Project 2: Augmenting functional information about human genes using probabilistic phylogenetic modeling

George G. Vega Yon
vegayon@usc.edu
Duncan Thomas Paul D. Thomas Paul Marjoram Huaiyu Mi John Morrison

Department of Preventive Medicine
University of Southern California

October 27, 2017

Agenda

On Genes and Trees

Model

Peeling algorithm

The `amcmc` R package

The `aphylo` R package

Preliminary Results

Concluding Remarks

Agenda

Overview

- A GO annotation is an association between a gene and a GO (Gene Ontology) term describing its function, e.g: A gene can be annotated with the GO term `GO:0016049`, which denotes *cellular growth*.

Overview

- A GO annotation is an association between a gene and a GO (Gene Ontology) term describing its function, e.g: A gene can be annotated with the GO term `GO:0016049`, which denotes *cellular growth*.
- Functional knowledge (e.g. Gene Ontology (GO) terms annotations) for human genes is very incomplete.

Overview

- A GO annotation is an association between a gene and a GO (Gene Ontology) term describing its function, e.g: A gene can be annotated with the GO term `GO:0016049`, which denotes *cellular growth*.
- Functional knowledge (e.g. Gene Ontology (GO) terms annotations) for human genes is very incomplete.
- Increase in association detection power using prior biological knowledge depends strongly on annotation completeness.

Overview

- A GO annotation is an association between a gene and a GO (Gene Ontology) term describing its function, e.g: A gene can be annotated with the GO term `GO:0016049`, which denotes *cellular growth*.
- Functional knowledge (e.g. Gene Ontology (GO) terms annotations) for human genes is very incomplete.
- Increase in association detection power using prior biological knowledge depends strongly on annotation completeness.
- Phylogenetic inference of annotations (i.e. using evolutionary trees) allows vast experimental knowledge in model systems (e.g. mouse, fruit fly, yeast) to augment human gene annotations.

Overview (cont.)

- Manual curation of GO terms is good but infeasible, e.g.:

Overview (cont.)

- Manual curation of GO terms is good but infeasible, e.g.: 1 to 2 people annotating ~4,000 families took roughly 4 years (6 years of FTE).

Overview (cont.)

- Manual curation of GO terms is good but infeasible, e.g.: 1 to 2 people annotating ~4,000 families took roughly 4 years (6 years of FTE).
- Today, we present a model that uses both:

Overview (cont.)

- Manual curation of GO terms is good but infeasible, e.g.: 1 to 2 people annotating ~4,000 families took roughly 4 years (6 years of FTE).
- Today, we present a model that uses both:
  - Existing gene functional annotations, and
  - Phylogenetic trees

  to infer annotations on un-annotated genes in a *probabilistic way* (so it is not a 0/1 prediction).

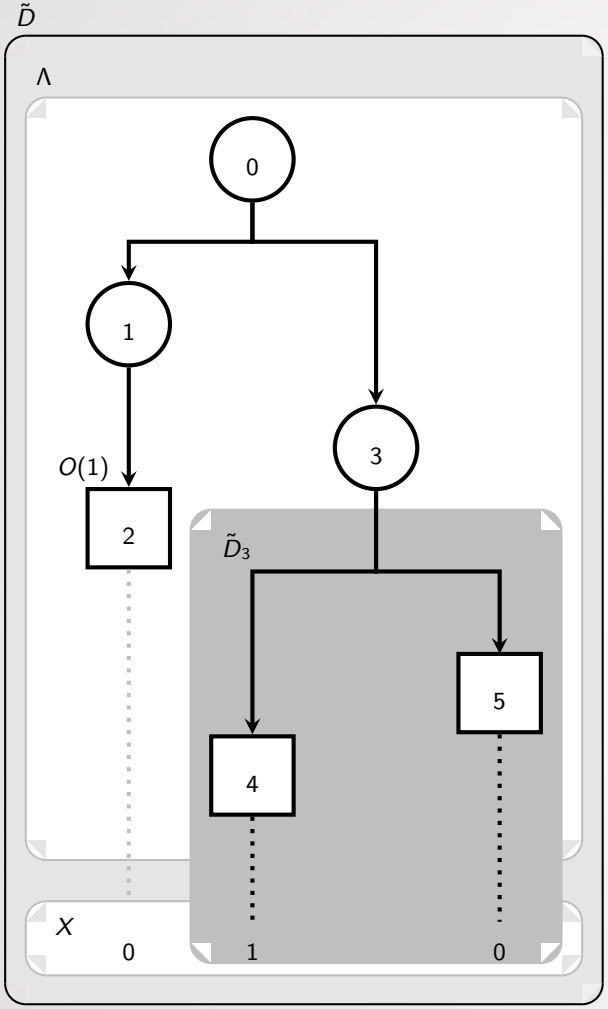
Overview (cont.)

- Manual curation of GO terms is good but infeasible, e.g.: 1 to 2 people annotating ~4,000 families took roughly 4 years (6 years of FTE).
- Today, we present a model that uses both:
  - Existing gene functional annotations, and
  - Phylogenetic trees

  to infer annotations on un-annotated genes in a *probabilistic way* (so it is not a 0/1 prediction).
- This predicted functional information will serve as prior covariates in Projects 1 and 3.

Agenda

Some definitions



Symbol	Description
$\tilde{D}$	Observed Annotated Tree
Λ	Partially ordered phylogenetic tree (PO tree)
$O(n)$	Offspring of node n
$\tilde{D}_n$	n-induced Annotated Sub-tree
X	True Annotation
X_{obs}	Experimental annotation

Where

$$x_{obs\,lp} = \begin{cases} 1 & \text{if the function is believed to be present} \\ 0 & \text{if the function is believed to be absent} \\ 9 & \text{if we don't have information for this node} \end{cases}$$

more details

A probabilistic model of function propagation

1. For any given node, we can write down the probability of observing a *functional state* as a function of some model parameters and its offspring.

A probabilistic model of function propagation

1. For any given node, we can write down the probability of observing a *functional state* as a function of some model parameters and its offspring.
2. This version of our model has five parameters (probabilities):

A probabilistic model of function propagation

1. For any given node, we can write down the probability of observing a *functional state* as a function of some model parameters and its offspring.
2. This version of our model has five parameters (probabilities):
   - 2.1 Root node had a function: π,
   - 2.2 Gain of function: μ_0,
   - 2.3 Loss of function: μ_1.
   - 2.4 Misclassification of:
     - A missing function as present, ψ_0, and
     - A present function as missing, ψ_1

A probabilistic model of function propagation

1. For any given node, we can write down the probability of observing a *functional state* as a function of some model parameters and its offspring.
2. This version of our model has five parameters (probabilities):
   - 2.1 Root node had a function: π,
   - 2.2 Gain of function: μ_0,
   - 2.3 Loss of function: μ_1.
   - 2.4 Misclassification of:
     - A missing function as present, ψ_0, and
     - A present function as missing, ψ_1

   All five parameters are assumed to be equal across functions, this is, $\pi, \mu_0, \mu_1, \psi_0$, and ψ_1 are assumed to be independent of the functions that are analyzed.

A probabilistic model of function propagation

1. For any given node, we can write down the probability of observing a *functional state* as a function of some model parameters and its offspring.
2. This version of our model has five parameters (probabilities):
   - 2.1 Root node had a function: π,
   - 2.2 Gain of function: μ_0,
   - 2.3 Loss of function: μ_1.
   - 2.4 Misclassification of:
     - A missing function as present, ψ_0, and
     - A present function as missing, ψ_1

   All five parameters are assumed to be equal across functions, this is, $\pi, \mu_0, \mu_1, \psi_0$, and ψ_1 are assumed to be independent of the functions that are analyzed.
3. In this presentation, we will focus on the case that we are dealing with a single function.

Agenda

Peeling (pruning) phylogenies (Felsenstein, 1973, 1981)

Given an experimentally annotated phylogenetic tree, the likelihood computation on a single function is as follows.

Peeling (pruning) phylogenies (Felsenstein, 1973, 1981)

Given an experimentally annotated phylogenetic tree, the likelihood computation on a single function is as follows.

1. Create an matrix P of size $|N| \times 2$,

Peeling (pruning) phylogenies (Felsenstein, 1973, 1981)

Given an experimentally annotated phylogenetic tree, the likelihood computation on a single function is as follows.

1. Create an matrix P of size $|N| \times 2$,
2. For node $n \in \{\text{peeling sequence}\}$ (from leafs to root) do:

Peeling (pruning) phylogenies (Felsenstein, 1973, 1981)

Given an experimentally annotated phylogenetic tree, the likelihood computation on a single function is as follows.

1. Create an matrix P of size $|N| \times 2$,
2. For node $n \in \{\text{peeling sequence}\}$ (from leafs to root) do:
   - 2.1 For $x_n \in \{0, 1\}$ do:

     Set $P_{n,x_n} = \begin{cases} \Pr(X_n = x_n \mid X_{obs\,n} = X_{obs\,n}) & \text{If n is a leaf} \\ L\left(X_n = x_n \,\middle|\, \tilde{D}_n\right) & \text{otherwise} \end{cases}$
   - 2.2 Next n

Peeling (pruning) phylogenies (Felsenstein, 1973, 1981)

Given an experimentally annotated phylogenetic tree, the likelihood computation on a single function is as follows.

1. Create an matrix P of size $|N| \times 2$,
2. For node $n \in \{\text{peeling sequence}\}$ (from leafs to root) do:
   - 2.1 For $x_n \in \{0, 1\}$ do:

     Set $P_{n,x_n} = \begin{cases} \Pr(X_n = x_n \mid X_{obs\,n} = X_{obs\,n}) & \text{If n is a leaf} \\ L\left(X_n = x_n \,\middle|\, \tilde{D}_n\right) & \text{otherwise} \end{cases}$
   - 2.2 Next n
3. At this point the matrix P should be completely filled, we can compute

$$L\left(\psi, \mu, \pi \middle| \tilde{D}\right) = \pi L\left(X_0 = 1 \,\middle|\, \tilde{D}_0\right) + (1 - \pi) L\left(X_0 = 0 \,\middle|\, \tilde{D}_0\right)$$

Peeling (pruning) phylogenies (Felsenstein, 1973, 1981)

Given an experimentally annotated phylogenetic tree, the likelihood computation on a single function is as follows.

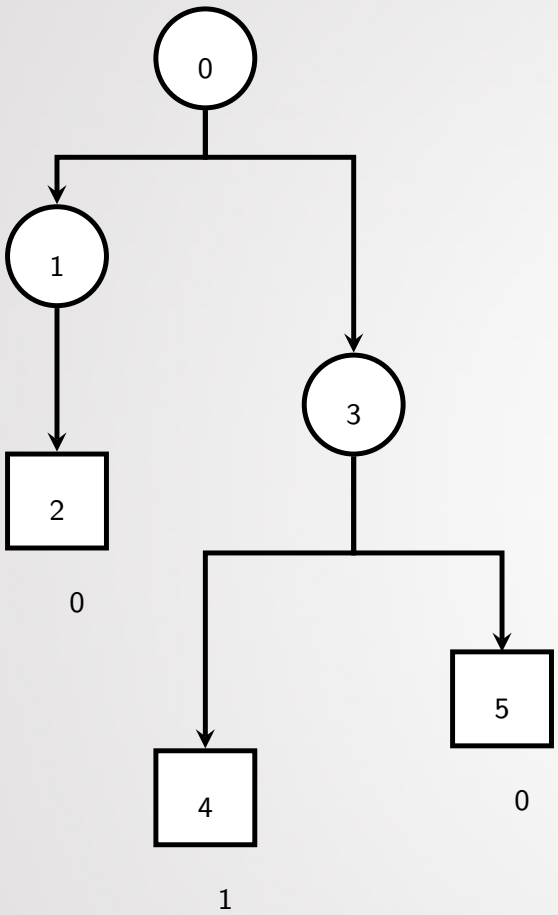
1. Create an matrix P of size $|N| \times 2$,
2. For node $n \in \{\text{peeling sequence}\}$ (from leafs to root) do:
   - 2.1 For $x_n \in \{0, 1\}$ do:

     Set $P_{n,x_n} = \begin{cases} \Pr(X_n = x_n \mid X_{obs\,n} = X_{obs\,n}) & \text{If n is a leaf} \\ L\left(X_n = x_n \,\middle|\, \tilde{D}_n\right) & \text{otherwise} \end{cases}$
   - 2.2 Next n
3. At this point the matrix P should be completely filled, we can compute

$$L\left(\psi, \mu, \pi \middle| \tilde{D}\right) = \pi L\left(X_0 = 1 \,\middle|\, \tilde{D}_0\right) + (1 - \pi) L\left(X_0 = 0 \,\middle|\, \tilde{D}_0\right)$$

Let's see an example! more details

Peeling algorithm

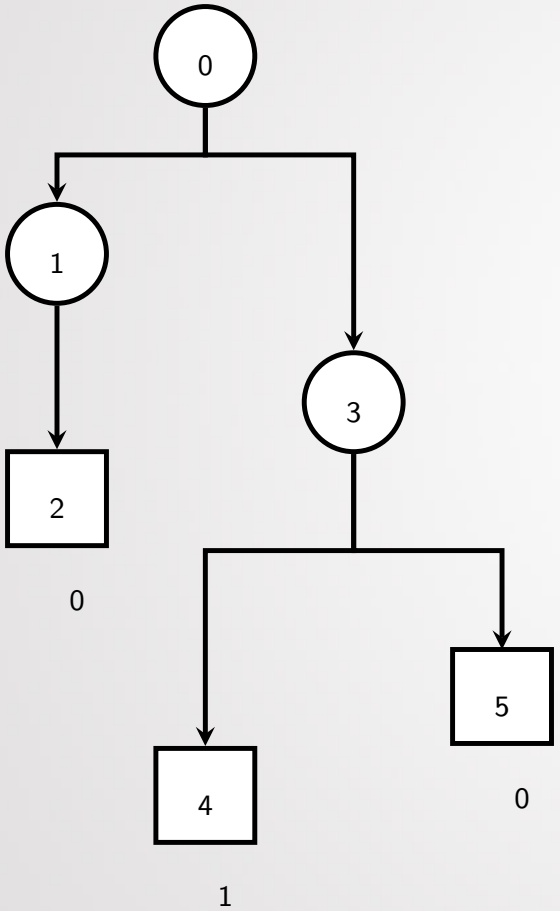


- Let's calculate the likelihood of observing this tree with the following parameters:

$$\begin{aligned} \text{Mislabeling a 0} : \psi_0 &= 0.05 \\ \text{Mislabeling a 1} : \psi_1 &= 0.01 \\ \text{Functional gain} : \mu_0 &= 0.04 \\ \text{Functional loss} : \mu_1 &= 0.01 \\ \text{Root node has the function} : \pi &= 0.05 \end{aligned}$$

Peeling algorithm (cont. 1)

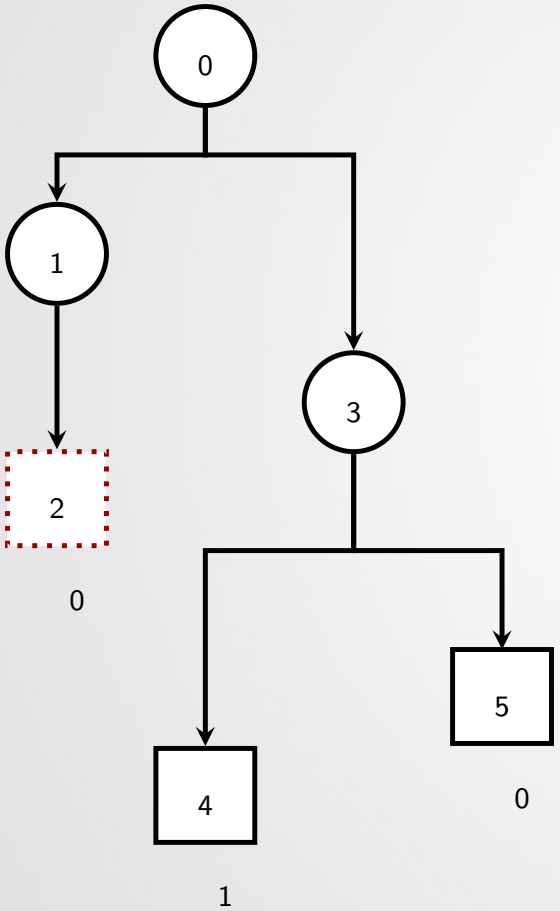
$\psi_0 = 0.05$ $\psi_1 = 0.01$ $\mu_0 = 0.04$ $\mu_1 = 0.01$ $\pi = 0.05$



	State 0	State 1
0		
1		
2		
3		
4		
5		

Peeling algorithm (cont. 1)

$\psi_0 = 0.05$ $\psi_1 = 0.01$ $\mu_0 = 0.04$ $\mu_1 = 0.01$ $\pi = 0.05$

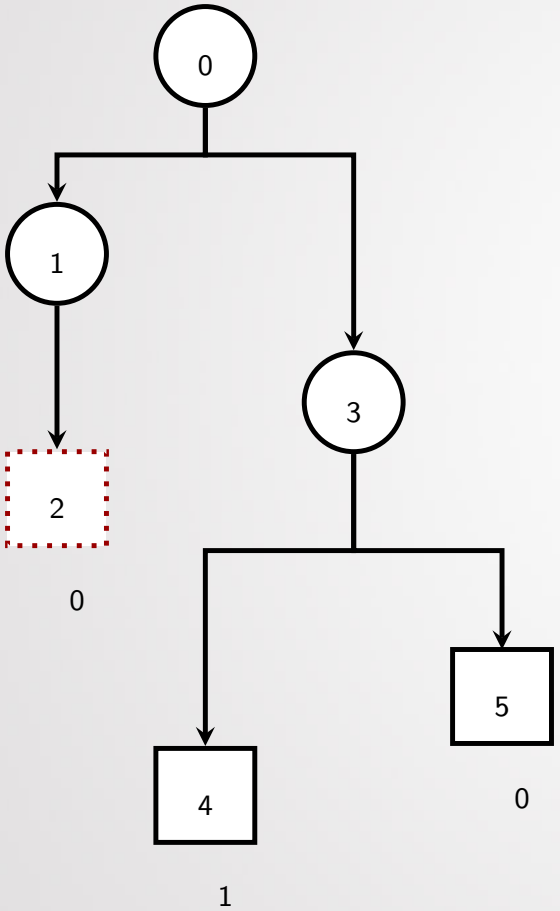


	State 0	State 1
0		
1		
2	0.9500	
3		
4		
5		

$\Pr(X_{obs2} = 0 \mid X_2 = 0) = 1 - \psi_0 = 0.95$

Peeling algorithm (cont. 1)

$\psi_0 = 0.05$ $\psi_1 = 0.01$ $\mu_0 = 0.04$ $\mu_1 = 0.01$ $\pi = 0.05$



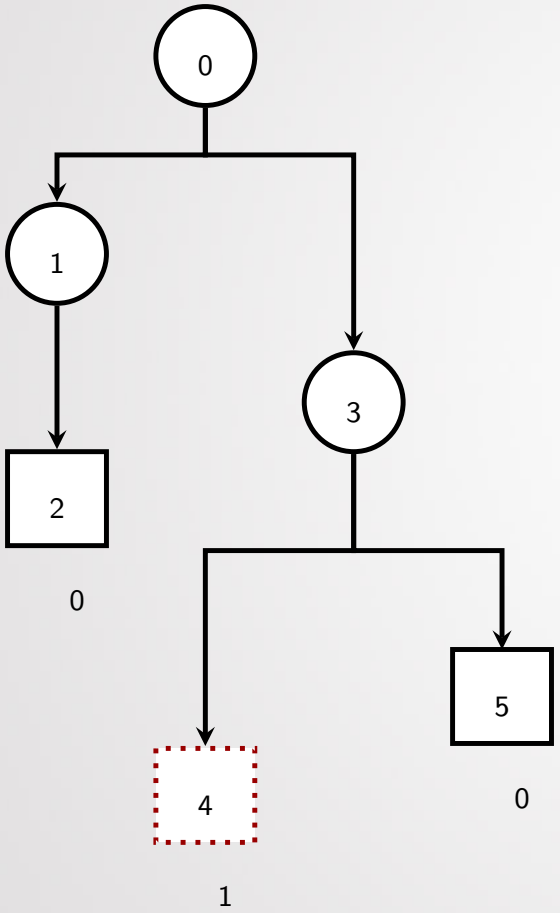
	State 0	State 1
0		
1		
2	0.9500	0.0100
3		
4		
5		

$\Pr(X_{obs2} = 0 \mid X_2 = 0) = 1 - \psi_0 = 0.95$

$\Pr(X_{obs2} = 0 \mid X_2 = 1) = \psi_1 = 0.01$

Peeling algorithm (cont. 1)

$\psi_0 = 0.05$ $\psi_1 = 0.01$ $\mu_0 = 0.04$ $\mu_1 = 0.01$ $\pi = 0.05$



	State 0	State 1
0		
1		
2	0.9500	0.0100
3		
4	0.0500	
5		

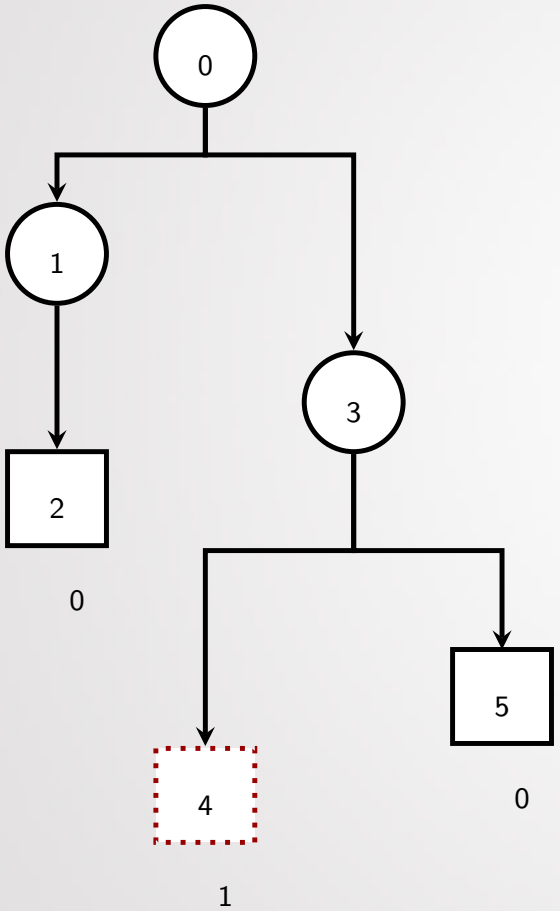
$\Pr(X_{obs2} = 0 \mid X_2 = 0) = 1 - \psi_0 = 0.95$

$\Pr(X_{obs2} = 0 \mid X_2 = 1) = \psi_1 = 0.01$

$\Pr(X_{obs4} = 1 \mid X_4 = 0) = \psi_0 = 0.05$

Peeling algorithm (cont. 1)

$\psi_0 = 0.05$ $\psi_1 = 0.01$ $\mu_0 = 0.04$ $\mu_1 = 0.01$ $\pi = 0.05$



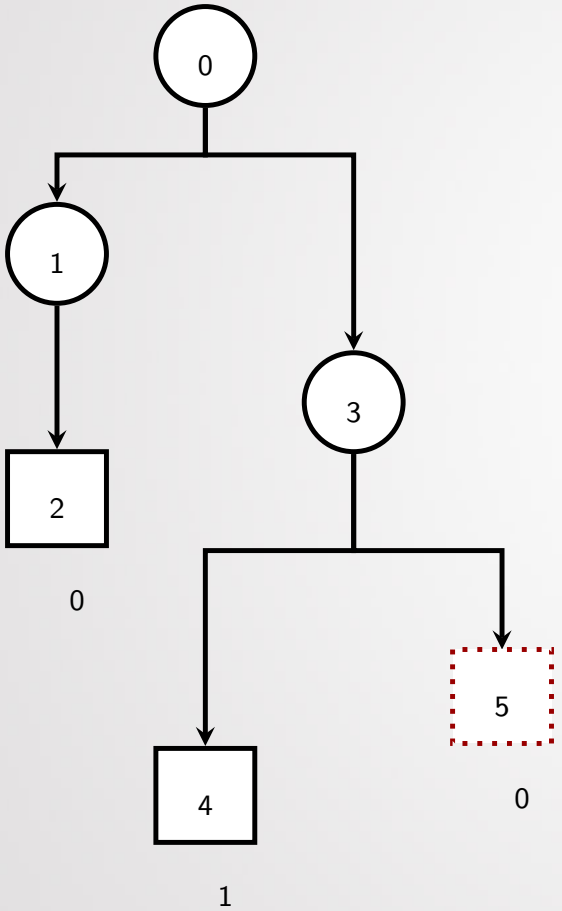
	State 0	State 1
0		
1		
2	0.9500	0.0100
3		
4	0.0500	0.9900
5		

$$\Pr(X_{obs2} = 0 \mid X_2 = 0) = 1 - \psi_0 = 0.95$$
$$\Pr(X_{obs2} = 0 \mid X_2 = 1) = \psi_1 = 0.01$$

$$\Pr(X_{obs4} = 1 \mid X_4 = 0) = \psi_0 = 0.05$$
$$\Pr(X_{obs4} = 1 \mid X_4 = 1) = 1 - \psi_1 = 0.99$$

Peeling algorithm (cont. 1)

$\psi_0 = 0.05$ $\psi_1 = 0.01$ $\mu_0 = 0.04$ $\mu_1 = 0.01$ $\pi = 0.05$



	State 0	State 1
0		
1		
2	0.9500	0.0100
3		
4	0.0500	0.9900
5	0.9500	

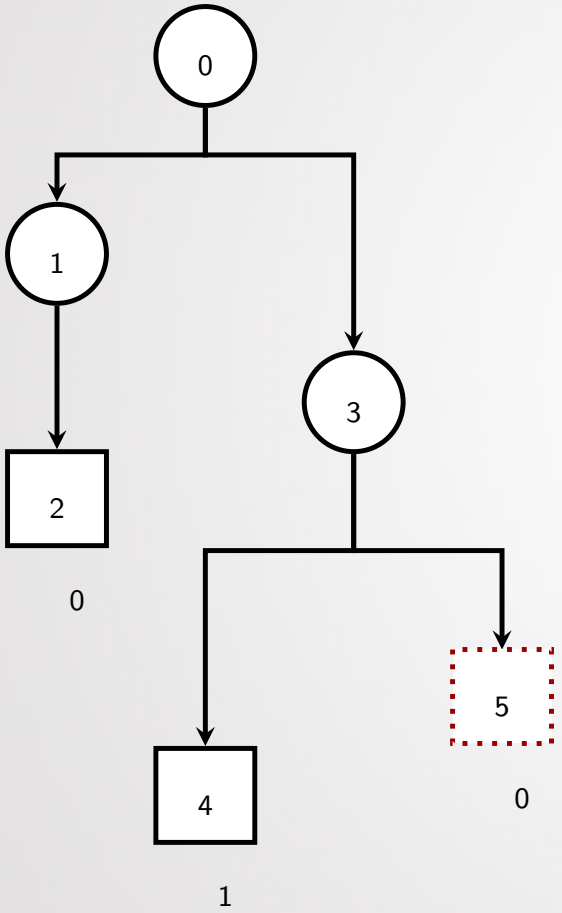
$$\Pr(X_{obs2} = 0 \mid X_2 = 0) = 1 - \psi_0 = 0.95$$
$$\Pr(X_{obs2} = 0 \mid X_2 = 1) = \psi_1 = 0.01$$

$$\Pr(X_{obs4} = 1 \mid X_4 = 0) = \psi_0 = 0.05$$
$$\Pr(X_{obs4} = 1 \mid X_4 = 1) = 1 - \psi_1 = 0.99$$

$$\Pr(X_{obs5} = 0 \mid X_5 = 0) = 1 - \psi_0 = 0.95$$

Peeling algorithm (cont. 1)

$\psi_0 = 0.05$ $\psi_1 = 0.01$ $\mu_0 = 0.04$ $\mu_1 = 0.01$ $\pi = 0.05$



	State 0	State 1
0		
1		
2	0.9500	0.0100
3		
4	0.0500	0.9900
5	0.9500	0.0100

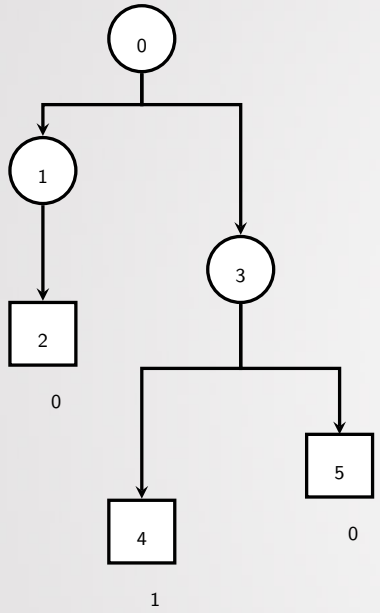
$$\Pr(X_{obs2} = 0 \mid X_2 = 0) = 1 - \psi_0 = 0.95$$
$$\Pr(X_{obs2} = 0 \mid X_2 = 1) = \psi_1 = 0.01$$

$$\Pr(X_{obs4} = 1 \mid X_4 = 0) = \psi_0 = 0.05$$
$$\Pr(X_{obs4} = 1 \mid X_4 = 1) = 1 - \psi_1 = 0.99$$

$$\Pr(X_{obs5} = 0 \mid X_5 = 0) = 1 - \psi_0 = 0.95$$
$$\Pr(X_{obs5} = 0 \mid X_5 = 1) = \psi_1 = 0.01$$

Peeling algorithm (cont. 2)

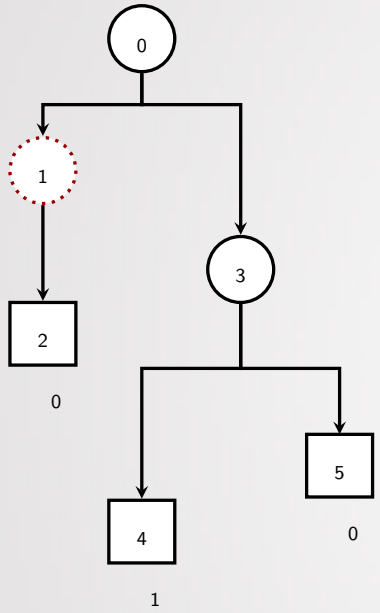
$\psi_0 = 0.05$ $\psi_1 = 0.01$ $\mu_0 = 0.04$ $\mu_1 = 0.01$ $\pi = 0.05$



	State 0	State 1
0		
1		
2	0.9500	0.0100
3		
4	0.0500	0.9900
5	0.9500	0.0100

Peeling algorithm (cont. 2)

$$\psi_0 = 0.05 \qquad \psi_1 = 0.01 \qquad \mu_0 = 0.04 \qquad \mu_1 = 0.01 \qquad \pi = 0.05$$

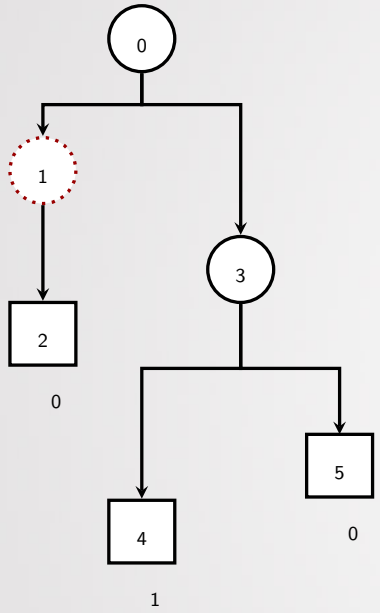


	State 0	State 1
0		
1		
2	0.9500	0.0100
3		
4	0.0500	0.9900
5	0.9500	0.0100

$$\mathsf{L}\left(X_1 = 0 \mid \tilde{D}_1\right) = \Pr\left(X_{obs2} = 0 \mid X_2 = 0\right)(1 - \mu_0) + \Pr\left(X_{obs2} = 0 \mid X_2 = 1\right)\mu_0$$

Peeling algorithm (cont. 2)

$$\psi_0 = 0.05 \qquad \psi_1 = 0.01 \qquad \mu_0 = 0.04 \qquad \mu_1 = 0.01 \qquad \pi = 0.05$$

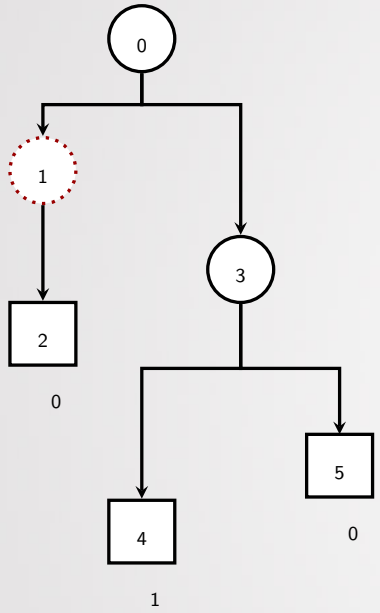


	State 0	State 1
0		
1	0.9124	
2	0.9500	0.0100
3		
4	0.0500	0.9900
5	0.9500	0.0100

$$\mathsf{L}\left(X_1 = 0 \mid \tilde{D}_1\right) = \Pr\left(X_{obs2} = 0 \mid X_2 = 0\right)(1 - \mu_0) + \Pr\left(X_{obs2} = 0 \mid X_2 = 1\right)\mu_0$$
$$= 0.9500 \times 0.96 + 0.0100 \times 0.04 = 0.9124$$

Peeling algorithm (cont. 2)

$$\psi_0 = 0.05 \qquad \psi_1 = 0.01 \qquad \mu_0 = 0.04 \qquad \mu_1 = 0.01 \qquad \pi = 0.05$$



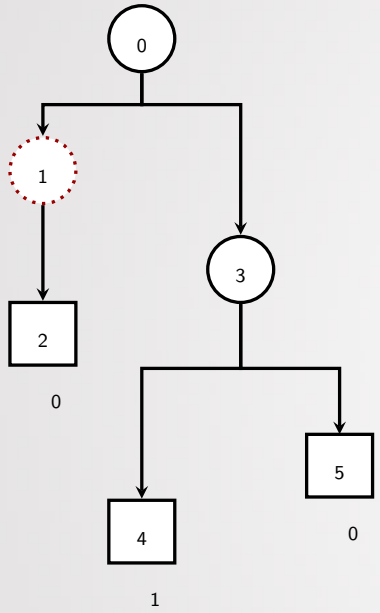
	State 0	State 1
0		
1	0.9124	
2	0.9500	0.0100
3		
4	0.0500	0.9900
5	0.9500	0.0100

$$\mathrm{L}\left(X_1 = 0 \mid \tilde{D}_1\right) = \Pr\left(X_{obs2} = 0 \mid X_2 = 0\right)(1-\mu_0) + \Pr\left(X_{obs2} = 0 \mid X_2 = 1\right)\mu_0$$
$$= 0.9500 \times 0.96 + 0.0100 \times 0.04 = 0.9124$$

$$\mathrm{L}\left(X_1 = 1 \mid \tilde{D}_1\right) = \Pr\left(X_2 = 0 \mid X_{obs2} = 0\right)\mu_1 + \Pr\left(X_2 = 1 \mid X_{obs2} = 0\right)(1-\mu_1)$$

Peeling algorithm (cont. 2)

$$\psi_0 = 0.05 \qquad \psi_1 = 0.01 \qquad \mu_0 = 0.04 \qquad \mu_1 = 0.01 \qquad \pi = 0.05$$



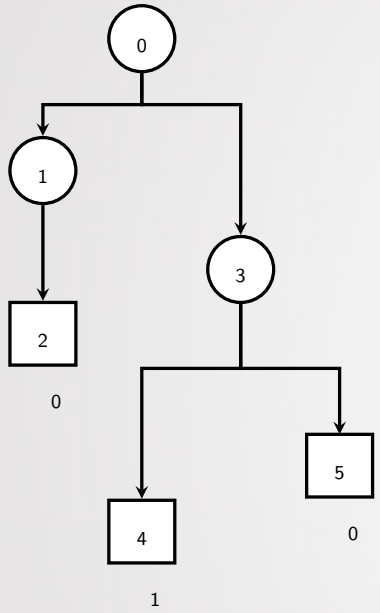
	State 0	State 1
0		
1	0.9124	0.0194
2	0.9500	0.0100
3		
4	0.0500	0.9900
5	0.9500	0.0100

$$\mathrm{L}\left(X_1 = 0 \mid \tilde{D}_1\right) = \Pr\left(X_{obs2} = 0 \mid X_2 = 0\right)(1-\mu_0) + \Pr\left(X_{obs2} = 0 \mid X_2 = 1\right)\mu_0$$
$$= 0.9500 \times 0.96 + 0.0100 \times 0.04 = 0.9124$$

$$\mathrm{L}\left(X_1 = 1 \mid \tilde{D}_1\right) = \Pr\left(X_2 = 0 \mid X_{obs2} = 0\right)\mu_1 + \Pr\left(X_2 = 1 \mid X_{obs2} = 0\right)(1-\mu_1)$$
$$= 0.9500 \times 0.01 + 0.0100 \times 0.99 = 0.0194$$

Peeling algorithm (cont. 3)

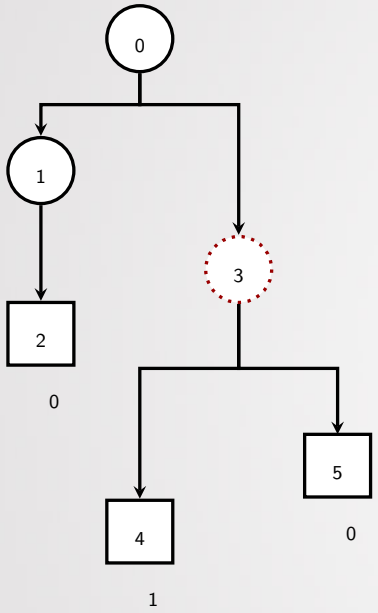
$\psi_0 = 0.05$ $\psi_1 = 0.01$ $\mu_0 = 0.04$ $\mu_1 = 0.01$ $\pi = 0.05$



	State 0	State 1
0		
1	0.9124	0.0194
2	0.9500	0.0100
3		
4	0.0500	0.9900
5	0.9500	0.0100

Peeling algorithm (cont. 3)

$$\psi_0 = 0.05 \qquad \psi_1 = 0.01 \qquad \mu_0 = 0.04 \qquad \mu_1 = 0.01 \qquad \pi = 0.05$$

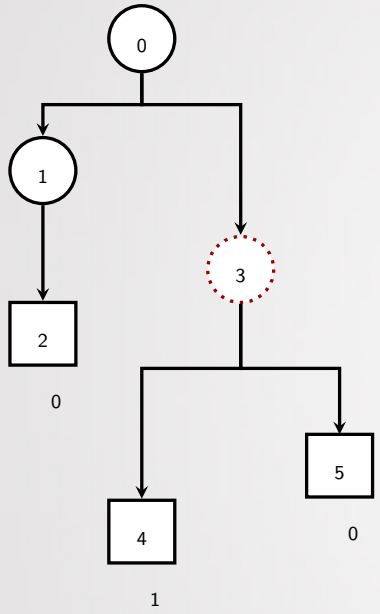


	State 0	State 1
0		
1	0.9124	0.0194
2	0.9500	0.0100
3		
4	0.0500	0.9900
5	0.9500	0.0100

$$\mathrm{L}\left(X_3 = 0 \mid \tilde{D}_3\right) = \prod_{m \in \{4,5\}} \sum_{x_m \in \{0,1\}} \mathrm{L}\left(X_m = x_m \mid \tilde{D}_m\right) \Pr\left(X_m = x_m \mid X_3 = 0\right)$$

Peeling algorithm (cont. 3)

$$\psi_0 = 0.05 \qquad \psi_1 = 0.01 \qquad \mu_0 = 0.04 \qquad \mu_1 = 0.01 \qquad \pi = 0.05$$



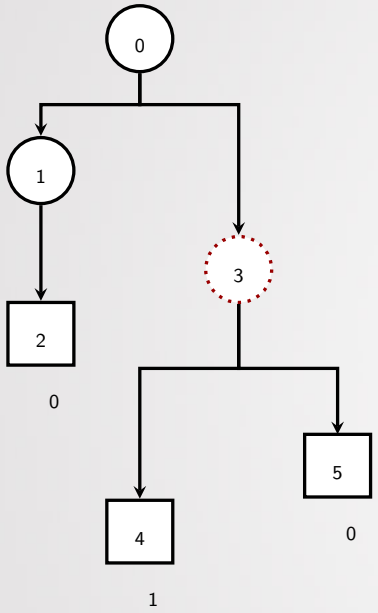
	State 0	State 1
0		
1	0.9124	0.0194
2	0.9500	0.0100
3		
4	0.0500	0.9900
5	0.9500	0.0100

$$\mathrm{L}\left(X_3 = 0 \mid \tilde{D}_3\right) = \prod_{m \in \{4,5\}} \sum_{x_m \in \{0,1\}} \mathrm{L}\left(X_m = x_m \mid \tilde{D}_m\right) \Pr\left(X_m = x_m \mid X_3 = 0\right)$$

$$= \left(0.05(1-\mu_0) + 0.99 \times \mu_0\right) \times \left(0.95(1-\mu_0) + 0.01 \times \mu_0\right)$$

Peeling algorithm (cont. 3)

$$\psi_0 = 0.05 \qquad \psi_1 = 0.01 \qquad \mu_0 = 0.04 \qquad \mu_1 = 0.01 \qquad \pi = 0.05$$



	State 0	State 1
0		
1	0.9124	0.0194
2	0.9500	0.0100
3		
4	0.0500	0.9900
5	0.9500	0.0100

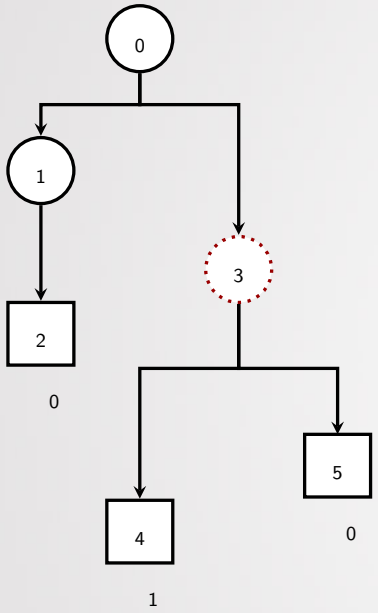
$$\mathrm{L}\left(X_3 = 0 \mid \tilde{D}_3\right) = \prod_{m \in \{4,5\}} \sum_{x_m \in \{0,1\}} \mathrm{L}\left(X_m = x_m \mid \tilde{D}_m\right) \Pr\left(X_m = x_m \mid X_3 = 0\right)$$

$$= \left(0.05(1-\mu_0) + 0.99 \times \mu_0\right) \times \left(0.95(1-\mu_0) + 0.01 \times \mu_0\right)$$

$$= (0.05(1-0.04) + 0.99 \times 0.04) \times (0.95(1-0.04) + 0.01 \times 0.04)$$

Peeling algorithm (cont. 3)

$$\psi_0 = 0.05 \qquad \psi_1 = 0.01 \qquad \mu_0 = 0.04 \qquad \mu_1 = 0.01 \qquad \pi = 0.05$$



	State 0	State 1
0		
1	0.9124	0.0194
2	0.9500	0.0100
3	0.0799	
4	0.0500	0.9900
5	0.9500	0.0100

$$\mathrm{L}\left(X_3 = 0 \mid \tilde{D}_3\right) = \prod_{m \in \{4,5\}} \sum_{x_m \in \{0,1\}} \mathrm{L}\left(X_m = x_m \mid \tilde{D}_m\right) \Pr\left(X_m = x_m \mid X_3 = 0\right)$$

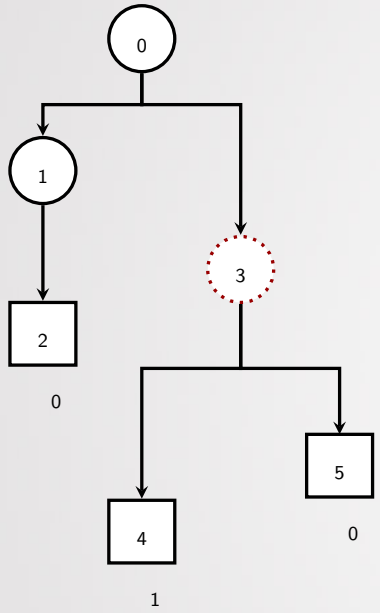
$$= \left(0.05(1-\mu_0) + 0.99 \times \mu_0\right) \times \left(0.95(1-\mu_0) + 0.01 \times \mu_0\right)$$

$$= (0.05(1-0.04) + 0.99 \times 0.04) \times (0.95(1-0.04) + 0.01 \times 0.04)$$

$$= 0.0799$$

Peeling algorithm (cont. 3)

$$\psi_0 = 0.05 \qquad \psi_1 = 0.01 \qquad \mu_0 = 0.04 \qquad \mu_1 = 0.01 \qquad \pi = 0.05$$



	State 0	State 1
0		
1	0.9124	0.0194
2	0.9500	0.0100
3	0.0799	0.0190
4	0.0500	0.9900
5	0.9500	0.0100

$$\mathrm{L}\left(X_3 = 0 \mid \tilde{D}_3\right) = \prod_{m \in \{4,5\}} \sum_{x_m \in \{0,1\}} \mathrm{L}\left(X_m = x_m \mid \tilde{D}_m\right) \Pr\left(X_m = x_m \mid X_3 = 0\right)$$

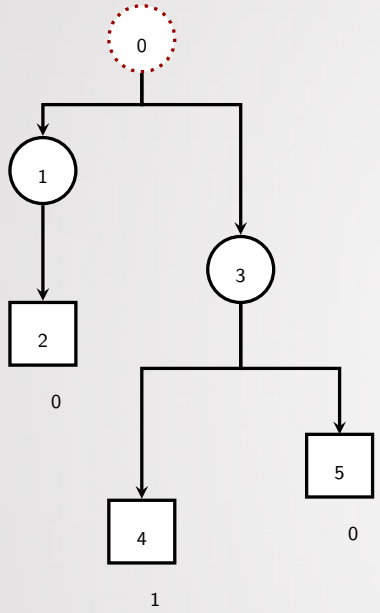
$$= \left(0.05(1-\mu_0) + 0.99 \times \mu_0\right) \times \left(0.95(1-\mu_0) + 0.01 \times \mu_0\right)$$

$$= (0.05(1-0.04) + 0.99 \times 0.04) \times (0.95(1-0.04) + 0.01 \times 0.04)$$

$$= 0.0799$$

Peeling algorithm (cont. 3)

$$\psi_0 = 0.05 \qquad \psi_1 = 0.01 \qquad \mu_0 = 0.04 \qquad \mu_1 = 0.01 \qquad \pi = 0.05$$



	State 0	State 1
0	0.0679	0.0006
1	0.9124	0.0194
2	0.9500	0.0100
3	0.0799	0.0190
4	0.0500	0.9900
5	0.9500	0.0100

$$\mathrm{L}\left(X_3 = 0 \mid \tilde{D}_3\right) = \prod_{m\in\{4,5\}} \sum_{x_m\in\{0,1\}} \mathrm{L}\left(X_m = x_m \mid \tilde{D}_m\right) \Pr\left(X_m = x_m \mid X_3 = 0\right)$$

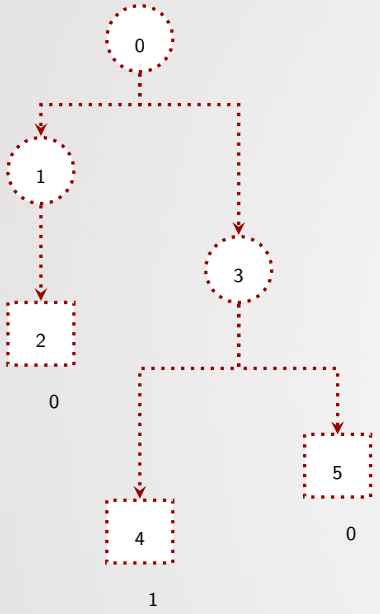
$$= \left(0.05(1-\mu_0) + 0.99 \times \mu_0\right) \times \left(0.95(1-\mu_0) + 0.01 \times \mu_0\right)$$

$$= (0.05(1-0.04) + 0.99 \times 0.04) \times (0.95(1-0.04) + 0.01 \times 0.04)$$

$$= 0.0799$$

Peeling algorithm (cont. 3)

$$\psi_0 = 0.05 \qquad \psi_1 = 0.01 \qquad \mu_0 = 0.04 \qquad \mu_1 = 0.01 \qquad \pi = 0.05$$



	State 0	State 1
0	0.0679	0.0006
1	0.9124	0.0194
2	0.9500	0.0100
3	0.0799	0.0190
4	0.0500	0.9900
5	0.9500	0.0100

$$\mathsf{L}\left(X_3 = 0 \mid \tilde{D}_3\right) = \prod_{m \in \{4,5\}} \sum_{x_m \in \{0,1\}} \mathsf{L}\left(X_m = x_m \mid \tilde{D}_m\right) \Pr\left(X_m = x_m \mid X_3 = 0\right)$$

$$= \left(0.05(1-\mu_0) + 0.99 \times \mu_0\right) \times \left(0.95(1-\mu_0) + 0.01 \times \mu_0\right)$$

$$= (0.05(1 - 0.04) + 0.99 \times 0.04) \times (0.95(1 - 0.04) + 0.01 \times 0.04)$$

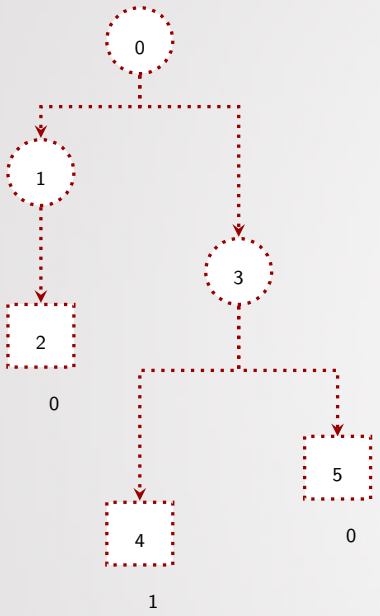
$$= 0.0799$$

Finally, the likelihood of this tree is:

$$\mathsf{L}\left(\psi, \mu, \Pi \mid \tilde{D}\right) = (1-\pi)\mathsf{L}\left(X_0 = 0 \mid \tilde{D}_0\right) + \pi \mathsf{L}\left(X_0 = 1 \mid \tilde{D}_0\right)$$

Peeling algorithm (cont. 3)

$$\psi_0 = 0.05 \qquad \psi_1 = 0.01 \qquad \mu_0 = 0.04 \qquad \mu_1 = 0.01 \qquad \pi = 0.05$$



	State 0	State 1
0	0.0679	0.0006
1	0.9124	0.0194
2	0.9500	0.0100
3	0.0799	0.0190
4	0.0500	0.9900
5	0.9500	0.0100

$$\mathsf{L}\left(X_3 = 0 \mid \tilde{D}_3\right) = \prod_{m\in\{4,5\}} \sum_{x_m\in\{0,1\}} \mathsf{L}\left(X_m = x_m \mid \tilde{D}_m\right) \Pr\left(X_m = x_m \mid X_3 = 0\right)$$

$$= \left(0.05(1-\mu_0) + 0.99 \times \mu_0\right) \times \left(0.95(1-\mu_0) + 0.01 \times \mu_0\right)$$

$$= (0.05(1-0.04) + 0.99 \times 0.04) \times (0.95(1-0.04) + 0.01 \times 0.04)$$

$$= 0.0799$$

Finally, the likelihood of this tree is:

$$\mathsf{L}\left(\psi, \mu, \Pi \mid \tilde{D}\right) = (1-\pi)\mathsf{L}\left(X_0 = 0 \mid \tilde{D}_0\right) + \pi\mathsf{L}\left(X_0 = 1 \mid \tilde{D}_0\right)$$

$$= (1-0.05) \times 0.0679 + 0.05 \times 5.5619 \times 10^{-4} = 0.0646$$

Agenda

Yet another MCMC package

You may be wondering why, well:

1. Allows running multiple chains simultaneously (parallel)
2. Overall faster than other Metrop MCMC algorithms (from our experience)
3. Planning to include other types of kernels (the Handbook of MCMC)
4. Implements reflective boundaries random-walk kernel

Example: MCMC

```r
# Loading the packages
library(amcmc)
library(coda) # coda: Output Analysis and Diagnostics for MCMC

# Defining the ll function (data was already defined)
ll <- function(x, D) {
  x <- log(dnorm(D, x[1], x[2]))
  sum(x)
}

ans <- MCMC(
  # Ll function and the starting parameters
  ll, c(mu=1, sigma=1),
  # How many steps, thinning, and burn-in
  nbatch = 1e4, thin=10, burnin = 1e3,
  # Kernel parameters
  scale = .1, ub = 10, lb = c(-10, 0),
  # How many parallel chains
  nchains = 4,
  # Further arguments passed to ll
  D=D
  )
```

Example: MCMC (cont. 1)

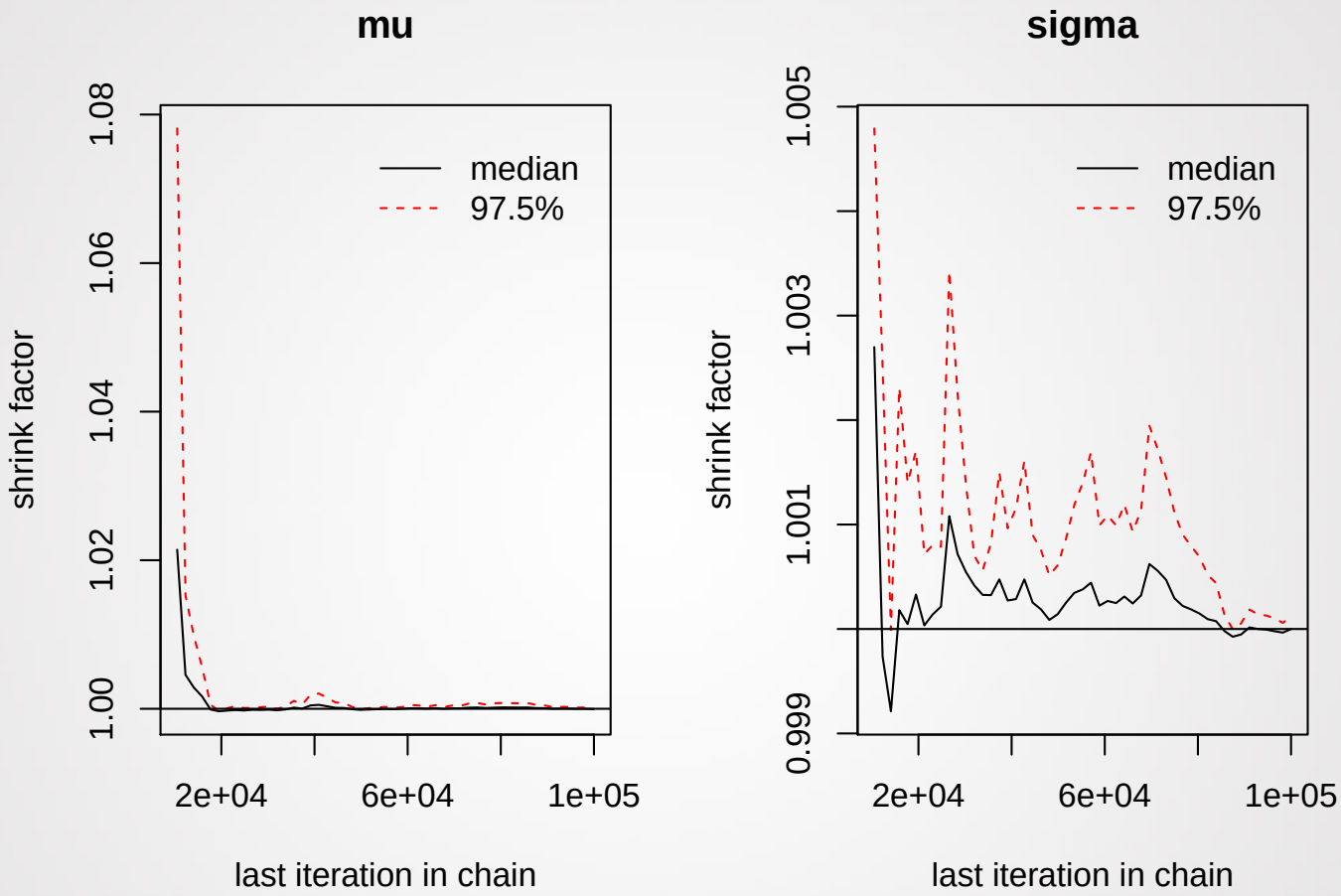


Figure 1: Gelman diagnostic for convergence. The closer to 1, the better the convergence. Rule of thumb: A chain has a reasonable convergence if it has a Potential Scale Reduction Factor (PSRF) below 1.15.

Example: MCMC (cont. 2)

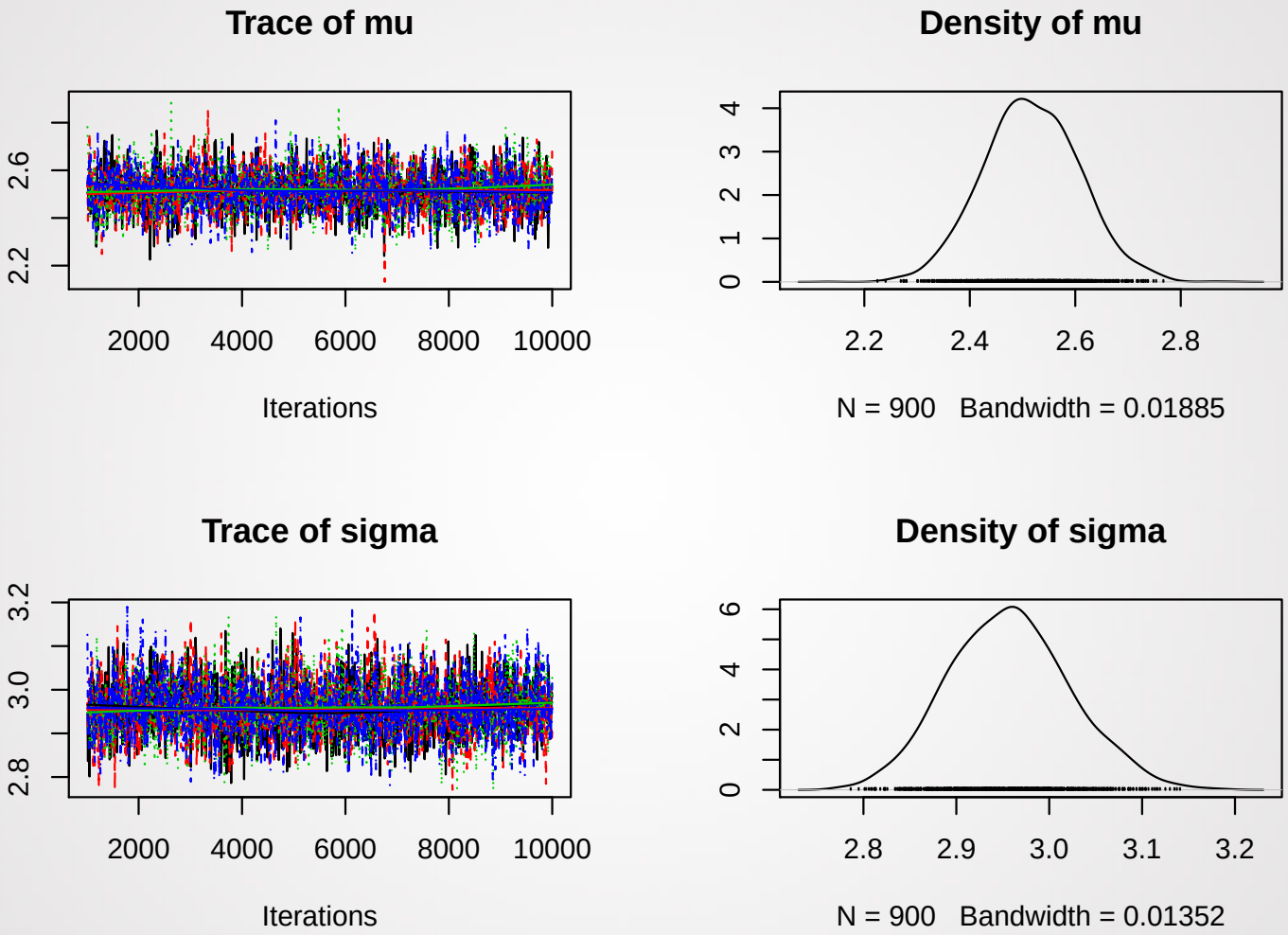


Figure 2: Posterior distribution

Agenda

aphylo in a nutshell

- Provides a representation of *annotated* partially ordered trees.

aphylo in a nutshell

- Provides a representation of *annotated* partially ordered trees.
- Interacts with the ape package (most used Phylogenetics R package with ~25K downloads/month)

aphylo in a nutshell

- Provides a representation of *annotated* partially ordered trees.
- Interacts with the ape package (most used Phylogenetics R package with ~25K downloads/month)
- Implements the loglikelihood calculation of our model (with C++ under-the-hood).

aphylo: Simulating Trees

```
set.seed(80)
tree <- sim_tree(5)
tree
```

```
##
## A PARTIALLY ORDERED PHYLOGENETIC TREE
##
##   # Internal nodes: 4
##   # Leaf nodes    : 5
##
##   Leaf nodes labels:
##     4, 5, 6, 7, 8.
##
##   Internal nodes labels:
##     0, 1, 2, 3.
```

```
atree <- sim_annotated_tree(
  tree = tree, P = 2,
  psi  = c(.05, .05),
  mu   = c(.2, .1),
  Pi   = .01
  )

atree
```

```
##
## A PARTIALLY ORDERED PHYLOGENETIC TREE
##
##   # Internal nodes: 4
##   # Leaf nodes    : 5
##
##   Leaf nodes labels:
##     4, 5, 6, 7, 8.
##
##   Internal nodes labels:
##     0, 1, 2, 3.
##
## ANNOTATIONS:
##      fun0000 fun0001
```

aphylo: Visualizing annotated data

```
plot(atree)
```

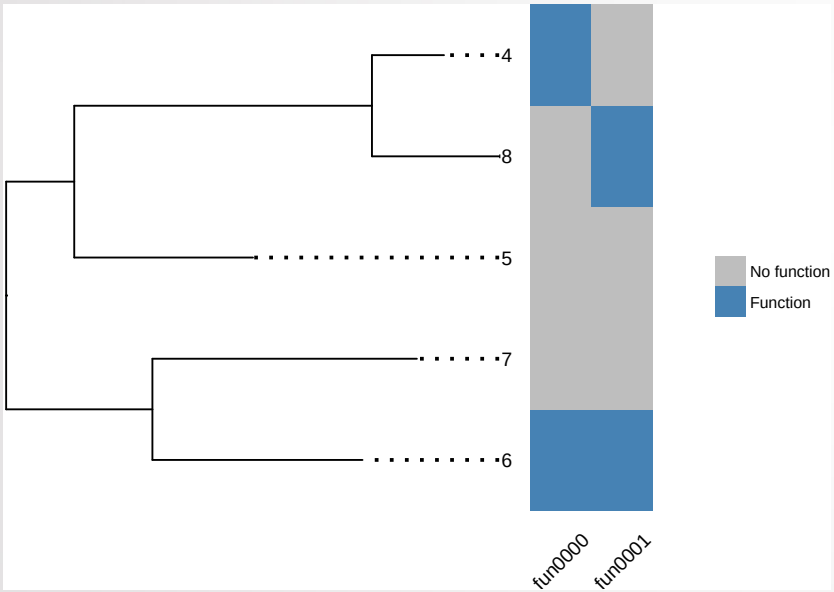


Figure 3: Visualization of annotations and tree structure.

```
plot_LogLike(atree)
```

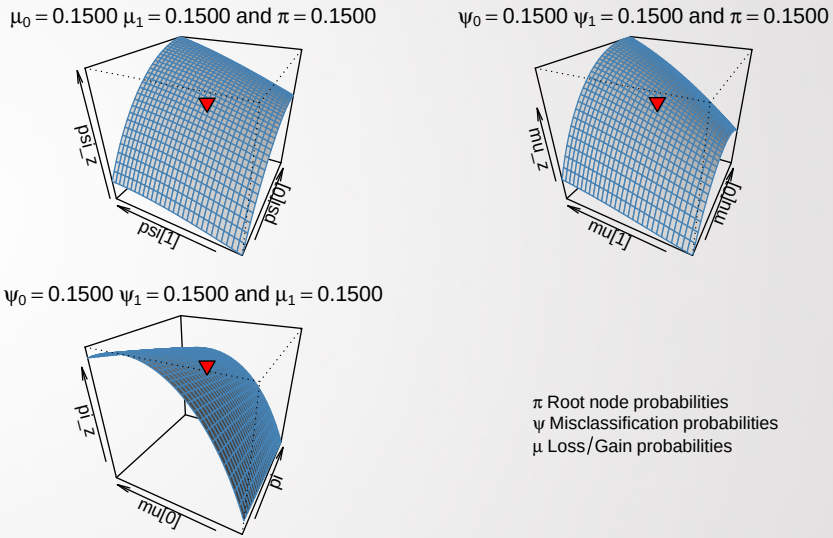


Figure 4: LogLikelihood surface of the simulated data

aphylo: Tree peeling

- The peeling algorithm requires visiting all nodes in a tree.

aphylo: Tree peeling

- The peeling algorithm requires visiting all nodes in a tree.
- The fact is, we don't need to go through branches with no annotations, as these are uninformative.

aphylo: Tree peeling

- The peeling algorithm requires visiting all nodes in a tree.
- The fact is, we don't need to go through branches with no annotations, as these are uninformative. So we can prune them, e.g.:

aphylo: Tree peeling

- The peeling algorithm requires visiting all nodes in a tree.
- The fact is, we don't need to go through branches with no annotations, as these are uninformative. So we can prune them, e.g.:

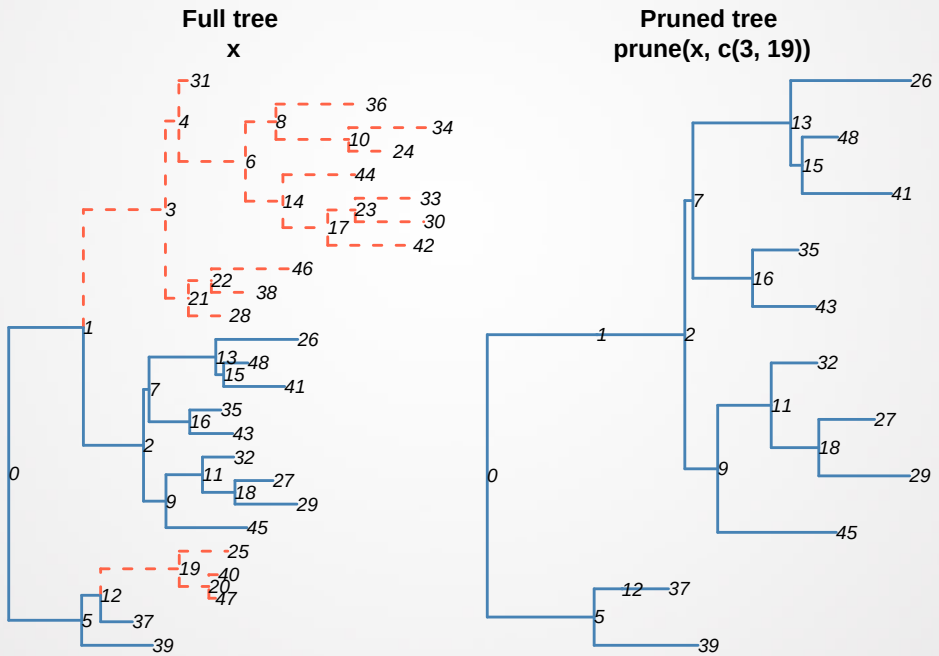


Figure 5: Peeling trees. In the original none of the leaf nodes under 3 and 9 have annotations. After peeling those branches, we go from having 49 nodes, to have 21

aphylo: Reading PantherDB data

```
# Reading the data
path <- system.file("tree.tree", package="aphylo")
dat <- read_panther(path)

# The tree
dat$tree
```

```
##
## Phylogenetic tree with 145 tips and 107 internal nodes.
##
## Tip labels:
##  AN5:MONBE|Gene=28576|UniProtKB=A9V8K6, AN7:SCHPO|PomBase=SPAC25B8.12c|UniProtKB=Q9UTA6
## Node labels:
##  AN0, AN1, AN2, AN3, AN4, AN6, ...
##
## Rooted; includes branch lengths.
```

aphylo: Reading PantherDB data (cont.)

```
# Extra annotations
head(dat$internal_nodes_annotations)
```

```
##     branch_length type          ancestor duplication
## AN0            NA    S              LUCA       FALSE
## AN1         0.057    S Archaea-Eukaryota       FALSE
## AN2         0.244    S         Eukaryota       FALSE
## AN3         0.436    S          Unikonts       FALSE
## AN4         0.417    S      Opisthokonts       FALSE
## AN6         0.684    D              <NA>        TRUE
```

aphylo: Predictions of the model

- Posterior probability:

$$\Pr\left(x_n = 1 \,\middle|\, \tilde{D}\right) = \frac{\Pr\left(\tilde{D} \,\middle|\, x_n = 1\right)}{\Pr\left(\tilde{D} \,\middle|\, x_n = 1\right) + \Pr\left(\tilde{D} \,\middle|\, x_n = 0\right) \frac{\left(1 - \Pr(x_n = 1)\right)}{\Pr(x_n = 1)}} \quad (1)$$

aphylo: Predictions of the model

- Posterior probability:

$$\Pr\left(x_n = 1 \,\middle|\, \tilde{D}\right) = \frac{\Pr\left(\tilde{D} \,\middle|\, x_n = 1\right)}{\Pr\left(\tilde{D} \,\middle|\, x_n = 1\right) + \Pr\left(\tilde{D} \,\middle|\, x_n = 0\right) \frac{\left(1 - \Pr(x_n = 1)\right)}{\Pr(x_n = 1)}} \quad (1)$$

Where

$$\Pr\left(x_n = 1\right) = \pi \Pr\left(x_n = 1 \mid x_0 = 1\right) + (1 - \pi) \Pr\left(x_n = 1 \mid x_0 = 0\right)$$

aphylo: Predictions of the model

- Posterior probability:

$$\Pr\left(x_n = 1 \,\middle|\, \tilde{D}\right) = \frac{\Pr\left(\tilde{D} \,\middle|\, x_n = 1\right)}{\Pr\left(\tilde{D} \,\middle|\, x_n = 1\right) + \Pr\left(\tilde{D} \,\middle|\, x_n = 0\right) \frac{\left(1 - \Pr(x_n = 1)\right)}{\Pr(x_n = 1)}} \quad (1)$$

Where

$$\Pr\left(x_n = 1\right) = \pi \Pr\left(x_n = 1 \mid x_0 = 1\right) + (1 - \pi) \Pr\left(x_n = 1 \mid x_0 = 0\right)$$

And

$$\begin{bmatrix} \Pr\left(x_n = 0 \mid x_0 = 0\right) & \Pr\left(x_n = 1 \mid x_0 = 0\right) \\ \Pr\left(x_n = 0 \mid x_0 = 1\right) & \Pr\left(x_n = 1 \mid x_0 = 1\right) \end{bmatrix} \approx \begin{bmatrix} 1 - \hat{\mu}_0 & \hat{\mu}_0 \\ \hat{\mu}_1 & 1 - \hat{\mu}_1 \end{bmatrix}^{dist_{0n}}$$

aphylo: Predictions of the model (cont.)

Gene	Posterior Prob
AN208:THEMA|EnsemblGenome=TM_0651|UniProtKB=Q9WZB9	0.10
AN22:PLAF7|EnsemblGenome=PFL1270w|UniProtKB=Q8I5F4	0.94
AN161:STRR6|EnsemblGenome=spr0263|UniProtKB=Q8DR95	0.22
AN168:BACSU|EnsemblGenome=BSU11140|UniProtKB=P70947	0.23
AN166:LISMO|Gene=CAD00341|UniProtKB=Q8Y515	0.60
AN192:BACSU|EnsemblGenome=BSU39550|UniProtKB=P54947	0.95

Table 2: Predicted probabilities for a subset of leafs of a phylogenetic tree using the `predict()` function after estimating the model parameters. The function analized was simulated on a phylogenetic tree from PantherDB.

aphylo: How good is our prediction

- Quality of the prediction

aphylo: How good is our prediction

- Quality of the prediction

$$\delta\left(X_{obs\,H}, \hat{X}_H\right) = \sum_{h,u \in H} \left[(x_{obs\,h} - \hat{x}_h)^2 (x_{obs\,u} - \hat{x}_u)^2\right]^{1/2} w_{hu}$$

aphylo: How good is our prediction

- Quality of the prediction

$$\delta\left(X_{obs\,H}, \hat{X}_H\right) = \sum_{h,u \in H} \left[(x_{obs\,h} - \hat{x}_h)^2 (x_{obs\,u} - \hat{x}_u)^2\right]^{1/2} w_{hu}$$

Which, assuming $\hat{x} \sim \text{Bernoulli}(\alpha)$, has expected value

$$\mathrm{E}(\delta) = \sum_{h,u \in H} w_{hu} \sum_{\hat{x}_h, \hat{x}_u \in \{0,1\}} \Pr(\hat{x}_h)\Pr(\hat{x}_u) \left[(x_{obs\,h} - \hat{x}_h)^2 (x_{obs\,u} - \hat{x}_u)^2\right]^{1/2}$$

```
prediction_score(ans)
```

```
## PREDICTION SCORE: ANNOTATED PHYLOGENETIC TREE
## Observed : 0.06
## Random   : 0.25
## ------------------------------------------------------------------------
## Values standarized to range between 0 and 1, 0 being best.
```

aphylo: How good is our prediction (cont. 1)

```
plot(prediction_score(ans), main="")
```

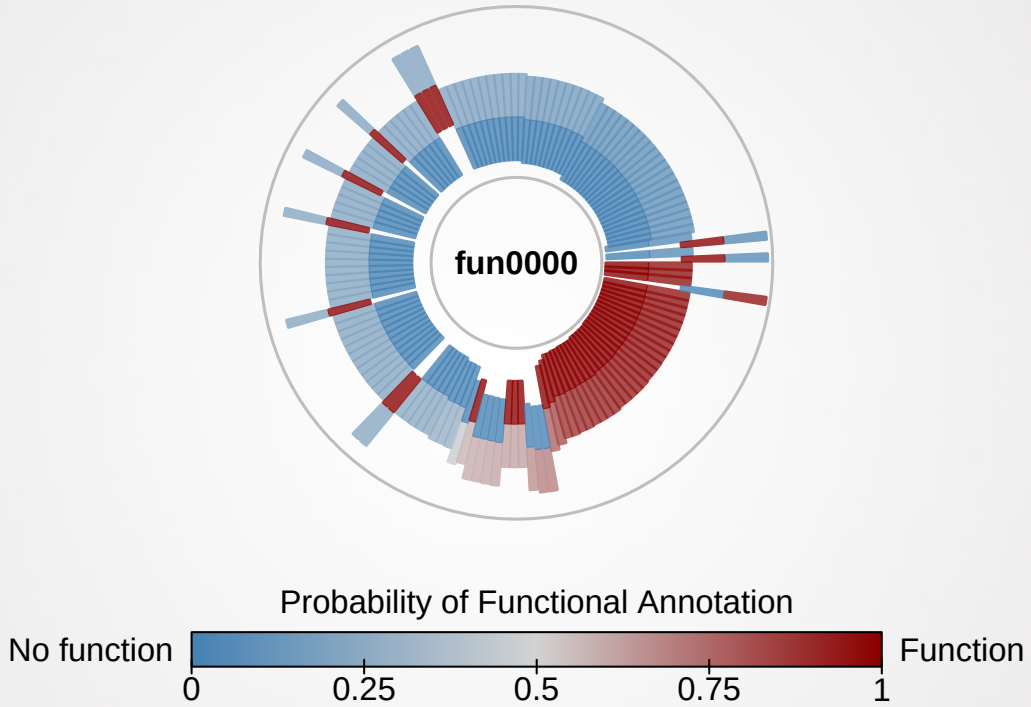


Figure 6: Predicted versus Observed values. Each slice of the pie represents a gene, the outer half of a slice is the predicted value, while the inner half is the observed value. Good predictions will coincide in color and show the slice closer to the center of the plot.

Agenda

A simulation study

Setup

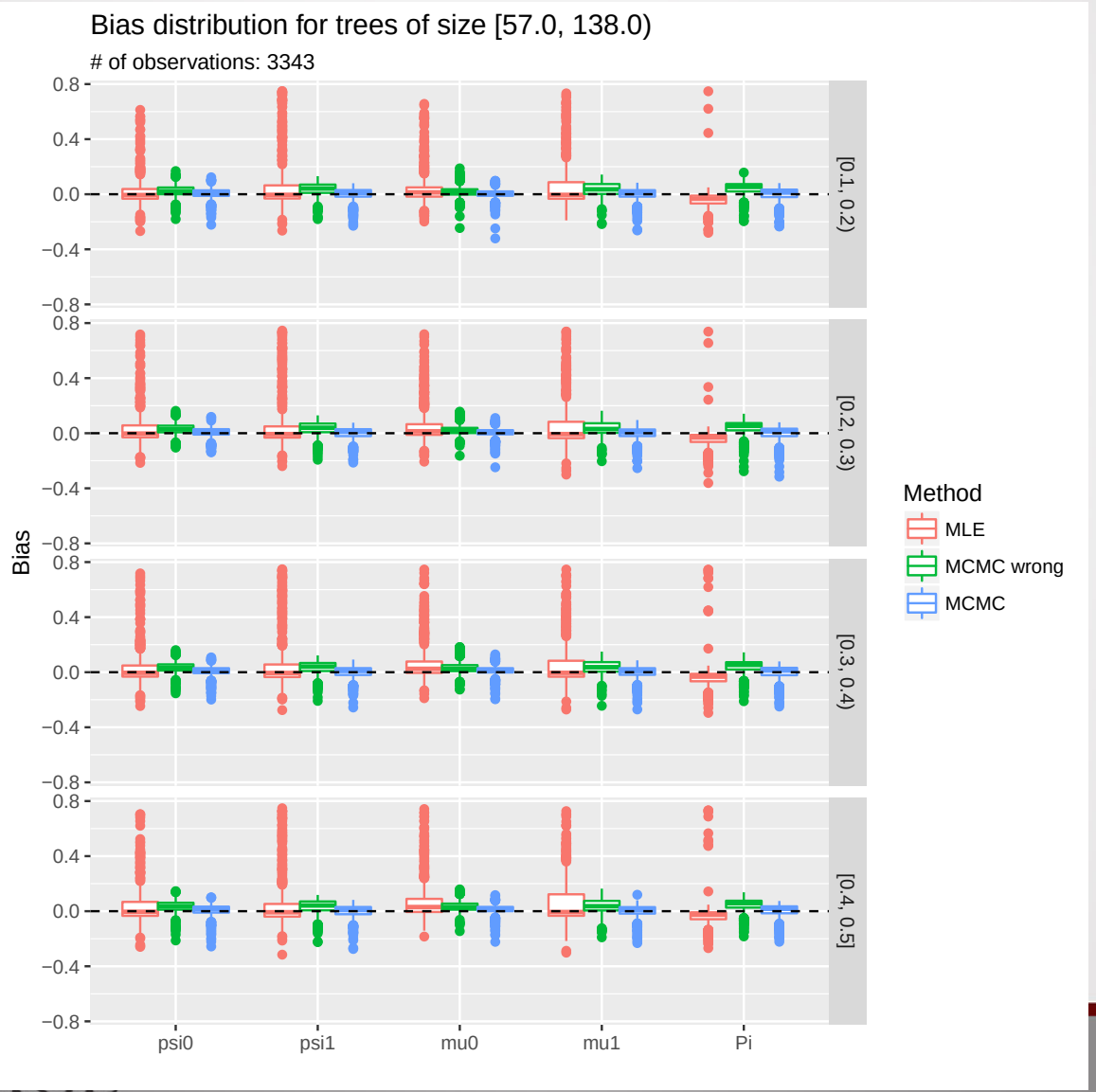
- Simulation study using ~13,000 families from PantherDB
- Using a Beta 1/20 prior, we simulated annotations:
  - Draw a set of the parameters $\{\psi_0, \psi_1, \mu_0, \mu_1, \pi\}$,
  - Simulated annotations using our model's Data Generating Process,
  - Randomly removed $p \in [.1, .5]$ proportion of annotations.
- With that data, we did parameter estimation and computed prediction scores using
  - MLE
  - MCMC with the right prior (Beta 1/20), and
  - MCMC with the wrong prior (Beta 1/10, twice the mean as the right prior).

  Both MCMC algorithms ran for 5×10^5 iterations, burn-in of 1×10^4, thinning of 100, and 5 chains.

more details

A simulation study

Bias



A simulation study

Prediction scores

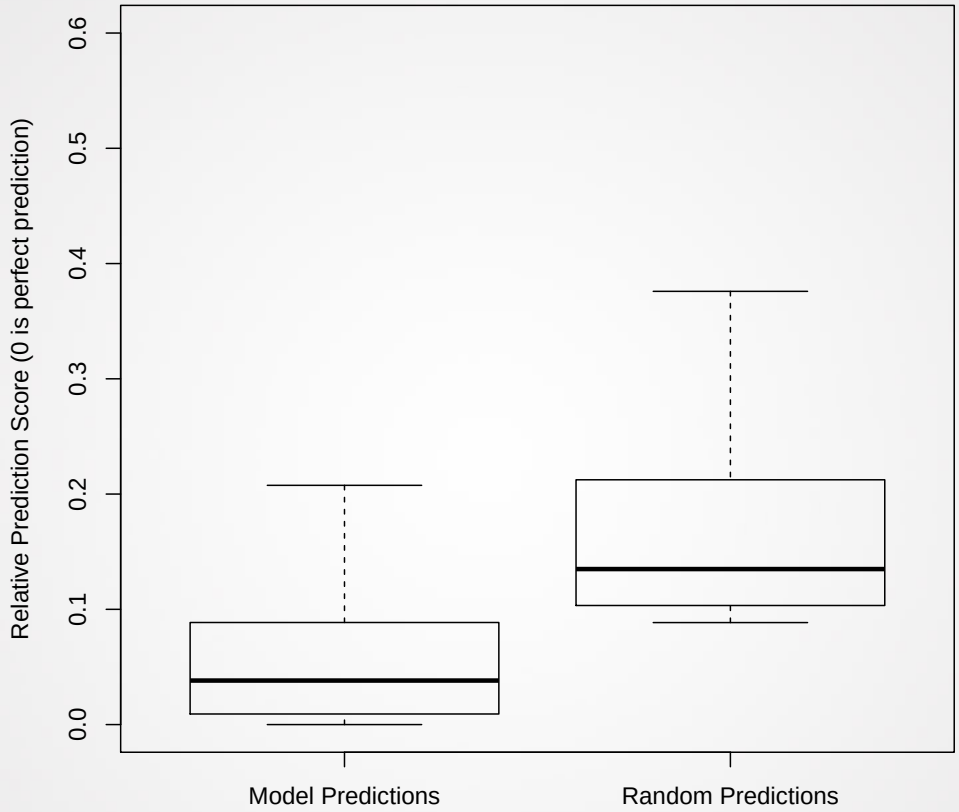


Figure 7: Distribution of prediction scores. The random prediction scores were computed analytically with parameter $p = 0.3$ (as resulting from the DGP).

Concluding Remarks

- A parsimonious model of gene functions: easy to apply on a large scale (we already ran some simulations using all 13,000 trees from PantherDB... and it took us less than 1 week with 10 processors only).

Concluding Remarks

- A parsimonious model of gene functions: easy to apply on a large scale (we already ran some simulations using all 13,000 trees from PantherDB... and it took us less than 1 week with 10 processors only).
- Already implemented, we are currently in the stage of writing the paper and setting up the simulation study.

Concluding Remarks

- A parsimonious model of gene functions: easy to apply on a large scale (we already ran some simulations using all 13,000 trees from PantherDB... and it took us less than 1 week with 10 processors only).
- Already implemented, we are currently in the stage of writing the paper and setting up the simulation study.
- For the next steps, we are evaluating whether to include or how to include:

Concluding Remarks

- A parsimonious model of gene functions: easy to apply on a large scale (we already ran some simulations using all 13,000 trees from PantherDB... and it took us less than 1 week with 10 processors only).
- Already implemented, we are currently in the stage of writing the paper and setting up the simulation study.
- For the next steps, we are evaluating whether to include or how to include:
  - Type of node: speciation, duplication, horizontal transfer.
  - Branch lengths
  - Correlation structure between functions
  - Using Taxon Constraints to improve predictions
  - Hierarchical model: Use fully annotated trees by curators as prior information.

Thank you!

Project 2: Augmenting functional information about human genes using probabilistic phylogenetic modeling

George G. Vega Yon
vegayon@usc.edu
Duncan Thomas Paul D. Thomas Paul Marjoram Huaiyu Mi John Morrison

Department of Preventive Medicine
University of Southern California

October 27, 2017

Formal definitions

go back

1. Phylogenetic tree: In our case, we talk about partially ordered phylogenetic tree, in particular, $\Lambda \equiv (N, E)$ is a tuple of nodes N, and edges

$$E \equiv \{(n, m) \in N \times N : n \mapsto m, n < m\}$$

2. Offspring of n: $O(n) \equiv \{m \in N : (n, m) \in E, n \in N\}$
3. Parent node of m: $r(m) \equiv \{n \in N : (n, m) \in E, m \in N\}$
4. Leaf nodes: $L(\Lambda) \equiv \{m \in N : O(m) = \{\emptyset\}\}$
5. Annotations: Given P functions, $X \equiv \{x_n \in \{0, 1\}^P : n \in L(\Lambda)\}$
6. Annotated Phylogenetic Tree $D \equiv (\Lambda, X)$
7. Observed Annotated Annotations $X_{obs} = \{x_{obs\,l}\}_{l \in L(\Lambda)}$,
8. Experimentally Annotated Phylogenetic Tree $\tilde{D} \equiv (\Lambda, X_{obs})$

Leaf node probabilities

go back

- The probability of the leaf nodes having annotations x_l conditional on the observed annotation is

$$\Pr(X_{obs\,l} = x_{obs\,l} \mid X_l = x_l) = \begin{cases} \psi & \text{if } x_{obs\,l} \neq x_l \\ 1 - \psi & \text{otherwise} \end{cases} \tag{2}$$

Where ψ can be either ψ_0 (mislabelling a zero), or ψ_1 (mislabelling a one).

Internal node probabilities

go back

- In the case of the internal nodes, the probability of a given state is defined in terms of the gain/loss probabilities

$$\Pr\left(X_n = x_l \,\middle|\, X_{r(n)} = x_{r(n)}\right) = \begin{cases} \mu & \text{if } x_n \neq x_{r(n)} \\ 1-\mu & \text{otherwise} \end{cases}$$

  Where μ can be either μ_0 (gain), or μ_1 (loss).

- Assuming independence accross offspring, we can write

$$\mathrm{L}\left(X_n = x_n \,\middle|\, \tilde{D}_n\right) = \prod_{m \in O(n)} \sum_{x_m \in \{0,1\}} \mathrm{L}\left(X_m = x_m \,\middle|\, \tilde{D}_m\right) \Pr\left(X_m = x_m \mid X_n = x_n\right) \quad (3)$$

  Notice that if m is a leaf node, then $\mathrm{L}\left(X_m = x_m \,\middle|\, \tilde{D}_m\right) = \Pr\left(X_m = x_m \mid X_{obs\,m} = x_{obs\,m}\right)$.

Likelihood of the tree

go back

- Once the computation reaches the root node, $n = 0$, equations (2) and (3): Allow us writing the likelihood of the entire tree

$$\mathrm{L}\left(\psi, \mu, \pi \middle| \tilde{D}\right) = \pi \mathrm{L}\left(X_0 = 1 \,\middle|\, \tilde{D}_0\right) + (1-\pi)\mathrm{L}\left(X_0 = 0 \,\middle|\, \tilde{D}_0\right) \quad (4)$$

A simulation study

Convergence

go back

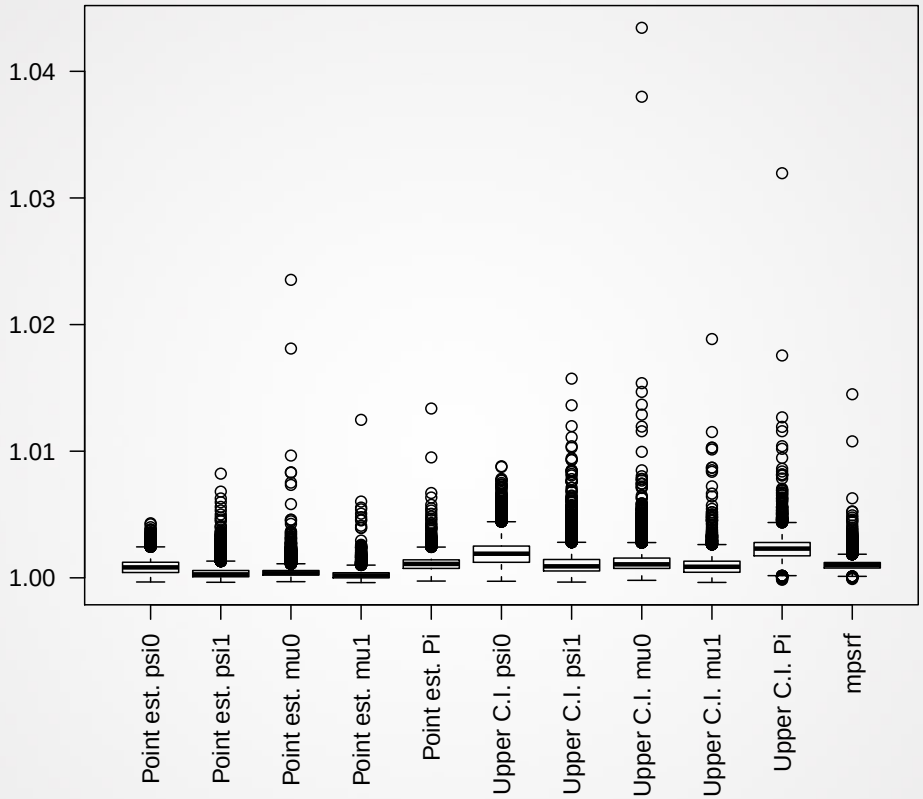


Figure 8: Gelman diagnostic for convergence. The closer to 1, the better the convergence. Rule of thumb: A chain has a reasonable convergence if it has a Potential Scale Reduction Factor (PSRF) below 1.15.