biases towards prior knowledge about the number of actual clusters, we picked five biclusters for approaches assuming a fixed number of biclusters. We utilized the identical settings except for the number of concealed units (biclusters). The performance was tested by comparing known classes of samples in the datasets with sample sets found by biclustering using the consensus score, the score is evaluated for sample clusters rather than biclusters. Table 2 summarizes the findings of the biclustering. RFN biclustering outperformed all other approaches in two of the three datasets and came in second in the third (significantly according to a McNemar test of accurate samples in clusters).

Table 2: Results on the (A) breast cancer, (B) multiple tissue samples, (C) DLBCL datasets

	(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

1000 Genomes datasets

In this investigation, RFN was employed to identify IBD DNA fragments. A DNA segment is IBD if it is identical in two or more persons because they inherited it from the same ancestor, that is, the segment has the same ancestral origin in these individuals. In a genotype matrix (Hochreiter, 2013; Povysil and Hochreiter, 2014, 2016), which has individuals as row elements and genomic SNVs as column elements, biclustering is well-suited to detect such IBD segments (Manavalan & Ganapathy, 2014). The minor allele of a particular SNV is usually present in a particular individual, so entries in the genotype matrix usually count how many times the minor allele of that SNV is present in that individual. Individuals that share an IBD segment are similar because they share minor alleles of SNVs (tag SNVs) inside the IBD segment, hence IBD segments can be thought of as biclusters.

We used 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/ (last accessed 31 October 2014)]. This collection contains low-coverage whole genome sequences from 2504 people from the continent's major ethnic groups (Africans, East Asians, South Asians, Europeans, and Admixed Americans). Individuals with cryptic first-degree relationships to others were deleted, leaving a final dataset of 2493 people (see Povysil and Hochreiter, 2016; Manavalan & Bynagari, 2015). The Max Planck Institute for Evolutionary Anthropology (Meyer et al., 2012; Prufer et al., 2014) provided high-coverage genomes of the Altai Neanderthal and Denisovan (http://cdna.eva.mpg.de/denisova/ (2 February 2021, date last accessed) and http://cdna.eva.mpg.de/neandertal/altai/, 23 May 2021, date last accessed). We also used data from the 1000 genomes project, which included the sequence of the reconstructed common ancestor of human, chimp, gorilla, orangutan, macaque, and marmoset genomes.

We limited our study to SNVs, excluding repeat regions and CpGs, as did Povysil and Hochreiter (2016). We deleted common and private SNVs before the analysis since RFN IBD identification is dependent on low frequency and rare variations (minor allele frequency 0.05). After that, all chromosomes were separated into 10 000 SNV intervals with 5000 SNV overlap between adjacent intervals. IBD segments were derived from biclusters after RFN was applied to unphased genotyping data. We used the same approach as Povysil and Hochreiter (2016), limiting the analysis

to SNVs and excluding repeat areas and CpGs. We eliminated common and private SNVs from the analysis since RFN IBD detection relies on low frequency and rare variants (minor allele frequency 0.05). Following that, all chromosomes were split into 10 000 SNV intervals with adjacent intervals overlapping by 5000 SNVs. The unphased genotyping data was subjected to RFN, and IBD segments were recovered from biclusters.

The cumulative sum of minor allele presences of individuals who share the IBD segment and tag SNVs retrieved by RFN is used to identify actual IBD segments from random finds. True IBD segments should have an IBD score that is proportional to the number of persons multiplied by the number of tag SNVs. We use $10E^5$ randomly picked DNA segments of the same size as the identified segment to construct the empirical distribution of IBD scores to estimate the significance of a result. Under the H0 distribution, we may calculate the P-value of our discovered IBD segments (Bynagari, 2014). Following that, we extract the genotyping matrix, which includes the sampled people as well as a number of SNVs equal to the number of SNVs between the first and last tag SNV of the IBD segment, starting with the sampled start SNV. Finally, we take a random sample of tag SNVs from these SNVs and compute the IBD score as previously explained. Figure 3 shows an IBD segment with a significantly significant IBD score (P-value < 1E-5).

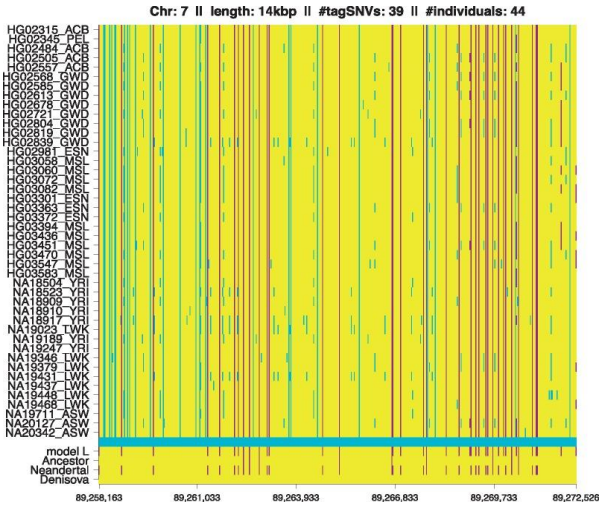

Figure 3: Example of an IBD segment matching

We discovered about > 1.5 million IBD segments in the 1000 Genomes Project Phase 3 data. Only Africans shared over 70% of the IBD segments, while individuals from all five continents shared < 1%. IBD segments discovered using RFN require less postprocessing than those found with HapFABIA, which was utilized in Hochreiter (2013) and Povysil and Hochreiter (2016). This is because RFNs can extract many more biclusters and thus IBD segments in a single run. As a result, difficulties created by HapFABIA's iterative method can be avoided. We compared the discovered IBD segments with the relevant ancient genomes as described by Povysil and Hochreiter to acquire insights into the genetic links between humans, Neanderthals, and Denisovans (2016). To identify IBD segments originating from this ancestor from those resulting from later interbreedings, we removed segments that were already present in the reconstructed ancestral sequence of all primates. We were able to confirm that Africans and Neanderthals/ Denisovans share a surprising number of IBD segments (see Fig. 3 for an example of an IBD segment that fits the Neanderthal genome).

Only Africans have Neanderthal- and Denisova-matching IBD segments, which are clearly shorter than IBD segments shared by non-Africans and ancient genomes (5500 versus 12 500 bp for Neanderthal- and Denisova-matching segments, respectively). Because shorter segments are thought to be older than longer ones (Povysil and Hochreiter, 2014), this suggests very early interbreedings inside Africa involving Neanderthals, Denisovans, and contemporary African ancestors (Povysil and Hochreiter, 2016).

CONCLUSION

On fake and real-world datasets, we introduced RFNs for biclustering and compared them to 13 existing biclustering approaches. RFN considerably outperformed all its competitors, including FABIA, on 400 benchmark datasets containing artificially implanted biclusters. RFN biclustering performed twice as well as all other approaches on three gene expression datasets with previously validated ground-truth. RFN identified IBD segments that earlier IBD detection algorithms had failed to find using data from the 1000 Genomes Project. These discovered parts back up the theory that human ancestors interbred with other ancient hominins in Africa. RFN biclustering is designed for big datasets with sparse coding, a large number of coding units, and different membership assignments. As a result, RFN biclustering outperforms FABIA and has the potential to become the new state-of-the-art biclustering method.

REFERENCES

Ahmed, A. A. A. (2012). Disclosure of Financial Reporting and Firm Structure as a Determinant: A Study on the Listed Companies of DSE. ASA University Review, 6(1), 43-60. https://doi.org/10.5281/zenodo.4008273

Ahmed, A. A. A., & Dey, M. M. (2009). Corporate Attribute and the Extent of Disclosure: A Study of Banking Companies in Bangladesh. Proceedings of the 5th International Management Accounting Conference (IMAC), OCT 19-21, 2009, UKM, Kuala Lumpur, MALAYSIA, Pages: 531-553. https://publons.com/publon/11427801/

Ahmed, A. A. A., & Dey, M. M. (2010). Accounting Disclosure Scenario: An Empirical Study of the Banking Sector of Bangladesh. Accounting and Management Information Systems, 9(4), 581-602. https://doi.org/10.5281/zenodo.4008276

Azad, M. R., Khan, W., & Ahmed, A. A. A. (2011). HR Practices in Banking Sector on Perceived Employee Performance: A Case of Bangladesh. Eastern University Journal, 3(3), 30–39. https://doi.org/10.5281/zenodo.4043334

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.

Bynagari, N. B. (2014). Integrated Reasoning Engine for Code Clone Detection. *ABC Journal of Advanced Research*, *3*(2), 143-152. https://doi.org/10.18034/abcjar.v3i2.575

Bynagari, N. B. (2015). Machine Learning and Artificial Intelligence in Online Fake Transaction Alerting. *Engineering International*, *3*(2), 115-126. https://doi.org/10.18034/ei.v3i2.566

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.

Donepudi, P. K. (2014). Voice Search Technology: An Overview. *Engineering International*, *2*(2), 91-102. https://doi.org/10.18034/ei.v2i2.502

Donepudi, P. K. (2014a). Technology Growth in Shipping Industry: An Overview. *American Journal of Trade and Policy*, *1*(3), 137-142. https://doi.org/10.18034/ajtp.v1i3.503

Donepudi, P. K. (2015). Crossing Point of Artificial Intelligence in Cybersecurity. *American Journal of Trade and Policy*, *2*(3), 121-128. https://doi.org/10.18034/ajtp.v2i3.493

Ganapathy, A. (2015). AI Fitness Checks, Maintenance and Monitoring on Systems Managing Content & Data: A Study on CMS World. *Malaysian Journal of Medical and Biological Research*, *2*(2), 113-118. https://doi.org/10.18034/mjmbr.v2i2.553

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.

Manavalan, M. (2014). Fast Model-based Protein Homology Discovery without Alignment. *Asia Pacific Journal of Energy and Environment*, *1*(2), 169-184. https://doi.org/10.18034/apjee.v1i2.580

Manavalan, M., & Bynagari, N. B. (2015). A Single Long Short-Term Memory Network can Predict Rainfall-Runoff at Multiple Timescales. *International Journal of Reciprocal Symmetry and Physical Sciences*, *2*, 1–7. Retrieved from https://upright.pub/index.php/ijrsps/article/view/39

Manavalan, M., & Bynagari, N. B. (2015). A Single Long Short-Term Memory Network can Predict Rainfall-Runoff at Multiple Timescales. *International Journal of Reciprocal Symmetry and Physical Sciences*, *2*, 1–7. Retrieved from https://upright.pub/index.php/ijrsps/article/view/39

Manavalan, M., & Ganapathy, A. (2014). Reinforcement Learning in Robotics. *Engineering International*, *2*(2), 113-124. https://doi.org/10.18034/ei.v2i2.572

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 IBDSegments 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.

Pru¨fer, 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.

Rouf, M. A., Hasan, M. S., & Ahmed, A. A. A. (2014). Financial Reporting Practices in the Textile Manufacturing Sectors of Bangladesh. *ABC Journal of Advanced Research*, *3*(2), 125-136. https://doi.org/10.18034/abcjar.v3i2.38

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.

--0--