Computer Science and Artificial Intelligence Laboratory

Technical Report

MIT-CSAIL-TR-2007-037 July 6, 2007

Hierarchical Dirichlet Process-Based Models For Discovery of Cross-species Mammalian Gene Expression

Georg K. Gerber, Robin D. Dowell, Tommi S. Jaakkola, and David K. Gifford

massachusetts institute of technology, cambridge, ma 02139 usa — www.csail.mit.edu

Hierarchical Dirichlet Process-Based Models For Discovery of Cross-species Mammalian Gene Expression Programs

Georg K. Gerber[1,2], Robin D. Dowell[1],
Tommi S. Jaakkola[1], David K. Gifford[1,3,*]

[1]MIT Computer Science and Artificial Intelligence Laboratory, Cambridge, MA
[2]Harvard-MIT Division of Health Sciences and Technology, Cambridge, MA
[3]Whitehead Institute for Biomedical Research, Cambridge, MA
[*]gifford@mit.edu, 32 Vassar Street G542, Cambridge, MA 02139

July 1, 2007

Abstract

An important research problem in computational biology is the identification of *expression programs*, sets of co-activated genes orchestrating physiological processes, and the characterization of the functional breadth of these programs. The use of mammalian expression data compendia for discovery of such programs presents several challenges, including: 1) cellular inhomogeneity within samples, 2) genetic and environmental variation across samples, and 3) uncertainty in the numbers of programs and sample populations. We developed GeneProgram, a new unsupervised computational framework that uses expression data to simultaneously organize genes into overlapping programs and tissues into groups to produce maps of inter-species expression programs, which are sorted by generality scores that exploit the automatically learned groupings. Our method addresses each of the above challenges by using a probabilistic model that: 1) allocates mRNA to different expression programs that may be shared across tissues, 2) is hierarchical, treating each tissue as a sample from a population of related tissues, and 3) uses Dirichlet Processes, a non-parametric Bayesian method that provides prior distributions over numbers of sets while penalizing model complexity. Using real gene expression data, we show that GeneProgram outperforms several popular expression analysis methods in recovering biologically interpretable gene sets. From a large compendium of mouse and human expression data, GeneProgram discovers 19 tissue groups and 100 expression programs active in mammalian tissues. Our method automatically constructs a comprehensive, body-wide map of expression programs and characterizes their functional generality. This map can be used for guiding future biological experiments, such as discovery of genes for new drug targets that exhibit minimal "cross-talk" with unintended organs, or genes that maintain general physiological responses that go awry in disease states. Further, our method is general, and can be applied readily to novel compendia of biological data.

1 Introduction

The great anatomic and physiologic complexity of the mammalian body arises from the coordinated expression of genes. A fundamental challenge in computational biology is the identification of sets of co-activated genes in a given biological context and the characterization of the functional breadth of such sets. Understanding of the functional generality of gene sets has both practical and theoretical utility. Sets of genes that are very specific to a particular cell type or organ may be useful as diagnostic markers or drug targets. In contrast, sets of genes that are active across diverse cell types can give us insight into unexpected developmental and functional similarities among tissues. While there has been considerable effort in systems biology to understand the structure and organization of co-expressed sets of genes in isolated tissues in the context of pathological processes, such as cancer and infection [37, 43, 57, 71], relatively little attention has been given to this task in the context of normal physiology throughout the entire body [63, 65]. By analyzing gene expression in this latter context, we can gain an understanding of baseline gene expression programs and characterize the specificity of such programs in reference to organism-wide physiological processes.

In this work, we use a large compendium of human and mouse body-wide gene expression data from representative normal tissue samples to discover automatically a set of biologically interpretable expression programs and to characterize quantitatively the specificity of each program. Large genome-wide mammalian expression data compendia present several new challenges that do not arise when analyzing data from simpler organisms. First, tissue samples usually represent collections of diverse cell-types mixed together in different proportions. Even if a sample consists of a relatively homogenous cell population, the cells can still behave asynchronously, due to factors such as microenvironments within the tissue that receive different degrees of perfusion. Second, each tissue sample is often from a different individual, so that the compendium represents a patchwork of samples from different genetic and environmental backgrounds. Finally, the number of expression programs and distinct cell populations present in a compendium is effectively unknown *a priori*.

We present a novel methodology, GeneProgram, designed for analyzing large compendia of mammalian expression data, which simultaneously compresses sets of genes into expression programs and sets of tissues into groups. Specific features of our algorithm address each of the above issues relating to analysis of compendia of mammalian gene expression data. First, our method handles tissue inhomogeneity by allocating the total mRNA recovered from each tissue to different gene expression programs, which may be shared across tissues. The number of expression programs used by a tissue therefore relates to its functional homogeneity. We address the second issue, of tissue samples coming from different individuals, by explicitly modeling each tissue as a sample from a population of related tissues. That is, related tissues are assumed to use similar expression programs and to similar extents, but the precise number of genes and the identity of genes used from each program may vary in each sample. Additionally, populations of related tissues are discovered automatically, and provide a natural means for characterizing the generality of expression programs. Finally, uncertainty in the numbers of tissue groups and expression programs is handled by using a non-parametric Bayesian technique, Dirichlet Processes, which provides prior distributions over numbers of sets.

To understand the novel contributions of the GeneProgram algorithm, it is useful to view our framework in the context of a lineage of unsupervised learning algorithms that have previously been applied to gene expression data. These algorithms are diverse, and can be classified according to various features, such as whether they use matrix factorization methods [2], heuristic scoring functions [14], generative probabilistic

models [62], statistical tests [58, 68], or some combinations of these methods [6, 17]. The simplest methods, such as K-means clustering, assume that all genes in a cluster are co-expressed across all tissues, and that there is no overlap among clusters. Next in this lineage are biclustering algorithms [14, 68, 13, 44, 69], which assume that all genes in a bicluster are co-expressed across a subset rather than across all tissues. In many such algorithms, genes can naturally belong to multiple biclusters.

GeneProgram is based on two newer unsupervised learning frameworks, the *topic model* [23, 30] and the Hierarchical Dirichlet Process mixture model [70]. The topic model formalism allows GeneProgram to further relax the assumptions of typical biclustering methods, through a probabilistic model in which each gene in an expression program has a (potentially) different chance of being co-expressed in a subset of tissues. The hierarchical structure of our model, which encodes the assumption that groups of tissues are more likely to use similar sets of expression programs in similar proportions, also provides advantages. Hierarchical models tend to be more robust to noise, because statistical strength is "borrowed" from items in the same group for estimating the parameters of clusters. Additionally, hierarchical models can often be interpreted more easily—in the context of the present application, the inferred expression programs will tend to be used by biologically coherent sets of tissues. Finally, through the Dirichlet Process mixture model formalism, GeneProgram automatically infers the numbers of gene expression programs and tissue groups. Because this approach is fully Bayesian, the numbers of mixture components can be effectively integrated over during inference, and the complexity of the model is automatically penalized. This is in contrast to previous methods that either require the user to specify the number of clusters directly or produce as many clusters as are deemed significant with respect to a heuristic or statistical score without providing a global complexity penalty. We note that Medvedovic *et al.* have also applied Dirichlet Process mixture models to gene expression analysis, but not in the context of topic models, Hierarchical Dirichlet Processes, or mammalian data [46].

As with previous methods [3, 8, 64, 75], we leverage the power of cross-species information to discover biologically relevant sets of co-expressed genes. However, these previous analyses generally required genes to be co-expressed across large sets of experiments [8, 64, 75, 41]. In contrast, GeneProgram uses expression data more flexibly, and is thus able to produce a refined picture of gene expression across species: expression programs may be used by only a subset of tissues, and may be unique to one species or shared across multiple species; tissue groups are similarly flexible. This probabilistic view of expression programs captures the intuition that the general structure of many programs is evolutionarily conserved, but some genes may be interchanged or lost.

The remainder of this paper is organized as follows. In Section 2, we present background material on ordinary and Hierarchical Dirichlet Process mixture models, which are a core component of the GeneProgram probability model. In Section 3, we provide a detailed description of the GeneProgram algorithm and probability model. In Section 4, we apply GeneProgram to the Novartis Gene Atlas v2 [65], consisting of expression data for 79 human and 61 mouse tissues. Using this data set, we compare GeneProgram's ability to recover biologically relevant gene sets to that of biclustering methods, and produce a body-wide map of expression programs organized by their functional generality scores. Finally, in Section 5, we discuss the significance of our results and comment on possible future research directions.

2 Dirichlet Processes

The task of assigning data to clusters is a classic problem in machine learning and statistics. A common approach to this problem is to construct a model in which data is generated from a mixture of probability distributions.

In finite mixture models, data is assumed to arise from a mixture with a pre-determined number of components [45]. The difficulty with such models is that the appropriate number of mixture components is not known *a priori* for many modeling applications. This issue is generally addressed by constructing a series of models with different numbers of components, and evaluating each model using some quality score [45].

An alternate, fully Bayesian approach is to build an *infinite* mixture model, in which the number of mixture components is potentially unlimited, and is itself a random variable that is part of the overall model. Obviously, only a finite number of mixture components can have data assigned to them. However, we still imagine the data as arising from an infinite number of components; as more data is collected, more components may be used to model the data more accurately. Thus, the infinite mixture model is a nonparametric model, in the sense that the number of model parameters grows with the amount of data. The challenge with such a model is how to place an appropriate prior on the infinite number of mixture component parameters and weights.

The Dirichlet Process (DP), a type of stochastic process first introduced in the 1960's [24] and originally of mostly theoretical interest [21, 22], has recently become an important modeling tool as a prior distribution for infinite mixture models. In this section, we will introduce the main concepts of DPs necessary to understand the GeneProgram model. In this regard, we will focus on a constructive definition of DPs in the context of priors for infinite mixture models. This development, which avoids measure theory, closely parallels that presented by [49] and [54].

A recent extension to the standard DP model is the Hierarchical Dirichlet Process (HDP), in which dependencies are specified among a set of DPs by arranging them in a tree structure [70]. HDPs are useful as priors for hierarchical mixture models, in which data is arranged into populations that preferentially share the usage of mixture components. In this section, we will discuss the original HDP formulation by Teh *et al.* in the context of infinite mixture models.

The use of DPs for real-world applications is predicated on practical inference methods. A great advance in this regard has been the development of efficient Markov Chain Monte Carlo (MCMC) methods for approximate inference for infinite mixture models using DP priors [60, 50, 54]. Although other approximate inference methods have been developed [47, 9, 39], MCMC remains the most widely used and versatile method. In particular, efficient MCMC schemes have been developed for HDP models [70], and can be readily extended for the GeneProgram model. Thus, our discussion of DP inference in this section will be restricted to MCMC methods.

The remainder of this section is organized as follows. First, we describe how Dirichlet Processes arise as priors in terms of the infinite limit of mixture models. Next, we describe the extension of DPs to HDPs. Finally, we describe basic MCMC sampling schemes for DPs and HDPs.

2.1 Probability models

2.1.1 Bayesian finite mixture models

We begin by defining a typical Bayesian finite mixture model, which we will subsequently extend to the infinite case. Figure 1 depicts the model using standard graphical model notation with plates. The model consists of J mixture components, where each component j has associated with it a mixture weight denoted π_j and a parameter vector denoted $\boldsymbol{\theta}_j$. Assume we have N data points denoted x_i, where $1 \leq i \leq N$. Each data point is assigned to a mixture component via an indicator variable z_i, i.e., the probability that data point i is assigned to component j is $\mathrm{p}(\mathsf{z}_i = j \mid \boldsymbol{\pi}) = \pi_j$ or $\mathsf{z}_i \mid \boldsymbol{\pi} \sim \text{Multinomial}(\cdot \mid \boldsymbol{\pi})$. The conditional likelihood for each data point may then be written as:

$$\mathrm{p}(\mathsf{x}_i \mid \mathsf{z}_i = j, \boldsymbol{\theta}) = F(\mathsf{x}_i \mid \boldsymbol{\theta}_j)$$

Here, $F(\cdot \mid \cdot)$ is a probability density function parameterized by $\boldsymbol{\theta}$.

To complete the model, we need to define prior distributions over the parameters. We will assume that the component parameters are drawn i.i.d. from some base distribution H, i.e., $\boldsymbol{\theta}_j \sim H(\cdot)$. We also need to specify a prior distribution for the weight parameters. As is typical for Bayesian mixture models, we will assume a symmetric Dirichlet prior on the mixture weights, i.e., $\boldsymbol{\pi} \mid J, \alpha \sim \text{Dirichlet}(\cdot \mid \alpha/J)$. One consequence of using a symmetric prior is that it is not sensitive to the order of the component parameters. Note that the Dirichlet prior is conjugate to the multinomially distributed weights, so that the posterior is also a Dirichlet distribution.

To summarize, our J-dimensional mixture model is defined as:

$$\begin{aligned}
\boldsymbol{\pi} \mid \alpha, J &\sim \text{Dirichlet}(\cdot \mid \alpha/J) \\
\boldsymbol{\theta}_j \mid H &\sim H(\cdot) \\
\mathsf{z}_i \mid \boldsymbol{\pi} &\sim \text{Multinomial}(\cdot \mid \boldsymbol{\pi}) \\
\mathsf{x}_i \mid \mathsf{z}_i = j, \boldsymbol{\theta} &\sim F(\cdot \mid \boldsymbol{\theta}_j)
\end{aligned}$$

In mixture models, we are primarily interested in knowing which component each data point i has been assigned to—the weights $\boldsymbol{\pi}$ are to some extent "nuisance" parameters. It is possible to derive closed form expressions for the data point assignment variable posterior distributions with the mixture weights integrated out. These posterior distributions will be particularly useful in the extension to the infinite mixture model. Note that although the assignment variables $\mathbf{z}$ are conditionally independent given the weights, they become dependent if we integrate out the weights (i.e., the probability of assigning a data point to a particular component depends on the assignments of all other data points). As it turns out, the probability of assigning data point i to some component j given assignments of all other data points can be written as a simple closed form expression (see [54]):

$$\mathrm{p}(\mathsf{z}_i = j \mid \mathbf{z}_{-i}, \alpha, J) = \int \mathrm{p}(\mathsf{z}_i = j \mid \boldsymbol{\pi}) \mathrm{p}(\boldsymbol{\pi} \mid \mathbf{z}_{-i}, \alpha, J) d\boldsymbol{\pi}$$

$$\mathrm{p}(\boldsymbol{\pi} \mid \mathbf{z}_{-i}, \alpha, J) \propto \mathrm{p}(\mathbf{z}_{-i} \mid \boldsymbol{\pi}) \mathrm{p}(\boldsymbol{\pi} \mid \alpha, J)$$

$$\Rightarrow \mathrm{p}(\mathsf{z}_i = j \mid \mathbf{z}_{-i}, \alpha, J) \propto \int \mathrm{p}(\mathsf{z}_i = j \mid \boldsymbol{\pi}) \mathrm{p}(\mathbf{z}_{-i} \mid \boldsymbol{\pi}) \mathrm{p}(\boldsymbol{\pi} \mid \alpha, J) d\boldsymbol{\pi}$$

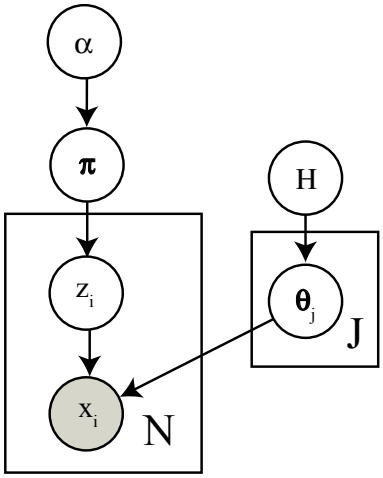


Figure 1: A graphical model depiction of a finite mixture model with J mixture components and N data items. Circles represent variables, and arrows denote dependencies among variables. Vectors are depicted with bold type, and observed variables are shown inside shaded circles. Rectangles represent plates, or repeated sub-structures in the model.

$$\Rightarrow \mathrm{p}(\mathsf{z}_i = j \mid \mathbf{z}_{-i}, \alpha, J) \propto \frac{n_j^{-i} + \alpha/J}{N - 1 + \alpha} \tag{1}$$

Here, $\mathbf{z}_{-i}$ denotes the assignments of all data excluding data point i, and n_j^{-i} denotes the number of data points assigned to component j excluding data point i. The second line of the derivation follows simply from Bayes' theorem. The final line of the derivation follows from conjugacy between the Dirichlet prior on the weights and the multinomial distribution on the assignment variables. Thus, the density function under the integral is that of a non-symmetrical Dirichlet distribution, allowing us to derive the final closed form expression.

2.1.2 Infinite mixture models and Dirichlet Processes

In this subsection we show how the Dirichlet Process arises as a prior for infinite mixture models.

Figure 2 depicts an infinite mixture model using standard graphical model notation with plates. As can be seen from the figure, the model is almost structurally identical to the finite version. The distinguishing feature is that the weight and parameter vectors are now infinite dimensional.

The challenge with this model is then to define an appropriate prior for the infinite dimensional parameters and weights. As with any mixture model, the infinite dimensional weights must sum to one. A probability distribution that generates such weights is the *stick-breaking* distribution, denoted Stick(α), where α is a scaling or concentration parameter (discussed in more detail below). This distribution is defined constructively. Intuitively, we imagine starting with a stick of unit length and breaking it at a random point. We retain one of the pieces, and break the second piece again at a random point. This process is repeated infinitely, producing a set of random weights that sum to one with probability one [60]. To be more precise,

the jth weight π_j is constructed as:

$$\pi'_j \mid \alpha \sim \text{Beta}(1, \alpha)$$

$$\pi_j = \pi'_j \prod_{l=1}^{j-1} (1 - \pi'_l)$$

The infinite mixture model can be constructed using the stick-breaking distribution as a prior on the mixture weights and the base distribution H as a prior on the component parameters. This can be summarized as:

$$\boldsymbol{\pi} \mid \alpha \sim \text{Stick}(\alpha)$$

$$\boldsymbol{\theta}_j \mid H \sim H(\cdot)$$

$$\mathsf{z}_i \mid \boldsymbol{\pi} \sim \text{Multinomial}(\cdot \mid \boldsymbol{\pi})$$

$$\mathsf{x}_i \mid \mathsf{z}_i = j, \boldsymbol{\theta} \sim F(\cdot \mid \boldsymbol{\theta}_j)$$

Note that this construction produces a vector $\boldsymbol{\pi}$ with a countably infinite number of dimensions, whose components all sum to one, and H is sampled independently a countably infinite number of times to generate the mixture component parameter values.

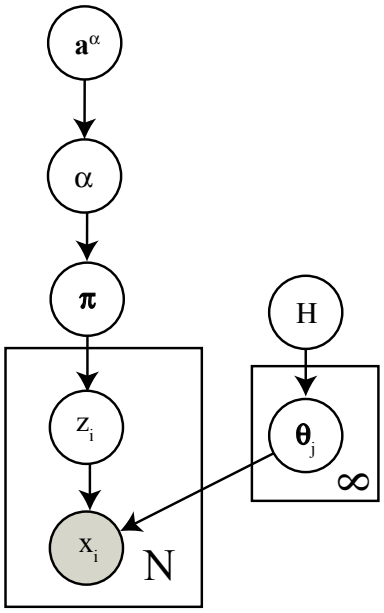


Figure 2: A graphical model depiction of the infinite mixture model. Circles represent variables, and arrows denote dependencies among variables. Vectors are depicted with bold type, and observed variables are shown inside shaded circles. Rectangles represent plates, or repeated sub-structures in the model.

To establish the connection between Dirichet Processes and the model described above, we consider the distribution over all possible component parameter values for the infinite mixture model. This distribution

will be non-zero at a countably infinite number of values. Formally, we denote this distribution by G and can write it as:

$$G(\boldsymbol{\psi}) = \sum_{j=1}^{\infty} \pi_j \delta(\boldsymbol{\psi} - \boldsymbol{\theta}_j)$$

Here, ψ is an arbitrary parameter value, and $\delta(\cdot)$ is the standard delta-function, which is non-zero only when its argument is zero.

Each distribution G thus constructed can be viewed as a sample from a stochastic process, which can in fact be proven to be the Dirichlet Process (see [36] and [60]). In general, we will characterize a Dirichlet Process by a scalar parameter α, called the concentration parameter, and a base distribution H. A sample from a Dirichlet Process, which we denote $G \mid \alpha, H \sim \text{DP}(\alpha, H)$, is thus a distribution that is non-zero over a countably infinite number of values (with probability one). As we have seen, each sample effectively parameterizes an infinite dimensional mixture model.

The concentration parameter α affects the expected number of mixture components containing data items when the DP is used as a prior for the infinite mixture model. As shown in [5], the expected number of non-empty mixture components J depends only on α and the number of data points N:

$$E[J \mid \alpha, N] = \alpha \sum_{l=J-1}^{N} \frac{1}{\alpha + l - 1} \approx \alpha \ln \left(\frac{N + \alpha}{\alpha} \right)$$

Thus, we see that the number of non-empty components grows approximately as the logarithm of the size of the data set. Further, we see that the number of components grows as the concentration parameter α increases.

To make our model fully Bayesian, we would like to treat the concentration parameter α as a random variable and place a prior on it. The Gamma distribution is commonly used as a prior for α, in part because efficient inference is possible with this prior, and also because appropriate parameter choices result in a relatively uninformative prior [49]. Thus, we place a Gamma prior on α with hyperparameters $\boldsymbol{a}^{\alpha}$, i.e., $\alpha \mid \boldsymbol{a}^{\alpha} \sim \text{Gamma}(a_1^{\alpha}, a_2^{\alpha})$.

2.1.3 Hierarchical Dirichlet Process models

In this section, we describe the Hierarchical Dirichlet Process (HDP) models introduced by Teh *et al.* [70]. As in the previous section on DPs, we will present HDPs in terms of priors for infinite mixture models. We will describe only a two-level hierarchical model for clarity; additional levels are simply added by applying the model construction process recursively.

Figure 3 depicts a basic HDP using standard graphical model notation with plates. In HDP models, we assume that data is divided into T subsets, each consisting of N_t data points denoted x_{ti}, where $1 \leq t \leq T$ and $1 \leq i \leq N_t$. Each such data set division is modeled by an infinite mixture model with weights $\boldsymbol{\pi}_t$ and component assignment variables z_{ti}. These infinite mixture models are not independent; the mixtures share component parameters $\boldsymbol{\theta}$ and a common Dirichlet Process prior.

The dependencies among the infinite mixture models can be understood in terms of a construction using the stick-breaking distribution. Beginning at the top of the model, we imagine drawing a sample G from a

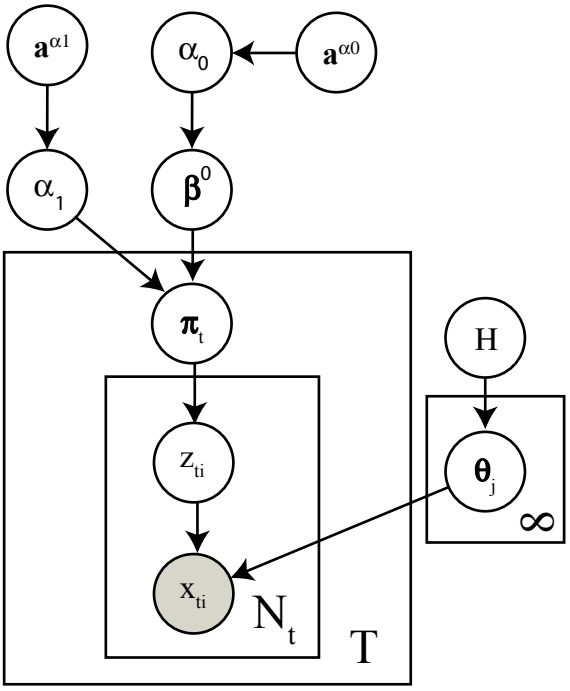


Figure 3: A graphical model depiction of the Hierarchical Dirichlet Process represented as an infinite mixture model. Circles represent variables, and arrows denote dependencies among variables. Vectors are depicted with bold type, and observed variables are shown inside shaded circles. Rectangles represent plates, or repeated sub-structures in the model.

Dirichlet Process, i.e., $G \mid \alpha_0, H \sim \text{DP}(\alpha_0, H)$. Recall that we can write this sample as:

$$G(\boldsymbol{\psi}) = \sum_{j=1}^{\infty} \beta_j^0 \delta(\boldsymbol{\psi} - \boldsymbol{\theta}_j)$$

Here, $\boldsymbol{\theta}_j$ are drawn i.i.d. from the base distribution H, and $\boldsymbol{\beta}^0 \mid \alpha_0 \sim \text{Stick}(\alpha_0)$.

We next form a second DP using the sample G itself as a base distribution, i.e., we construct $\text{DP}(\alpha_1, G)$. We then generate i.i.d. samples from this DP for each of the T sub-models, i.e., $G_t \mid \alpha_1, G \sim \text{DP}(\alpha_1, G)$. Each sample can be written as:

$$G_t(\boldsymbol{\psi}) = \sum_{j=1}^{\infty} \pi_{tj} \delta(\boldsymbol{\psi} - \boldsymbol{\theta}_j)$$

Notice that these distributions must necessarily be non-zero only at the same points $\boldsymbol{\theta}_j$ as G is. We have now constructed a set of T dependent infinite mixture models, where each model has separate (but dependent) weights $\boldsymbol{\pi}_t$ and shared component parameters $\boldsymbol{\theta}$.

It can be shown that the weights $\boldsymbol{\pi}_t$ can be constructed via a stick-breaking process using the top-level

weights $\boldsymbol{\beta}^0$ (see [70]):

$$\pi'_{tj} \sim \text{Beta}\left(\alpha_1\beta_j^0, \alpha_1\left(1 - \sum_{l=1}^{j}\beta_l^0\right)\right)$$

$$\pi_{tj} = \pi'_{tj}\prod_{l=1}^{j-1}(1-\pi'_{tl})$$

2.2 Markov Chain Monte Carlo approximate inference

2.2.1 Single level infinite mixture models

Markov Chain Monte Carlo (MCMC) algorithms are general tools for approximating posterior distributions of models. With these methods, one alternately samples from the distributions for subsets of variables conditioned on the remaining variables. Given some mild constraints on the model distributions, the approximation converges to the true posterior distribution in the large sample limit [26]. The utility of MCMC methods hinges on the ability to sample from a set of conditional distributions more efficiently than sampling from the full posterior.

In the case of infinite mixture models using a DP prior, sampling can be made efficient by exploiting a "trick" that requires tracking of only a finite number of non-empty mixture components and the data points already assigned to them. Figure 4 presents the overall MCMC sampling scheme for single level infinite mixture models.

Repeat for all data items $i = 1 \ldots N$:
  Sample z_i, the assignment of the data item to a mixture component, from its posterior, i.e., $\mathrm{p}(\mathsf{z}_i \mid \mathbf{z}_{-i}, \alpha, \boldsymbol{\theta})$
    If the data item has been assigned to a new component, sample a new mixture component parameter $\boldsymbol{\theta}_*$ from its posterior

Repeat for all non-empty mixture components $j = 1 \ldots J$:
  Sample the component parameter $\boldsymbol{\theta}_j$ from its posterior

Sample the DP concentration parameter α from its posterior

Figure 4: One iteration of the basic MCMC sampling scheme for an infinite mixture model using a Dirichlet Process prior.

The key MCMC sampling step for Dirichlet Processes involves picking assignments of data points to mixture components. We sample the assignment of a data point i conditioned on the other variables from the distribution given by:

$$\mathrm{p}(\mathsf{z}_i \mid \mathbf{z}_{-i}, \alpha, \boldsymbol{\theta}, \mathbf{x}) \propto \mathrm{p}(\mathsf{z}_i \mid \mathbf{z}_{-i}, \alpha)\mathrm{p}(\mathbf{x} \mid \mathbf{z}, \boldsymbol{\theta}) \tag{2}$$

The proportionality simply follows from Bayes' theorem. Recall from equation 1 that for finite mixture

models, we can write $\mathrm{p}(\mathsf{z}_i = j \mid \mathbf{z}_{-i}, \alpha, J)$ in closed form:

$$\mathrm{p}(\mathsf{z}_i = j \mid \mathbf{z}_{-i}, \alpha, J) \propto \frac{n_j^{-i} + \alpha/J}{N - 1 + \alpha}$$

For the case of infinite mixture models, and in which $n_j^{-i} > 0$ (i.e., the jth component of the mixture is non-empty), it can be proven that this distribution converges to (see [54]):

$$\mathrm{p}(\mathsf{z}_i = j \mid \mathbf{z}_{-i}, \alpha) \propto \frac{n_j^{-i}}{N - 1 + \alpha} \tag{3}$$

For infinite mixture models, we must consider the probability that a data point does not belong to one of the mixture components containing other data points. That is, we will need to calculate $\mathrm{p}(\mathsf{z}_i \neq \mathsf{z}_l, \forall\, l \neq i \mid \mathbf{z}_{-i}, \alpha)$. It can be proven that this probability is given by (see [54]):

$$\mathrm{p}(\mathsf{z}_i \neq \mathsf{z}_l, \forall\, l \neq i \mid \mathbf{z}_{-i}, \alpha) \propto \frac{\alpha}{N - 1 + \alpha} \tag{4}$$

We can thus combine equations 2, 3 and 4 to obtain the posterior distributions for the assignment variables:

$$\mathrm{p}(\mathsf{z}_i = j \mid \mathbf{z}_{-i}, \alpha, \boldsymbol{\theta}, \mathbf{x}) \propto \frac{n_j^{-i}}{N - 1 + \alpha} \mathrm{p}(\mathsf{x}_i \mid \boldsymbol{\theta}_j) \quad \text{for } n_j^{-i} > 0 \tag{5}$$

$$\mathrm{p}(\mathsf{z}_i \neq \mathsf{z}_l, \forall\, l \neq i \mid \mathbf{z}_{-i}, \alpha, \boldsymbol{\theta}, \mathbf{x}) \propto \frac{\alpha}{N - 1 + \alpha} \int F(\mathsf{x}_i \mid \boldsymbol{\psi}) H(\boldsymbol{\psi}) d\boldsymbol{\psi} \tag{6}$$

Thus, for each iteration, we sample the mixture component assignments for all data points using equations 5 and 6. For the first J components already containing data items, we use equation 5 to compute the assignment probability. We use equation 6 to compute the probability of assigning the data point to a new mixture component. Notice that in equation 6, we integrate over the mixture component parameters, as any component parameters are possible for a new component. Sampling is most efficient when $F(\cdot)$ and $H(\cdot)$ are conjugate. However, in cases of non-conjugacy of these distributions, Monte Carlo methods may be used [50, 54].

We also need to sample from the posterior for the concentration parameter α. It can be shown that the conditional distribution for α is given by (see [49]):

$$\mathrm{p}(\alpha \mid J, N, \boldsymbol{a}^\alpha) \propto \alpha^{a_1^\alpha + J - 1} e^{-a_2^\alpha \alpha} \mathrm{B}(\alpha, N)$$

Here, $\mathrm{B}(\cdot, \cdot)$ is the standard Beta function defined as:

$$\mathrm{B}(u, v) = \frac{\Gamma(u)\Gamma(v)}{\Gamma(u + v)} = \int_0^1 \eta^{u-1} (1 - \eta)^{v-1} d\eta$$

Escobar and West describe an efficient sampling scheme for α [20]. They noted that $\mathrm{p}(\alpha \mid J, N)$ can be written as a marginalization over an auxiliary variable η:

$$\mathrm{p}(\alpha \mid J, N, \boldsymbol{a}^\alpha) \propto \int_0^1 \mathrm{p}(\alpha, \eta \mid J, N, \boldsymbol{a}^\alpha) d\eta$$

$$\mathrm{p}(\alpha, \eta \mid J, N, \boldsymbol{a}^{\alpha}) \propto \alpha^{a_1^{\alpha}+J-1} e^{-a_2^{\alpha}\alpha} \eta^{\alpha-1} (1-\eta)^{N-1}$$

From the joint distribution, we can see that:

$$\mathrm{p}(\alpha \mid \eta, J, N, \boldsymbol{a}^{\alpha}) \propto \mathrm{Gamma}(\alpha \mid a_1^{\alpha} + J - 1, a_2^{\alpha} - \ln \eta)$$

$$\mathrm{p}(\eta \mid \alpha, J, N) \propto \mathrm{Beta}(\eta \mid \alpha, N)$$

Thus, by sampling from the above two conditional distributions, we can sample from the posterior for α to update the concentration parameter during the MCMC sampling iterations.

2.2.2 Hierarchical Dirichlet Process models

Teh *et al.* described an MCMC method for HDP infinite mixture models that uses auxiliary variables to make sampling from the conditional distributions efficient [70]. Figure 5 provides an overview of the sampling scheme.

Repeat for all data subsets $t = 1 \ldots T$ and data items $i = 1 \ldots N$:
  Sample z_{ti}, the assignment of data item i from subset t to a mixture component, from its posterior, i.e., $\mathrm{p}(\mathsf{z}_{ti} \mid \mathbf{z}_{-i}, \boldsymbol{\beta}^0, \boldsymbol{\theta}, \mathbf{x}, \alpha_1)$
    If the data item has been assigned to a new component, sample a new top-level mixture weight β_*^0 from the stick-breaking distribution and a new mixture component parameter $\boldsymbol{\theta}_*$ from its posterior

Repeat for all non-empty mixture components $j = 1 \ldots J$:
  Sample the component parameter $\boldsymbol{\theta}_j$ from its posterior

Sample the top-level mixture weights $\boldsymbol{\beta}^0$ from their posterior

Sample the concentration parameters α_0 and α_1 from their posteriors

Figure 5: One iteration of the basic MCMC sampling scheme for the Hierarchical Dirichlet Process mixture model with two levels.

The first task is to sample the data point assignment variables, $\mathbf{z}$. The method for this is similar to that used for ordinary Dirichlet Process mixture models. We begin by considering a finite mixture model of dimension J and integrating out the individual mixture weights $\boldsymbol{\pi}_t$ to obtain the conditional probability of $\mathbf{z}$ given $\boldsymbol{\beta}^0$:

$$\mathrm{p}(\mathbf{z} \mid \boldsymbol{\beta}^0, \alpha_1) = \prod_{t=1}^{T} \frac{\Gamma(\alpha_1)}{\Gamma(\alpha_1 + N_t)} \prod_{j=1}^{J} \frac{\Gamma(\alpha_1 \beta_j^0 + n_{tj})}{\Gamma(\alpha_1 \beta_j^0)} \tag{7}$$

Here, N_t denotes the number of data items in subset t, and n_{tj} represents the number of data items from subset t assigned to mixture component j. It can be shown that in the limit of an infinite mixture model, the conditional probability has a particularly simple form:

$$\mathrm{p}(\mathsf{z}_{ti} = j \mid \mathbf{z}_{-i}, \boldsymbol{\beta}^0, \alpha_1) \propto \alpha_1 \beta_j^0 + n_{tj}^{-i}$$

By combining this with the conditional likelihood for data points, $F(\cdot \mid \cdot)$, we obtain the posterior distribution for assigning data points to mixture components:

$$\mathrm{p}(\mathsf{z}_{ti} = j \mid \mathbf{z}_{-i}, \boldsymbol{\beta}^0, \boldsymbol{\theta}, \mathbf{x}, \alpha_1) \propto (\alpha_1 \beta_j^0 + n_{tj}^{-i}) F(\mathsf{x}_{ti} \mid \boldsymbol{\theta}_j) \tag{8}$$

This equation holds if j is a non-empty component. The posterior distribution for assigning a data point to a new component is given by:

$$\mathrm{p}(\mathsf{z}_{ti} \neq \mathsf{z}_{tl} \; \forall \; t, l \neq i \mid \mathbf{z}_{-i}, \boldsymbol{\beta}^0, \boldsymbol{\theta}, \mathbf{x}, \alpha_1) \propto (\alpha_1 \beta_*^0) \int F(\mathsf{x}_{ti} \mid \boldsymbol{\psi}) H(\boldsymbol{\psi}) d\boldsymbol{\psi} \tag{9}$$

Here, we define $\beta_*^0 = 1 - \sum_{l=1}^{J} \beta_l^0$, where there are J components with data points assigned to them. As with ordinary DPs, Monte Carlo methods may be used if $F(\cdot \mid \cdot)$ and $H(\cdot)$ are non-conjugate distributions.

So, to sample the data point assignments we use equations 8 and 9. If a data point is assigned to a new component, we must also generate a new weight β_{J+1}^0 using the stick-breaking distribution, i.e., we sample $b \sim \mathrm{Beta}(1, \alpha_0)$ and set $\beta_{J+1}^0 \leftarrow b\beta_*^0$.

To sample from the model posterior, we also must sample the top-level weights $\boldsymbol{\beta}^0$. The method for this relies on a "trick" using auxiliary variables. For the derivation, we need to use a general property of ratios of Gamma functions given by:

$$\frac{\Gamma(n+a)}{\Gamma(a)} = \sum_{m=0}^{n} s(n,m) a^m \tag{10}$$

Here, n and a are natural numbers. In equation 10, the ratio of Gamma functions has been expanded into a polynomial with a coefficient $s(n,m)$ for each term. These coefficients are called unsigned Stirling numbers of the first kind, which count the permutations of n objects having m permutation cycles (see [1]). By definition, $s(0,0) = 1$, $s(n,0) = 0$, $s(n,n) = 1$ and $s(n,m) = 0$ for $m > n$. Additional coefficients are then computed recursively using the equation $s(n+1,m) = s(n,m-1) + ns(n,m)$.

Note that the $\boldsymbol{\beta}^0$ weights in the conditional probability $\mathrm{p}(\mathbf{z} \mid \boldsymbol{\beta}^0)$ in equation 7 occur as arguments of ratios of Gamma functions. These ratios can be expanded to yield polynomials in the $\boldsymbol{\beta}^0$ weights:

$$\frac{\Gamma(\alpha_1 \beta_j^0 + n_{tj})}{\Gamma(\alpha_1 \beta_j^0)} = \sum_{m_{tj}=0}^{n_{tj}} s(n_{tj}, m_{tj}) (\alpha_1 \beta_j^0)^{m_{tj}} \tag{11}$$

An efficient sampling method can be derived by introducing $\mathbf{m}$ as auxiliary variables. The conditional distributions for sampling $\mathbf{m}$ and $\boldsymbol{\beta}^0$ can be shown to be:

$$\mathrm{p}(\mathsf{m}_{tj} = m \mid \mathbf{z}, \mathbf{m}_{-tj}, \boldsymbol{\beta}^0) \propto s(n_{tj}, m) (\alpha_1 \beta_j^0)^m \tag{12}$$

$$\mathrm{p}(\boldsymbol{\beta}^0 \mid \mathbf{z}, \mathbf{m}) \propto (\beta_*^0)^{\alpha_0 - 1} \prod_{j=1}^{J} \beta_j^{\sum_t m_{tj} - 1} \propto \mathrm{Dirichlet}(\sum_t m_{t1}, \ldots, \sum_t m_{tJ}, \alpha_0) \tag{13}$$

Finally, we need to sample the concentration parameters α_0 and α_1 for the HDP. As with the regular DP model, we will assume Gamma priors on the concentration parameters.

For α_0, it can be shown that:

$$\mathrm{p}(\mathsf{J} = J \mid \alpha_0, \mathbf{m}) \propto s(M, J) \alpha_0^J \frac{\Gamma(\alpha_0)}{\Gamma(\alpha_0 + M)}$$

Here, $M = \sum_t \sum_j m_{tj}$ and J is the number of non-empty mixture components. Combining the above equation with the prior for α_0 yields the conditional probability for α_0, which can be sampled using the same method as described for sampling concentration parameters for regular DPs.

Sampling α_1 requires the introduction of two additional auxiliary variables $\mathbf{w}$ and $\mathbf{b}$. The following update equations can then be derived:

$$\mathrm{p}(\mathsf{w}_t \mid \alpha_1) \propto w_t^{\alpha_1}(1-w_t)^{N_t-1}$$

$$\mathrm{p}(\mathsf{b}_t \mid \alpha_1) \propto \left(\frac{N_t}{\alpha_1}\right)^{b_t}$$

$$\mathrm{p}(\alpha_1 \mid \mathbf{w}, \mathbf{b}, \boldsymbol{a}^{\alpha_1}) \propto \mathrm{Gamma}(a_1^{\alpha_1} + \sum_{t=1}^{T}(M_t - b_t), a_2^{\alpha_1} - \sum_{t=1}^{T} \log w_t)$$

Here, $a_1^{\alpha_1}$ and $a_2^{\alpha_1}$ are the hyperparameters for the Gamma prior on α_1 and $M_t = \sum_{j=1}^{J} m_{tj}$.

3 The GeneProgram algorithm and probability model

3.1 Algorithm overview

The GeneProgram algorithm consists of data integration (pre-processing), model inference, and distribution summary steps as depicted in Figure 6. Data integration makes data from multiple species comparable and discretizes it in preparation for input to the model. The first data integration step combines replicates and normalizes microarray data to make measurements of gene expression comparable across tissues. The second data integration step uses a pre-defined homology map to convert species specific gene identifiers into meta-gene identifiers. Meta-genes are virtual genes that correspond one-to-one with genes in each species. Some genes do not have counterparts in other species, and these are filtered out. In the final data integration step, continuous expression measurements are discretized. The model inference step seeks to discover underlying expression programs and tissue groups in the data probabilistically. To accomplish this, we use Markov Chain Monte Carlo (MCMC) sampling to estimate the model posterior probability distribution. Each posterior sample describes a configuration of expression programs and tissue groups for the entire data set; more probable configurations tend to occur in more samples. The final step of the algorithm is model summarization, which produces consensus descriptions of expression programs and tissue groups from the posterior samples.

3.2 The probability model

3.2.1 Intuitive overview

We can understand the GeneProgram probability model intuitively as a series of "recipes" for constructing the gene expression of tissues. Figure 7 presents a cartoon of this process, in which we imagine that we are generating the expression data for the digestive tract of a person. The digestive tract is composed of a variety of cell types, with cells of a given type living in different microenvironments, and thus expressing somewhat different sets of genes. We can envision each cell in an organ choosing to express a subset of genes from relevant expression programs; some programs will be shared among many cell types and others will be more

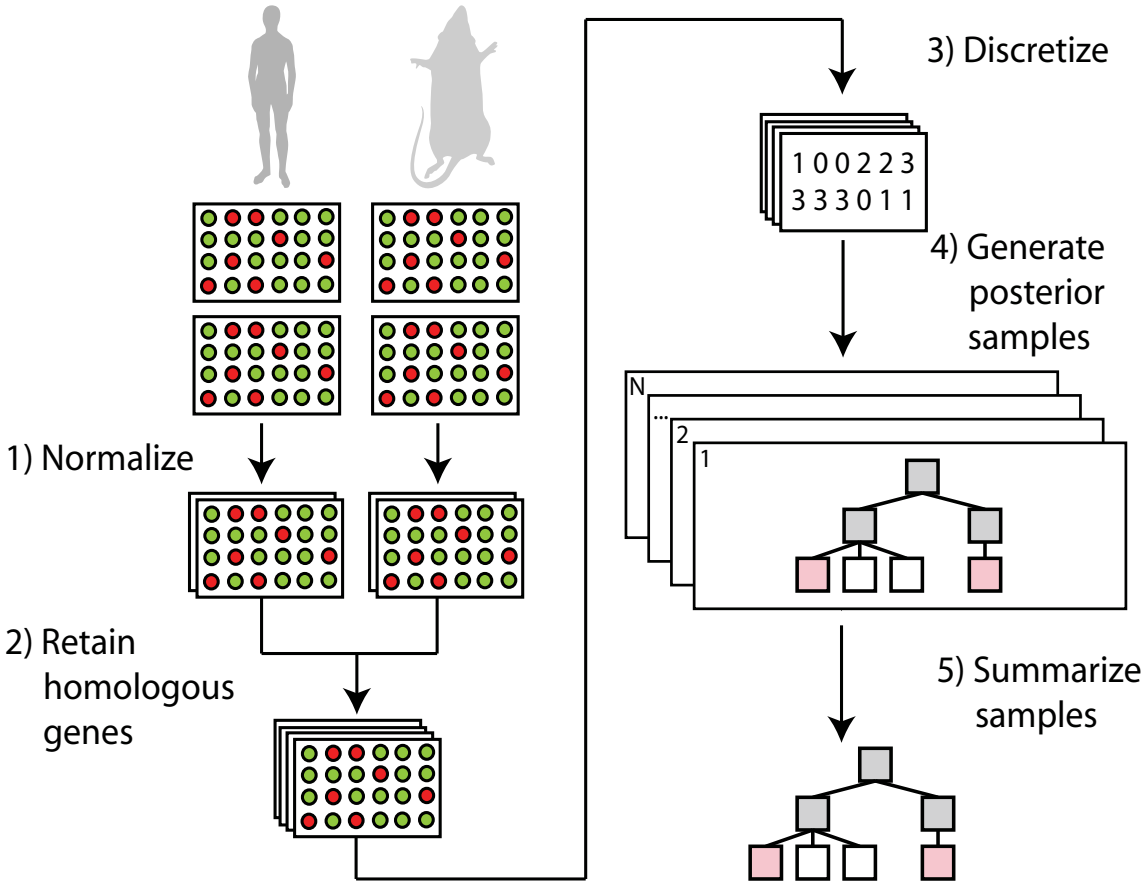


Figure 6: GeneProgram algorithm steps. The main steps of the algorithm are: data integration (steps 1-3), model inference (step 4), and posterior sample summarization (step 5). See the text for details.

specific. As we move along the digestive tract, the cell types present will change and different expression programs will become active. However, based on the similar physiological functions of the tissues of the digestive tract, we expect more extensive sharing of expression programs than we would between dissimilar organs such as the brain and kidneys. As can be seen in Figure 7, the final steps of our imaginary data generation experiment involve organ dissection, homogenization, cell lysis and nucleic acid extraction, to yield the total mRNA expressed in the tissue, which is then measured on a DNA microarray.

The conceptual experiment described above for "constructing" collections of mRNA molecules from tissues is analogous to the *topic model*, a probabilistic method developed for information retrieval applications [30, 10] and also applied to other domains, such as computer vision [66, 67] and haploinsufficiency profiling [23]. In topic models for information retrieval applications, documents are represented as unordered collections of words, and documents are decomposed into sets of related words called topics that may be shared across documents. In hierarchical versions of such models, documents are further organized into categories and topics are preferentially shared within the same category. In the GeneProgram model, a unit of mRNA detectable on a microarray is analogous to an individual word in the topic model. Related tissue populations (tissue groups) are analogous to document categories, tissues are analogous to documents, and topics are analogous to expression programs.

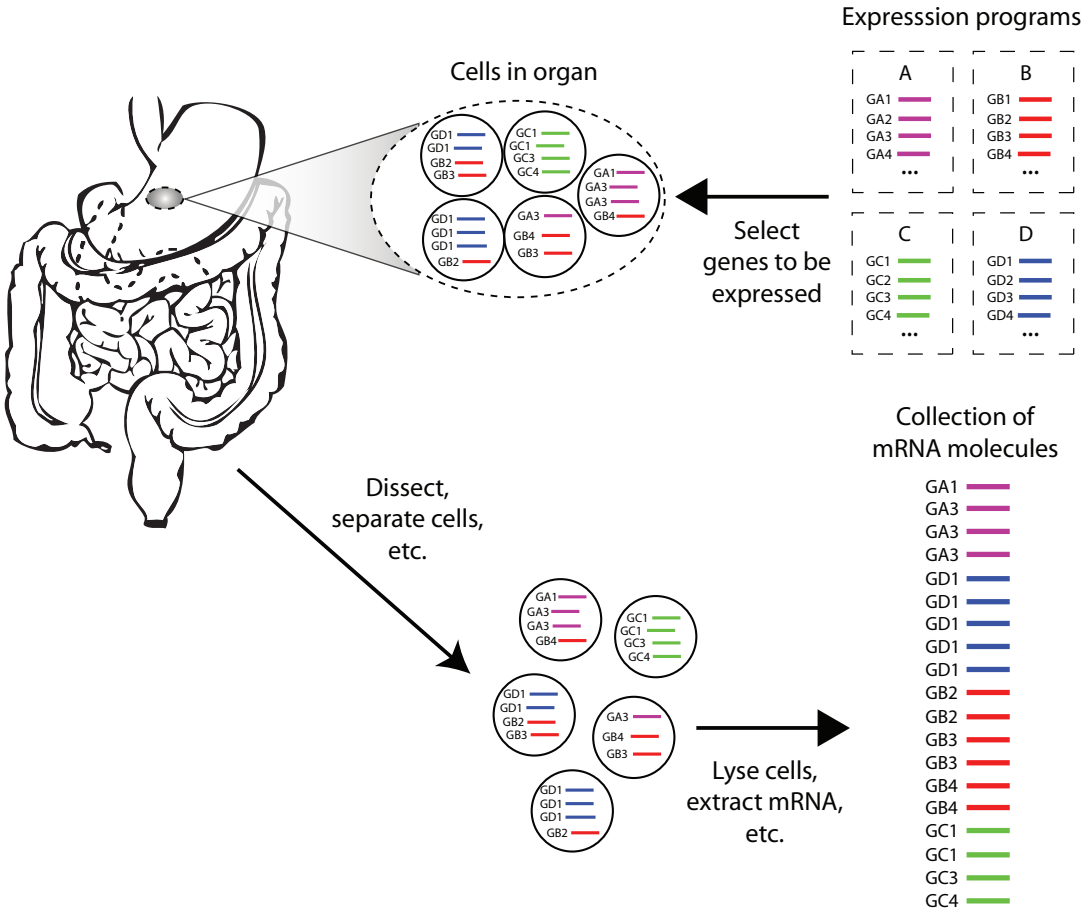


Figure 7: Conceptual overview of the data generation process for gene expression in mammalian tissues. The GeneProgram probability model can be thought of as a series of "recipes" for constructing the gene expression of tissues, as depicted in this cartoon example for a digestive tract. In the upper right, four expression programs (labeled A-D) are shown, consisting of sets of genes (e.g., GA1 represents gene 1 in program A). Cells (circles) throughout the digestive tract choose genes to be expressed probabilistically from the programs. The biological experimenter than collects mRNA by dissecting out the appropriate organ, taking a tissue sample, homogenizing it, lysing cells, and extracting the nucleic acids.

GeneProgram handles uncertainty in the numbers of expression programs and tissue groups by using a model based on Hierarchical Dirichlet Processes [70]. We note that in the original Hierarchical Dirichlet Processes formulation [70], data items were required to be manually assigned to groups. The GeneProgram model extends this work, automatically determining the number of groups and tissue memberships in the groups.

The GeneProgram probability model consists of a three-level hierarchy of Dirichlet Processes, as depicted in Figure 8 part A. Tissues are at the lowest level in the hierarchy. Each tissue is characterized by a mixture (weighted combination) of expression programs that is used to describe the observed gene expression levels in the tissue. An expression program represents a set of cross-species meta-genes that are co-activated to varying extents, as depicted in Figure 8 part B. When a tissue uses an expression program, the homology map translates meta-genes into the appropriate species specific genes. Tissues differ in terms of which expression programs they employ and how the programs are weighted. The middle level of the hierarchy consists of tissue groups, in which each group represents tissues that are similar in their use of expression programs. The highest and root level in the hierarchy describes a base level mixture of expression programs that is not tissue or group specific.

Each node in our hierarchical model maintains a mixture of gene expression programs, and the mixtures at the level below are constructed on the basis of those above. Thus, a tissue is decomposed into a collection of gene expression programs, which are potentially shared across the entire model, but are more likely to be shared by related tissues (those in the same tissue group). Because our model uses Dirichlet Processes, the numbers of both expression programs and tissue groups are not fixed and may vary with each sample from the model posterior distribution. In the next section, we describe the GeneProgram probability model in detail.

3.2.2 Formal model description

The GeneProgram model consists of three levels of DPs. Starting from the leaves these are: tissues, tissue groups, and the root. Each expression program corresponds to a mixture component of the HDP. Because the model is hierarchical, the expression programs are shared by all DPs in the model. An expression program specifies a multinomial distribution over meta-genes. Discrete expression levels are treated analogously to word occurrences in documents in topic models. Thus, a tissue's vector of gene expression levels is converted into a collection of expression events, in which the number of events for a given gene equals the expression level of that gene in the tissue. The model assumes that each gene expression event in a tissue is independently generated by an expression program. In the original HDP formulation [70], the entire tree structure was assumed to be pre-specified. We extend this work, by allowing the model to learn the number

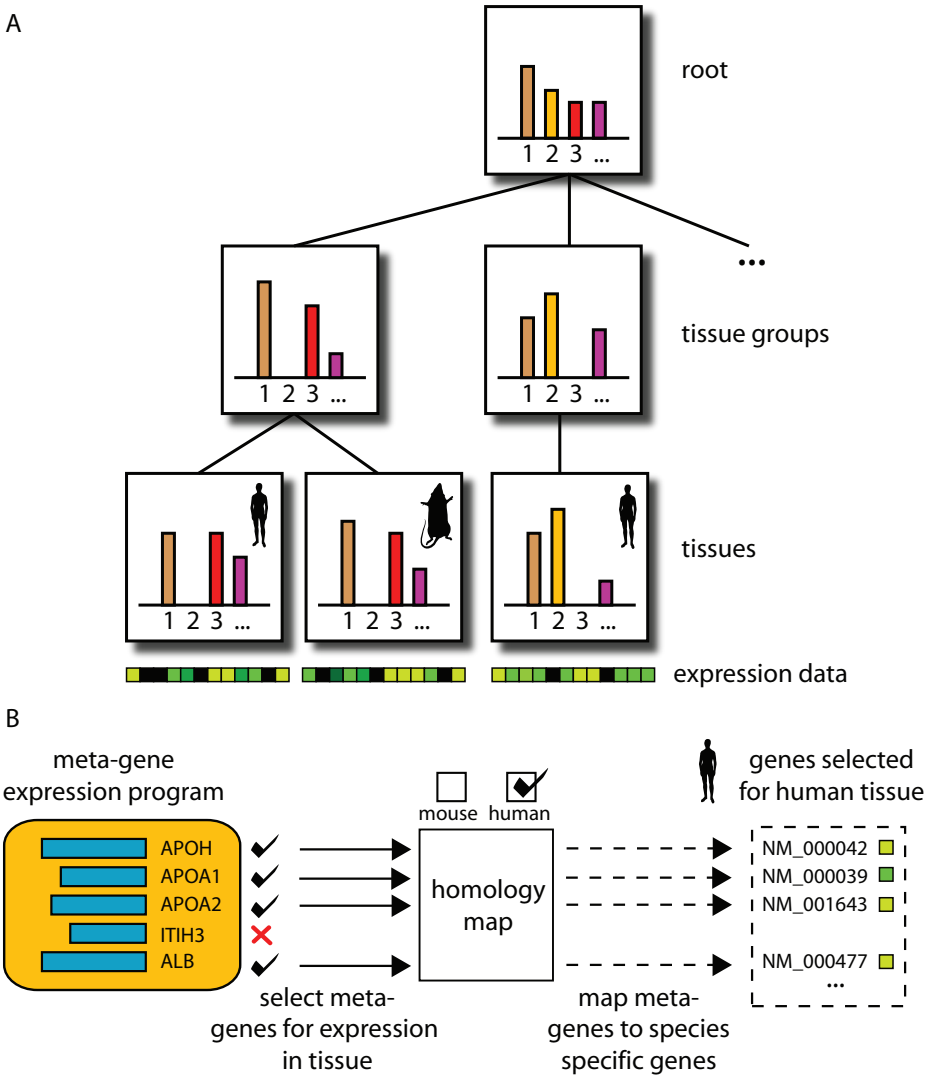

A
root
1 2 3 ...
tissue groups
tissues
expression data
B
meta-gene
expression program
APOH
APOA1
APOA2
ITIH3
ALB
mouse
human
homology
map
genes selected
for human tissue
NM_000042
NM_000039
NM_001643
NM_000477
select meta-
genes for expression
in tissue
map meta-
genes to species
specific genes

Figure 8: *(Part A)* Overview of the GeneProgram probability model. The model is based on Hierarchical Dirichlet Process mixture models, a non-parametric Bayesian method. The model consists of a three-level hierarchy of Dirichlet Processes. Each node describes a weighted mixture of expression programs (each colored bar represents a different program; heights of bars = mixture weights). The mixtures at each level are constructed on the basis of the parent mixtures above. Tissues are at the leaves of the hierarchy, and may be from either species. The observed gene expression of each tissue is characterized by a vector of discretized expression levels of species specific genes (row of small shaded squares below each tissue). *(Part B)* Example of a gene expression program. A gene expression program specifies a set of cross-species meta-genes that are co-activated to varying extents in a subset of tissues. On the left is a simple program containing five meta-genes (colored bars = expression frequencies). In this example, a human tissue uses the gene expression program, choosing four meta-genes from the set with various levels of expression. The homology map (center) translates the meta-genes into species specific genes (right).

of groups and the memberships of tissues in these groups. Thus, groups themselves are generated by a DP, which uses samples from the root process DP as its base distribution.

Figure 9 depicts the model using graphical model notation with plates and Table 1 summarizes the random variables in the model.

We will begin by describing the model at the level of observed data, and then move up the hierarchy. Assume that there are T tissues and G meta-genes. For simplicity, we will assume that there are also G genes for each species and that the ordering of genes uniquely determines the cross-species mapping. Thus, in the following discussion, genes and meta-genes are used interchangeably. The expression data associated with each tissue t consists of a G-dimensional vector $\mathbf{e}_t$ of discrete expression levels, i.e., $\mathsf{e}_{tg} \in \{0, 1, \ldots, E\}$ is the expression level of gene g in tissue t, where there are E possible discrete expression levels.

A tissue's vector of gene expression levels is converted into a collection of expression events, in which the number of events for a given gene equals the expression level of that gene in the tissue. This representation of expression levels as an unordered "bag of expression events" is analogous to the representation of words in a document as a "bag of words" in topic models. To be precise, let $\mathbf{x}_t$ denote a set of expression events for tissue t, and define a mapping ω from $\mathbf{x}_t$ to genes, where $\omega(\mathsf{x}_{ti}) = g$ iff $\mathsf{e}_{tg} > 0$. The vector $\mathbf{x}_t$ will have N_t elements, where $N_t = \sum_{g=1}^{G} e_{tg}$, i.e., as many elements as there are discrete expression events in the tissue.

The model assumes that each gene expression event in a tissue is independently generated by an expression program. The variable z_{ti} assigns gene expression events to programs, i.e., $\mathsf{z}_{ti} = j$ indicates that x_{ti} was generated from the jth expression program. An expression program is a multinomial probability distribution over genes. To be precise, let $\boldsymbol{\theta}_j$ represent a parameter vector of size G for expression program j. Then, the probability of generating expression event x_{ti} corresponding to gene g given that it is assigned to expression program j is $\mathrm{p}(\omega(\mathsf{x}_{ti}) = g \mid \mathsf{z}_{ti} = j, \boldsymbol{\theta}_j) = \theta_{jg}$. We use a symmetric Dirichlet prior for $\boldsymbol{\theta}_j$ with parameter λ, and a Gamma prior for λ with hyperparameter vector $\boldsymbol{a}^{\lambda}$.

The mixing probabilities over expression programs are generated by the DPs in the hierarchy. To be precise, let $\boldsymbol{\pi}_t$ denote the mixing probabilities at the leaf level in the DP hierarchy. That is, $\boldsymbol{\pi}_t$ denotes the mixing probabilities over expression programs for tissue t, i.e., $\mathrm{p}(\mathsf{z}_{ti} = j \mid \boldsymbol{\pi}_t) = \pi_{tj}$. Let $\boldsymbol{\beta}^k$ denote the mixing probabilities at the middle level in the DP hiearchy. That is, $\boldsymbol{\beta}^k$ denotes the mixing probabilities over

Var.	Dim.	Description	Cond. distribution or prior
x_{ti}	1	Expression event i in tissue t; corresponds directly to observed data.	Multinomial, given the assignment to expression program j.
z_{ti}	1	Assignment variable of an expression event to an expression program, i.e., $\mathsf{z}_{ti} = j$ indicates that expression event i in tissue t is assigned to expression program j.	Generated from mixing probabilities over expression programs for the tissue, i.e., $\mathrm{p}(\mathsf{z}_{ti} = j \mid \boldsymbol{\pi}_t) = \pi_{tj}$.
$\boldsymbol{\pi}_t$	∞	Mixing probabilities over expression programs for tissue t.	DP, given the assignment of the tissue to group k, its parent DP mixing probabilities, and its concentration parameter, i.e., $\boldsymbol{\pi}_t \mid \mathsf{q}_t = k, \alpha_1, \boldsymbol{\beta}^k \sim \mathrm{DP}(\alpha_1, \boldsymbol{\beta}^k)$.
$\boldsymbol{\beta}^k$	∞	Mixing probabilities over expression programs at the level of tissue group k; middle level in the DP hierarchy.	DP, given its parent mixing probabilities and concentration parameters, i.e., $\boldsymbol{\beta}^k \mid \alpha_0, \boldsymbol{\beta}^0 \sim \mathrm{DP}(\alpha_0, \boldsymbol{\beta}^0)$.
$\boldsymbol{\beta}^0$	∞	Root level mixing probabilities in the DP heterarchy.	DP, generated from the stick-breaking distribution given its concentration parameter, i.e., $\boldsymbol{\beta}^0 \mid \alpha_0 \sim \mathrm{Stick}(\alpha_0)$.
$\boldsymbol{\theta}_j$	G	Parameters for expression, program j, describing a multinomial distribution over G meta-genes.	Dirichlet distribution prior (parameterized by λ).
λ	1	Pseudo-count parameter for a symmetric Dirichlet distribution.	Gamma distribution prior with a two-dimensional hyperparameter vector $\boldsymbol{a}^\lambda$.
q_t	1	Assignment variable of tissues to groups, i.e., $\mathsf{q}_t = k$ indicates that tissue t belongs to tissue group k.	Generated from mixing probabilities over tissue groups, i.e, $\mathrm{p}(\mathsf{q}_t = k \mid \boldsymbol{\epsilon}) = \epsilon_k$.
$\boldsymbol{\epsilon}$	∞	Mixing probabilities over the tissue groups.	DP, generated from the stick-breaking prior given its concentration parameter, i.e, $\boldsymbol{\epsilon} \mid \gamma \sim \mathrm{Stick}(\gamma)$.
α_1	1	Concentration parameter for $\boldsymbol{\pi}_t$.	Gamma distribution prior with two-dimensional hyperparameter vector $\boldsymbol{a}^{\alpha_1}$.
α_0	1	Concentration parameter for $\boldsymbol{\beta}^0$ and $\boldsymbol{\beta}^k$.	Gamma distribution prior with two-dimensional hyperparameter vector $\boldsymbol{a}^{\alpha_0}$.
γ	1	Concentration parameter for $\boldsymbol{\epsilon}$.	Gamma distribution prior with two-dimensional hyperparameter vector $\boldsymbol{a}^\gamma$.

Table 1: Summary of random variables in the GeneProgram model. The columns are: variable name (vectors are in bold type), dimensions of the variable, description, and the conditional or prior distribution on the variable.

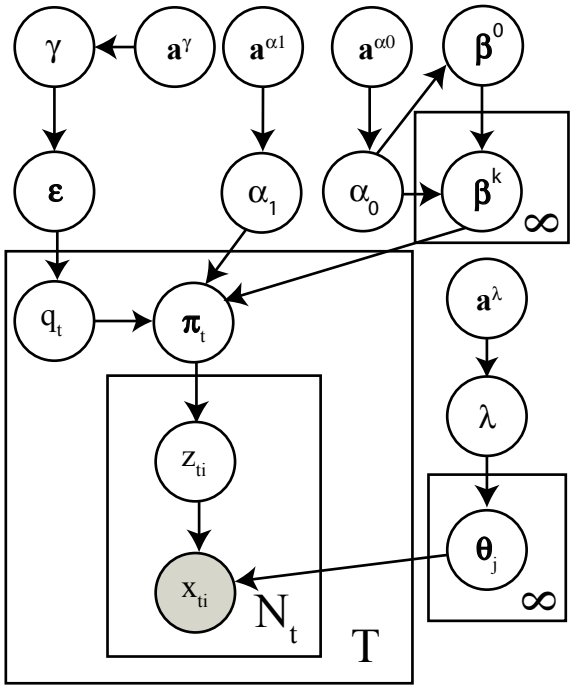


Figure 9: The GeneProgram model is depicted using graphical model notation with plates. Circles represent variables, and arrows denote dependencies among variables. Vectors are depicted with bold type, and observed variables are shown inside shaded circles. Rectangles represent plates, or repeated sub-structures in the model. See the text and Table 1 for details.

expression programs at the level of tissue group k. Finally, we let $\boldsymbol{\beta}^0$ denote the root-level mixing probabilities. In the stick-breaking construction for HDP models, it is assumed that root level mixing probabilities are generated by the stick-breaking distribution, i.e., $\boldsymbol{\beta}^0 \mid \alpha_0 \sim \text{Stick}(\alpha_0)$, where $\alpha_0 \sim \text{Gamma}(\boldsymbol{a}^{\alpha_0})$. The hierarchical structure of the model then implies that $\boldsymbol{\beta}^k$ is conditionally distributed as a Dirichlet Process, i.e., $\boldsymbol{\beta}^k \mid \alpha_0, \boldsymbol{\beta}^0 \sim \text{DP}(\alpha_0, \boldsymbol{\beta}^0)$, where we assume that $\boldsymbol{\beta}^k$ also uses concentration parameter α_0.

The tissue level expression program mixing probabilities $\boldsymbol{\pi}_t$ depend on the group that the tissue is assigned to. The variable q_t assigns tissues to groups, i.e., $\mathsf{q}_t = k$ indicates that tissue t belongs to tissue group k and $\text{p}(\mathsf{q}_t = k \mid \boldsymbol{\epsilon}) = \epsilon_k$, where $\boldsymbol{\epsilon}$ represents mixing probabilities over the tissue groups. The mixing probabilities $\boldsymbol{\epsilon}$ over tissue groups are also modeled using a Dirichlet Process. That is, $\boldsymbol{\epsilon} \mid \gamma \sim \text{Stick}(\gamma)$, where γ is a concentration parameter with $\gamma \sim \text{Gamma}(\boldsymbol{a}^{\gamma})$. Given an assignment of tissue t to group k, the tissue level mixing probabilities over expression programs $\boldsymbol{\pi}_t$ are then generated from the middle level mixing probabilities $\boldsymbol{\beta}^k$. That is, $\boldsymbol{\pi}_t \mid \mathsf{q}_t = k, \alpha_1, \boldsymbol{\beta}^k \sim \text{DP}(\alpha_1, \boldsymbol{\beta}^k)$, where α_1 is a concentration parameter with hyperparameters $\boldsymbol{a}^{\alpha_1}$, i.e., $\alpha_1 \sim \text{Gamma}(\boldsymbol{a}^{\alpha_1})$. This completes our formal description of the GeneProgram probability model.

3.3 Model inference

The posterior distribution for the model is approximated via Markov Chain Monte Carlo (MCMC) sampling using the follow steps:

1. Sample each assignment of an expression event to an expression program, z_{ti}; create new expression programs as necessary.
2. Sample $\boldsymbol{\beta}^0$ and $\boldsymbol{\beta}^k$ and auxiliary variables for all tissue groups.
3. Sample tissue group assignments q_t for all tissues; create new tissue groups as necessary.
4. Sample concentration parameters α_0, α_1, and γ.
5. Sample expression program Dirichlet prior parameter λ.

Steps 1, 2, and 4 are identical to those described by Teh *et al.* in their auxiliary variable sampling scheme [70] (see Section 2.2 for further details). Note that $\mathsf{x}_{ti} \mid \mathsf{z}_{ti} = j, \boldsymbol{\theta}_j \sim \text{Multinomial}(\boldsymbol{\theta}_j)$, and $\boldsymbol{\theta}_j$ is Dirichlet distributed, allowing us to integrate out $\boldsymbol{\theta}_j$ when computing the posterior for z_{ti}. This means that we do not need to represent $\boldsymbol{\theta}_j$ explicitly during sampling. In step 3, we must compute the posteriors for tissue group assignments. This can be written as:

$$\mathrm{p}(\mathsf{q}_t = k \mid \mathbf{z}_t, \mathbf{q}_{-t}, \alpha_0, \gamma, \boldsymbol{\beta}^k) \propto \mathrm{p}(\mathsf{q}_t = k \mid \mathbf{q}_{-t}, \gamma) \prod_{i=1}^{N_t} \int \mathrm{p}(\mathsf{z}_{ti} \mid \boldsymbol{\pi}_t) \mathrm{p}(\boldsymbol{\pi}_t \mid \boldsymbol{\beta}^k, \alpha_0) d\boldsymbol{\pi}_t$$

Here, $\mathbf{q}_{-t}$ denotes all tissue group assignments excluding tissue t. Note that because the conditional distributions for z_{ti} and $\boldsymbol{\pi}_t$ are conjugate, the integral in the above equation can be computed in closed form. Step 5 uses the auxiliary variable sampling method for resampling the parameter for a symmetric Dirichlet prior, as detailed in [20].

We implemented the sampling scheme in Java. Inference was always started with all data assigned to a single expression program. We burned in the sampler for 100,000 iterations, and then collected relevant posterior distribution statistics from 50,000 samples. We set the hyperparameters for all concentration parameters to 10^{-8} to produce vague prior distributions. Both hyperparameters for the Gamma prior on λ were set to 1.0, biasing λ toward a unit pseudo-count Dirichlet distribution.

3.4 Summarizing the model posterior probability distribution

3.4.1 Overview

In order to produce interpretable results, GeneProgram needs to create a summary of the model posterior distribution that was approximated using MCMC sampling.

The final step of the GeneProgram algorithm summarizes the approximated model posterior probability distribution with *consensus tissue groups* (CTGs) and *recurrent expression programs* (REPs). The posterior distributions of Dirichlet Process mixture models are particularly challenging to summarize because the number of mixture components may differ for each sample. Previous approaches for summarizing Dirichlet Process mixture model components have used pair-wise co-clustering probabilities as a similarity measure for input into an agglomerative clustering algorithm [46]. This method is feasible if there are a relatively

small number of items to be clustered, and we employ it for producing consensus tissue groups. However, this method is not feasible for summarizing expression programs in large data sets because of the number of pair-wise probabilities that would need to be calculated for each sample.

We developed a novel method for summarization of the model posterior distribution, which discovers recurrent expression programs by combining information from similar expression programs that reoccur across posterior samples. Our method is based on the observation that each expression program is significantly used by only a limited number of tissues. Thus, this limited set of tissues serves as a unique signature that allows us to track the expression program across model posterior samples. A recurrent expression program is summarized by the average frequency of expression of meta-genes across many model posterior samples.

3.4.2 Detailed description of recurrent expression programs and consensus tissue groups

CTGs are constructed by first computing the empirical probability that a pair of tissues will be assigned to the same tissue group. The empirical co-grouping probabilities are then used as pair-wise similarity measures in a standard bottom-up agglomerative hierarchical clustering algorithm using complete linkage (e.g., as discussed in [19]). To be precise, let S denote the total number of samples, and $q_t^{(l)}$ the tissue group assignment for tissue t in sample l. The empirical co-grouping probability for tissues t and r is then:

$$\widehat{p}_{tr} = \sum_{l=1}^{S} \mathrm{I}(q_t^{(l)} = q_r^{(l)})/S$$

Here, $I(\cdot)$ is the indicator function.

Clustering is stopped using a pre-defined cut-off c_{tg} to produce the final CTGs. We used a cut-off of c_{tg} = 0.90 to produce strongly coherent groups. However, we note that the empirical co-grouping probabilities tend to be either very small or close to 1.0, rendering our results relatively insensitive to the choice of c_{tg}.

REPs consist of sets of tissues and genes that appear together with significant probability in expression programs across multiple samples. For each expression program in each sample, a set of *index tissues* is determined based on the extent of overlap of genes in the program and those expressed by the tissue (significance is determined using the hypergeometric distribution). To be precise, let $J^{(s)}$ be the number of expression programs used in sample s. Let $\eta_{tj}^{(s)}$ denote the number of genes expressed in tissue t and assigned to expression program j in sample s, i.e., $\eta_{tj}^{(s)} = |\{\omega(x_{ti}) : z_{ti}^{(s)} = j\}|$. We use the hypergeometric distribution to compute a *p*-value, $v_{tj}^{(s)}$, for each tissue and expression program pair:

$$v_{tj}^{(s)} = 1 - \text{HyperCDF}(\eta_{tj}^{(s)} - 1, G, \sum_{l=1}^{J^{(s)}} \eta_{tl}^{(s)}, \sum_{l=1}^{T} \eta_{lj}^{(s)})$$

Here, HyperCDF denotes the cumulative distribution function for the hypergeometric distribution. We use the *p*-values, $v_{tj}^{(s)}$, to compute the index tissues $V_j^{(s)}$ for expression program j in sample s, i.e., $V_j^{(s)} = \{t : v_{tj}^{(s)} < c_1\}$, i.e., the set of all tissues whose *p*-values for expression program j are below a threshold c_1 in sample s. We used a *p*-value threshold c_1 of 5%.

A hash table using the index tissues enables the algorithm to efficiently determine whether an expression program has already occurred in previous samples. If it has not, a new REP is instantiated; otherwise the

expression program is merged into the appropriate REP. Statistics are tracked for each REP, including the number of samples it occurs in, its average weighting in the tissue's mixture over programs, and average expression levels of species specific genes and meta-genes in the program. To be precise, let S_j denote the number of samples in which REP j occurs. Then, the empirical mean expression level for gene g in REP j is defined as:

$$\widehat{e}_{gj} = \frac{\sum_{s=1}^{S} \sum_{t,i} \mathrm{I}(z_{ti}^{(s)} = j)}{|V_j| S_j} \text{ s.t. } t \in V_j^{(s)}, \omega(x_{ti}) = g$$

The empirical mean gene occurrence for gene g in recurrent expression program j is defined as:

$$\widehat{o}_{gj} = \frac{\sum_{s=1}^{S} \sum_{t} \mathrm{I}\left(\sum_{i} \mathrm{I}(z_{ti}^{(s)} = j) > 0\right)}{|V_j| S_j} \text{ s.t. } t \in V_j^{(s)}, \omega(x_{ti}) = g$$

The empirical mean tissue weighting for tissue t in recurrent expression program j is defined as:

$$\widehat{w}_{tj} = \frac{\sum_{s=1}^{S} \eta_{tj}^{(s)}}{N_t S}$$

After all samples have been collected, several post-processing steps are then performed, including filtering out infrequently occurring REPs and genes, and merging of similar REPs. We filtered out REPs that occurred in fewer than 50% of samples, and filtered out genes with $\widehat{o}_{gj}$ scores less than 5%. The final merging step uses the same agglomerative procedure described for CTGs. In this case, the similarity measure is the fraction of genes shared by REPs. Only common index tissues are retained in merging two REPs. Merging is stopped when the similarity measure is less than a cut-off of 50%.

3.5 Expression data discretization

Expression data input into GeneProgram was first discretized using a mutual information-based greedy agglomerative merging algorithm, essentially as described in Hartemink *et al.* [32]. In brief, continuous expression levels are first uniformly discretized into a large number of levels. The algorithm then repeatedly finds the best two adjacent levels to merge by minimizing the reduction in the pair-wise mutual information between all expression vectors. The appropriate number of levels to stop at is determined by choosing the inflection point on the curve obtained by plotting the score against the number of levels. In this case, we obtained three levels.

For completeness, we describe the discretization algorithm here. We begin by initializing the algorithm with sets of expression levels for each tissue. We denote gene i in tissue t by g_{ti}, where there are T tissues. Let $r(g_{ti})$ denote the rank of gene i in tissue t based on the continuous expression value of the gene. To initialize the algorithm, we begin by assigning genes in each tissue t to an ordered set $\Lambda_t^{(0)}$ of N_L discrete expression levels that induce uniform bins on the gene rankings for the tissue. That is, $\Lambda_t^{(0)} = (L_{t1}^{(0)}, \ldots, L_{tN_L}^{(0)})$, where $g_{ti} \in L_{tl}^{(0)}$ iff $l - 1 < r(g_{ti}) N_L / G_t \leq l$. Here, G_t is the number of genes in tissue t that are considered expressed (e.g., expression values greater than some threshold).

Each iteration consists of a set of trial merges, in which adjacent levels are merged and a score is computed. For iteration q and for each trial h, the adjacent levels h and $h+1$ are merged, forming a new set of levels with one less element, i.e., $(L_{t1}^{(q-1)}, \ldots, L_{th}^{(q-1)} \bigcup L_{t(h+1)}^{(q-1)}, L_{t(h+2)}^{(q-1)}, \ldots, L_{t(N_L-q)}^{(q-1)})$. Let $\boldsymbol{e}_t^{(qh)}$

denote the discrete vector of expression levels for tissue t for iteration q of the algorithm and trial merge h. That is, $e_{ti}^{(qh)} = l$ iff g_{ti} is in level l for trial merge h and g_{ti} is expressed in the tissue (otherwise, we set $e_{ti}^{(qh)} = 0$). The score for a trial merge h is the mutual information between all pairs of vectors of discretized expression data, i.e., $S_h^q = \sum_{t_1=1}^{T-1} \sum_{t_2>t_1} \mathrm{MI}(\boldsymbol{e}_{t_1}^{(qh)}, \boldsymbol{e}_{t_2}^{(qh)})$. At each iteration, the single merge operation that produces the highest score is retained. Note that because the algorithm is greedy, its run-time is $O(N_L^2 T^2)$.

4 GeneProgram discovered biologically relevant tissue groups and expression programs in a large compendium of human and mouse body-wide gene expression data

Our objective was to apply GeneProgram to a large compendium of mammalian gene expression data, both to compare our method's performance against that of other algorithms, as well as to explore the biological relevance of discovered tissue groups and expression programs. In this regard, we used the Novartis Gene Atlas v2 [65], consisting of genome-wide expression measurements for 79 human and 61 mouse tissues. This dataset was chosen because it contains a large set of relatively high-quality expression experiments, with body-wide samples representative of normal tissues measured on similar microarray platforms. Further, the data is from two species, potentially allowing for the discovery of higher quality cross-species gene expression programs.

4.1 Data set pre-processing

All arrays in the data set were first processed using the GC content-adjusted robust multi-array algorithm (GC-RMA) [73]. To correct for probe specific intensity differences, the intensity of each probe was normalized by dividing by its geometric mean in the 31 matched tissues. For genes represented by more than one probe, we used the maximum of the normalized intensities. A gene was considered expressed if its normalized level was greater than 2.0 and was called present in one or more replicates of the MAS5 Absent/Present calls [33].

We identified pairs of related genes using Homologene (build 47) [72], which attempts to find homologous gene sets among the completely sequenced eukaryotic genomes by using a taxonomic tree, conserved gene order, and measures of sequence similarity. Of the approximately 16,000 homologous human-mouse pairs identified by Homologene, 9851 gene pairs appear in the Gene Atlas v2.

4.2 GeneProgram discovered 19 consensus tissue groups and 100 recurrent expression programs

Figure 10 depicts all 19 tissue groups. Supplemental online Table 1 [27] provides a summary and supplemental online Table 2 [28] contains the full data for all 100 expression programs. The tissue groups were of various sizes, ranging from 1–38 tissues (median of 4). Expression program sizes ranged from 12–292 meta-genes (median of 72) and 1–38 tissues (median of 4). A large fraction (67%) of meta-genes appeared in at least one expression program and 31% were shared by several expression programs. Forty-two percent of tissue groups and 33% of expression programs contained at least one tissue from each species. It is

important to realize that the number of cross-species tissue groups and expression programs was limited by the data set: only 62 out of the 140 tissue samples could be directly paired between species and some key tissues with distinct functions, such as the stomach and eye, were represented in only one species.

4.3 GeneProgram automatically assigned tissues to biologically relevant groups

To provide a quantitative assessment of the biological relevance of sets of tissues, we manually classified tissues into 10 high-level, physiologically based categories and then calculated an enrichment score for each discovered tissue group using the hypergeometric distribution. See the supplemental online material for [29] for the complete manually derived tissue categories. To correct for multiple hypothesis tests, we used the procedure of Benjamini and Hochberg [7] with a false-discovery rate cut-off of 0.05.

Seventy-nine percent of tissue groups had significant enrichment scores, and in all such cases, the score was significant for only a single category (see Figure 10). For instance, tissue group "L," which was significantly enriched only for the "hematological/immune" category, consisted exclusively of human immune cells such as natural killer cells, and CD4+ and CD8+ T-cells. As another example, tissue group "B," significantly enriched only for the "neural" category, consisted exclusively of neural tissues from both species. We note that GeneProgram discovered these groups in a wholly unsupervised manner, and that many of the groups clearly represent a more refined picture of the data than the 10 broad categories we had manually compiled.

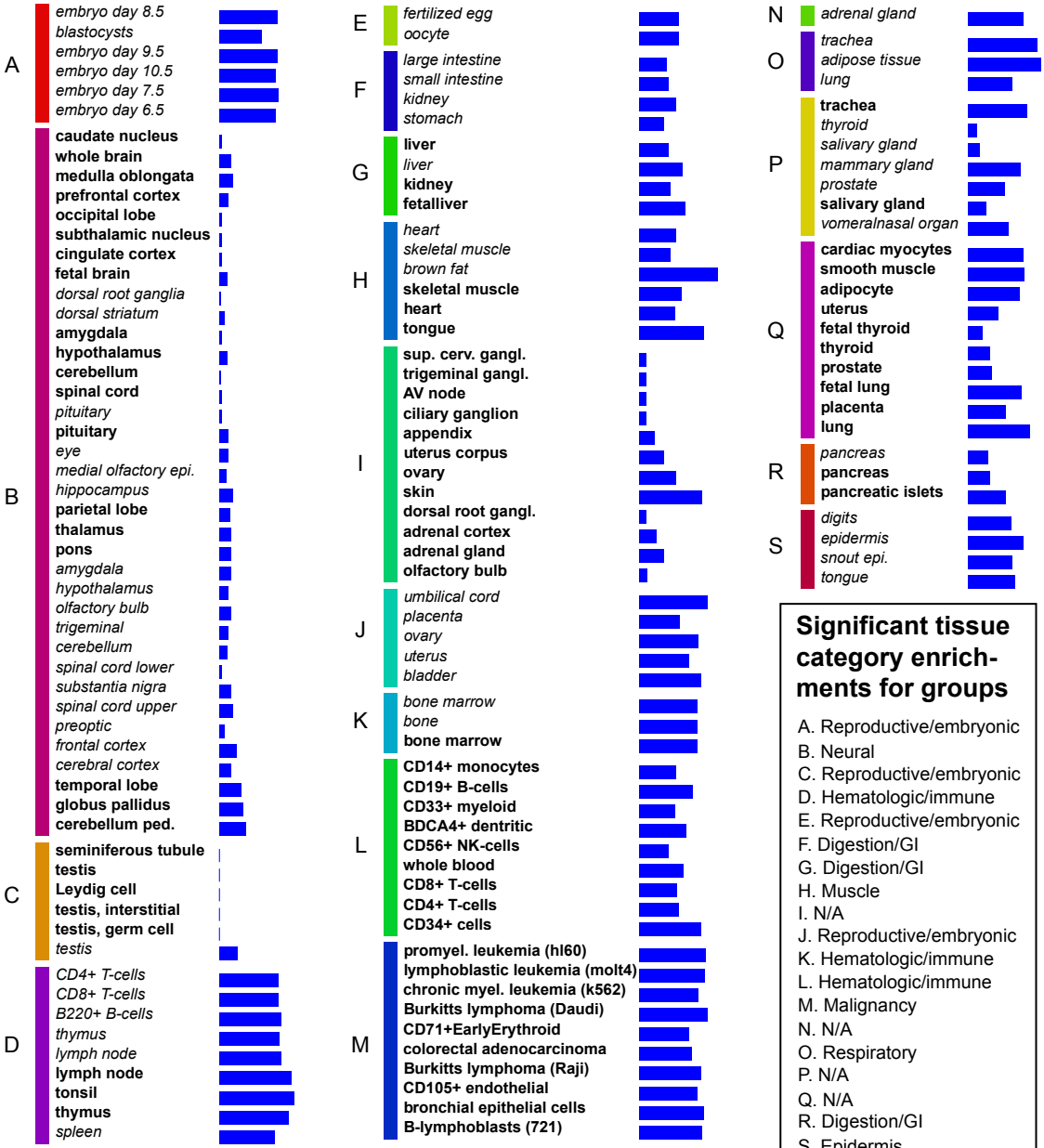
4.4 GeneProgram outperformed biclustering algorithms in the discovery of biologically relevant gene sets

Because expression programs characterize both genes and tissues, we used both Gene Ontology (GO) categories [4] and the 10 manually derived tissue categories to assess GeneProgram's ability to recover biologically relevant gene sets and to compare this performance to that of two biclustering algorithms, Samba [68, 61] and a non-negative matrix factorization (NMF) implementation [12]. We chose these two algorithms for comparison because they are popular in the gene expression analysis community, they have previously outperformed other biclustering algorithms, and available implementations are capable of handling large data sets.

We mapped genes to GO annotations using RefSeq identifiers from the May 2004 (hg17) and August 2005 (mm7) assemblies of human and mouse genomes [4, 72]. For calculating enrichments, we used both mouse and human GO annotations from the biological process categories with between 5 and 200 genes. Enrichment score calculation and correction for multiple hypothesis tests were the same as described in Section 4.3.

As Table 2 shows, GeneProgram clearly outperformed the other two algorithms in the tissue dimension (60% of expression programs significantly enriched for tissue categories, versus 10% for Samba and 20% for NMF). GeneProgram outperformed NMF and had equivalent performance to Samba in the gene dimension (61% of expression programs significantly enriched for GO categories, versus 62% for Samba and 27% for NMF).

Figure 11 shows the same trends using correspondence plots, which are sensitive, graphical methods for comparing biclustering algorithms [69]. These plots depict log p-values on the horizontal axis and the fraction of biclusters with p-values below a given value on the vertical axis. Depicted p-values are from the most abundant class for each bicluster (i.e., that with the largest number of genes or tissue in the



Significant tissue category enrich-ments for groups

A. Reproductive/embryonic
B. Neural
C. Reproductive/embryonic
D. Hematologic/immune
E. Reproductive/embryonic
F. Digestion/GI
G. Digestion/GI
H. Muscle
I. N/A
J. Reproductive/embryonic
K. Hematologic/immune
L. Hematologic/immune
M. Malignancy
N. N/A
O. Respiratory
P. N/A
Q. N/A
R. Digestion/GI
S. Epidermis

Figure 10: GeneProgram discovered 19 consensus tissue groups using gene expression data from 140 human and mouse tissue samples. The algorithm identified these groups in a wholly unsupervised manner. In each tissue group (denoted A-S), human tissues are designated with bold type and mouse tissues with italic type. Tissues were classified manually into 10 broad categories based on physiological function, and it was found that 79% of tissue groups were significantly enriched for at least one category (boxed legend, lower right corner). To the right of each tissue is a blue vertical bar depicting its weighted average of generality scores for expression programs, which provides a measure of the extent to which the tissue uses programs shared by diverse tissue types (see the text for details).

algorithm	**gene dimension** (GO category enrichment)	**tissue dimension** (manually derived category enrichment)
GeneProgram	61%	60%
Samba	62%	10%
NMF	27%	20%

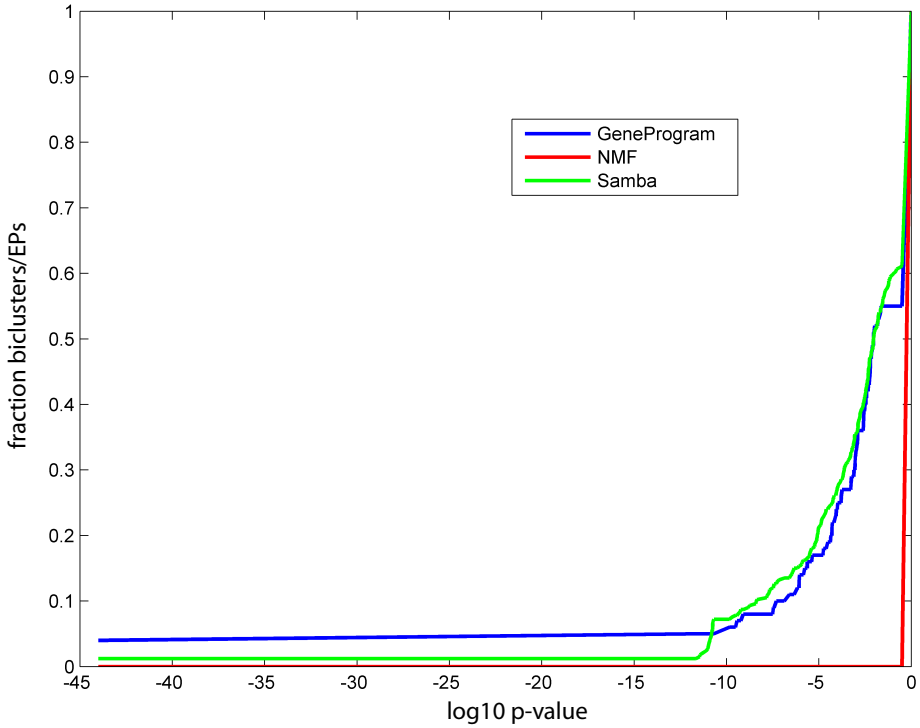
Table 2: Comparison of GeneProgram to biclustering algorithms for recovery of biologically interpretable gene sets. GeneProgram's ability to recover biologically interpretable gene sets from a large compendium of mammalian tissue gene expression data was compared against that of two popular biclustering algorithms, Samba and a non-negative matrix factorization (NMF) implementation. GeneProgram dominated the other two algorithms in the tissue dimension; it outperformed NMF and had equivalent performance to Samba in the gene dimension. Biological interpretability of gene sets was assessed using Gene Ontology (GO) categories in the gene dimension, and manually constructed categories in the tissue dimension. Each cell in the table shows the percentage of sets significantly enriched for at least one category in a given dimension (p-value < 0.05, corrected for multiple hypothesis tests).

overlap) and calculated using the hypergeometric distribution. Note that biclusters with large p-values are not significantly enriched for any class, and may represent noise.

These results suggest several performance trends related to features of the different algorithms. Samba generally appeared to be successful at finding relatively small sets of genes that are co-expressed in subsets of tissues, but had difficulty uncovering larger structures in data. Presumably, our algorithm's clear dominance of both Samba and NMF in the tissue dimension was partly attributable to GeneProgram's hierarchical model. Both of the other algorithms lack such a model, so the assignment of tissues to biclusters was not guided by global relationships among tissues.

We note also that the algorithms differed substantially in runtimes: Samba was fastest (approximately 3 hours), GeneProgram the next fastest (approximately 3 days), and NMF the slowest (approximately 6 days), with all software running on a 3.2 GHz Intel Xenon CPU. Although these runtime differences may be attributable in part to implementation details, it is worth noting that GeneProgram, a fully Bayesian model using MCMC sampling for inference, ran faster than the NMF algorithm, which uses a more "traditional" objective maximization algorithm to search for the appropriate number of biclusters.

A. Gene dimension



B. Tissue dimension

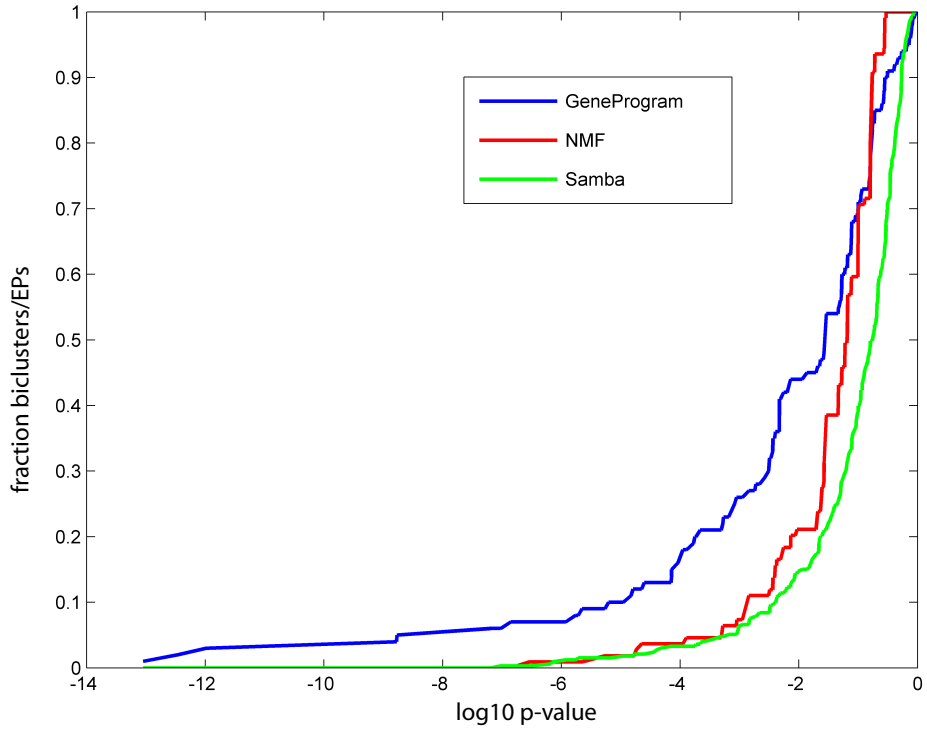


Figure 11: Correspondence plots comparing GeneProgram to biclustering algorithms. These plots compare GeneProgram's ability to recover biologically interpretable gene sets from a large compendium of mammalian tissue gene expression data against that of two popular biclustering algorithms, Samba and a non-negative matrix factorization (NMF) implementation. GeneProgram clearly dominated the other two algorithms in the tissue dimension; it outperformed NMF and had equivalent performance to Samba in the gene dimension. Biological interpretability of gene sets was assessed using Gene Ontology (GO) categories in the gene dimension, and manually derived high-level, physiologically based categories in the tissue dimension. The plots depict *p*-values (enrichment scores) on the horizontal axis and the fraction of biclusters with *p*-values below a given value on the vertical axis (the *p*-value for the most abundant class was used).

4.5 GeneProgram cross-species expression programs outperformed single-species programs in terms of biological relevance in both the gene and tissue dimensions

Seventy-nine percent of cross-species programs were significantly enriched for GO categories versus 52% of single-species programs, and 82% of cross-species programs were significantly enriched for the manually derived tissue categories versus 51% of single-species programs. These results suggest that combining data from both species was valuable for discovery of biologically relevant expression programs. However, this conclusion must be interpreted cautiously for the gene dimension, because GO annotations may be biased toward extensively studied genes that are expressed in both species.

It is also relevant to ask whether single species expression programs represent biologically important differences in gene expression between mice and humans. Unfortunately, substantial differences in how samples from the two species were obtained, prepared and experimentally analyzed were confounding factors. Nonetheless, some single-species expression programs appeared to reflect real biological differences between mice and humans. For instance, expression program 78 contained only mouse tissues, including general and snout epidermis. Interestingly, the program contained many keratin genes, which are components of hair fibers, and the Cochlin gene, which has been detected in spindle-shaped cells located along nerve fibers that innervate hair cells [55]; such structures are considerably more abundant in fur-covered mouse skin than in human skin.

4.6 Automatic inference of tissue groups resulted in significant improvements in model performance

We used cross-validation to analyze the importance of automatic tissue group inference in our model. We tested the full GeneProgram model versus a simplified version in which there were no groups and all tissues are attached directly to the root of the hierarchy.

We used 10-fold cross-validation on the 140 tissues; the order of the tissues was first randomly permuted so that there would be no bias toward selecting training sets from only a single species.

The *perplexity* was then calculated for each held-out tissue; perplexity is a measure commonly used for evaluating statistical language and information retrieval models [56]. In this context, it is inversely related to the predicted model likelihood of the expression data in the held-out tissue given the training data. Thus, smaller perplexity values indicate a better model fit.

The model was burned in with 100,000 iterations as described in Section 3.3. After burn-in, the model posterior was sampled ten times (we allowed 100 iterations between samples to reduce dependencies). For each of the ten samples, the held-out tissue t was then added back, the model was burned in for 10,000 iterations and 500 samples were generated to compute:

$$L_t^{(s)} = \sum_{i=1}^{N_t} \log \mathrm{p}(\mathsf{x}_{ti} \mid \boldsymbol{D})$$

Here, t denotes the tissue, s the sample, and $\boldsymbol{D}$ all the training data. An estimate of the tissue log-likelihood $\widehat{L}_t$ was then computed from the 5,000 $L_t^{(s)}$ samples using the harmonic mean method described by Kass and Raftery [38]. The tissue perplexity was then estimated as:

$$\text{perplexity} = 2^{\widehat{L}_t / N_t}$$

The full GeneProgram model consistently yielded reduced perplexity values compared to the simplified model, with a median perplexity reduction of 24%. Figure 12 shows a graph of these results. Perplexity reductions of 10% or greater have typically been considered significant [56]. Thus, we conclude that allowing the model to infer tissue groups automatically significantly improves performance.

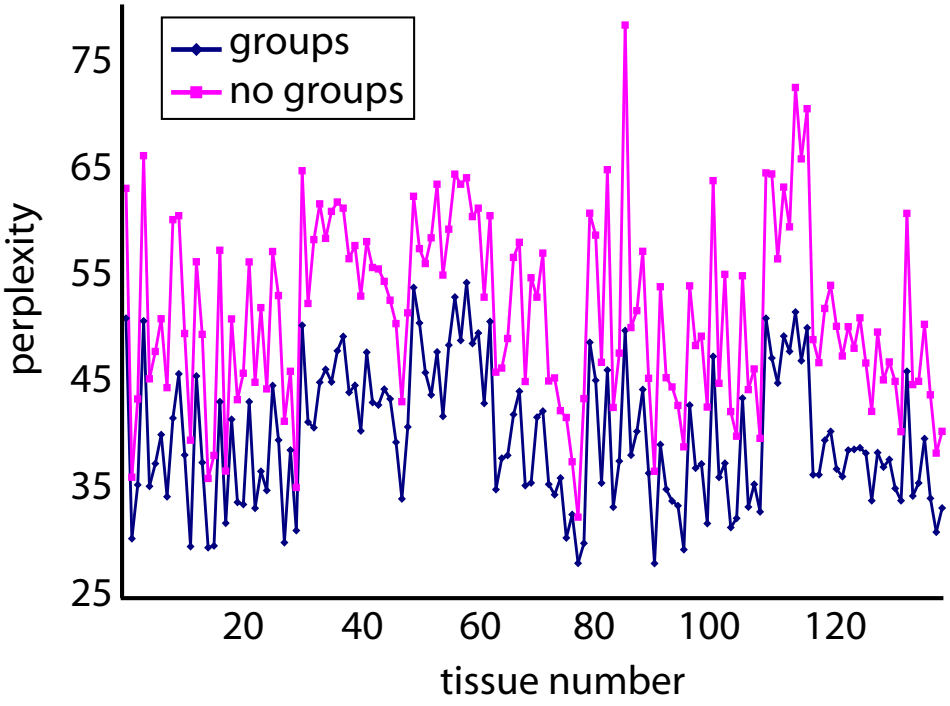


Figure 12: Automatic inference of tissue groups improves cross-validation performance of the GeneProgram model. We used 10-fold cross-validation to test the full GeneProgram model ("groups") versus a simplified version in which there are no groups and all tissues are attached directly to the root of the hierarchy ("no groups"). Each data point represents the calculated perplexity value for each held-out tissue (1-79 = human tissues, 80-140 = mouse tissues). Lower perplexity values indicate better model performance. The median reduction in perplexity for the full versus the simplified model was 24%.

4.7 The generality score quantified the functional specificity of expression programs and organized programs into a comprehensive body-wide map

We developed a score for assessing the functional generality of expression programs, and demonstrated its utility for automatically characterizing the spectrum of discovered programs—from gene sets involved in general physiologic processes to highly tissue-specific ones.

The *generality* score is the entropy of the normalized distribution of usage of an expression program by all tissues in each tissue group. Because the distribution employed for calculating the score is normalized, tissue groups that only use an expression program a relatively small amount will have little effect on the score. Thus, a high generality score indicates that an expression program is used fairly evenly across many tissue groups; a low score indicates the program is used by tissues from a small number of groups.

To be precise, let $\overline{q}_t$ denote the consensus tissue group (CTG) assignment for tissue t. We compute the usage for CTG k of recurrent expression program (REP) j as:

$$h_{kj} = \sum_{t=1}^{T} \widehat{w}_{tj} \mathrm{I}(\overline{q}_t = k)$$

The normalized usage is then computed as:

$$\widehat{h}_{kj} = \frac{h_{kj}}{\sum_{l=1}^{K} h_{lj}}$$

Here, K is the total number of CTGs. The generality score for REP j is then computed as:

$$\mathrm{GS}_j = -\sum_{k=1}^{K} \widehat{h}_{kj} \log \widehat{h}_{kj}$$

We note that the generality score requires a global organization of tissues into groups, rather than just the local associations of subsets of tissues with individual gene sets provided by biclustering algorithms. Because there is uncertainty in the number of tissue groups, GeneProgram's Dirichlet Process-based model provides a natural framework for computing the generality score.

4.7.1 Evaluation of the weighted average of generality scores across all expression programs for each tissue uncovered several trends relating to tissue function and anatomic location

Figure 10 depicts the weighted scores for all tissues. As is evident from this figure, some tissues types, including neural, testicular and thyroid samples, had very low average generality scores, presumably reflect-

ing the highly specialized functions of these tissues. In contrast, a number of other tissue types, including embryologic, hematologic progenitors, immune, malignant, epithelial and adipose samples, had very high average generality scores. In the case of embryologic and relatively undifferentiated malignant tissues, high scores presumably reflected the activation of large numbers of expression programs shared with many other types of tissues. The other high-scoring tissues mentioned also shared programs with many types of tissues, but this sharing may be attributed to both common biological functions as well as the fact that cells from these high-scoring tissues are found in many organs throughout the body.

We note that it is likely that some tissues had artificially high average generality scores due to sample contamination from anatomically nearby tissues. For instance, expression program 24, a program associated with muscle function, was used by fetal thyroid (3%), prostate (8%), lower spinal cord (4%), bone marrow (5%), and brown fat (12%). Each of these tissues is underneath substantial amounts of muscle, making contamination likely [48]. As another example, expression program 83 contained many genes involved in pancreatic function. However, this program was also used by mouse spleen (18%) and stomach (4%). Because the pancreas is anatomically proximal to both the stomach and spleen and can leave pancreatic tissue surrounding the duodenum as a result of its migration during development [53], contamination of these tissues seems likely.

4.7.2 Generality scores classified the functional specificity of individual expression programs

Figure 13 displays a histogram of generality scores for all expression programs (EPs) with non-zero scores. Based on the generality score, we divided expression programs into three broad categories: 1) general body-wide physiology, 2) specialized organ physiology, and 3) tissue specific. Below we provide illustrative examples from each category.

General body-wide physiology expression programs. EPs with high generality scores were involved in common physiological functions of cells present in a variety of tissues throughout the body. For instance, EP 13 (generality = 2.50, 25 tissues) contained many genes critical for DNA replication and EP 33 (generality = 2.34, 28 tissues) contained a striking number of genes involved with RNA processing, including numerous nuclear ribonucleoprotein components [40]. Interestingly, both EPs were used by many of the same tissues containing rapidly dividing cells, including embryologic, immune, and malignant tissues. Two additional examples include, EP 39 (generality = 2.88, 13 tissues), significantly enriched for genes involved in epithelial function, such as keratins and collagens; and, EP 24 (generality = 1.88, 15 tissues), significantly enriched for genes involved in general muscle function, including several known to be expressed in both cardiac and skeletal muscle such as alpha-actin-1 [31], myoglobin [25], and phosphoglycerate mutase isozyme M [18]. Interestingly, the tongue used both EPs 39 and 24 to a considerable extent, reflecting its mixed muscular and epithelial physiological functions.

Specialized organ physiology expression programs. EPs with intermediate generality scores were involved in specialized functions of a few closely related—but not necessarily anatomically proximate—tissues. For instance, EP 15 (generality = 1.44, 6 tissues) was significantly enriched for genes involved in erythropoiesis and was used primarily by adult bone marrow from both species and human fetal liver. Interestingly, the fetal liver is known to be critical for erythropoiesis during embryonic development, after which bone marrow becomes the predominant organ involved in this process [52]. Another example in this category includes EP 73 (generality = 0.93, 6 tissues), which was used by the kidney and liver in both species, and was enriched for genes involved in oxidative metabolism and gluconeogenesis. A final interesting example of this type is EP 88 (generality = 0.84, 3 tissues), which was used by the pituitary in both species and

to a smaller extent by human pancreatic islets (5%). This EP contained a number of specific genes involved with pituitary function such as PIT1 [51] and the prolactin precursor [15]. A literature search revealed that several of the genes contained in this EP are known to be shared between the pituitary and islets, including prohormone convertase I [76] and proopiomelanocortin preproprotein [11, 35]. However, many of the genes in EP 88 have not previously been characterized as shared between the two endocrine organs, and thus may constitute interesting future candidates for experimental biology work.

Tissue specific expression programs. Finally, EPs with very low generality scores were used by essentially a single type of tissue, and represented very specialized aspects of organ functions. For instance, EP 19 (generality = 0.0, 6 tissues) was used exclusively by testicular tissues in both species, and was significantly enriched for genes involved in spermatogenesis. Two additional examples clearly illustrate GeneProgram's ability to automatically allocate tissues' gene expression to both general and specific programs. EP 43 (generality = 0) was used exclusively by the eye and was highly enriched for lens and retina specific genes. The eye also used EP 39, the general epithelial program described above, reflecting its more prosaic components. EP 58 (generality = 0) was exclusively used by the heart in both species, and contained cardiac specific genes such as atrial natriuretic peptide [16] and cardiac troponin T [59]. The heart also used the general muscle topic, EP 24, described above. Finally, EP 53 (generality = 0.26, 38 tissues), which was significantly enriched for genes involved in neurotransmission, illustrates that the generality score can be low despite usage of a program by a large number of tissues. Neural tissues were very abundant in the data set (31% of all tissues); because GeneProgram collapsed these tissues into a small number of groups, the generality score for EP 53 accurately reflected the biological homogeneity of the exclusively neural tissues using the expression program.

5 Conclusion and discussion

We presented a new computational methodology, GeneProgram, specifically designed for analyzing large compendia of mammalian expression data. We applied our method to a large compendium of human and mouse body-wide gene expression data from representative normal tissue samples, and demonstrated that GeneProgram outperformed other methods in the discovery of biologically interpretable gene sets. We further showed that allowing the GeneProgram model to infer tissue groups automatically significantly improved performance. Using the data compendium, GeneProgram discovered 19 tissue groups and 100 expression programs active in mammalian tissues. We introduced an expression program generality score that exploits the tissue groupings automatically learned by GeneProgram, and showed that this score characterizes the functional spectrum of discovered expression programs.

GeneProgram encodes certain assumptions that differ from some previous methods for analyzing expression data and so merit further discussion. First, we model expression data in a semi-quantitative fashion, assuming that discrete levels of mRNA correspond to biologically interpretable expression differences. We believe this is appropriate because popular array technologies can only reliably measure semi-quantitative, relative changes in expression; many relevant consequences of gene expression are threshold phenomena [34, 42, 74]; and it is difficult to assign a clear biological interpretation to a full spectrum of continuous expression levels. Second, GeneProgram assumes that discrete "units" of mRNA are independently allocated to expression programs, which captures the phenomena that mRNA transcribed from the same gene can be translated into proteins that may participate in different biological processes throughout a cell or tissue. Independence of mRNA units is an unrealistic assumption, but this approximation, which is important

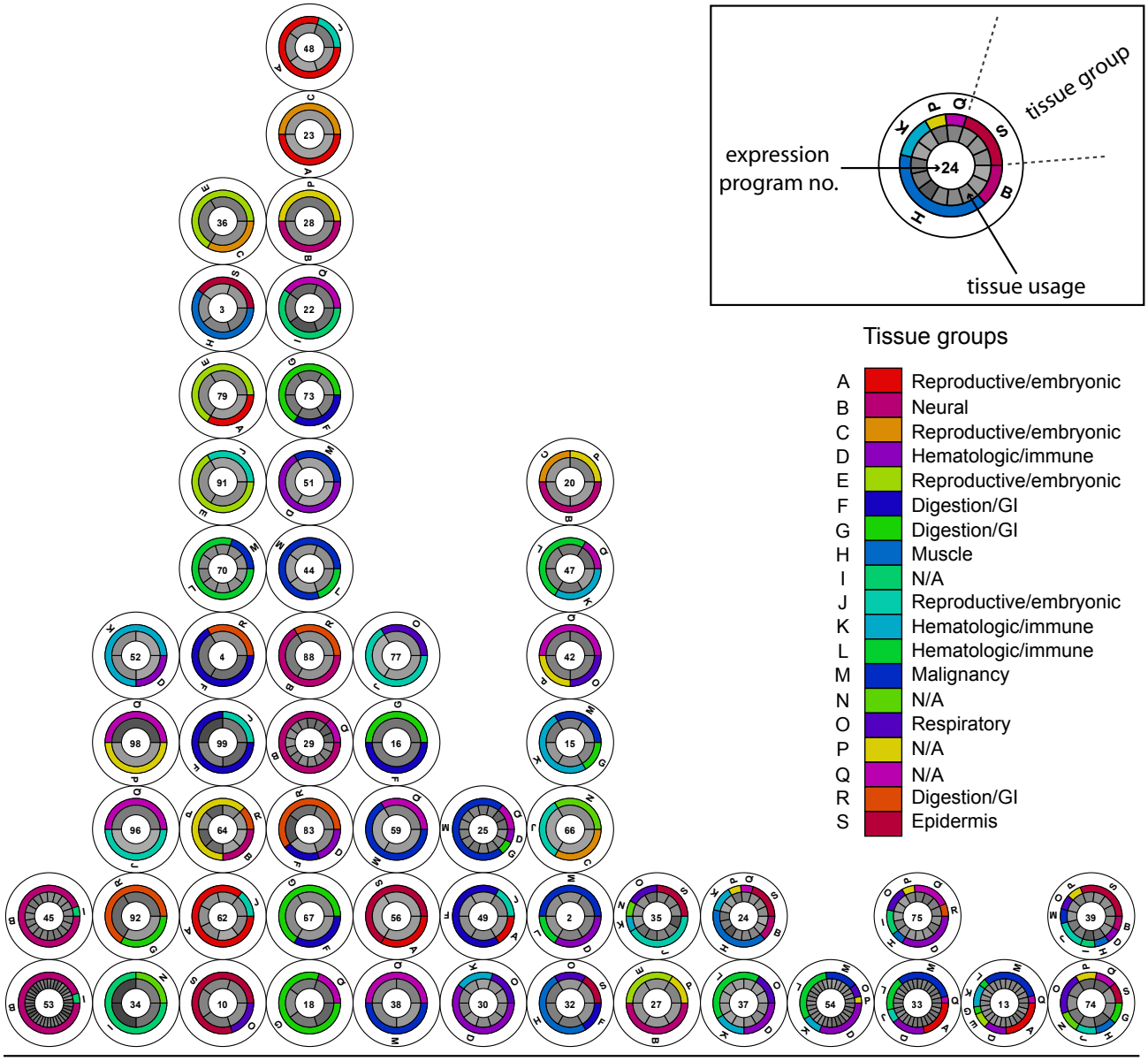

expression program no.
tissue group
tissue usage
24
Tissue groups
A Reproductive/embryonic
B Neural
C Reproductive/embryonic
D Hematologic/immune
E Reproductive/embryonic
F Digestion/GI
G Digestion/GI
H Muscle
I N/A
J Reproductive/embryonic
K Hematologic/immune
L Hematologic/immune
M Malignancy
N N/A
O Respiratory
P N/A
Q N/A
R Digestion/GI
S Epidermis
0.29
0.48
0.67
0.86
1.06
1.25
1.44
1.63
1.83
2.02
2.40
2.59
2.79
generality score

Figure 13: The generality score organized expression programs discovered by GeneProgram into a comprehensive body-wide map. A histogram using the generality score summarizes the functional specificity of the expression programs (EPs) discovered by GeneProgram in a large compendium of human and mouse gene expression data. The horizontal axis displays bins of generality scores. A high generality score indicates that an expression program is used fairly evenly across many tissue groups; a low score indicates the program is used by tissues from a small number of groups. Only EPs with non-zero scores are shown. EPs are depicted as numbered circles stacked within score bins. Two rings around each EP provide additional information. The innermost ring shows individual tissue usage percentages as shaded wedges (darker shading = higher usage). The outer ring depicts tissue groups, with arc sizes proportional to the number of tissues in the group using the EP. The boxed example shows EP 24 (generality = 1.88), which is used by 15 tissues from 6 groups. The legend below the boxed example depicts the broad physiological category that each tissue group was significantly enriched for.

for efficient inference, has worked well in practice for many other applications of topic models [23, 30, 10]. Finally, although GeneProgram does not directly model down-regulation of genes, it does capture this phenomenon implicitly in that a tissue's non-use of an expression program provides critical information for the algorithm. However, this approach does not take into account the magnitude of a gene's down-regulation or distinguish down-regulation from a lack of significant change in a gene's expression. GeneProgram can be usefully extended to take such information into account for application to datasets consisting of time-series or samples and controls, such as two-color microarray data.

Our method produced a comprehensive, body-wide map of expression programs active in mammalian physiology with several distinguishing features. First, by simultaneously using information across 140 tissue samples, GeneProgram was able to finely dissect the data, automatically splitting mRNA expressed in tissues among both general and specific programs. Second, because our model explicitly operates on probabilistically ranked gene sets throughout the entire inference process, rather than finding individual differentially expressed genes or merging genes into sets in pre-processing steps, our results are more robust to noise. Third, the fact that expression programs provide probabilistically ranked sets of genes also provides a logical means for prioritizing directed biological experiments. Fourth, because our model is fully Bayesian, providing a global penalty for model complexity including for the number of tissue groups and expression programs, the generated map represents a mathematically principled compression of gene expression information throughout the entire organism. Finally, although such a large, comprehensive map is inherently complicated, we believe that GeneProgram's automatic grouping of tissues and the associated expression program generality score aid greatly in its interpretation.

We believe that the features of the discovered map discussed above will make it particularly useful for guiding future biological experiments. Tissue-specific expression programs can provide candidate genes for diagnostic markers or drug targets that exhibit minimal "cross-talk" with unintended organs. General expression programs may be useful for identifying genes important in regulating and maintaining general physiological responses, which may go awry in disease states such as sepsis and malignancy. Both general and tissue-specific discovered programs contained many functionally unannotated genes, and in some cases the programs were shared among unexpected sets of tissues. Additionally, some such unannotated genes appear in cross-species expression programs, making them particularly attractive candidates for further bio-

logical characterization.

The map's utility can be further enhanced by adding new data as it becomes available, particularly body-wide tissue samples profiling gene expression in additional species. Further, our method is general, making it suitable for analyzing any large expression data compendium, including those relating to developmental or disease processes. Our framework is also flexible, and could accommodate other genome-wide sources of biological data in future work, such as DNA-protein binding or DNA sequence motif information. GeneProgram's ability to discover tissue groups and expression programs *de novo* using a principled probabilistic method, as well as its use of data in a semi-quantitative manner, makes it especially valuable for novel "meta-analysis" applications involving large data sets of unknown complexity in which direct fully quantitative comparisons are difficult.

References

[1] M. Abramowitz and I. A. Stegun. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. Dover, 1972.

[2] O. Alter, P. O. Brown, and D. Botstein. Singular value decomposition for genome-wide expression data processing and modeling. *Proc Natl Acad Sci U S A*, 97(18):10101–6, Aug 29 2000.

[3] O. Alter, P. O. Brown, and D. Botstein. Generalized singular value decomposition for comparative analysis of genome-scale expression data sets of two different organisms. *Proc Natl Acad Sci U S A*, 100(6):3351–6, Mar 18 2003.

[4] Anonymous. The Gene Ontology (GO) project in 2006. *Nucleic Acids Res*, 34(Database issue):D322–6, Jan 1 2006.

[5] C. Antoniak. Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *Annals of Statistics*, 2(6):1152–1174, 1974.

[6] A. Battle, E. Segal, and D. Koller. Probabilistic discovery of overlapping cellular processes and their regulation. *J Comput Biol*, 12(7):909–27, Sep 2005.

[7] Y. Benjamini and Y. Hochberg. Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *Journal of the Royal Statistical Society, Series B (Methodological)*, 57(1):289–300, 1995.

[8] S. Bergmann, J. Ihmels, and N. Barkai. Similarities and differences in genome-wide expression data of six organisms. *PLoS Biol*, 2(1):E9, Jan 2004.

[9] D. M. Blei and M. I. Jordan. Variational inference for Dirichlet process mixtures. *Journal of Bayesian Analysis*, 1:121–144, 2005.

[10] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[11] M. I. Borelli, F. E. Estivariz, and J. J. Gagliardino. Evidence for the paracrine action of islet-derived corticotropin-like peptides on the regulation of insulin release. *Metabolism*, 45(5):565–70, May 1996.

[12] J. P. Brunet, P. Tamayo, T. R. Golub, and J. P. Mesirov. Metagenes and molecular pattern discovery using matrix factorization. *Proc Natl Acad Sci U S A*, 101(12):4164–9, Mar 23 2004.

[13] P. Carmona-Saez, R. D. Pascual-Marqui, F. Tirado, J. M. Carazo, and A. Pascual-Montano. Biclustering of gene expression data by non-smooth non-negative matrix factorization. *BMC Bioinformatics*, 7:78, 2006.

[14] Y. Cheng and G. M. Church. Biclustering of expression data. In *Eighth International Conference on Intelligent Systems for Molecular Biology (ISMB)*, pages 93–103. AAAI Press, 2000.

[15] N. E. Cooke, D. Coit, J. Shine, J. D. Baxter, and J. A. Martial. Human prolactin: cDNA structural analysis and evolutionary comparisons. *J Biol Chem*, 256(8):4007–16, Apr 25 1981.

[16] A. J. de Bold. Atrial natriuretic factor: a hormone produced by the heart. *Science*, 230(4727):767–70, Nov 15 1985.

[17] D. Dueck, Q. D. Morris, and B. J. Frey. Multi-way clustering of microarray data using probabilistic sparse matrix factorization. *Bioinformatics*, 21 Suppl 1:i144–51, Jun 2005.

[18] Y. H. Edwards, S. Sakoda, E. Schon, and S. Povey. The gene for human muscle-specific phosphoglycerate mutase, PGAM2, mapped to chromosome 7 by polymerase chain reaction. *Genomics*, 5(4):948–51, Nov 1989.

[19] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *Proc Natl Acad Sci U S A*, 95(25):14863–8, Dec 8 1998.

[20] M. D. Escobar and M. West. Bayesian Density Estimation and Inference Using Mixtures. *Journal of the American Statistical Association*, 90(430):577–588, 1995.

[21] T. S. Ferguson. A Bayesian analysis of some nonparametric problems. *Annals of Statistics*, 1:209–230, 1973.

[22] T. S. Ferguson. Prior distributions on spaces of probability measures. *Annals of Statistics*, 2:615–629, 1974.

[23] P. Flaherty, G. Giaever, J. Kumm, M. I. Jordan, and A. P. Arkin. A latent variable model for chemogenomic profiling. *Bioinformatics*, 21(15):3286–93, Aug 1 2005.

[24] D. A. Freedman. On the asymptotic behavior of Bayes estimates in the discrete case. *Annals of Mathematical Statistics*, 34:1386–1403, 1963.

[25] D. J. Garry, G. A. Ordway, J. N. Lorenz, N. B. Radford, E. R. Chin, R. W. Grange, R. Bassel-Duby, and R. S. Williams. Mice without myoglobin. *Nature*, 395(6705):905–8, Oct 29 1998.

[26] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian Data Analysis*. Chapman & Hall, 2004.

[27] G. K. Gerber, R. D. Dowell, T. S. Jaakkola, and D. K. Gifford. *Table 1 (Supplemental): Summary of expression programs discovered by GeneProgram from Novartis Tissue Atlas v2 data, CSAIL Digial Work Product Archive*. http://hdl.handle.net/1721.1/37602.

[28] G. K. Gerber, R. D. Dowell, T. S. Jaakkola, and D. K. Gifford. *Table 2 (Supplemental): Complete data for all 100 expression programs discovered by GeneProgram from the Novartis Gene Atlas v2, CSAIL Digial Work Product Archive*. http://hdl.handle.net/1721.1/37603.

[29] G. K. Gerber, R. D. Dowell, T. S. Jaakkola, and D. K. Gifford. Automated discovery of functional generality of human gene expression programs. *PLoS Computational Biology*, to appear, 2007.

[30] T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proc Natl Acad Sci U S A*, 101 Suppl 1:5228–35, Apr 6 2004.

[31] P. Gunning, P. Ponte, L. Kedes, R. Eddy, and T. Shows. Chromosomal location of the co-expressed human skeletal and cardiac actin genes. *Proc Natl Acad Sci U S A*, 81(6):1813–7, Mar 1984.

[32] A. J. Hartemink, D. K. Gifford, T. S. Jaakkola, and R. A. Young. Using graphical models and genomic expression data to statistically validate models of genetic regulatory networks. In *Pac. Symp. on Biocomputing (PSB)*, pages 422–433, 2001.

[33] E. Hubbell, W. M. Liu, and R. Mei. Robust estimators for expression analysis. *Bioinformatics*, 18(12):1585–92, Dec 2002.

[34] D. A. Hume. Probability in transcriptional regulation and its implications for leukocyte differentiation and inducible gene expression. *Blood*, 96:2323–2328, 2000.

[35] A. Hummel, U. Lendeckel, H. Hahn von Dorsche, and H. Zuhlke. Presence and regulation of a truncated proopiomelanocortin gene transcript in rat pancreatic islets. *Biol Chem Hoppe Seyler*, 373(10):1039–44, Oct 1992.

[36] H. Ishwaran and L. James. Gibbs sampling methods for stick-breaking priors. *Journal of the American Statistical Association*, 96(453):161–173, 2001.

[37] R. G. Jenner and R. A. Young. Insights into host responses against pathogens from transcriptional profiling. *Nat Rev Microbiol*, 3(4):281–94, Apr 2005.

[38] R. E. Kass and A. E. Raftery. Bayes Factors. *Journal of the American Statistical Association*, 90(430):773–795, 1995.

[39] K. Kurihara, M. Welling, and N. Vlassis. Accelerated variational Dirichlet process mixtures. In *Neural Information Processing Systems (NIPS)*, Vancouver, B.C., 2006.

[40] T. Lehmeier, V. Raker, H. Hermann, and R. Luhrmann. cDNA cloning of the Sm proteins D2 and D3 from human small nuclear ribonucleoproteins: evidence for a direct D1-D2 interaction. *Proc Natl Acad Sci U S A*, 91(25):12317–21, Dec 6 1994.

[41] B. Y. Liao and J. Zhang. Evolutionary conservation of expression profiles between human and mouse orthologous genes. *Mol Biol Evol*, 23(3):530–40, Mar 2006.

[42] J. W. Little. Threshold effects in gene regulation: when some is not enough. *Proc Natl Acad Sci U S A*, 102(15):5310–1, 2005.

[43] M. Liu, S. J. Popper, K. H. Rubins, and D. A. Relman. Early days: genomics and human responses to infection. *Curr Opin Microbiol*, 9(3):312–9, Jun 2006.

[44] S. C. Madeira and A. L. Oliveira. Biclustering algorithms for biological data analysis: A survey. *IEEE Transactions on Computational Biology and Bioinformatics*, 1(1):24–45, 2004.

[45] G. McLachlan and D. Peel. *Finite Mixture Models*. John Wiley & Sons, 2000.

[46] M. Medvedovic, K. Y. Yeung, and R. E. Bumgarner. Bayesian mixture model based clustering of replicated microarray data. *Bioinformatics*, 20(8):1222–32, May 22 2004.

[47] T. Minka and Z. Ghahramani. Expectation propagation for infinite mixtures. In *Neural Information Processing Systems (NIPS)*, Vancouver, B.C., 2003.

[48] K. L. Moore and A. F. Dalley. *Clinically Oriented Anatomy*. Lippincott Williams & Wilkins, fourth edition, 1999.

[49] D. J. Navarro, T. L. Griffiths, M. Steyvers, and M. D. Lee. Modeling individual differences using Dirichlet processes. *Journal of Mathematical Psychology*, 50:101–122, 2006.

[50] R. M. Neal. Markov chain sampling methods for Dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, 9:249–265, 2000.

[51] K. Ohta, Y. Nobukuni, H. Mitsubuchi, T. Ohta, T. Tohma, Y. Jinno, F. Endo, and I. Matsuda. Characterization of the gene encoding human pituitary-specific transcription factor, Pit-1. *Gene*, 122(2):387–8, Dec 15 1992.

[52] S. H. Orkin and L. I. Zon. Genetics of erythropoiesis: induced mutations in mice and zebrafish. *Annu Rev Genet*, 31:33–60, 1997.

[53] T. Pieler and Y. Chen. Forgotten and novel aspects in pancreas development. *Biol Cell*, 98(2):79–88, Feb 2006.

[54] C. Rasmussen. The Infinite Gaussian Mixture Model. In *Neural Information Processing Systems (NIPS)*, 2000.

[55] N. G. Robertson, L. Lu, S. Heller, S. N. Merchant, R. D. Eavey, M. McKenna, Jr. Nadol J. B., R. T. Miyamoto, Jr. Linthicum F. H., J. F. Lubianca Neto, A. J. Hudspeth, C. E. Seidman, C. C. Morton, and J. G. Seidman. Mutations in a novel cochlear gene cause DFNA9, a human nonsyndromic deafness with vestibular dysfunction. *Nat Genet*, 20(3):299–303, Nov 1998.

[56] R. Rosenfeld. Two decades of statistical language modeling: Where do we go from here? *Proceedings of the IEEE*, 88(8), 2000.

[57] E. Segal, N. Friedman, N. Kaminski, A. Regev, and D. Koller. From signatures to models: understanding cancer using microarrays. *Nat Genet*, 37 Suppl:S38–45, Jun 2005.

[58] E. Segal, N. Friedman, D. Koller, and A. Regev. A module map showing conditional activity of expression modules in cancer. *Nat Genet*, 36(10):1090–8, Oct 2004.

[59] A. J. Sehnert, A. Huq, B. M. Weinstein, C. Walker, M. Fishman, and D. Y. Stainier. Cardiac troponin T is essential in sarcomere assembly and cardiac contractility. *Nat Genet*, 31(1):106–10, may 2002.

[60] J. Sethuraman. A Constructive Definition of Dirichlet Priors. *Statistica Sinica*, 4:639–650, 1994.

[61] R. Shamir, A. Maron-Katz, A. Tanay, C. Linhart, I. Steinfeld, R. Sharan, Y. Shiloh, and R. Elkon. EXPANDER–an integrative program suite for microarray data analysis. *BMC Bioinformatics*, 6:232, 2005.

[62] Q. Sheng, Y. Moreau, and B. De Moor. Biclustering microarray data by Gibbs sampling. *Bioinformatics*, 19 Suppl 2:II196–II205, Oct 2003.

[63] R. Shyamsundar, Y. H. Kim, J. P. Higgins, K. Montgomery, M. Jorden, A. Sethuraman, M. van de Rijn, D. Botstein, P. O. Brown, and J. R. Pollack. A DNA microarray survey of gene expression in normal human tissues. *Genome Biol*, 6:R22, 2005.

[64] J. M. Stuart, E. Segal, D. Koller, and S. K. Kim. A gene-coexpression network for global discovery of conserved genetic modules. *Science*, 302(5643):249–55, Oct 10 2003.

[65] A. I. Su, T. Wiltshire, S. Batalov, H. Lapp, K. A. Ching, D. Block, J. Zhang, R. Soden, M. Hayakawa, G. Kreiman, M. P. Cooke, J. R. Walker, and J. B. Hogenesch. A gene atlas of the mouse and human protein-encoding transcriptomes. *Proc Natl Acad Sci U S A*, 101(16):6062–7, Apr 20 2004.

[66] E. B. Sudderth, A. Torralba, W. T. Freeman, and A. S. Willsky. Describing Visual Scenes using Transformed Dirichlet Processes. In *Neural Information Processing Systems (NIPS)*, Vancouver, B.C., 2005.

[67] E. B. Sudderth, A. Torralba, W. T. Freeman, and A. S. Wilsky. Learning Hierarchical Models of Scences, Objects, and Parts. In *International Conf. on Computer Vision*, Beijing, China, 2005.

[68] A. Tanay, R. Sharan, and R. Shamir. Discovering statistically significant biclusters in gene expression data. *Bioinformatics*, 18 Suppl 1:S136–44, 2002.

[69] A. Tanay, R. Sharan, and R. Shamir. *Biclustering Algorithms: A Survey*. Computer and Information Science Series. Chapman & Hall/CRC, 2005.

[70] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical Dirichlet Processes. *Journal of the American Statistical Association*, 2006.

[71] R. Wadlow and S. Ramaswamy. DNA microarrays in clinical cancer research. *Curr Mol Med*, 5(1):111–20, Feb 2005.

[72] D. L. Wheeler, T. Barrett, D. A. Benson, S. H. Bryant, K. Canese, V. Chetvernin, D. M. Church, M. DiCuccio, R. Edgar, S. Federhen, L. Y. Geer, W. Helmberg, Y. Kapustin, D. L. Kenton, O. Khovayko, D. J. Lipman, T. L. Madden, D. R. Maglott, J. Ostell, K. D. Pruitt, G. D. Schuler, L. M. Schriml, E. Sequeira, S. T. Sherry, K. Sirotkin, A. Souvorov, G. Starchenko, T. O. Suzek, R. Tatusov, T. A. Tatusova, L. Wagner, and E. Yaschenko. Database resources of the National Center for Biotechnology Information. *Nucleic Acids Res*, 34(Database issue):D173–80, Jan 1 2006.

[73] Z. Wu and R. A. Irizarry. Stochastic models inspired by hybridization theory for short oligonucleotide arrays. *J Comput Biol*, 12(6):882–93, Jul-Aug 2005.

[74] Q. Zhang, M. E. Andersen, and R. B. Conolly. Binary gene induction and protein expression in individual cells. *Theor Biol Med Model*, 3, 2006.

[75] X. J. Zhou and G. Gibson. Cross-species comparison of genome-wide expression patterns. *Genome Biol*, 5(7):232, 2004.

[76] X. Zhu, A. Zhou, A. Dey, C. Norrbom, R. Carroll, C. Zhang, V. Laurent, I. Lindberg, R. Ugleholdt, J. J. Holst, and D. F. Steiner. Disruption of PC1/3 expression in mice causes dwarfism and multiple neuroendocrine peptide processing defects. *Proc Natl Acad Sci U S A*, 99(16):10293–8, Aug 6 2002.