
@inproceedings{Hu:Boyd-Graber:Daume-III:Ying-2013,
  Url = {http://umiacs.umd.edu/~jbg/docs/2013_coalescent.pdf}
  Title = {Binary to Bushy: Bayesian Hierarchical Clustering with the Beta Coalescent},
  Author = {Yuening Hu and Jordan Boyd-Graber and Hal Daumé III and Z. Irene Ying},
  Booktitle = {Neural Information Processing Systems},
  Year = {2013},
}

Links:
- Data [http://www.umiacs.umd.edu/~ynhu/code/CoalescentTree.zip]

Downloaded from [http://umiacs.umd.edu/~jbg/docs/2013_coalescent.pdf]
Abstract

Discovering hierarchical regularities in data is a key problem in interacting with large datasets, modeling cognition, and encoding knowledge. A previous Bayesian solution—Kingman’s coalescent—provides a probabilistic model for data represented as a binary tree. Unfortunately, this is inappropriate for data better described by bushier trees. We generalize an existing belief propagation framework of Kingman’s coalescent to the beta coalescent, which models a wider range of tree structures. Because of the complex combinatorial search over possible structures, we develop new sampling schemes using sequential Monte Carlo and Dirichlet process mixture models, which render inference efficient and tractable. We present results on synthetic and real data that show the beta coalescent outperforms Kingman’s coalescent and is qualitatively better at capturing data in bushy hierarchies.

1 The Need For Bushy Hierarchical Clustering

Hierarchical clustering is a fundamental data analysis problem: given observations, what hierarchical grouping of those observations effectively encodes the similarities between observations? This is a critical task for understanding and describing observations in many domains [1, 2], including natural language processing [3], computer vision [4], and network analysis [5]. In all of these cases, natural and intuitive hierarchies are not binary but are instead bushy, with more than two children per parent node. Our goal is to provide efficient algorithms to discover bushy hierarchies.

We review existing nonparametric probabilistic clustering algorithms in Section 2, with particular focus on Kingman’s coalescent [6] and its generalization, the beta coalescent [7, 8]. While Kingman’s coalescent has attractive properties—it is probabilistic and has edge “lengths” that encode how similar clusters are—it only produces binary trees. The beta coalescent (Section 3) does not have this restriction. However, naïve inference is impractical, because bushy trees are more complex: we need to consider all possible subsets of nodes to construct each internal nodes in the hierarchy.

Our first contribution is a generalization of the belief propagation framework [9] for beta coalescent to compute the joint probability of observations and trees (Section 3). After describing sequential Monte Carlo posterior inference for the beta coalescent, we develop efficient inference strategies in Section 4, where we use proposal distributions that draw on the connection between Dirichlet processes—a ubiquitous Bayesian nonparametric tool for non-hierarchical clustering—and hierarchical coalescents to make inference tractable. We present results on both synthetic and real data that show the beta coalescent captures bushy hierarchies and outperforms Kingman’s coalescent (Section 5).

2 Bayesian Clustering Approaches

Recent hierarchical clustering techniques have been incorporated inside statistical models; this requires formulating clustering as a statistical—often Bayesian—problem. Heller et al. [10] build
binary trees based on the marginal likelihoods, extended by Blundell et al. [11] to trees with arbitrary
branching structure. Ryan et al. [12] propose a tree-structured stick-breaking process to generate trees
with unbounded width and depth, which supports data observations at leaves and internal nodes. However, these models do not distinguish edge lengths, an important property in distinguishing how “tight” the clustering is at particular nodes.

Hierarchical models can be divided into complementary “fragmentation” and “coagulation” frame-
works [7]. Both produce hierarchical partitions of a dataset. Fragmentation models start with a single
partition and divide it into ever more specific partitions until only singleton partitions remain. Coagula-
tion frameworks repeatedly merge singleton partitions until only one partition remains. Pitman-Yor
diffusion trees [13], a generalization of Dirichlet diffusion trees [14], are an example of a bushy
fragmentation model, and they model edge lengths and build non-binary trees.

Instead, our focus is on bottom-up coalescent models [8], one of the coagulation models and
complementary to diffusion trees, which can also discover hierarchies and edge lengths. In this
model, \( n \) nodes are observed (we use both observed to emphasize that nodes are known and leaves
to emphasize topology). These observed nodes are generated through some unknown tree with latent
dges and unobserved internal nodes. Each node (both observed and latent) has a single
parent replaces the original nodes. This is called a coalescent event. This process repeats until there
is only one node left, and a complete tree structure \( \pi \) (Figure 1) is obtained.

Consider a set of \( n \) individuals observed at the present (time \( t = 0 \)). All individuals start in one of \( n \)
singleton sets. After time \( t_i \), a set of these nodes coalesce into a new node. Once a set merges, their
parent replaces the original nodes. This is called a coalescent event. This process repeats until there
is only one node left, and a complete tree structure \( \pi \) (Figure 1) is obtained.

Different coalescents are defined by different probabilities of merging a set of nodes. This is called
the coalescent rate, defined by a general family of coalescents: the lambda coalescent [7, 15]. We
represent the rate via the symbol \( \lambda^k_n \), the rate at which \( k \) out of \( n \) nodes merge into a parent node.
From a collection of \( n \) nodes, \( k \geq n \) can coalesce at some coalescent event (\( k \) can be different for
different coalescent events). The rate of a fraction \( \gamma \) of the nodes coalescing is given by \( \gamma^{-2} \Lambda(d \gamma) \),
where \( \Lambda(d \gamma) \) is a finite measure on \([0, 1]\). So \( k \) nodes merge at rate

\[
\lambda^k_n = \int_0^1 \gamma^{k-2}(1 - \gamma)^{n-k} \Lambda(d \gamma) \quad (2 \leq k \leq n).
\]

Choosing different measures yields different coalescents. A degenerate Dirac delta measure at 0
results in Kingman’s coalescent [6], where \( \lambda^k_n \) is 1 when \( k = 2 \) and zero otherwise. Because this
gives zero probability to non-binary coalescent events, this only creates binary trees.

Alternatively, using a beta distribution \( \text{BETA}(2 - \alpha, \alpha) \) as the measure \( \Lambda \) yields the beta coalescent.
When \( \alpha \) is closer to 1, the tree is bushier; as \( \alpha \) approaches 2, it becomes Kingman’s coalescent. If we
have \( n_{i-1} \) nodes at time \( t_{i-1} \) in a beta coalescent, the rate \( \lambda^k_{n_{i-1}} \) for a children set of \( k \) nodes at time
\( t_i \) and the total rate \( \lambda_{n_{i-1}} \) of any children set merging—summing over all possible mergers—is

\[
\lambda^k_{n_{i-1}} = \frac{\Gamma(k - \alpha)\Gamma(n_{i-1} - k + \alpha)}{\Gamma(2 - \alpha)\Gamma(\alpha)\Gamma(n_{i-1})} \quad \text{and} \quad \lambda_{n_{i-1}} = \sum_{k=2}^{n_{i-1}} \binom{n_{i-1}}{k} \lambda^k_{n_{i-1}}.
\]

Each coalescent event also has an edge length—duration—\( \delta_i \). The duration of an event comes from
an exponential distribution, \( \delta_i \sim \exp(\lambda_{n_{i-1}}) \), and the parent node forms at time \( t_i = t_{i-1} - \delta_i \).
Shorter durations mean that the children more closely resemble their parent (the mathematical basis
for similarity is specified by a transition kernel, Section 3).

Analogous to Kingman’s coalescent, the prior probability of a complete tree \( \pi \) is the product of all of
its constituent coalescent events \( i = 1, \ldots, m \), merging \( k_i \) children after duration \( \delta_i \),

\[
p(\pi) = \prod_{i=1}^m \left( \frac{p(k_i,n_{i-1})}{p(\delta_i,k_i,n_{i-1})} \cdot \frac{p(\delta_i,k_i,n_{i-1})}{\exp(-\lambda_{n_{i-1}} \delta_i)} \right) = \prod_{i=1}^m \lambda^k_{n_{i-1}} \cdot \exp(-\lambda_{n_{i-1}} \delta_i).
\]

\[\text{---1\textsuperscript{This is appropriate where the entirety of a population is known—both ancestors and descendants. We focus on the case where only the descendants are known. For a concrete example, see Section 5.2.}}\]
Algorithm 1 MCMC inference for generating a tree
1: for Particle \( s = 1, 2, \ldots, S \) do
2: Initialize \( n^s = n \), \( i = 0 \), \( t^0 = 0 \), \( w^0_\delta = 1 \).
3: Initialize the node set \( V^s = \{ \rho_0, \rho_1, \ldots, \rho_n \} \).
4: while \( \exists b \in \{ 1 \cdots S \} \) where \( n^s > 1 \) do
5: Update \( i = i + 1 \).
6: for Particle \( s = 1, 2, \ldots, S \) do
7: if \( n^s == 1 \) then
8: Continue.
9: Propose a duration \( \delta^s_i \) by Equation 10.
10: Set coalescent time \( t^s_i = t^s_{i-1} - \delta^s_i \).
11: Sample partitions \( \rho^s_i \) from DPMM.
12: Propose a set \( \rho^s_i \) according to Equation 11.
13: Update weight \( w^s_i \) by Equation 13.
14: Update \( n^s = n^s - |\rho^s_i| + 1 \).
15: Remove \( \rho^s_i \) from \( V^s \); add \( \rho^s_i \) to \( V^s \).
16: Compute effective sample size \( \text{ESS} \) [16].
17: if \( \text{ESS} < S/2 \) then
18: Resample particles [17].

3 Beta Coalescent Belief Propagation

The beta coalescent prior only depends on the topology of the tree. In real clustering applications, we also care about a node’s children and features. In this section, we define the nodes and their features, and then review how we use message passing to compute the probabilities of trees.

An internal node \( \rho_i \) is defined as the merger of other nodes. The children set of node \( \rho_i \), \( \rho_{c,i} \), coalesces into a new node \( \rho_i \equiv \cup_{b \in c_i} \rho_b \). This encodes the identity of the nodes that participate in specific coalescent events; Equation 3, in contrast, only considers the number of nodes involved in an event. In addition, each node is associated with a multidimensional feature vector \( y_i \).

Two terms specify the relationship between nodes’ features: an initial distribution \( p_0(y_b) \) and a transition kernel \( \kappa_{t_i,t_b}(y_i, y_b) \). The initial distribution can be viewed as a prior or regularizer for feature representations. The transition kernel encourages a child’s feature \( y_b \) (at time \( t_b \)) to resemble feature \( y_i \) (formed at \( t_i \)); shorter durations \( t_b - t_i \) increase the resemblance.

Intuitively, the transition kernel can be thought as a similarity score; the more similar the features are, the more likely nodes are. For Brownian diffusion (discussed in Section 4.3), the transition kernel follows a Gaussian distribution centered at a feature. The covariance matrix \( \Sigma \) is decided by the mutation rate \( \mu \) [18, 9], the probability of a mutation in an individual. Different kernels (e.g., multinomial, tree kernels) can be applied depending on modeling assumptions of the feature representations.

To compute the probability of the beta coalescent tree \( \pi \) and observed data \( x \), we generalize the belief propagation framework used by Teh et al. [9] for Kingman’s coalescent; this is a more scalable alternative to other approaches for computing the probability of a Beta coalescent tree [19]. We define a subtree structure \( \theta_i = \{ \theta_{i-1}, \delta_i, \rho_{c,i} \} \), thus the tree \( \theta_m \) after the final coalescent event \( m \) is a complete tree \( \pi \). The message for node \( \rho_i \) marginalizes over the features of the nodes in its children set.\(^2\) The total message for a parent node \( \rho_i \) is

\[
M_{\rho_i}(y_i) = Z_{\rho_i}^{-1}(x|\theta_i) \prod_{b \in \rho_i} \int \kappa_{t_i,t_b}(y_i, y_b) M_{\rho_b}(y_b) dy_b. \tag{4}
\]

where \( Z_{\rho_i}(x|\theta_i) \) is the local normalizer, which can be computed as the combination of initial distribution and messages from a set of children,

\[
Z_{\rho_i}(x|\theta_i) = \int p_0(y_i) \prod_{b \in \rho_i} \left( \int \kappa_{t_i,t_b}(y_i, y_b) M_{\rho_b}(y_b) dy_b \right) dy_i. \tag{5}
\]

\(^2\)When \( \rho_b \) is a leaf, the message \( M_{\rho_b}(y_b) \) is a delta function centered on the observation.
Recursively performing this marginalization through message passing provides the joint probability of a complete tree \( \pi \) and the observations \( x \). At the root,

\[
Z_{-\infty}(x|\theta_m) = \int p_0(y_{-\infty}) \kappa_{-\infty,t_m}(y_{-\infty},y_m) \mathcal{M}_{\rho_m}(y_m) dy_m dy_{-\infty}
\]

where \( p_0(y_{-\infty}) \) is the initial feature distribution and \( m \) is the number of coalescent events. This gives the marginal probability of the whole tree,

\[
p(x|\pi) = Z_{-\infty}(x|\theta_m) \prod_{i=1}^m Z_{\rho_i}(x|\theta_i),
\]

The joint probability of a tree \( \pi \) combines the prior (Equation 3) and likelihood (Equation 7),

\[
p(x, \pi) = Z_{-\infty}(x|\theta_m) \prod_{i=1}^m \lambda_{s_i}^{\delta_i} \exp(-\lambda_{s_i-1} \delta_i) \cdot Z_{\rho_i}(x|\theta_i).
\]

### 3.1 Sequential Monte Carlo Inference

Sequential Monte Carlo (SMC)—often called particle filters—estimates a structured sequence of hidden variables based on observations [20]. For coalescent models, this estimates the posterior distribution over tree structures given observations \( x \). Initially (\( i = 0 \)) each observation is in a singleton cluster;\(^1\) in subsequent particles (\( i > 0 \)), points coalesce into more complicated tree structures \( \theta_i \), where \( s \) is the particle index and we add superscript \( s \) to all the related notations to distinguish between particles. We use sequential importance resampling [21, SIR] to weight each particle \( s \) at time \( t_i \), denoted as \( w^s_{t_i} \).

The weights from SIR approximate the posterior. Computing the weights requires a conditional distribution of data given a latent state \( p(x|\theta^s_{t_i}) \), a transition distribution between latent states \( p(\theta^s_{t_i}|\theta^s_{t_i-1}) \), and a proposal distribution \( f(\theta^s_{t_i}|\theta^s_{t_i-1},x) \). Together, these distributions define weights

\[
w^s_{t_i} = w^s_{t_i-1} \frac{p(x|\theta^s_{t_i})p(\theta^s_{t_i}|\theta^s_{t_i-1})}{f(\theta^s_{t_i}|\theta^s_{t_i-1},x)}. \tag{9}
\]

Then we can approximate the posterior distribution of the hidden structure using the normalized weights, which become more accurate with more particles.

To apply SIR inference to belief propagation with the beta coalescent prior, we first define the particle space structure. The \( s^{th} \) particle represents a subtree \( \theta_{t_i-1}^s \) at time \( t_{i-1} \), and a transition to a new subtree \( \theta_i^s \) takes a set of nodes \( \rho_i^s \) from \( \theta_{t_i-1}^s \) and merges them at \( t_i \), where \( t_i = t_{i-1} + \delta_i \) and \( \theta_i^s = \{ \theta_{t_i-1}^s, \rho_i^s \} \). Our proposal distribution must provide the duration \( \delta_i \) and the children set \( \rho_i^s \) to merge based on the previous subtree \( \theta_{t_i-1}^s \).

We propose the duration \( \delta_i \) from the prior exponential distribution and propose a children set from the posterior distribution based on the local normalizers.\(^4\) This is the “priorpost” method in Teh et al. [9].

However, this approach is intractable. Given \( n_{t_i} \) nodes at time \( t_i \), we must consider all possible children sets \( \binom{n_{t_i-1}}{2} + \binom{n_{t_i-1}}{3} + \cdots + \binom{n_{t_i-1}}{n_{t_i-1}} \). The computational complexity grows from \( O(n_{t_i-1}^2) \) (Kingman’s coalescent) to \( O(2^n) \) (beta coalescent).

### 4 Efficiently Finding Children Sets with DPMM

We need a more efficient way to consider possible children sets. Even for Kingman’s coalescent, which only considers pairs of nodes, Gorur et al. [22] do not exhaustively consider all pairs. Instead, they use data structures from computational geometry to select the \( R \) closest pairs as their restriction set, reducing inference to \( O(n \log n) \). While finding closest pairs is a traditional problem in computational geometry, discovering arbitrary-sized sets is less studied.

\(^1\)The relationship between time and particles is non-intuitive. Time \( t \) goes backward with subsequent particles. When we use time-specific adjectives for particles, this is with respect to inference.

\(^4\)This is a special case of Section 4.2’s algorithm, where the restriction set \( \Omega \) is all possible subsets.
In this section, we describe how we use a Dirichlet process mixture model [23, DPMM] to discover a restriction set \(\Omega\), integrating DPMMs into the SMC proposal. We first briefly review what DPMMs are, describe why they are attractive, and then describe how we incorporate DPMMs in SMC inference.

The DPMM is defined by a concentration \(\beta\) and a base distribution \(G_0\). A distribution over mixtures is drawn from a Dirichlet process (DP): \(G \sim \text{DP}(\beta, G_0)\). Each observation \(x_i\) is assigned to a mixture component \(\mu_i\), drawn from \(G\). Because the Dirichlet process is a discrete distribution, observations \(i\) and \(j\) have the same mixture component \((\mu_i = \mu_j)\). When this happens, points are said to be in the same partition. Posterior inference can discover a distribution over partitions. A full derivation of these sampling equations appears in the supplemental material.

### 4.1 Attractive Properties of DPMMs

**DPMMs and Coalescents** Berestycki et al. [8] showed that the distribution over partitions in a Dirichlet process is equivalent to the distribution over coalescents’ allelic partitions—the set of members that have the same feature representation—when the mutation rate \(\mu\) of the associated kernel is half of the Dirichlet concentration \(\beta\) (Section 3). For Brownian diffusion, we can connect DPMM with coalescents by setting the kernel covariance \(\Sigma = \mu I\) to \(\Sigma = \beta / 2 I\).

The base distribution \(G_0\) is also related with nodes’ feature. The base distribution \(G_0\) of a Dirichlet process generates the probability measure \(G\) for each block, which generates the nodes in a block. As a result, we can select a base distribution which fits the distribution of the samples in coalescent process. For example, if we use Gaussian distribution for the transition kernel and prior, a Gaussian is also appropriate as the DPMM base distribution.

**Effectiveness as a Proposal** The necessary condition for a valid proposal [24] is that it should have support on a superset of the true posterior. In our case, the distribution over partitions provided by the DPMM considers all possible children sets that could be merged in the coalescent. Thus the new proposal with DPMM satisfies this requirement, and it is a valid proposal.

In addition, Chen [25] gives a set of desirable criteria for a good proposal distribution: accounts for outliers, considers the likelihood, and lies close to the true posterior. The DPMM fulfills these criteria. First, the DPMM provides a distribution over all partitions. Varying the concentration parameter \(\beta\) can control the length of the tail of the distribution over partitions. Second, choosing the base distribution of the DPMM appropriately models the feature likelihood; i.e., ensuring the DPMM places similar nodes together in a partition with high probability. Third, the DPMM qualitatively provides reasonable children sets when compared with exhaustively considering all children sets (Figure 2(c)).

### 4.2 Incorporating DPMM in SMC Proposals

To address the inference intractability in Section 3.1, we use the DPMM to obtain a distribution over partitions of nodes. Each partition contains clusters of nodes, and we take a union over all partitions to create a restriction set \(\Omega_i = \{\omega_{i1}, \omega_{i2}, \cdots \}\), where each \(\omega_{ij}\) is a subset of the \(n_{i-1}\) nodes. A standard Gibbs sampler provides these partitions (see supplemental).

With this restriction set \(\Omega_i\), we propose the duration time \(\delta_i^t\) from the exponential distribution and propose a children set \(\rho^{*}_{ij}\) based on the local normalizers

\[
f_i(\delta_i^t) = \lambda_{n_{i-1}} n_{i-1} \exp(-\lambda_{n_{i-1}} \delta_i^t) \quad (10)
\]

\[
f_i(\rho^{*}_{ij} | \delta_i^t, \theta_{i-1}^s) = \frac{Z_{\rho_i}(x|\theta_{i-1}^s, \delta_i^t, \rho^{*}_{ij})}{Z_0} \quad (11)
\]

where \(\Omega_i^s\) restricts the candidate children sets, \(\mathbb{1}\) is the indicator, and we replace \(Z_{\rho_i}(x|\theta_{i}^s)\) with \(Z_{\rho_i}(x|\theta_{i-1}^s, \delta_i^t, \rho^{*}_{ij})\) since they are equivalent here. The normalizer is

\[
Z_0 = \sum_{\rho^{*}_{i}} Z_{\rho_i}(x|\theta_{i-1}^s, \delta_i^t, \rho^{*}_{ij}) \cdot \mathbb{1} [\rho^{*}_{ij} \in \Omega_i^s] = \sum_{\rho^{*}_{i} \in \Omega_i^s} Z_{\rho_i}(x|\theta_{i-1}^s, \delta_i^t, \rho^{*}_{ij}). \quad (12)
\]

Applying the true distribution (the \(j^{th}\) multiplicand from Equation 8) and the proposal distribution (Equation 10 and Equation 11) to the SIR weight update (Equation 9),

\[
w_i^t = w_{i-1}^t \frac{\lambda_{n_{i-1}} n_{i-1} \sum_{\rho^{*}_{i} \in \Omega_i^s} Z_{\rho_i}(x|\theta_{i-1}^s, \delta_i^t, \rho^{*}_{ij})}{\lambda_{n_{i-1}} n_{i-1}}.
\]

5
where $|\rho_s^{ci}|$ is the size of children set $\rho_s^{ci}$; parameter $\lambda_{n_{ci}}$ is the rate of the children set $\rho_s^{ci}$ (Equation 2); and $\lambda_{n_{ci}-1}$ is the rate of all possible sets given a total number of nodes $n_{ci-1}$ (Equation 2).

We can view this new proposal as a coarse-to-fine process: DPMM proposes candidate children sets; SMC selects a children set from DPMM to coalesce. Since the coarse step is faster and filters “bad” children sets, the slower finer step considers fewer children sets, saving computation time (Algorithm 1). If $\Omega$, has all children sets, it recovers exhaustive SMC. We estimate the effective sample size [16] and resample [17] when needed. For smaller sets, the DPMM is sometimes impractical (and only provides singleton clusters). In such cases it is simpler to enumerate all children sets.

### 4.3 Example Transition Kernel: Brownian Diffusion

This section uses Brownian diffusion as an example for message passing framework. The initial distribution $p_i(y_i)$ of each node is $N(0, \infty)$; the transition kernel $\kappa_{t_i,t_{i+1}}(y_i, \cdot)$ is a Gaussian centered at $y_i$ with variance $(t_i-t_{i+1})\Sigma$, where $\Sigma = \mu I$, $\mu = \beta/2$, $\beta$ is the concentration parameter of DPMM. Then the local normalizer $Z_{\rho_i}(x|\theta_i)$ is

$$Z_{\rho_i}(x|\theta_i) = \int N(y_i; 0, \infty) \prod_{b \in E_i} N(y_i; \bar{y}_b, \Sigma(v_{y_b} + t_b - t_{i+1})dy_i, \quad (14)$$

and the node message $M_{\rho_i}(y_i)$ is normally distributed $M_{\rho_i}(y_i) \sim N(y_i; \bar{y}_{\rho_i}, \Sigma v_{\rho_i})$, where

$$v_{\rho_i} = \left( \sum_{b \in E_i} (v_{y_b} + t_b - t_{i+1})^{-1} \right)^{-1}, \quad \bar{y}_{\rho_i} = \left( \sum_{b \in E_i} \frac{\bar{y}_b}{v_{y_b} + t_b - t_{i+1}} \right) v_{\rho_i}.$$

### 5 Experiments: Finding Bushy Trees

In this section, we compare trees built by the beta coalescent (beta) against those built by Kingman’s coalescent (kingman) and hierarchical agglomerative clustering [26, hac] on both synthetic and real data. We show beta performs best and can capture data in more interpretable, bushier trees.

**Setup** The parameter $\alpha$ for the beta coalescent is between 1 and 2. The closer $\alpha$ is to 1, the bushier the tree is, and we set $\alpha = 1.2$.\textsuperscript{5} We set the mutation rate as 1, thus the DPMM parameter is initialized as $\beta = 2$, and updated using slice sampling [27]. All experiments use 100 initial iterations of DPMM inference with 30 more iterations after each coalescent event (forming a new particle).

**Metrics** We use three metrics to evaluate the quality of the trees discovered by our algorithm: purity, subtree and path length. The dendrogram purity score [28, 10] measures how well the leaves in a subtree belong to the same class. For any two leaf nodes, we find the least common subsumer node $s$ and—for the subtree rooted at $s$—measure the fraction of leaves with same class labels. The subtree score [9] is the ratio between the number of internal nodes with all children in the same class and the total number of internal nodes. The path length score is the average difference—over all pairs—of the lowest common subsumer distance between the true tree and the generated tree, where the lowest common subsumer distance is the distance between the root and the lowest common subsumer of two nodes. For purity and subtree, higher is better, while for length, lower is better. Scores are in expectation over particles and averaged across chains.

### 5.1 Synthetic Hierarchies

To test our inference method, we generated synthetic data with edge length (full details available in the supplemental material); we also assume each child of the root has a unique label and the descendants also have the same label as their parent node (except the root node).

We compared beta against kingman and hac by varying the number of observations (Figure 2(a)) and feature dimensions (Figure 2(b)). In both cases, beta is comparable to kingman and hac (no edge length). While increasing the feature dimension improves both scores, more observations do not: for synthetic data, a small number of observations suffice to construct a good tree.

\textsuperscript{5}With DPMM proposals, $\alpha$ has a negligible effect, so we elide further analysis for different $\alpha$ values.
Figure 2: Figure 2(a) and 2(b) show the effect of changing the underlying data size or number of dimension. Figure 2(c) shows that our DPMM proposal for children sets is comparable to an exhaustive enumeration of all possible children sets (enum).

To evaluate the effectiveness of using our DPMM as a proposal distribution, we compare exhaustively enumerating all children set candidates (enum) while keeping the SMC otherwise unchanged; this experiment uses ten data points (enum is completely intractable on larger data). Beta uses the DPMM and achieved similar accuracy (Figure 2(c)) while greatly improving efficiency.

5.2 Human Tissue Development

Our first real dataset is based on the developmental biology of human tissues. As a human develops, tissues specialize, starting from three embryonic germ layers: the endoderm, ectoderm, and mesoderm. These eventually form all human tissues. For example, one developmental pathway is \textit{ectoderm} → \textit{neural crest} → \textit{cranial neural crest} → \textit{optic vesicle} → \textit{cornea}. Because each germ layer specializes into many different types of cells at specific times, it is inappropriate to model this development as a binary tree, or with clustering models lacking path lengths.

Historically, uncovering these specialization pathways is a painstaking process, requiring inspection of embryos at many stages of development; however, massively parallel sequencing data make it possible to efficiently form developmental hypotheses based on similar patterns of gene expression.

To investigate this question we use the transcriptome of 27 tissues with known, unambiguous, time-specific lineages [29]. We reduce the original 182727 dimensions via principle component analysis [30, PCA]. We use five chains with five particles per chain.

Using reference developmental trees, \textbf{beta} performs better on all three scores (Table 1) because \textbf{beta} builds up a bushy hierarchy more similar to the true tree. The tree recovered by \textbf{beta} (Figure 3) reflects human development. The first major differentiation is the division of embryonic cells into three layers of tissue: endoderm, mesoderm, and ectoderm. These go on to form almost all adult organs and cells. The placenta (magenta), however, forms from a fourth cell type, the trophoblast; this is placed in its own cluster at the root of the tree. It also successfully captures ectodermal tissue lineage. However, mesodermic and endodermic tissues, which are highly diverse, do not cluster as well. Tissues known to secrete endocrine hormones (dashed borders) cluster together.

5.3 Clustering 20-newsgroups Data

Following Heller et al. [10], we also compare the three models on 20-newsgroups, a multilevel hierarchy first dividing into general areas (rec, space, and religion) before specializing into areas such as baseball or hockey. This true hierarchy is inset in the bottom right of Figure 4, and we assume each edge has the same length. We apply latent Dirichlet allocation [31] with 50 topics to this corpus, and use the topic distribution for each document as the document feature. We use five chains with eighty particles per chain.

\[ \text{http://qwone.com/~jason/20Newsgroups/} \]

\[ \text{We use “rec.autos”, “rec.sport.baseball”, “rec.sport.hockey”, “sci.space” newsgroups but also—in contrast to Heller et al. [10]—added “soc.religion.christian.”} \]
We would like to thank the anonymous reviewers for their helpful comments, and thank Héctor Corrada Bravo for pointing us to human tissue data. This research was supported by NSF grant #1018625. Any opinions, findings, conclusions, or recommendations expressed here are those of the authors and do not necessarily reflect the view of the sponsor.
References


