OPEN ACCESS Freely available online

PLOS ONE

A Framework for Regularized Non-Negative Matrix Factorization, with Application to the Analysis of Gene Expression Data

Leo Taslaman[1], Björn Nilsson[1,2]*

1 Department of Hematology and Transfusion Medicine, Lund University, Lund, Sweden, **2** Broad Institute, Cambridge, Massachusetts, United States of America

Abstract

Non-negative matrix factorization (NMF) condenses high-dimensional data into lower-dimensional models subject to the requirement that data can only be added, never subtracted. However, the NMF problem does not have a unique solution, creating a need for additional constraints (regularization constraints) to promote informative solutions. Regularized NMF problems are more complicated than conventional NMF problems, creating a need for computational methods that incorporate the extra constraints in a reliable way. We developed novel methods for regularized NMF based on block-coordinate descent with proximal point modification and a fast optimization procedure over the alpha simplex. Our framework has important advantages in that it (a) accommodates for a wide range of regularization terms, including sparsity-inducing terms like the L^1 penalty, (b) guarantees that the solutions satisfy necessary conditions for optimality, ensuring that the results have well-defined numerical meaning, (c) allows the scale of the solution to be controlled exactly, and (d) is computationally efficient. We illustrate the use of our approach on in the context of gene expression microarray data analysis. The improvements described remedy key limitations of previous proposals, strengthen the theoretical basis of regularized NMF, and facilitate the use of regularized NMF in applications.

Citation: Taslaman L, Nilsson B (2012) A Framework for Regularized Non-Negative Matrix Factorization, with Application to the Analysis of Gene Expression Data. PLoS ONE 7(11): e46331. doi:10.1371/journal.pone.0046331

Editor: Magnus Rattray, University of Manchester, United Kingdom

Received May 9, 2012; **Accepted** August 31, 2012; **Published** November 2, 2012

Funding: This work was supported by the Swedish Foundation for Strategic Research, the Swedish Children's Cancer Fund, the Swedish Scientific Council, Marianne and Marcus Wallenberg's Foundation, the Swedish Society of Medicine, and the BioCARE initiative. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing Interests: The authors have declared that no competing interests exist.

* E-mail: bjorn.nilsson@med.lu.se

Introduction

Given a data matrix A of size $m \times n$, the aim of NMF is to find a factorization $A = WH^T$ where W is a non-negative matrix of size $m \times k$ (the component matrix), H is a non-negative matrix of size $n \times k$ (the mixing matrix), and k is the number of components in the model. Because exact factorizations do not always exist, common practice is to compute an approximate factorization by minimizing a relevant loss function, typically

$$\begin{aligned} &\text{minimize} && \|A - WH^T\|_F^2 \\ &\text{subject to} && W, H \geq 0, \end{aligned} \tag{1}$$

where $\|\cdot\|_F$ is the Frobenius norm. Other loss functions include Kullback-Leibler's, Bregman's, and Csiszar's divergences [1–4]. Problem 1 has been well studied and several solution methods proposed, including methods based on alternating non-negative least squares [5,6], multiplicative updates [1,3,7,8], projected gradient descent [9–11], and rank-one residue minimization [12] (reviews in refs. [9,13]).

The NMF problem is computationally hard. Particularly, an important property is that the factorization is not unique, as every invertible matrix S satisfying $WS \geq 0$ and $S^{-1}H^T \geq 0$ will yield another non-negative factorization $(WS)(S^{-1}H^T)$ of the same matrix as WH^T (simple examples of S matrices include diagonal re-scaling matrices) [14]. To reduce the problem of non-uniqueness, additional constraints can be included to find solutions that are likely to be informative/relevant with respect to problem-specific prior knowledge. While prior knowledge can be expressed in different ways, the extra constraints often take the form of regularization constraints (regularization terms) that promote qualities like sparseness, smoothness, or specific relationships between components [13]. At the same time, the computational problem becomes more complicated, creating a need for computation methods that are capable of handling the regularization constraints in a robust and reliable way.

We developed a novel framework for regularized NMF. This framework represents an advancement in several respects: first, our starting point is a general formulation of the regularized NMF problem where the choice of regularization term is open. Our approach is therefore not restricted to a single type of regularization, but accommodates for a wide range of regularization terms, including popular penalties like the L^1 norm; second, we use an optimization scheme based on block-coordinate descent with proximal point modification. This scheme guarantees that the solution will always satisfy necessary conditions for optimality, ensuring that the results will have a well-defined numerical meaning; third, we developed a computationally efficient procedure to optimize the mixing matrix subject to the constraint that the scale of the solution can be

controlled exactly, enabling standard, scale-dependent regularization terms to be used safely. We evaluate our approach on high-dimensional data from gene expression profiling studies, and demonstrate that it is numerically stable, computationally efficient, and identifies biologically relevant features. Together, the improvements described here remedy important limitations of earlier proposals, strengthen the theoretical basis of regularized NMF and facilitate its use in applications.

Results

Regularized Non-negative Matrix Factorization with Guaranteed Convergence and Exact Scale Control

We consider the regularized NMF problem

$$\begin{array}{ll}\text{minimize} & \|A-WH^T\|_F^2+\lambda\mathcal{R}(W)\\ \text{subject to} & W,\,H\geq 0,\,1^TH=\alpha 1^T,\end{array} \tag{2}$$

where $\mathcal{R}(W)$ is a regularization term, $\lambda>0$ determines the impact of the regularization term, and $1^TH=\alpha 1^T$ is an extra equality constraint that enforces additivity to a constant $\alpha>0$ in the columns H. While we have chosen to regularize W and scale H, it is clear that the roles of the two factors can be interchanged by transposition. We assume that $\mathcal{R}(W)$ is convex and continuously differentiable, but do not make any additional assumptions about $\mathcal{R}$ at this stage. Thus, we consider a general formulation of regularized NMF where one factor is regularized, the scale of the solution is controlled exactly, and the choice of regularization term still open.

The equality constraint that locks the scale of H is critical. The reason is that common regularization terms are scale-dependent. For example, this is the case for $\mathcal{R}(\cdot)=\|\cdot\|_1$ (L^1/LASSO regularization), $\mathcal{R}(\cdot)=\|\cdot\|_F^2$ (L^2/Tikhonov regularization), and $\mathcal{R}(\cdot)=\|\Gamma\cdot\|_F^2$ (L^2 regularization with an inner operator Γ that encodes spatial or temporal relationships between variables). Scale-dependent regularization terms will pull W towards zero, and indirectly inflate the scale of H unboundedly. Locking the scale of the unregularized factor prevents this phenomenon.

To solve Problem 2, we explored an approach based on block coordinate descent (BCD). In general, the BCD method is useful for minimizing a function F when the coordinates can be partitioned into N blocks such that, at each iteration, F can be minimized (at low computational cost) with respect to the coordinates of one block while the coordinates in the other blocks are held fixed. The method can be expressed as the update rule

$$\mathbf{x}_i\leftarrow\underset{\mathbf{x}\in\Omega_i}{\arg\min}\,F(\ldots,\mathbf{x}_{i-1},\mathbf{x},\mathbf{x}_{i+1},\ldots),$$

where $\mathbf{x}_i$ and Ω_i denote the coordinates and domain of the ith block, respectively. The updates are applied to all coordinate blocks in cyclic order. In the case of NMF, there are three natural ways to define blocks: per-column, per-row, or per-matrix. We partition the coordinates of H per column whereas the partitioning of W depends on the anatomies of $\mathcal{R}$ and the subproblem solver (details below).

Regarding the convergence of BCD procedures, it can be shown that if the domain for the ith coordinate block, Ω_i, is compact and all subproblems are strictly convex (that is, Ω_i is convex and F is strictly convex over Ω_i), the sequence generated by a BCD procedure has at least one limit point and each limit point is a critical point of the original function F [15]. In this context, we say that an algorithm has converged if the current point is within a tolerance from a critical point (that is, a point (W,H) where the derivative of the objective function is non-negative in all feasible directions; the first-order necessary condition for optimality). If F is convex but no longer strictly convex in Ω_i, limit points are still guaranteed to exist but are not necessarily critical points (that is, the solution may not satisfy the first-order criterion for optimality).

In Problem 2, the clamping of the scale bounds H and, indirectly, also W. Hence, all Ω_i's are bounded. Because they are also closed, they are compact. However, subproblems that are not strictly convex may still occur. To guarantee solutions that represent critical points, we therefore need to safeguard against non-strict convexity in the BCD subproblems. To this end, we add a *proximal point term* to objective functions of subproblems that are not known to be strictly convex beforehand. A proximal point term penalizes the Euclidean distance to the previous point in Ω_i, makes the subproblems strictly convex, and guarantees that limit points of the generated sequence are critical points of the *original* function F [16]. The BCD updates change to

$$\mathbf{x}_i\leftarrow\underset{\mathbf{x}\in\Omega_i}{\arg\min}\,F(\ldots,\mathbf{x}_{i-1},\mathbf{x},\mathbf{x}_{i+1},\ldots)+\delta_i\|\mathbf{x}-\mathbf{x}_i\|^2,$$

where $\delta_i\|\mathbf{x}-\mathbf{x}_i\|^2$ is the proximal point term and $\delta_i\geq 0$ a small number which can be zero if F is known to be strictly convex in Ω_i (in this case the proximal point term is not needed).

Optimizing the Mixing Coefficients

We developed an efficient procedure to optimize each block (column) of the mixing matrix H. The procedure itself is given in Algorithm 1. This section describes the proof.

The constraints $H\geq 0$ and $1^TH=\alpha 1$ imply that columns of H must lie in the α-*simplex*, defined as

$$\Delta^\alpha=\left\{\mathbf{x}\in\mathbb{R}_+^n\;|\;\sum_{i=1}^n x_i=\alpha\right\}.$$

Geometrically, this is the intersection of the non-negative orthant and a hyperplane with normal vector 1^T and offset α from the origin. The set is convex and also compact, meaning the conditions for a BCD to converge discussed in the previous section are satisfied.

We first derive general optimality criteria for convex functions on Δ^α. Let $f:\Delta^\alpha\rightarrow\mathbb{R}$ be convex and differentiable. By definition, $\mathbf{x}\in\Delta^\alpha$ is a minimum of f, if and only if the directional derivative at $\mathbf{x}$ is non-negative in every feasible direction

$$\nabla f^T(\mathbf{y}-\mathbf{x})\geq 0,\;\forall\mathbf{y}\in\Delta^\alpha. \tag{3}$$

Considering the special cases $\mathbf{y}=(0,\ldots,0,y_i=\alpha,0,\ldots,0)^T$, we see that

$$\frac{\partial f}{\partial x_i}\geq\frac{1}{\alpha}\nabla f^T\mathbf{x},i=1,\ldots,n \tag{4}$$

must hold if $\mathbf{x}$ is a minimum. However, the converse also holds. Assuming that Equation 4 holds and letting $\mathbf{y}$ be an arbitrary point in Δ^α, we have

$$\nabla f^T \mathbf{y} \geq \frac{1}{\alpha} \nabla f^T \mathbf{x} \sum_i y_i = \nabla f^T \mathbf{x}.$$

Hence, Equation 3 and Equation 4 are equivalent. Moreover, Equation 4 can be rephrased as

$$\min_k \frac{\partial f}{\partial x_k} \geq \frac{1}{\alpha} \nabla f^T \mathbf{x}.$$

This is interesting because the fact that $\mathbf{x} \in \Delta^\alpha$ implies that the reversed inequality also holds

$$\min_k \frac{\partial f}{\partial x_k} = \frac{1}{\alpha} \sum_i \left(x_i \min_k \frac{\partial f}{\partial x_k} \right) \leq \frac{1}{\alpha} \nabla f^T \mathbf{x},$$

meaning we have inequality in both directions, meaning x is a minimum if and only if

$$\min_k \frac{\partial f}{\partial x_k} = \frac{1}{\alpha} \nabla f^T \mathbf{x}.$$

The right-hand side of this equation is a weighted average of partial derivatives. Because the weights are non-negative and the smallest partial derivative is included when forming this average, all partial derivatives that correspond to non-zero coordinates of $\mathbf{x}$ must equal the smallest partial derivative at $\mathbf{x}$. Taken together, x is a minimum of a convex function $f : \Delta^\alpha \to \mathbb{R}$ if and only if

$$\frac{\partial f}{\partial x_i} = \min_k \frac{\partial f}{\partial x_k}, \quad \forall i \in \mathcal{P} = \{j \mid x_j > 0\} \tag{5}$$

where $\mathcal{P}$ denotes the indices of the non-zero coordinates in $\mathbf{x}$. This somewhat surprising result sets the stage for the development of an efficient way to minimize the columns of H.

We next connect Equations 2 and 5 using a rank-one residue approach. Rewriting WH^T, we have

$$WH^T = \sum_j \mathbf{w}_j \mathbf{h}_j^T \text{ and defining } R_i = A - \sum_{j \neq i} \mathbf{w}_j \mathbf{h}_j^T,$$

the subproblem of updating a column $\mathbf{h}_i$ becomes

$$\mathbf{h}_i \leftarrow \arg\min_{\mathbf{h} \in \Delta^\alpha} \|R_i - \mathbf{w}_i \mathbf{h}^T\|_F^2 + \delta_{\mathbf{h}_i} \|\mathbf{h} - \mathbf{h}_i\|^2,$$

which is the same as

$$\mathbf{h}_i \leftarrow \arg\min_{\mathbf{h} \in \Delta^\alpha} \frac{1}{2} \|\mathbf{h}\|_2^2 - \mathbf{h}^T \mathbf{v}, \tag{6}$$

where $\mathbf{v}$ denotes the constant vector $(R_i^T \mathbf{w}_i + \delta_{\mathbf{h}_i} \mathbf{h}_i)/(\|\mathbf{w}_i\|_2^2 + \delta_{\mathbf{h}_i})$. The key to solving this problem efficiently lies in the observation that $\mathbf{h}$ can be solved directly when the indices of the non-zero variables are known. To see this, assume for a while that $\mathcal{P}$ is given and let f be the above objective function of Problem 6. Because f is convex, Equation 5 implies that all its partial derivatives with respect to the non-zero variables share a common value, that is

$$h_j - v_j = C, \; j \in \mathcal{P}$$

for some C at the minimum. Summing over j and using the fact that $\mathbf{h} \in \Delta^\alpha$, we can solve for C

$$C = (\alpha - \sum_{j \in \mathcal{P}} v_j)/\#\mathcal{P}$$

meaning $h_j = C + v_j$, $j \in \mathcal{P}$, $h_j = 0$, $j \notin \mathcal{P}$. Thus, all that remains is a way to find $\mathcal{P}$. Although this may seem like a problem with a complexity of $O(2^n)$ at first sight, it turns out that $\mathcal{P}$ must correspond to the indices of the $\#\mathcal{P}$ largest coordinates of $\mathbf{v}$. To see this, assume that $\mathbf{h}$ is a minimum and that there exist indices $a \in \mathcal{P}$ and $b \notin \mathcal{P}$ such that $\mathbf{v}_a < \mathbf{v}_b$. Then, the entries h_a and h_b could be swapped to obtain another feasible vector that would yield a smaller objective function value in Equation 6, contradicting that $\mathbf{h}$ is a minimum. Hence, the only remaining question is how many coordinates are non-zero at the minimum. This question can be resolved by computing C and the partial derivatives for different values of $\#\mathcal{P}$ until Equation 5 is satisfied. This procedure can be implemented as a linear $O(n)$ search (Algorithm 1) and is amenable to speed-ups when used iteratively (Discussion).

Optimizing the Components

Unlike the optimization of H, which is independent of $\mathcal{R}$, the optimization of W depends on the choice of $\mathcal{R}$. We next give W optimization procedures for three common types of regularization:

Sparseness regularization. A common way to enforce sparsity is to penalize the L^1 norm, the closest convex L^p relaxation of the L^0 penalty (the number of non-zero elements). To optimize W with $\mathcal{R}(W) = \|W\|_1$, one possibility is to use the rank-one residue approach. Rewriting WH^T as a sum of rank-one matrices and considering the Karush-Kuhn-Tucker (KKT) conditions, it is easy to show that the BCD update for the column/block $\mathbf{w}_i$ is given by

$$\mathbf{w}_i \leftarrow \frac{[R\mathbf{h}_i - \frac{\lambda}{2} 1]_+}{\|\mathbf{h}_i\|_2^2},$$

where $[\cdot]_+$ denotes truncation of vector elements at zero. Another possibility is to view W as a single block, in which case the minimization can be rewritten as a non-negative least squares problem (this follows directly from the KKT conditions) that can be solved efficiently using for example the Fast Non-Negative Least Squares algorithm (FNNLS) [17].

Tikhonov regularization. We next consider L^2 regularization with $\mathcal{R} = \|\Gamma W\|_F^2$ where Γ is an $m \times m$ filter matrix. This type of regularization is used to impose various types of smoothing, for example by using $\Gamma = I$ or various difference operators, like $\Gamma_{(i,i)} = 1$, $\Gamma_{(i,i+1)} = -1$, and $\Gamma_{(i,j)} = 0$ elsewhere. Partitioning the coordinates per column and using a rank-one residue approach, the column-wise BCD updates become

$$\mathbf{w}_i \leftarrow \arg\min_{\mathbf{w} \geq 0} \|R - \mathbf{w}\mathbf{h}_i^T\|_F^2 + \lambda \|\Gamma \mathbf{w}\|^2.$$

Expanding the norm and removing constant terms, we get

$$\mathbf{w}_i \leftarrow \underset{\mathbf{w}\geq 0}{\arg\min}\ \|\mathbf{w}\|^2\|\mathbf{h}_i\|^2 - 2\langle \mathbf{w},\ \infty R\mathbf{h}_i\rangle + \lambda\|\Gamma\mathbf{w}\|^2, \qquad (7)$$

which is a non-negative least squares problem. To see this, let L^TL be the Cholesky decomposition of $\|\mathbf{h}_i\|^2 I + \lambda\Gamma^T\Gamma$ and consider

$$\mathbf{w}_i \leftarrow \underset{\mathbf{w}\geq 0}{\arg\min}\ \|L\mathbf{w} - L^{-T}R\mathbf{h}_i\|^2. \qquad (8)$$

Expanding Equation 8 and removing the constant term, we recover Equation 7. Hence, we can solve Equation 8 which can be done using non-negative least squares algorithms that start from the normal equations and do not require explicit Cholesky decomposition [17–19].

Related base vector regularization. In some applications, certain base vectors are known to be closer to each other. For example, this type of regularization may be motivated in the reconstruction of cell type-specific gene expression profiles from gene expression profiles of compound tissues, where the gene expression patterns of related cell types can be expected to be similar. One way to incorporate such information is to penalize the squared distance between base vectors that are known to be related. The objective function becomes

$$\|A - WH^T\|_F^2 + \lambda \sum_{(i,j)\in\mathcal{N}} \|\mathbf{w}_i - \mathbf{w}_j\|^2$$

where the set $\mathcal{N}$ defines pairs of adjacent vectors, encoded as a matrix N where each column defines a pair (i, j) by having elements that are λ at position i and j and 0 elsewhere. The objective function can then be written as

$$\|[A\ 0] - W[H^T\ N]\|_F^2,$$

the minimum of which with respect to W can again be found using FNNLS or other non-negative least squares algorithms.

Computational Efficiency

To illustrate its use, we implemented our method with L^1 norm-induced sparseness regularization (Algorithm 2; denoted rNMF), and applied it to sets of gene expression profiles of blood disorders (Table 1). For comparison, we considered two previously published methods [20,21]. These methods are relevant as control methods as they also seek to perform NMF with L^1 regularization and exact scale control. Other sparse NMF methods have been published (Discussion), but solve different formulations and, hence, are less relevant as controls in this context. Out of the two selected control methods, we found the method in [21] to be the most efficient, making it a representative control method. Each data set was analyzed with different numbers of components (k = 5,10, and 15) and regularization parameter values (λ selected to yield 25%, 50%, and 75% zeroes in W; the value needed to achieve a specific degree of sparsity varies between data sets).

Throughout, rNMF was 1.5 to 3.0 times faster per iteration and converged considerably faster (Table 1 and Figure 1a). The method also exhibited robust closing of the KKT conditions, illustrating that the theoretical prediction that solutions represent critical points holds numerically in practice (Figure 1b).

Analysis of Gene Expression Data

To illustrate the use of our approach in a practical situation, we applied rNMF to the Microarray Innovations in LEukemia (MILE) data set [22,23], containing 2096 gene expression profiles of bone marrow samples from patients with a range of blood disorders (Affymetrix Human U133 Plus 2.0 arrays; 54612 genes expression values/probes per sample). We applied rNMF to the MILE data with varying numbers of components ($k = 10$, 20 and 30) and varying degrees of sparsity (λ chosen to yield 50%, 75%, and 90% sparsity in W). To illustrate the effect of sparsity regularization, we also analyzed the data using conventional NMF (equivalent to setting $\lambda = 0$).

Now, it is well known that the bone marrow morphology varies considerably between disorders and between patients, especially in terms of the abundances of various classes of blood cells. It is also known that different classes of blood cells exhibit distinct gene expression patterns [24]. Much of the variation in the data will therefore be caused by fluctuations in cell type abundances and by differences in gene expression between cell types. Because rNMF and NMF are driven by variation, it is reasonable to assess the biological relevance of the results by testing whether the components contain gene expression features belonging to specific classes of blood cells. To this end, we used gene set enrichment testing, a statistical technique that is widely used in genomics to annotate high-dimensional patterns. In essence, statistically significant enrichment for a gene/probe set in a component means that the genes/probes comprising the set have higher coordinate values (at the set level) in a component than would be expected by chance (*c.f.*, ref. [25,26]). We defined sets of marker genes for all major classes of blood cells (Materials and Methods), and tested for enrichment of each of these sets in each component using the program RenderCat [25].

As illustrated in Figure 2a, enrichments of cell type markers were identified in all rNMF components except the weakest ones. Enrichments of markers for almost all major cell types in the bone marrow were detected in at least one component. In some components, enrichments of markers belonging to multiple cell types were detected. In these cases, the detected cell types belonged to the same developmental lineages (and hence have similar gene expression patterns). For example, this can be seen in Figure 2a where w_0, w_1, and w_2 are enriched for features from multiple myeloid cell types and w_{10} and w_{14} enriched for features from multiple lymphoid cell types. Together, the results support that the components are biologically relevant.

Conventional NMF also generated components with enrichments of cell type-specific markers. Interestingly, however, we observed differences as to which components did and did not capture cell type-specific features. As shown in Figure 2a, strong components generated by rNMF could usually be annotated and components that could not be annotated were usually the weakest ones. With conventional NMF, this pattern was generally not seen. Instead, as shown in Figure 2b, strong components could often not be annotated, suggesting that conventional NMF did not enrich for cell type-specific features. A likely explanation could be that there are relatively few cell type-specific markers compared to the number of genes in the genome, and that limiting the cardinality of components by including L^1 regularization promotes the identification of small sets instead of broader features that are less specific.

Discussion

Non-negative matrix factorization has been previously suggested as a valuable tool for analysis of various types of genomic data, particularly gene expression data [27–31]. The rationale is that gene expression is an inherently non-negative quantity. In this case, NMF allows the data to be expressed in their natural scale,

Table 1. Time (seconds) needed to complete one update of all coordinates and to reach convergence in sets of gene expression data from blood disorders.

		rNMF		control	
Data set, reference	**Data size**	**iteration**	**convergence**	**iteration**	**convergence**
Acute Myeloid Leukemia [36]	22283×293	0.75	21.7	2.2	219.4
Acute Myeloid Leukemia [37]	54613×461	3.95	128.8	10.2	>600
Acute Myeloid Leukemia [38]	44692×162	0.96	17.3	1.5	163.6
Acute Lymphoblastic Leukemia [39]	22215×288	0.94	17.8	2.3	245.7
Multiple Myeloma [40]	54613×320	3.04	29.1	6.4	>600

All methods were implemented in C++ and identically initialized. Timings obtained on a 2.30 GHz Intel Core i7 2820QM CPU with 16 GB RAM. For convergence, we required a relative decrease in the objective function less than 10^{-4} in successive iterations. Throughout, $\alpha=1$ and $\delta=10^{-5}$.
doi:10.1371/journal.pone.0046331.t001

thereby avoiding re-normalization by row-centering as is needed by dimension-reduction techniques based on correlation matrices (*e.g.*, principal component analysis).

We developed methods that enable robust and efficient solution of a range of regularized NMF problems and tested these methods in the context of gene expression data analysis. The key component of our approach is an efficient procedure to optimize the mixing coefficients H over the α-simplex, enabling the scale of the solution to be explicitly controlled. Further, our approach separates the task of optimizing H and optimizing W. This has three advantages. First, the optimization of H becomes independent of the regularization term, meaning the same algorithm (Algorithm 1) can always be used. Second, as exemplified by the L^1 regularization case, the optimization of W is simplified, at least with standard regularization terms. Third, a proximal point term can be included, guaranteeing convergence towards critical points, ensuring that the results will always have well-defined numerical meaning [16]. Experimentally, we have illustrated that our method is computationally efficient and capable of enhancing the identification of biologically relevant features from gene expression data by incorporating prior knowledge.

Previous work on regularized NMF is limited compared with previous work on conventional NMF. A straightforward formulation is

$$\begin{aligned} &\text{minimize} && \|A-WH^T\|_F^2+\lambda_1\mathcal{R}_1(W)+\lambda_2\mathcal{R}_2(H) \\ &\text{subject to} && W,\,H\geq 0, \end{aligned} \tag{9}$$

where are the functions $\mathcal{R}_1$ and $\mathcal{R}_2$ enforce the regularization constraints, and the parameters $\lambda_1,\lambda_2>0$ control the impact of the regularization terms [13]. This formulation allows regularization of both factors and basic computation methods can be derived for some choices of $\mathcal{R}_1$ and $\mathcal{R}_2$ by extending conventional NMF methods [32,33]. However, balancing λ_1 and λ_2 against each other is often difficult and simultaneous regularization of both factors is rarely wanted. More commonly, the goal is to regularize one of the factors. For example, to get sparse component vectors, an L^1 penalty can be imposed on W whereas H does not have to be regularized. In Equation 9, single-factor regularization would correspond to setting λ_1 or λ_2 to zero. Again, with standard scale-dependent regularization terms, this will pull the regularized factor towards zero and inflate the unregularized factor unboundedly. Scale-independent penalty terms have been proposed [34], but these are non-convex and therefore complicate optimization with respect to the regularized factor. One could also attempt to control

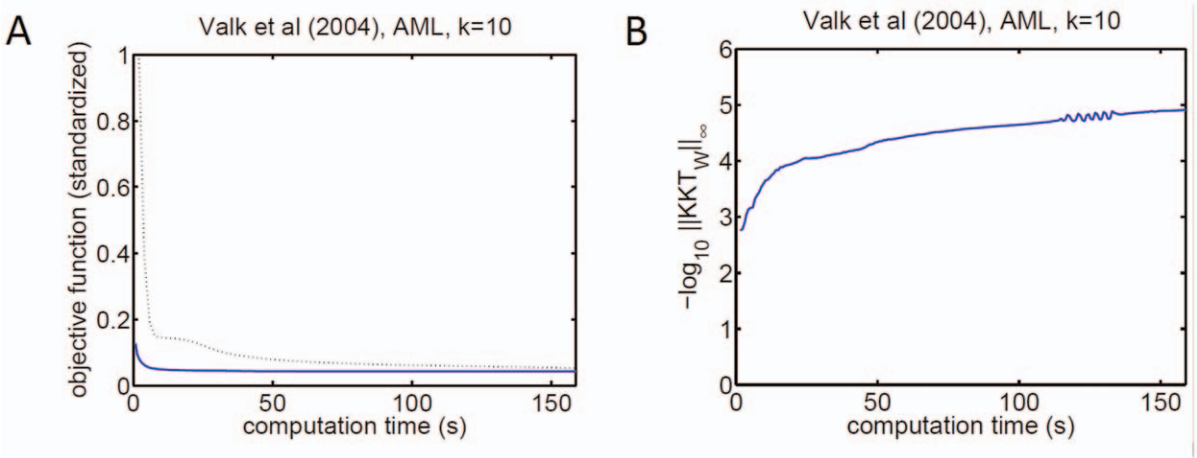

Figure 1. Convergence of rNMF on real data. Left: The objective function decreases faster with rNMF (blue) than the control method (dashed). We standardized the objective function by dividing it by the squared Frobenius norm of A. Right: As predicted theoretically, rNMF closes the KKT conditions (y axis indicates the negative logarithm of the max-norm of the KKT condition matrix for W, that is $(\mathrm{K}KT_W)_{ij}=\min(\delta f/\delta w_{ij},w_{ij})$ which should approach the zero matrix). The results in this figure were obtained for gene expression profiles of Acute Myeloid Leukemia [36], $k=10$, and λ set to yield about 50% sparsity. This example is representative as similar results were obtained for other data sets and parameter choices.
doi:10.1371/journal.pone.0046331.g001

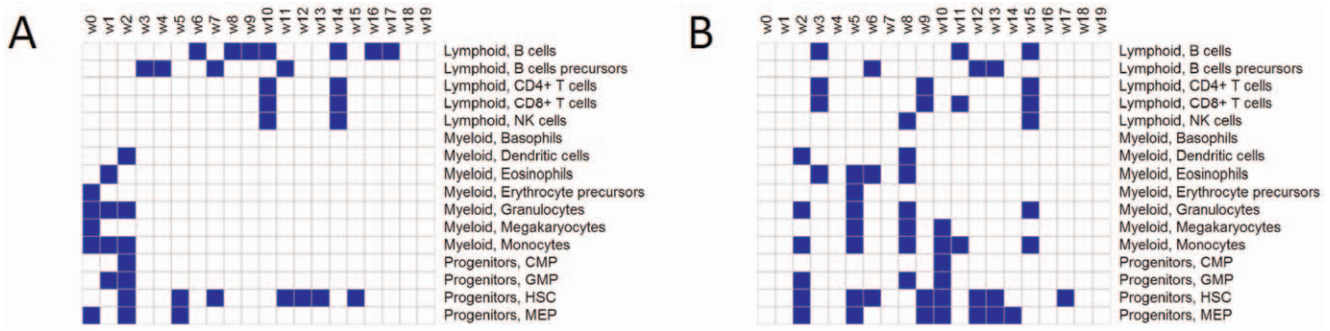

Figure 2. Application to gene expression microarray data from blood disorders. Columns indicate components, rows classes of blood cells. Blue cells indicate significant enrichment of cell type-specific markers (as detected by gene set enrichment testing; $p<10^{-5}$) in the component generated by rNMF with 90% sparsity (**a**) and conventional NMF (**b**). The components have been ordered by strength (defined as L^2 norm of $w_k h_k^T$) with w_0 denoting the strongest component. As discussed in detail in Results, strong components generated by rNMF capture cell type-related gene expression features more clearly than conventional NMF.
doi:10.1371/journal.pone.0046331.g002

the scale of the unregularized factor within the framework of Problem 9 by choosing $\mathcal{R}_i(\cdot)=\|\cdot 1-1\|_F^2$ or $\mathcal{R}_i(\cdot)=\|\cdot^T 1-1\|_F^2$ [13]. However, this again requires balancing of λ_1 against λ_2 which is difficult, and, moreover, the scale can only be controlled approximately. Another ad hoc approach could be to compensate for the pull of the regularization term by standardizing the column norms of W or H between iterations. This is equivalent to inserting a diagonal matrix D and its inverse between the factors. This operation is safe in conventional NMF because the value of the objective function will not change. With a regularization term, however, column standardization is unsafe: although the value of the fitting term $\|A-WH^T\|_F^2$ will not change, the value of the regularization term may, meaning the objective function may increase between iterations. To control the scale exactly, [20] proposed a truncated gradient descent method and [21] a multiplicative update method, and studied regularization with respect to sparsity. These methods represent the closest predecessors of our approach and were therefore used as control methods.

When it comes to the convergence, the strongest proved result for conventional NMF is guaranteed convergence to critical points. Some conventional NMF methods always find critical points, for example alternating non-negative least squares. By contrast, regularized NMF methods are less well characterized. To our knowledge, the only regularized NMF method that is known to guarantee critical point solutions is an alternating non-negative least squares method that solves Problem 9 when $\mathcal{R}_1$ is the squared L^1 norm and $\mathcal{R}_2$ is the L^2-norm [32]. Methods based on Lee-Seung's multiplicative descent method do not guarantee critical points [13], nor do current exact-scale methods [20,21].

In conclusion, we have presented a new framework for regularized NMF. Our approach has advantages in that it accommodates for a wide range of regularization terms, guarantees solutions that satisfy necessary conditions for optimality, allows the scale of the solution to be controlled exactly, is computationally efficient, and enables decomposition of gene expression data subject to knowledge priors. Hopefully, this study, along with other efforts, will further the development of methods to analyze complex high-dimensional data.

Materials and Methods

Microarray data sets generated on Affymetrix microarrays were retrieved from NCBI Gene Expression Omnibus (http://www.ncbi.nlm.nih.gov/gds; accession numbers GSE1159, GSE6891, GSE12417, GSE13159, GSE19784, and GSE28497). Because NMF assumes an additive model, non-log transformed gene expression values were used throughout the experiments. Sets of cell type-specific markers were inferred by use of the d-map compendium containing gene expression profiles of all major classes of blood cells sorted by flow cytometry (Affymetrix U133A arrays) [24]. One set per cell type was inferred by comparing d-map profiles belonging to this cell type to all others using Smyth's moderated t-test [35], selecting the top 100 probes as markers (results agreeing with those shown were obtained using the top 50, 150, 200 and 250 probes). Gene set enrichment testing was performed using the program RenderCat [25].

URLs

A C++ implementation is available at http://www.broadinstitute.org/~bnilsson/rNMF.rar.

Algorithm 1

Pseudocode to optimize a column $\mathbf{h}_i$ in H, given R_i, $\mathbf{w}_i$, the current $\mathbf{h}_i$, and the proximal point parameter δ. Note that in the if clause, the first condition $j==n$ asserts that the program never tries to reach v_{n+1} whereas the second asserts that C is the minimal value of the partial derivatives. Because $\mathbf{v}$ is sorted in descending order, and $\partial f/\partial h_j$ equals C if $j\in\mathcal{P}$ and $-v_j$ otherwise, it is sufficent to compare C with $-v_{j+1}$.

$\mathbf{v}=(R_i^T\mathbf{w}_i+\delta\mathbf{h}_i)\ /\ (\|\mathbf{w}_i\|_2^2+\delta)$
$[\mathbf{v},J]=sort(\mathbf{v},'descend')$
$\alpha=1.0$
for $j=1$ **to** n **do**
  $\alpha=\alpha-v_i$
  $C=\alpha/j$
  if $j==n$ **or** $C\leq-\mathbf{v}(j+1)$ **then**
    break
  end if
end for
$\mathbf{h}(j+1:n)=0$
$\mathbf{h}(J)=\mathbf{h}$
return h

Algorithm 2

Pseudocode for the complete rNMF procedure with $\mathcal{R}(W)=||W||_1$. To change the type of regularization, change the $\mathbf{w}_i$ update. Note that the rank-one residual R is updated cumulatively to save computations.

$\delta = 10^{-4}$
$R = A - WH^T$
Repeat
  for $i = 1$ **to** k **do**
    $R = R + \mathbf{w}_i \mathbf{h}_i^T$
    $\mathbf{h}_i = \text{Algorithm1}(R, \mathbf{w}_i, \mathbf{h}_i, \delta)$
    $\mathbf{w}_i = [R\mathbf{h}_i - \frac{\lambda}{2}\mathbf{1}]_+ / ||\mathbf{h}_i||_2^2$
    $R = R - \mathbf{w}_i \mathbf{h}_i^T$
  end for
until stopping criterion is reached
return (W, H)

Author Contributions

Conceived and designed the experiments: BN. Performed the experiments: BN LT. Analyzed the data: BN LT. Wrote the paper: BN LT. Supervised the project and designed the software used in the analysis: BN.

References

1. Lee DD, Seung HS (2000) Algorithms for non-negative matrix factorization. In: Advances in Neural Information Processing Systems 13: Proceedings of the 2000 Conference. MIT Press, pp. 556–562.
2. Battenberg E, Wessel D (2009) Accelerating non-negative matrix factorization for audio source separation on multi-core and many-core architectures. In: 10th International Society for Music Information Retrieval Conference (ISMIR 2009).
3. Dhillon I, Sra S (2005) Generalized non-negative matrix approximations with Bregman divergences. In: Proceedings of the Neural Information Processing Systems (NIPS) Conference, Vancouver.
4. Cichocki A, Zdunek R, Amari S (2006) Csiszar's divergences for non-negative matrix factorization: family of new algorithms. In: Proceedings of the 6th International Conference on Independent Components Analysis and Blind Signal Separation.
5. Paatero P, Tapper U (1994) Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. Environmetrics 5: 111–126.
6. Paatero P (1997) Least squares formulation of robust non-negative factor analysis. Chemometrics and Intelligent Laboratory Systems 37: 23–35.
7. Lee DD, Seung HS (1999) Learning the parts of objects by non-negative matrix factorization. Nature 401: 788–791.
8. Gonzalez E, Zhang Y (2005) Accelerating the Lee-Seung algorithm for non-negative matrix factorization. Technical report, Rice University.
9. Chu M, Diele F, Plemmons R, Ragni S (2005) Optimality, computation, and interpretation of nonnegative matrix factorizations. SIAM Journal on Matrix Analysis.
10. Lin C (2007) Projected gradient methods for non-negative matrix factorization. Neural Computation 19: 2756–2779.
11. Zdunek R, Cichocki A (2008) Fast nonnegativematrix factorization algorithms using projected gradient approaches for large-scale problems. Computational Intelligence and Neuroscienc.
12. Ho ND (2008) Nonnegative matrix factorization algorithms and applications. Ph.D. thesis, Université Catholique De Louvain.
13. Berry M, Browne M, Langville A, Pauca V, Plemmons R (2006) Algorithms and applications for approximate non-negative matrix factorization. Preprint.
14. Donoho D, Stodden V (2004) When does non-negative matrix factorization give a correct decomposition into parts. Cambridge, MA: MIT Press.
15. Bertsekas DP (1999) Nonlinear programming. Athena Scientific, 2 edition.
16. Grippo L, Sciandrone M (2000) On the convergence of the block nonlinear gauss–seidel method under convex constraints. Operations Research Letters 26: 127–136.
17. Benthem MHV, Keenan MR (2004) Fast algorithm for the solution of large-scale non-negativityconstrained least squares problems. Journal of chemometrics 18: 441–450.
18. Bro R, Jong SD (1998) A fast non-negativity-constrained least squares algorithm. Journal of chemometrics 11: 393–401.
19. Lawson CL, Hanson RJ (1974) Solving Least Squares Problems. Series in automatic computing. Prentice-Hall, 158–164 pp.
20. Hoyer PO (2002) Non-negative sparse coding. In: Neural Networks for Signal Processing XII (Proc. IEEE Workshop on Neural Networks for Signal Processing). pp. 557–565.
21. Eggert J, Körner E (2004) Sparse coding and NMF. In: Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on. volume 4, 2529–2533.
22. Kohlmann A, Kipps TJ, Rassenti LZ, Downing JR, Shurtleff SA, et al. (2008) An international standardization programme towards the application of gene expression profiling in routine leukaemia diagnostics: the microarray innovations in leukemia study prephase. Br J Haematol 142: 802–807.
23. Haferlach T, Kohlmann A, Wieczorek L, Basso G, Kronnie GT, et al. (2010) Clinical utility of microarray-based gene expression profiling in the diagnosis and subclassification of leukemia: report from the international microarray innovations in leukemia study group. J Clin Oncol 28: 2529–2537.
24. Novershtern N, Subramanian A, Lawton LN, Mak RH, Haining WN, et al. (2011) Densely interconnected transcriptional circuits control cell states in human hematopoiesis. Cell 144: 296–309.
25. Nilsson B, Håkansson P, Johansson M, Nelander S, Fioretos T (2007) Threshold-free high-power methods for the ontological analysis of genomewide gene-expression studies. Genome Biology 8.
26. Subramanian A (2005) Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. Proc Natl Acad Sci U S A 102: 15545–50.
27. Brunet JP, Tamayo P, Golub TR, Mesirov JP (2004) Metagenes and molecular pattern discovery using matrix factorization. Proc Natl Acad Sci U S A 101: 4164–4169.
28. Carrasco DR, Tonon G, Huang Y, Zhang Y, Sinha R, et al. (2006) High-resolution genomic profiles define distinct clinico-pathogenetic subgroups of multiple myeloma patients. Cancer Cell 9: 313–325.
29. Frigyesi A, Höglund M (2008) Non-negative matrix factorization for the analysis of complex gene expression data: identification of clinically relevant tumor subtypes. Cancer Inform 6: 275–292.
30. Devarajan K (2008) Nonnegative matrix factorization: an analytical and interpretive tool in computational biology. PLoS Comput Biol 4: e1000029.
31. Tamayo P, Scanfeld D, Ebert BL, Gillette MA, Roberts CWM, et al. (2007) Metagene projection for cross-platform, cross-species characterization of global transcriptional states. Proc Natl Acad Sci U S A 104: 5959–64.
32. Kim H, Park H (2007) Sparse non-negative matrix factorization via alternating non-negativityconstrained least squares for microarray data analysis. Bioinformatics 23: 1495–1502.
33. Pauca V, Piper J, Plemmons R (2005) Non-negative matrix factorization for spectral data analysis. Linear algebra and its applications.
34. Hoyer PO (2004) Non-negative matrix factorization with sparseness constraints. Journal of Machine Learning Research 5: 1457–1469.
35. Smyth G (2004) Linear models and empirical bayes methods for assessing differential expression in microarray experiments. Statistical Applications in Genetics and Molecular Biology 3.
36. Valk PJM, Verhaak RGW, Beijen MA, Erpelinck CAJ, van Waalwijk van Doorn-Khosrovani SB, et al. (2004) Prognostically useful gene-expression profiles in acute myeloid leukemia. N Engl J Med 350: 1617–1628.
37. Verhaak RGW, Wouters BJ, Erpelinck CAJ, Abbas S, Beverloo HB, et al. (2009) Prediction of molecular subtypes in acute myeloid leukemia based on gene expression profiling. Haematologica 94: 131–134.
38. Metzeler KH, Hummel M, Bloomfield CD, Spiekermann K, Braess J, et al. (2008) An 86-probeset gene-expression signature predicts survival in cytogenetically normal acute myeloid leukemia. Blood 112: 4193–4201.
39. Coustan-Smith E, Song G, Clark C, Key L, Liu P, et al. (2011) New markers for minimal residual disease detection in acute lymphoblastic leukemia. Blood 117: 6267–6276.
40. Broyl A, Hose D, Lokhorst H, de Knegt Y, Peeters J, et al. (2010) Gene expression profiling for molecular classification of multiple myeloma in newly diagnosed patients. Blood 116: 2543–2553.