

TOWARDS A MOLECULAR-LEVEL UNDERSTANDING OF BIOLOGICAL PROCESSES BY UNSUPERVISED ANALYSIS OF GENE EXPRESSION DATA

Liviu BADEA

National Institute for Research and Development in Informatics (ICI)
E-mail: badea@ici.ro

Biological systems are exceptionally complex processors of information, in which digital and analog processes are inextricably intertwined. While the analog processes are hard to measure on a genomic scale, we observe that the digital nature of genomic information has allowed the recent explosion of high throughput microarray data, which has enabled new breakthroughs in our understanding of the control of gene expression. However, despite its enormous potential, microarray data has proved difficult to analyze, due to a number of domain-specific problems. In the context of an unsupervised analysis (clustering) of microarray data, we show that these problems can be elegantly dealt with by using nonnegative matrix factorizations (NMF) to cluster genes and samples simultaneously while allowing for bicluster overlaps. Moreover, we address the instability of NMF by employing Positive Tensor Factorization to perform a two-way meta-clustering of the biclusters produced in several different clustering runs. The application of our approach to a large lung cancer dataset proved computationally tractable and was able to recover the histological classification of the various cancer subtypes represented in the dataset.

Key words: bioinformatics, microarrays, gene expression data analysis, clustering, nonnegative matrix factorization, meta-clustering, positive tensor factorization.

1. THE RELATIONSHIP BETWEEN BIOLOGY AND INFORMATICS

We are currently entering a new era of “rational” medicine and drug design, based on fundamental molecular-level knowledge about the biological processes involved in the normal functioning of organisms, as well as in various diseases. In the pharmaceutical industry, time, cost and throughput constraints have begun to significantly limit the much needed development of new drugs for many frequent diseases. The genomic revolution in biology and other high-throughput technologies have recently enabled a more rational drug design, leading to new compounds that can successfully combat several previously intractable diseases (such as Gleevec, which is effective against chronic myeloid leukemia).

In spite of such notable successes, the tasks faced by this domain are huge, especially due to the daunting complexity of biological systems and processes. The sheer size of the relevant data and knowledge makes their processing by human subjects impossible and thus requires the use of computers. (The NCBI databases [1] store sequence information for more than 56 billion base-pairs, while the number of relevant biomolecular databases and resources on the Web exceeds 500.)

However, the use of computers in this domain is not limited to the storage and management of huge collections of data. The integrated in-depth analysis of this data is far more complex and important for extracting biological knowledge as well as for producing experimentally testable hypotheses. In fact, it has become apparent that computer science is “to biology what mathematics is to physics” (Harold Morowitz). In the following, we intend to elaborate on this idea. Figure 1 below schematically suggests that in the same way in which physics uses mathematics to represent models of the physical world, biology uses informatics for representing the knowledge about biological entities and processes. In fact, the most important breakthroughs in physics are related to the development of quantitative mathematical models for physical phenomena. Therefore, since biology is *in principle* “included” in physics (via chemistry perhaps), we may *naively* expect the mathematical models of physical processes to be directly usable for modelling biological

phenomena. Unfortunately, from a *practical* point of view, nothing can be more remote from the truth. While the laws of physics are simple, very general and relatively few, the “laws” of biology are complex, quite specific and very numerous. (Although the laws of biology are based on the laws of physics, they include a large number of “frozen accidents”¹ or spontaneously broken symmetries, which explain their complexity and specificity.) Therefore, in practice we can hardly use the inclusion 1 and the detailed physical models from Figure 1 to model biological processes.²

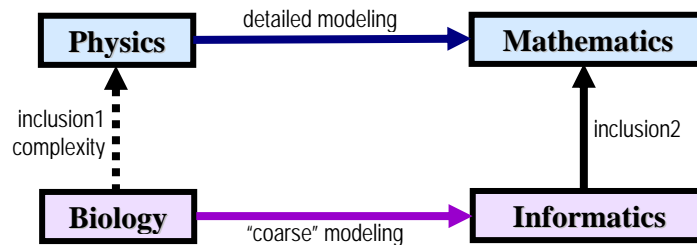


Fig. 1. The relationship between biology, informatics, physics and mathematics

Modelling biological processes therefore involves developing *incomplete models* based on *experimental data* (rather than first principles). Not surprisingly, the most significant breakthroughs in biology are linked to the breakthroughs in experimental techniques, especially the high-throughput technologies that have witnessed an explosive growth in the past decade.

2. DIGITAL CODES FOR ANALOG BIOLOGICAL PROCESSORS

Biological systems are exceptionally complex *processors of information*, capable of adaptation to their environment as well as of replication and evolution. However, they are quite different from man-made information processors, which are entirely “digital” processors. In the following, we argue that biological systems employ digital codes that encode and control analog processors, the latter being probably more adaptable to a diverse environment. This distinction between digital and analog processors in biological systems is essential for distinguishing the various aspects of biological systems that can be measured using high-throughput technology, and which will be discussed in the next sections.

Indeed, most human information processing artefacts (ranging from the simplest digital computer applications to the most sophisticated artificial intelligence programs) face significant problems related to the *interface with their environment*. This issue, which has only recently begun to be perceived as one of the main bottlenecks towards the development of more adaptable intelligent systems, is due to the simple fact that any digital mapping of the real world will either be too simplistic (and thus incapable of adaptation and evolution), or too detailed to be processable by human-developed software (e.g. robotic vision is way behind the capabilities of the human visual system in terms of recognizing objects and their movements in complex scenes).

On the other hand, biological systems use a digital encoding of their own *structure* (the DNA), which enables a high replicative fidelity and a correspondingly tight control of their evolution, but they are operating in an exceedingly complex “analog” world and thus make use of exceptionally well adapted analog “processors”. Digital and analog processors are intertwined inextricably, to the point that one would not work without the other. More precisely, the exceptional adaptation of the analog processors is due to the evolutionary mechanisms that are only possible thanks to the digital encoding in DNA of the structure of the biological systems, while the functioning and maintenance of the digital code itself is realised with the help of tremendously complex analog processes.

¹ Such as the genetic code.

² A systems biology model of an isolated eukaryotic cell would involve huge numbers of parameters of chemical reactions (e.g. kinetical constants), an enormously complicated molecular description of the subcellular structures, etc. All of this is well beyond the capabilities of human knowledge engineers and biologists, as well as the processing capabilities of the fastest supercomputers.

The “digital code” of biological systems is based on the so-called genetic code, i.e. the correspondence between DNA codons³ and aminoacid components of proteins, which is conserved for all biological systems on Earth. The differences between the various living beings, their individual “digital codes” are given by the precise sequences of bases that make up their genomes.

The last decade has witnessed an explosive growth of organisms whose genomes have been completely sequenced, ranging from viruses and bacteria to complex multi-cellular eukaryotes such as Homo Sapiens. This explosive growth was facilitated by the digital nature of these codes, as opposed to the much more complex molecular biology of the “analog” processes,⁴ which are currently only incompletely known, despite the much more extensive research efforts spent in the domain of molecular cell biology.

As previously mentioned, constructing *complete* mathematical models for multi-cellular eukaryotes usable for simulation and prediction is a daunting task, as they would have to cover not just the metabolic, signalling and gene expression control networks, but also their enormously complex interactions. Although such complete models are way beyond present technology and knowledge, there are aspects and subprocesses that are within the reach of current high-throughput experimental techniques.

3. HIGH-THROUPT MEASUREMENT OF GENE EXPRESSION AND MICROARRAY DATA ANALYSIS

An essential aspect of cellular function is represented by the varying amounts of protein produced by the various genes. Although a high-throughput⁵ direct measurement of protein levels is complicated and very expensive, genome-wide high-throughput measurements of mRNA levels are comparatively easy due to the *digital* nature of base-pair complementarity between mRNA and genomic DNA. *Microarrays* (or *gene chips*) [2] allow the simultaneous measurement of mRNA levels for virtually all genes of an organism, exploiting the gene sequences known from previous whole genome sequencing projects. They are therefore ideal for studying the *control of gene expression* in various experimental conditions and states (for instance between normal and diseased tissues).

The advent of microarray technology has allowed a revolutionary transition from the exploration of the expression of a handful of genes to that of entire genomes. However, despite its enormous potential, *microarray data has proved difficult to analyze*, mainly due to the following domain-specific problems:

- (P1) the huge numbers of genes involved (up to a few tens of thousands) compared to the small number of samples (tens to a few hundreds),
- (P2) the high experimental noise levels specific to this technology,
- (P3) the large number of factors that influence gene expression (many of which are *not* at the mRNA/transcriptone level) as well as the complexity of their interactions,
- (P4) the inability of most currently used clustering algorithms to naturally deal with overlapping clusters (most produce non-overlapping clusters – a serious limitation in this domain, since a gene is typically involved in several biological processes),
- (P5) the difficulty in clustering genes and samples simultaneously, as well as
- (P6) the instability of the resulting clusters w.r.t. the initialization of the algorithm.

In the following, we make the biologically plausible simplifying assumption that the overlap of biological processes is *additive*

$$X_{sg} = \sum_c X(s, g | c) \quad (1)$$

where X_{sg} is the expression level of gene g in data sample s , while $X(s, g | c)$ is the expression level of g in s due to biological process c . We also assume that $X(s, g | c)$ is multiplicatively decomposable into the

³ i.e. triplets formed out of the four bases A, T, C, G.

⁴ These involve complex protein interactions, cellular compartments and the associated molecule trafficking, cell signaling and transport, synthesis and degradation of various metabolites and proteins, basic cellular processes such as metabolism, transcription, translation, DNA repair, etc.

⁵ By “high-throughput” measurements we mean measurements for large numbers of genes or proteins. Genome-wide measurements cover virtually all genes of an organism.

expression level A_{sc} of the biological process (cluster) c in sample s and the membership degree S_{cg} of gene g in c :

$$X(s, g | c) = A_{sc} \cdot S_{cg} \quad (2)$$

Of course, biological processes are frequently *non-linear*, but the large numbers of associated parameters and the lack of knowledge regarding their precise values lead to potential *overfitting* problems, which are alleviated by the linearity assumption above.

Also note that the semantics of *membership degrees* employed in this paper differs from the related notions from both probabilistic and fuzzy logics, in that overlapping biological processes are additively superposable (for a more concrete example, see also the discussion regarding the adeno-squamous cases from Section 6).

4. CLUSTERING VIA NONNEGATIVE MATRIX FACTORIZATIONS

Combining (1) and (2) leads to a reformulation of our clustering problem as a *nonnegative factorization* of the $n_s \times n_g$ (samples \times genes) gene expression matrix X as a product of an $n_s \times n_c$ (samples \times clusters) matrix A and an $n_c \times n_g$ (clusters \times genes) matrix S :

$$X_{sg} \approx \sum_c A_{sc} \cdot S_{cg} \quad (3)$$

with the additional nonnegativity constraints:

$$A_{sc} \geq 0, S_{cg} \geq 0. \quad (4)$$

(Expression levels and membership degrees cannot be negative.)

More formally, this can be cast as a constrained optimization problem:

$$\min C(A, S) = \frac{1}{2} \|X - A \cdot S\|_F^2 = \frac{1}{2} \sum_{s,g} (X - A \cdot S)_{sg}^2 \quad (5)$$

subject to the nonnegativity constraints (4), and could be solved using Lee and Seung's seminal *Nonnegative Matrix Factorization (NMF)* algorithm [3,4] (ε is a small regularization parameter):

NMF(X, A_0, S_0) \rightarrow (A, S)

$A \leftarrow A_0, S \leftarrow S_0$ (typically A_0, S_0 are initialized randomly)

loop until convergence

$$S_{cg} \leftarrow S_{cg} \frac{(A^T \cdot X)_{cg}}{(A^T \cdot A \cdot S)_{cg} + \varepsilon} \quad ; \quad A_{sc} \leftarrow A_{sc} \frac{(X \cdot S^T)_{sc}}{(A \cdot S \cdot S^T)_{sc} + \varepsilon}$$

As explained above, such a factorization can be viewed as a “soft” clustering algorithm allowing for *overlapping clusters*, since we may have several significant S_{cg} entries on a given column g of S (so a gene g may “belong” to several clusters c).

Many other clustering and dimensionality reduction algorithms (such as k-means / Vector Quantization, Principal Components Analysis, etc.) can be viewed as matrix factorizations (3), the differences arising from the different constraints imposed on the matrix factors [4]. For example, in the case of k-means, clusters are non-overlapping, so each column of S is required to have a single nonzero element, while PCA constrains the columns of A and the rows of S to be orthogonal (without the nonnegativity constraints), thus leading to distributed “representations”. On the other hand, the nonnegativity constraints (4) imposed by NMF produce more sparse factorizations (i.e. with fewer significant elements of A and S). Note that the nonnegativity and the orthogonality constraints are mutually incompatible.

Also note that the NMF factorization (3) is non-unique, since it is invariant under the following scalings of the rows of A and columns of S :

$$A \leftarrow A \cdot D, S \leftarrow D^{-1} \cdot S,$$

where D is a positive diagonal matrix⁶ with elements $d_c = \sqrt{\sum_g S_{cg}^2}$.

Theoretical and experimental evidence suggests that NMF is preferable to other factorization methods for clustering gene expression data, as it deals successfully with problems (P1), (P2), (P4), and (P5) mentioned in Section 3. Problem (P3) is due to an intrinsic limitation of transcriptomic data – solving it would require additional types of high-throughput data⁷ that are generally unavailable at this time since they involve the “analog” processes mentioned in Section 2. We next address problem (P6), the instability of the clusters w.r.t. the initialization of the algorithm.

5. STABLE FACTORIZATIONS USING META-CLUSTERING WITH POSITIVE TENSOR FACTORIZATIONS

As previously mentioned, virtually all clustering or factorization algorithms are affected by the *instability* of the resulting clusters w.r.t. the initialization of the algorithm (as in the case of *NMF*, *k-means*, *fuzzy k-means* [5]), or w.r.t. slight differences in the input dataset as a result of resampling the initial data (e.g. for hierarchical clustering). This is not surprising if we adopt a unifying view of clustering as a constrained optimization problem, since the fitness landscape of such a complex problem may involve many different local minima into which the algorithm may get caught when started off from different initial states.

Although such an *instability* seems hard to avoid, we may be interested in the clusters that keep reappearing in the majority of the runs of the algorithm. This is related to the problem of *combining multiple clustering systems*, which is the unsupervised analog of the classifier combination problem but involves solving an additional so-called *cluster correspondence* problem, which amounts to finding the best matches between clusters generated in different runs.

The cluster correspondence problem can also be cast as an unsupervised optimization problem, which can be solved by a *meta-clustering algorithm*. Choosing an appropriate meta-clustering algorithm for dealing with this problem crucially depends on the precise notion of cluster correspondence.

Since a very strict notion of *perfect one-to-one correspondence* between the clusters of each pair of clustering runs may be too tough to be realized in most practical cases, we could look for clusters that are most *similar* (although not necessarily identical) across all runs. This is closest to performing something similar to single-linkage hierarchical clustering on the sets of clusters produced in the various clustering runs, with the additional constraint of allowing in each meta-cluster no more than a single cluster from each individual run. Unfortunately, this constraint will render the meta-clustering algorithm highly unstable. Thus, while trying to address the instability of (object-level) clustering using meta-level clustering, we end up with instability in the meta-clustering algorithm itself. Therefore, a “softer” notion of cluster correspondence is needed.

Note that allowing for cluster overlap in the NMF algorithm alleviates but does not completely eliminate the instability of clustering, since the optimization problem (5), (4) is non-convex. In particular, the NMF algorithm produces different factorizations (biclusters) $(A^{(i)}, S^{(i)})$ for different initializations, so meta-clustering the resulting “soft” clusters might be needed to obtain a more stable set of clusters. However, using a “hard” meta-clustering algorithm would once again entail an unwanted instability.

In a previous paper [6], we have shown that a generalization of NMF called *Positive Tensor Factorization (PTF)* [7] is precisely the tool needed for meta-clustering “soft”, potentially overlapping *biclusters* produced in different clustering runs by fuzzy k-means or NMF. This not only alleviates the instability of a “hard” meta-clustering algorithm, but also produces a “base” set of “*bicluster prototypes*”, out of which all biclusters of all individual runs can be recomposed, despite the fact that they may not correspond to identically reoccurring clusters in all individual runs. In the next Section, we demonstrate that this approach can be successfully used for biclustering a large lung cancer gene expression dataset.

⁶ Note that such positive diagonal matrices are the most general positive matrices whose inverses are also positive (thereby preserving the nonnegativity of A and S under the above transformation).

⁷ Covering for example genome-wide promoter activation data for all known transcription factors and genes, high-throughput measurements of phosphorylation states of signalling molecules, large-scale metabolomic data, etc.

In the following, we use NMF for object-level clustering and PTF for meta-clustering. This unified approach solves in an elegant manner both the clustering and the cluster correspondence problem. More precisely, we first run NMF as object-level clustering r times:

$$X \approx A^{(i)} \cdot S^{(i)} \quad i = 1, \dots, r \quad (6)$$

where X is the gene expression matrix to be factorized (samples \times genes), $A^{(i)}$ (samples \times clusters) and $S^{(i)}$ (clusters \times genes).

To allow the comparison of membership degrees S_{cg} for different clusters c , we scale the rows of $S^{(i)}$ to unit norm by taking advantage of the above-mentioned scaling invariance of the factorization (6): $A(i) \leftarrow A(i) \cdot D$, $S(i) \leftarrow D^{-1} \cdot S(i)$, where D is a positive diagonal matrix with elements $d_c = \sqrt{\sum_g S_{cg}^{(i)2}}$.

Next, we perform *meta-clustering* of the resulting *biclusters* ($A^{(i)}, S^{(i)}$). This is in contrast with as far as we know all existing meta-clustering approaches, which take only one dimension into account (either the object- or the sample dimension). Although such *one-way* approaches work well in many cases, they will fail whenever two clusters correspond to very similar sets of genes, while differing along the sample dimension.

In the following, we show that a slight generalization of NMF, namely *Positive Tensor Factorization* (PTF) [7] can be successfully used to perform *two-way* meta-clustering, taking both the gene and the sample dimensions into account.

Naively, one would be tempted to try clustering the biclusters⁸ $A_c^{(i)} \cdot S_c^{(i)}$ instead of the gene clusters $S_c^{(i)}$, but this is practically infeasible in most real-life datasets because it involves factorizing a matrix of size $r \cdot n_c \times n_s \cdot n_g$. On closer inspection, however, it turns out that it is not necessary to construct this full-blown matrix – actually we are searching for a *Positive Tensor Factorization* of this matrix⁹

$$A_{sc}^{(i)} \cdot S_{cg}^{(i)} \approx \sum_{k=1}^{n_c} \alpha_{ck}^{(i)} \cdot \beta_{sk} \cdot \gamma_{kg} \quad (7)$$

The indices in (7) have the following domains: s – samples, g – genes, c – clusters, k – metaclusters. To simplify the notation, we merge the indices i and c into a single index (ic):

$$A_{s(ic)} \cdot S_{(ic)g} \approx \sum_{k=1}^{n_c} \alpha_{(ic)k} \cdot \beta_{sk} \cdot \gamma_{kg} \quad (7')$$

Note that β and γ are the “unified” versions of $A^{(i)}$ and $S^{(i)}$ respectively. More precisely, the columns β_k of β and the corresponding rows γ_k of γ make up a *base set of bicluster prototypes* $\beta_k \cdot \gamma_k$ out of which all biclusters of all individual runs can be recomposed, while α encodes the (*bi*)cluster-*metacluster* correspondence.

Ideally (in case of a perfect one-to-one correspondence of biclusters across runs), we would expect the rows of α to contain a single significant entry $\alpha_{(ic),m(i,c)}$, so that each bicluster $A_c^{(i)} \cdot S_c^{(i)}$ corresponds to a single bicluster prototype $\beta_{m(i,c)} \cdot \gamma_{m(i,c)}$ (where $m(i,c)$ is a function of i and c):

$$A_c^{(i)} \cdot S_c^{(i)} = \alpha_{(ic),m(i,c)} \cdot \beta_{m(i,c)} \cdot \gamma_{m(i,c)}. \quad (8)$$

Additionally, each metacluster m should contain no more than a single bicluster from each run, i.e. there should be no significant entries $\alpha_{(ic''),m}$ and $\alpha_{(ic''),m}$ with $c' \neq c''$.

Although it could be easily solved by a hard meta-clustering algorithm, such an ideal cluster correspondence is only very seldom encountered in practice, mainly due to the *instability* of most clustering algorithms. Thus, instead of such a perfect correspondence (8), we settle for a weaker one (7') in which the

⁸ $A_c^{(i)}$ is the column c of $A^{(i)}$, while $S_c^{(i)}$ is the row c of $S^{(i)}$.

⁹ By solving the constrained optimization problem $\min C(\alpha, \beta, \gamma) = \frac{1}{2} \sum_{i,c,s,g} \left(A_{sc}^{(i)} S_{cg}^{(i)} - \sum_{k=1}^{n_c} \alpha_{ck}^{(i)} \beta_{sk} \gamma_{kg} \right)^2$ s.t. $\alpha, \beta, \gamma \geq 0$.

rows of α can contain several significant entries, so that all biclusters $A_c^{(i)} \cdot S_c^{(i)}$ are recovered as combinations of bicluster prototypes $\beta_k \cdot \gamma_k$.

The nonnegativity constraints of PTF meta-clustering are essential both for allowing the interpretation of $\beta_k \cdot \gamma_k$ as bicluster prototypes, as well as for obtaining sparse factorizations. (In practice, the rows of α tend to contain typically one or only very few significant entries.)

The factorization (7') can be computed using the following multiplicative update rules (the proofs are straightforward generalizations of those for NMF and can also be found e.g. in [7]):

$$\begin{aligned}\alpha &\leftarrow \alpha * \frac{(A^T \cdot \beta) * (S \cdot \gamma^T)}{\alpha \cdot [(\beta^T \cdot \beta) * (\gamma \cdot \gamma^T)]} \\ \beta &\leftarrow \beta * \frac{A \cdot [\alpha * (S \cdot \gamma^T)]}{\beta \cdot [(\alpha^T \cdot \alpha) * (\gamma \cdot \gamma^T)]} \\ \gamma &\leftarrow \gamma * \frac{[\alpha * (A^T \cdot \beta)]^T \cdot S}{[(\alpha^T \cdot \alpha) * (\beta^T \cdot \beta)]^T \cdot \gamma}\end{aligned}\tag{9}$$

where ‘*’ and ‘—’ denote element-wise multiplication and division of matrices, while ‘·’ is ordinary matrix multiplication.

After convergence of the PTF update rules, we make the prototype gene clusters directly comparable to each other by normalizing the rows of γ to unit norm, as well as the columns of α such that $\sum_{i,c} \alpha_{(ic)k} = r$ (r being the number of runs) and then run NMF initialized with (β, γ) to produce the final factorization $X \approx A \cdot S$.

Before addressing real-world gene expression datasets, we evaluated our algorithm on *synthetic datasets* that match as closely as possible real microarray data. Clusters were modelled using a hidden-variable graphical model, in which each hidden variable A_c corresponds to the cluster of genes influenced by A_c (clusters can overlap since an observable variable X_g can be influenced by several hidden variables A_c).

We observed that although all algorithms produce quite low relative errors $\varepsilon_{rel} = \|X - A \cdot S\| / \|X\|$ (under 16%)¹⁰, they behave quite differently when it comes to recovering the original clusters. In a certain way, the *match* of the recovered clusters with the original ones is more important than the relative error (see [6] for the definition of the *match* between two sets of possibly *overlapping* clusters). On the synthetic datasets, PTF consistently outperformed the other meta-clustering algorithms in terms of recovering the original clusters. Also, among all *object-level* clustering algorithms tried (k-means, fuzzy k-means and NMF), only NMF behaved consistently well. The conceptual elegance of the combination of NMF as object-level clustering and PTF as meta-clustering thus pays off in terms of performance.

6. METACLUSTERING A LUNG CANCER GENE EXPRESSION DATASET

We applied our meta-clustering approach to a large lung cancer microarray dataset available from the Meyerson lab at Harvard. Using HG-U95Av2 Affymetrix oligonucleotide microarrays, Bhattacharjee et al. [8] have measured mRNA expression levels of 12600 genes in 186 lung tumor samples (139 adenocarcinomas, 21 squamous cell lung carcinomas, 6 small cell lung cancers, 20 pulmonary carcinoids) and 17 normal lung samples (203 samples in total). Whereas the non-adeno classes are more or less well defined histologically, adenocarcinomas are very heterogeneous, with poorly defined histological and molecular level sub-classifications, despite the large variability in survival times and responsiveness to medication. We applied our algorithm to the *full* dataset and used the histological classification of the non-adeno samples (provided in the supplementary material to the original paper) as a gold standard for the

¹⁰ Except for fuzzy k-means which misbehaves for large numbers of clusters.

evaluation of the biclustering results. To eliminate the bias towards genes with high expression values, the gene expression matrix was normalized by separate scalings of the genes to equalize their norms.¹¹

Although nonnegative factorizations have the advantage of obtaining sparse and easily interpretable decompositions, they cannot directly account for gene down-regulation. To deal with gene down-regulation in the context of NMF, we extended the gene expression matrix with new “down-regulated genes” $g' = \text{pos}(\text{mean}(g_{\text{normal}}) - g)$ associated to the original genes g , where $\text{mean}(g_{\text{normal}})$ is the average of the gene over the *normal* samples, while $\text{pos}(\cdot)$ is the Heaviside step function.

To avoid overfitting, we estimated the number of clusters n_c as the number of dimensions around which the change in relative error $d\varepsilon/dn_c$ of the factorization of the real data “reaches from above” the change in relative error obtained for a randomized dataset (similar to [9], see Figure 2).

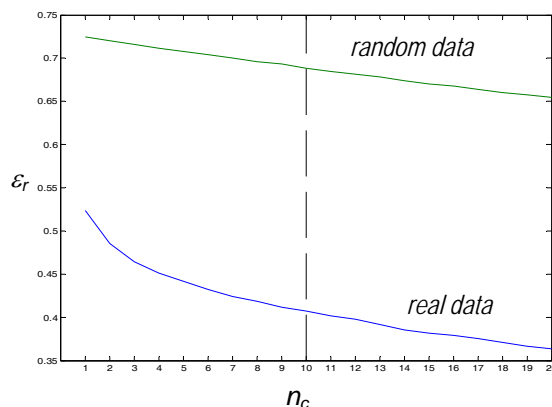


Figure 2. Determining the dimensionality of the dataset

We then used our metaclustering algorithm to factorize the extended gene expression matrix X by running PTF over 20 NMF runs with the number of clusters determined above ($n_c=10$). (The matrix X has 203 rows (samples) and $2 \times 3529 = 7058$ columns, i.e. extended genes.) Figure 3 shows the resulting sample cluster matrix A . Note that the algorithm has recovered the non-adenos sample clusters with high accuracy, despite the very large number of variables (genes), many of these potentially irrelevant in this problem. More precisely, the clusters 6, 7, 8 on the diagonal of A correspond to the classes ‘squamous’, ‘small cell’ and ‘normal’ respectively, while clusters 9 and 10 are two subtypes of carcinoids (which, like adenomas, are heterogeneous and form two partially overlapping clusters). Note that unlike most clustering methods, our approach allows for overlapping clusters. The accuracy of the sample cluster overlaps can be tested for example in the case of the samples AD341, AD275, AD234 and AD241, which were classified by histopathologists as adeno-squamous and also appear in the overlap of our ‘squamous’ cluster with other ‘adeno’ clusters. Similarly, the overlap between the small cell and squamous sample clusters corresponds to mixed small cell-squamous cases, which are mentioned in the literature.

The gene clusters S recovered genes with well known involvement in the lung cancer subtypes under study. For example, the squamous cluster contained numerous keratin genes (keratins 6A, 5, 17, 14, 13, 16, 19), typical for squamous differentiation, the keratinocyte-specific protein stratifin, the p53 tumor suppressor analog TP73L, etc. Moreover, we observed that the genes with large S_{cg} tend to be differentially expressed between the classes (according to a t-test), although the class information was never provided to the algorithm.

The Table below shows the relative reconstruction errors $\varepsilon = \|X - A \cdot S\|_F / \|X\|_F$ for k-means, fuzzy k-means,¹² NMF and PTF (we display the mean, STD and min errors for 20 clustering runs of each algorithm and clustering dimensions 5, 10, 14 and 20). PTF meta-clustering exhibits slightly smaller relative errors

¹¹ Genes with nearly constant and very low expression values (average expression levels < 30 and standard deviation < 50) had been discarded, leaving 3529 genes that are significantly expressed in the lung cancer samples.

¹² For both plain and fuzzy k-means, A is constructed from the cluster membership function, while S is given by the cluster centers.

than the best runs of k-means, fuzzy k-means and NMF, and the improvement also increases slightly with the number of clusters.

n_c	k-means mean(STD)	k-means best run	Fcm mean(STD)	Fcm best run	NMF mean(STD)	NMF best run	PTF
5	0.4460 (0.0010)	0.4445	0.4459 (0.0007)	0.4451	0.4408 (0.0004)	0.4406	0.4406
10	0.4247 (0.0030)	0.4196	0.4219 (0.0017)	0.4184	0.4062 (0.0005)	0.4056	0.4052
14	0.4151 (0.0035)	0.4104	0.4111 (0.0025)	0.4068	0.3866 (0.0009)	0.3855	0.3849
20	0.4002 (0.0045)	0.3936	0.3978 (0.0039)	0.3895	0.3642 (0.0006)	0.3634	0.362

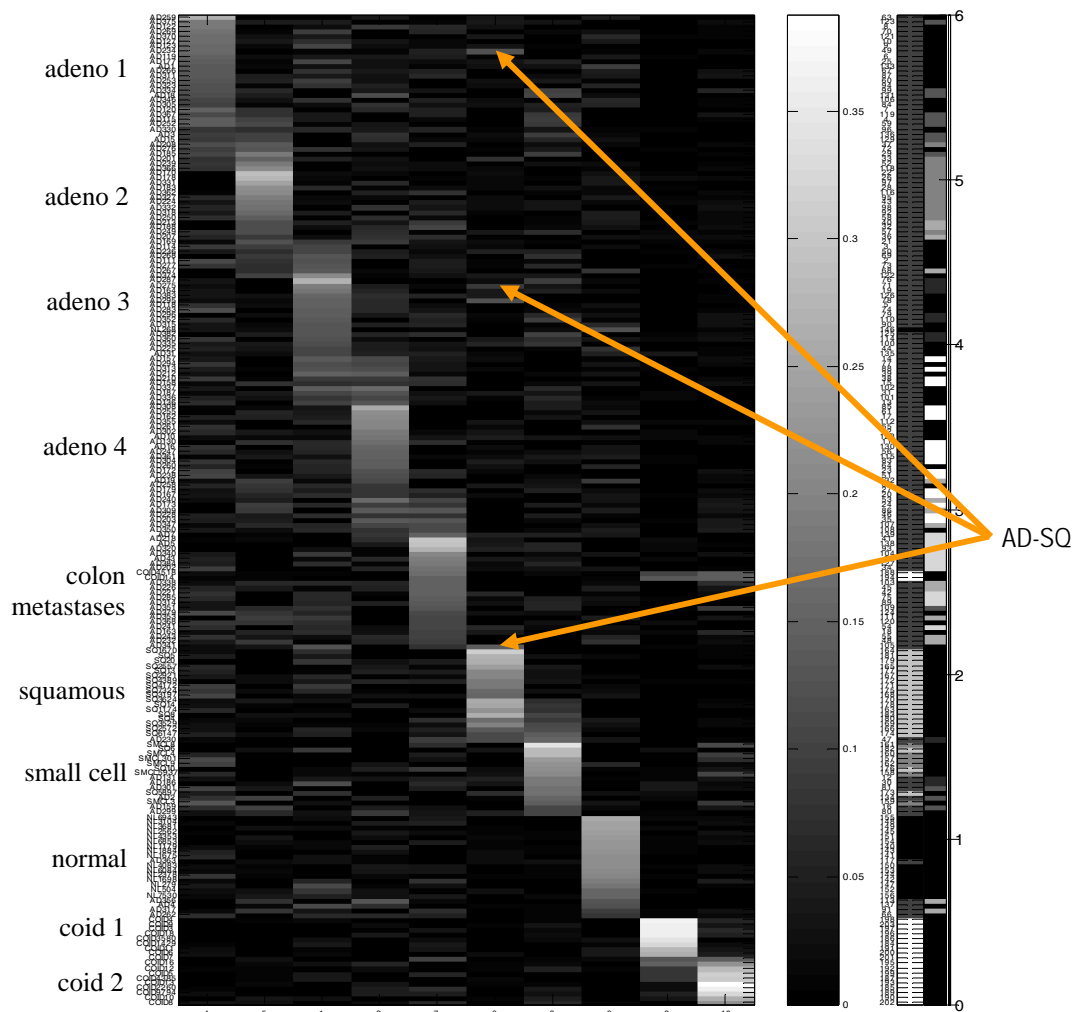


Figure 3. The sample clusters (matrix A – normalized columns)

Fuzzy k-means clustering required a very delicate fine-tuning of the fuzzy exponent for obtaining non-trivial clusters: we used a fuzzy exponent of 1.1, whereas a slightly higher value of 1.15 produced only trivial, non-informative clusters. However, such a small fuzzy exponent leads to very categorical membership degrees even for very small differences in distance between gene profiles, so the results are similar to those of hard clustering (plain k-means). This is probably due to the different interpretations of cluster overlap in fuzzy k-means and NMF respectively: whereas fuzzy k-means views overlaps in terms of

membership degrees, NMF and PTF interpret overlaps as *mixtures* (as in the case of the adeno-squamous samples – see also assumptions (1) and (2) above).

However, much more important than a small improvement in error is the *stability* of the resulting clusters. All studied methods recovered the non-adeno sample clusters satisfactorily (with differences in the adeno clusters that cannot be judged based on current histological evidence). To study the variability of the gene clusters S in different runs of each algorithm, we computed the average relative differences $\|S_i - S_j\|/\|S_i\|$ between pairs of gene cluster matrices S_i obtained in 20 different runs (of each algorithm) for $n_c=10$ clusters, as well as the corresponding *mismatches* between gene clusters matrices. We display the cluster mismatches for progressively larger cutoff thresholds¹³ to show that the differences between clusters obtained in different runs involve not just the small, but also the large coefficients of S .

	k-means mean(STD)	Fcm mean(STD)	NMF mean(STD)	best NMF mean(STD)	PTF mean(STD)
relative difference	0.1755 (0.0238)	0.0803 (0.0265)	0.3117 (0.0599)	0.1591 (0.0939)	0.0354 (0.0167)
mismatch $S > \theta_{0g}$	0.1730 (0.0204)	0.0748 (0.0225)	0.1457 (0.0296)	0.0747 (0.0433)	0.0165 (0.0080)
mismatch $S > 2\theta_{0g}$	0.3345 (0.0973)	0.1354 (0.0987)	0.2720 (0.0579)	0.1397 (0.0836)	0.0269 (0.0130)
mismatch $S > 3\theta_{0g}$	0.7121 (0.1658)	0.3215 (0.1863)	0.3059 (0.0885)	0.1228 (0.0738)	0.0283 (0.0130)

As the inter-run variability of S is quite large for all clustering methods tried¹⁴, except PTF (e.g. 31% for NMF with $n_c=10$), using such clustering algorithms for determining gene clusters is highly unreliable. On the other hand, PTF is preferable to the other methods due to its increased stability (only about 3% variability of S).

Moreover, PTF is preferable to fuzzy k-means in clustering gene expression data since it is able to reconstruct gene profiles of samples that represent *mixtures* of frequently occurring profiles. For example, the Meyerson dataset studied here contains numerous samples with expression profiles similar to a squamous profile SQ, as well as other samples with a different, adeno profile AD (by a *gene expression profile* we mean a set of gene expression values for all genes represented on the microarray chip). These two different sample groups will lead to two distinct columns of A representing the SQ and AD profiles. However, the Meyerson dataset also contains *adeno-squamous* samples with a mixed AD + SQ profile, which can be easily represented by NMF and PTF factorizations, but not by fuzzy or plain k-means.

7. CONCLUSIONS AND RELATED WORK

In this paper we argue that biological systems are exceptionally complex processors of information, in which digital and analog processes are inextricably intertwined. While the analog processes are hard to measure on a genomic scale, we observed that the digital nature of genomic information has allowed the recent explosion of high throughput microarray data, which has enabled new breakthroughs in our understanding of the control of gene expression.

However, despite its enormous potential, microarray data has proved difficult to analyze, due to a number of domain-specific problems. In the context of an unsupervised analysis (clustering) of microarray data, we showed that these problems can be elegantly dealt with by using nonnegative matrix factorizations (NMF) to cluster genes and samples simultaneously while allowing for bicluster overlaps. Moreover, we were able to address the instability of NMF by employing Positive Tensor Factorization to perform a two-way meta-clustering of the biclusters produced in several different clustering runs.

¹³ Cluster membership degrees S_{cg} were considered significant if they exceeded the thresholds $\theta_{0g} = 1/\sqrt{n_g}$, $2\theta_{0g}$ and $3\theta_{0g}$ respectively. Note that the rows of S are normalized to unit norm.

¹⁴ It also increases with the number of clusters (results not shown).

The application of our approach to a large lung cancer dataset proved computationally tractable and was able to recover the histological classification of the various cancer subtypes represented in the data-set.

A detailed review of the clustering methods applicable to gene expression data is out of the scope of this paper, due to space constraints. Briefly, our approach is significantly different from other biclustering approaches, such as Cheng and Church's [11], which is based on a simpler additive model that is not scale invariant (and thus problematic in the case of gene expression data). On the other hand, approaches based on singular value decompositions, or the Iterative Signature Algorithm [12], tend to produce holistic decompositions as opposed to the more parts-based ones obtained here (holistic decompositions being typically hard to interpret in this domain). Closest to our approach are [9] and [10]. Kim and Tidor [9] used NMF decompositions for analyzing a yeast gene expression compendium, but their approach still suffers from the instability of NMF. On the other hand, Brunet et al. [10] used NMF for *non-overlapping* iterative clustering of samples, rather than *biclustering* as we do.

In this paper we show that nonnegative decompositions such as NMF and PTF can be combined in a non-trivial way to obtain an improved meta-clustering algorithm for *gene expression data*. The approach deals with *overlapping clusters* and alleviates the annoying *instability* of currently used algorithms by using an advanced two-way meta-clustering technique based on *tensor* (rather than matrix) factorizations.

It is encouraging that PTF recovers the main known lung cancer subtypes, including subtle classifications of certain samples in overlapping classes (adeno-squamous), in a large dataset in which 70% of the samples represent the poorly characterized adenocarcinoma.

And although the improvements in error obtained by PTF are only marginal, it leads to increased stability of the gene clusters (which are extremely important for determining the genes causing the disease). Moreover, PTF proves more adequate in this domain than other methods like fuzzy k-means, due to its interpretation of cluster overlaps as mixtures, fuzzy k-means being extremely sensitive to minute changes in the fuzzy exponent.

REFERENCES

1. NCBI. *The National Center for Biotechnology Information* www.ncbi.nih.gov
2. SCHENA M, SHALON D, DAVIS RW, BROWN PO. 1995. *Quantitative Monitoring Of Gene-expression Patterns With A Complementary-DNA Microarray*. *Science* 270: (5235) 467-470 Oct 20 1995.
3. LEE D.D., H.S. SEUNG. *Algorithms for non-negative matrix factorization*. Proceedings NIPS*2000, MIT Press, 2001.
4. LEE D.D., H.S. SEUNG. *Learning the parts of objects by non-negative matrix factorization*. *Nature*, vol. 401, no. 6755, pp. 788-791, 1999.
5. BEZDEK J.C. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, 1981.
6. BADEA L. *Clustering and metaclustering with Nonnegative Matrix Decomposition*. Proceedings of the European Conference on Machine Learning ECML-05, Lecture Notes in Artificial Intelligence LNAI 3720, pp. 10-22, Springer Verlag 2005.
7. WELLING M., WEBER M. *Positive tensor factorization*. *Pattern Recognition Letters* 22(12): 1255-1261.
8. BHATTACHARJEE et al. *Classification of human lung carcinomas by mRNA expression profiling reveals distinct adenocarcinoma subclasses*. *PNAS* 2001 Nov. 20; 98(24):13790-5.
9. KIM P.M., TIDOR B. *Subsystem identification through dimensionality reduction of large-scale gene expression data*. *Genome Research* 2003 Jul;13(7):1706-18.
10. BRUNET J.P. et al. *Metagenes and molecular pattern discovery using matrix factorization*. *PNAS* 101(12):4164-9, 2004.
11. CHENG Y. CHURCH G. *Biclustering of expression data*. *Proc. ISMB-2000*, 93-103.
12. BERGMANN S, et al. *Iterative signature algorithm for the analysis of large-scale gene expression data*. *Phys. Rev. E*. Mar. 2003; 67.

Received August 28, 2007