# Process modeling day 1 slides

Renata Diaz

# Sections

# Theory and motivation of process modeling

# Theory and motivation of process modeling

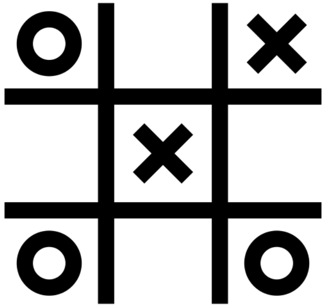
1. What do we *mean* by process modeling, anyway?
2. What are the *applications* of process modeling for ecological and evolutionary dynamics?
3. What are the *limitations* of a process modeling approach?

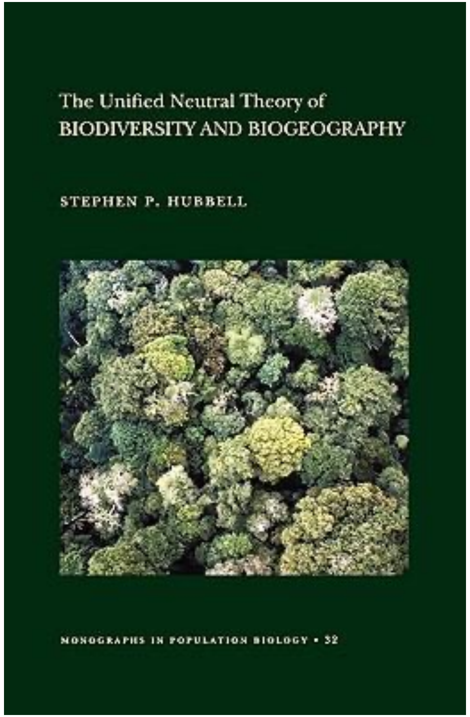
# What do we *mean* by process modeling, anyway?

- Have you ever worked with or encountered a process model?

# What do we *mean* by process modeling, anyway?

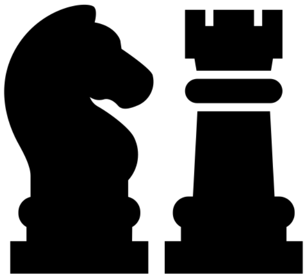
- Process models are **games…**
  - Scenarios play out according to rules
  - Outcomes depend on the rules + chance

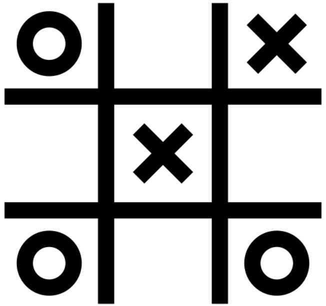




# What do we *mean* by process modeling, anyway?

- Games may be simple or complex
  - (Even simple games can be complex!)
  - Not necessarily deterministic
  - Not necessarily solvable analytically





# What are the applications of process models for eco-evo?

- Nearly unlimited flexibility for exploring processes involving…

stochasticity

large temporal/spatial scales

context dependence

feedbacks

multiple levels of organization

complexity

# What are the applications of process models for eco-evo?

- Use case: hypothesis exploration

How would I expect X to affect Y?

# What are the applications of process models for eco-evo?

- Use case: null models

How would I expect my system to look,
at random?

# What are the applications of process models for eco-evo?

- Use case: large swaths of time or space

How will this system look in 1000 years, under different scenarios?

# What are the applications of process models for eco-evo?

- Use case: explaining empirical data

What generative processes are (not) consistent with empirical observations?

# What are the applications of process models for eco-evo?

- Can you think of an application for a process model in your area of interest?

# What are the *limitations* of a process modeling approach?

# What are the *limitations* of a process modeling approach?

read the rules!!!

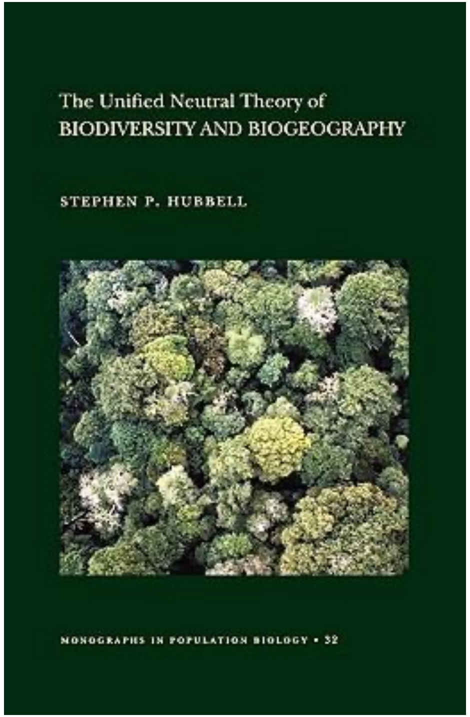
computationally expensive

pattern != process

"model identifiability"

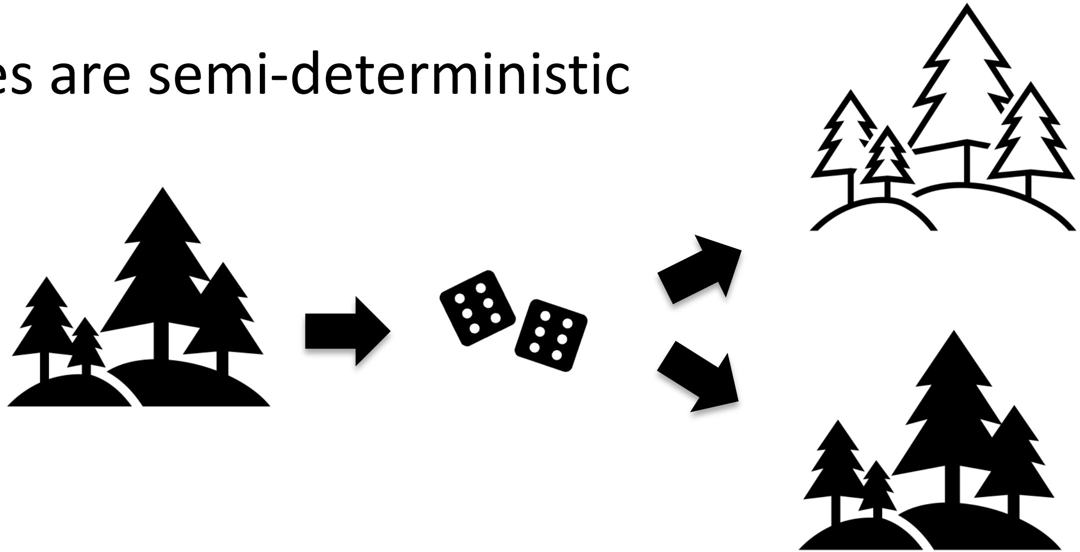
# An example: Hubbell's Neutral Theory

1. How do we fit UNTB into a process model framework?
2. What are the rules and outcomes of UNTB?
3. Let's play the game!

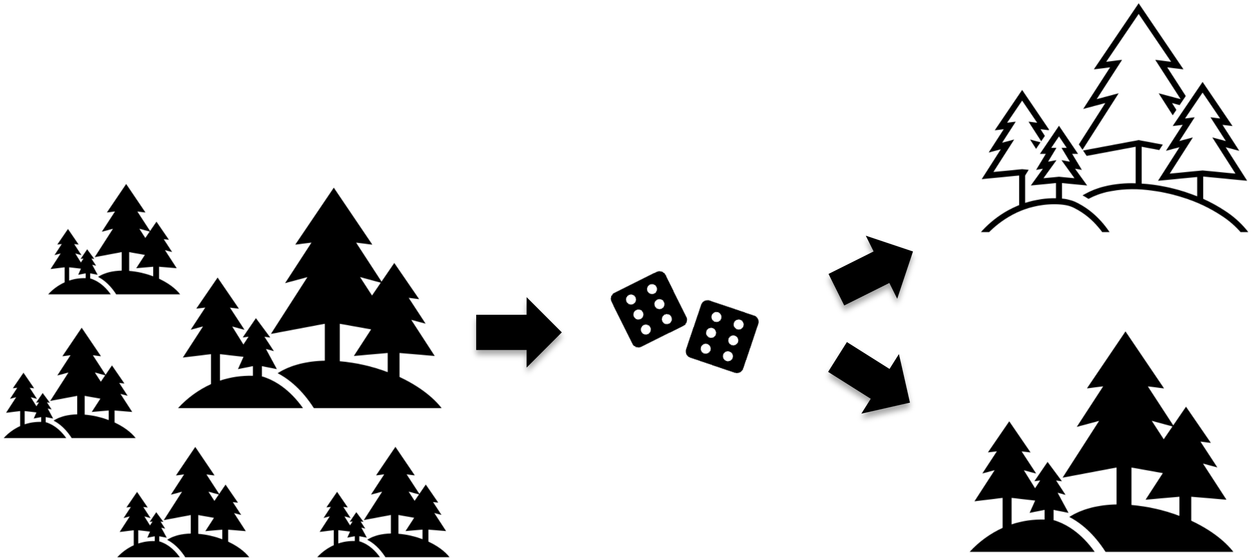


# How do we fit UNTB into a process model framework?

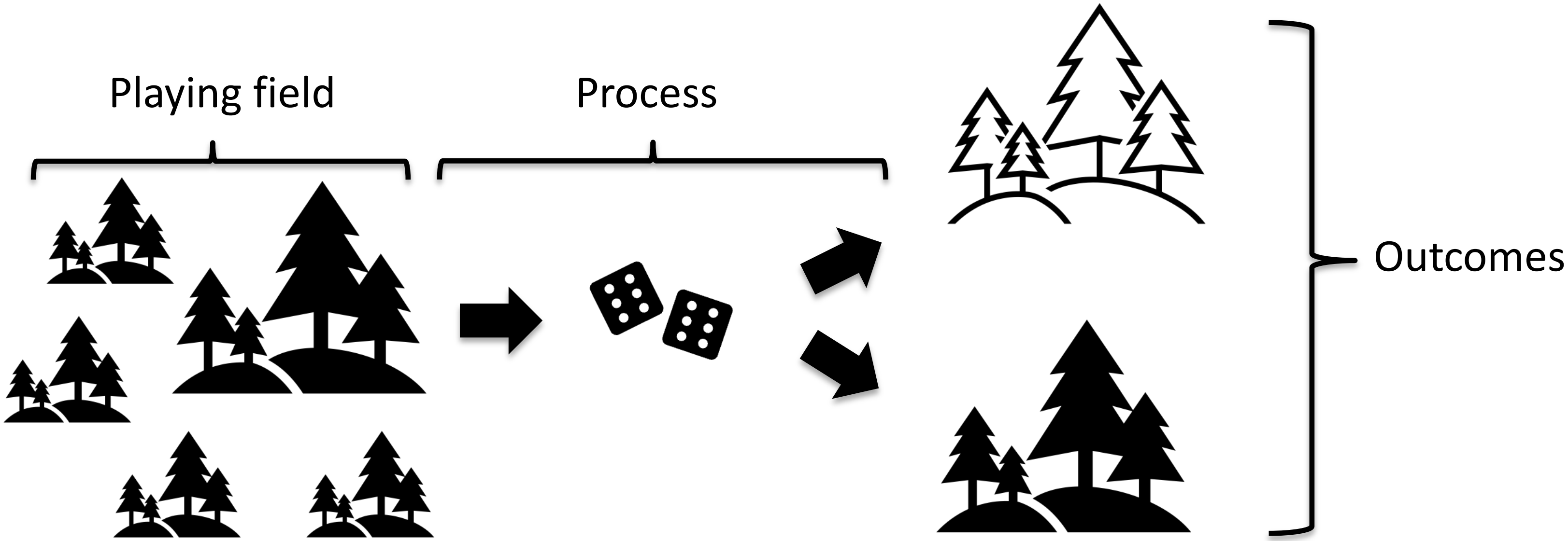
- Entities: Ecological communities made of individuals
- Individuals die, give birth, immigrate, and speciate according to rules
- Model outcomes are semi-deterministic



# What are the rules and outcomes of neutral theory?



# What are the rules and outcomes of neutral theory?



# The playing field

Metacommunity

Local community

All-time species list

# The process

Metacommunity

All-time species list

Local community

Each time step, an individual from the local community dies.

# The process

Metacommunity

All-time species list

Local community

Each time step, an individual from the local community dies.

They are replaced via either a **local birth** or **immigration** from the metacommunity.

# The process

Metacommunity

All-time species list

Local community

Each time step, an individual from the local community dies.

They are replaced via either a **local birth** or **immigration** from the metacommunity.

Sometimes, a speciation event occurs and a new species is added.

# The process

Metacommunity

All-time species list

Local community

Each time step, an individual from the local community dies.

They are replaced via either a **local birth** or **immigration** from the metacommunity.

Sometimes, a speciation event occurs and a new species is added.

This repeats.

# The parameters

Metacommunity

All-time species list

Local community

*Jm*: The number of individuals in the metacommunity
*Sm*: The number of species in the metacommunity
*J*: The number of individuals in the local community

*m*: The probability that an immigration event occurs

*v*: The probability that a speciation event occurs

# The process

Metacommunity (size = *Jm*)

 $\frac{1}{J}$

An individual is chosen to die.

All-time species list

Local community (size = *J*)

# The process

Metacommunity (size = *Jm*)

$\frac{1}{J}$

An individual is chosen to die.

All-time species list

Local community (size = *J*)

# The process

Metacommunity (size = *Jm*)

All-time species list

Local community (size = *J*)

$\frac{1}{J}$

An individual is chosen to die.

$\longrightarrow$

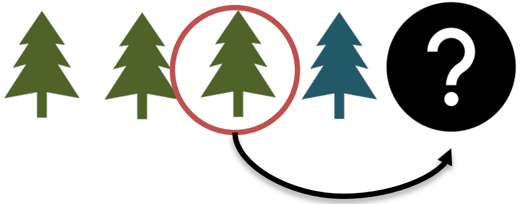
A **birth** or **immigration** event occurs

# The process

Metacommunity (size = *Jm*)

All-time species list

Local community (size = *J*)



$\frac{1}{J}$

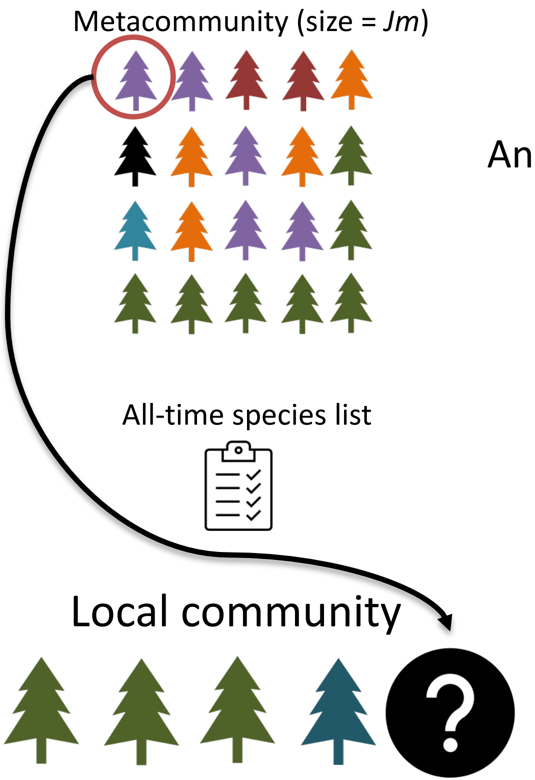
An individual is chosen to die.

→

A **birth** or **immigration** event occurs

If **birth**, a parent is chosen from the **local** community.

# The process




$\frac{1}{J}$

An individual is chosen to die.

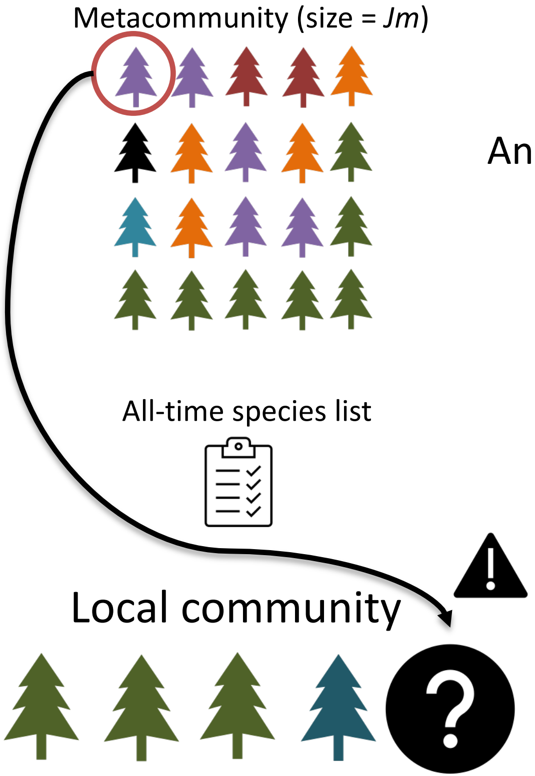
→

A **birth** or **immigration** event occurs

If **birth**, a parent is chosen from the **local** community.

If **immigration**, a parent is chosen from the **meta** community.

# The process




$\frac{1}{J}$

An individual is chosen to die.

→

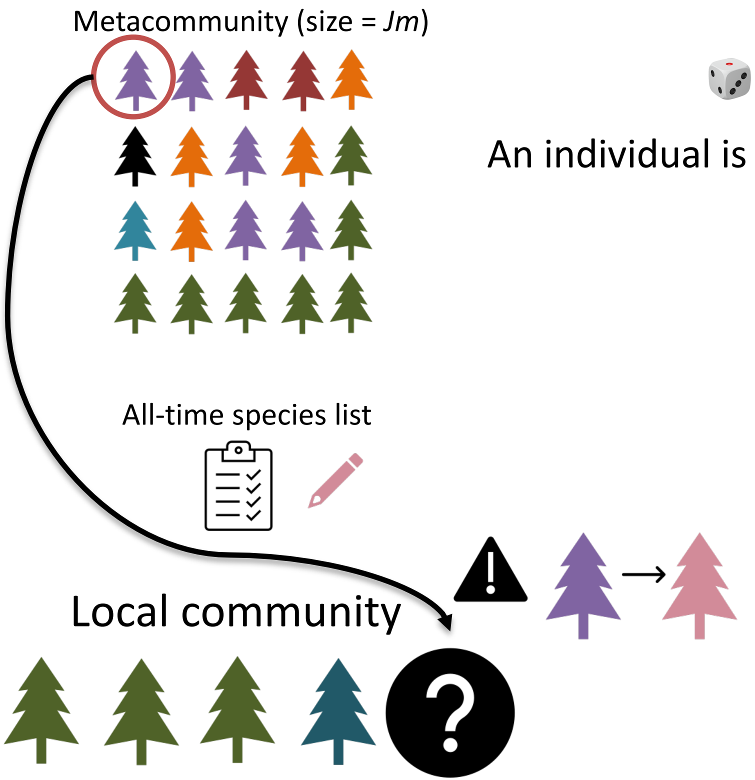
A **birth** or **immigration** event occurs

If **birth**, a parent is chosen from the **local** community.

If **immigration**, a parent is chosen from the **meta** community.

Sometimes, a **speciation** event occurs.

# The process




$\frac{1}{J}$

An individual is chosen to die. →

A **birth** or **immigration** event occurs

If **birth**, a parent is chosen from the **local** community.

If **immigration**, a parent is chosen from the **meta** community.

Sometimes, a **speciation** event occurs.

If so, the new offspring is a **new species,** and a new species joins the all-time list.

# The process


$\frac{1}{J}$

An individual is chosen to die.

→

A **birth** or **immigration** event occurs

If **birth**, a parent is chosen from the **local** community.

If **immigration**, a parent is chosen from the **meta** community.

Sometimes, a **speciation** event occurs.

If not, the new offspring is the **same species** as its parent.

If so, the new offspring is a **new species,** and a new species joins the all-time list.

# The process

Metacommunity (size = *Jm*)

All-time species list

Local community (size = *J*)

$\frac{1}{J}$

An individual is chosen to die.

→

A **birth** or **immigration** event occurs

If **birth**, a parent is chosen from the **local** community.

If **immigration**, a parent is chosen from the **meta** community.

Sometimes, a **speciation** event occurs.

If not, the new offspring is the **same species** as its parent.

If so, the new offspring is a **new species,** and a new species joins the all-time list.

The new offspring replaces the dead individual in the local community.

# The process

Metacommunity (size = *Jm*)

All-time species list

Local community (size = *J*)

$\frac{1}{J}$

An individual is chosen to die.

→

*m*

A **birth** or **immigration** event occurs

If **birth**, a parent is chosen $\frac{1}{J}$ from the **local** community.

If **immigration**, a parent is chosen $\frac{1}{Jm}$ from the **meta** community.

*v*

Sometimes, a **speciation** event occurs.

If not, the new offspring is the **same species** as its parent.

If so, the new offspring is a **new species,** and a new species joins the all-time list.

The new offspring replaces the dead individual in the local community.

Time goes on.

# The process

Metacommunity (size = *Jm*)

All-time species list

Local community (size = *J*)

$\frac{1}{J}$
An individual is chosen to die.

$m$
A **birth** or **immigration** event occurs

If **birth**, a parent is chosen $\frac{1}{J}$ from the **local** community.

If **immigration**, a parent is chosen $\frac{1}{Jm}$ from the **meta** community.

$v$
Sometimes, a **speciation** event occurs.

If not, the new offspring is the **same species** as its parent.

If so, the new offspring is a **new species,** and a new species joins the all-time list.

The new offspring replaces the dead individual in the local community.

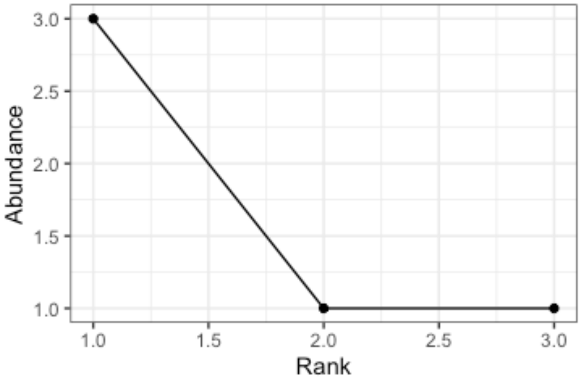
Time goes on.

# The outcomes

Local community (size = *J*)

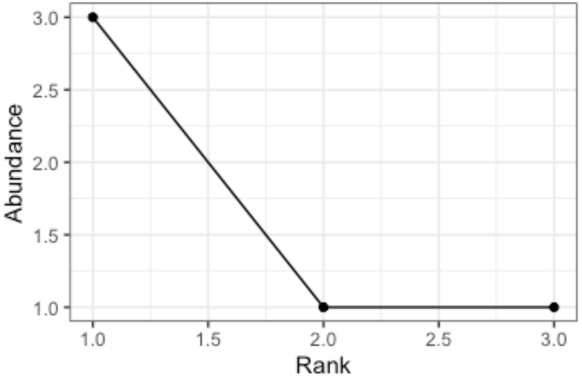
# The outcomes

Local community (size = *J*)



# The outcomes

Local community (size = *J*)




| Q | Hill number |
|---|---|
| 0 | 3 |
| 1 | 2.58 |
| 2 | 2.27 |

Coffee break.

(Then, we'll play!)

Break to play neutral games.

(Not on a computer.)

How did that go?

How did that go?

How could we make this more efficient?

How did that go?

How could we make this more efficient?

Break to code up UNTB in R.

# The process

Metacommunity (size = *Jm*)

All-time species list

Local community (size = *J*)

$\frac{1}{J}$

An individual is chosen to die.

$m$

A **birth** or **immigration** event occurs

If **birth**, a parent is chosen $\frac{1}{J}$ from the **local** community.

If **immigration**, a parent is chosen $\frac{1}{Jm}$ from the **meta** community.

$v$

Sometimes, a **speciation** event occurs.

If not, the new offspring is the **same species** as its parent.

If so, the new offspring is a **new species,** and a new species joins the all-time list.

The new offspring replaces the dead individual in the local community.

Time goes on.

Break to explore UNTB parameter settings in R.

# How did that go?

# Inferring parameters from results in UNTB

# Inferring parameters from results in UNTB

1. What do we mean by inferring parameters from outcomes?
2. How do we approach this for UNTB?
3. What are the challenges we run into?

# What do we mean by inferring parameters from outcomes?

- Assuming the *processes* in a model accurately describe the processes that generated some data***…

*** This is a big assumption!

# What do we mean by inferring parameters from outcomes?

- Assuming the *processes* in a model accurately describe the processes that generated some data***…

  *…we can use our knowledge of the model to guess the parameter settings that generated a specific outcome*.

*** This is a big assumption!

# What do we mean by inferring parameters from outcomes?

- Assuming the *processes* in a model accurately describe the processes that generated some data***…

  *…we can use our knowledge of the model to guess the parameter settings that generated a specific outcome*.

- This is the backbone of likelihood-free inference (coming up soon!)

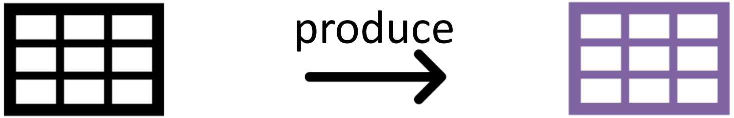
*** This is a big assumption!

# The (general) model structure

Run simulations over a wide range of parameter settings.

Input parameters
(*m, v, J, Jm, Sm*)

Outcome variables
(hill0, hill1, hill2)



# The (general) model structure

Run simulations over a wide range of parameter settings.

Input parameters
(*m, nu, J, Jm, Sm*)

produce →

Outcome variables
(hill0, hill1, hill2)

Fit a model of the form
$parameters \sim results$

Outcome variables
(hill0, hill1, hill2)

predict →

Input parameters
(*m, nu, J, Jm, Sm*)

# The (general) model structure

Run simulations over a wide range of parameter settings.

Input parameters
(*m, nu, J, Jm, Sm*)

produce

Outcome variables
(hill0, hill1, hill2)

Fit a model of the form
$parameters \sim results$

Outcome variables
(hill0, hill1, hill2)

predict

Input parameters
(*m, nu, J, Jm, Sm*)

Use this model to estimate the **parameter values** that produced **observed outcomes**

Focal outcome variables
(hill0 = …, hill1 = …, …)

estimate

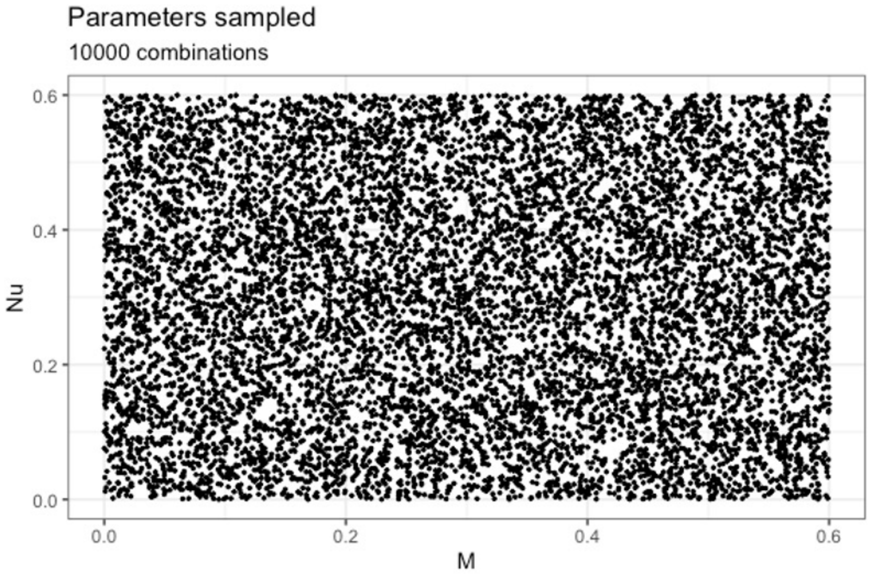
Generating parameters
(*m = …, nu = …, …*)

# An example: predicting *M* and *Nu* from UNTB

# Run simulations over a range of parameters

Constant parameters

| | Parameter | Value |
|---|---|---|
| 1 | Jm | 10000 |
| 2 | Sm | 1000 |
| 3 | J | 1000 |
| 4 | Timesteps | 1000 |



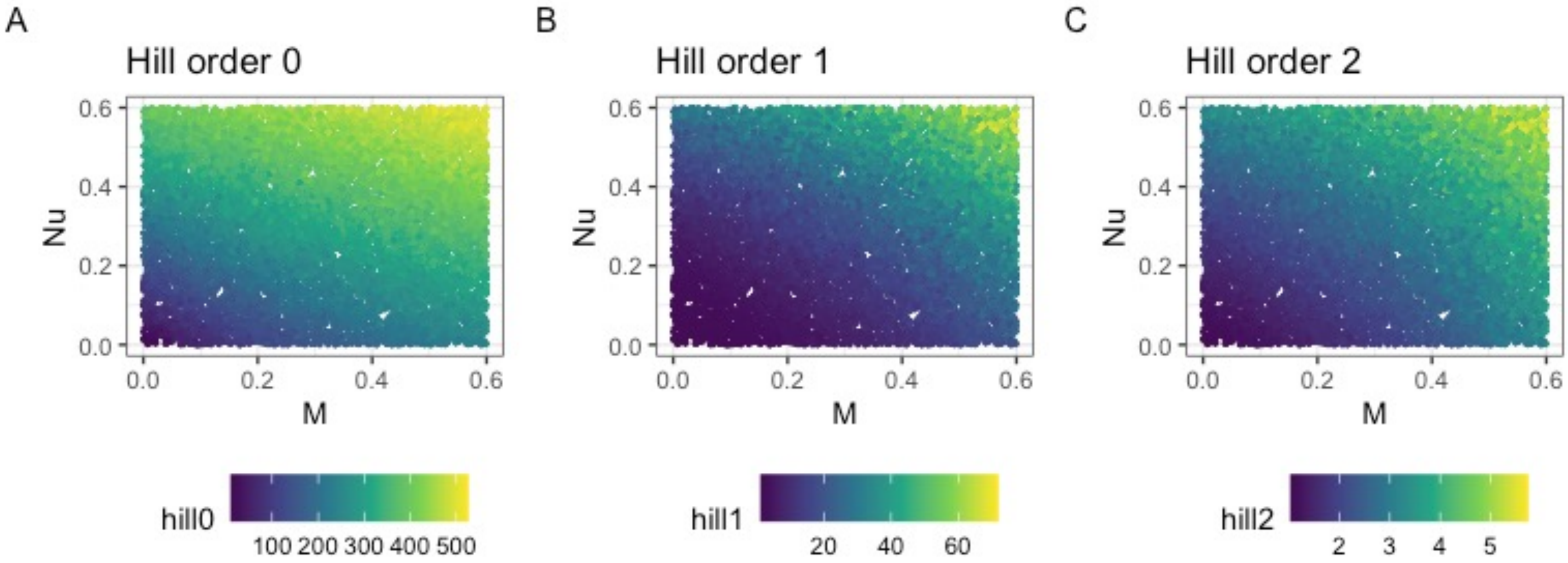
# Collect results

| | Jm | Sm | J | timesteps | Nu | M | hill0 | hill1 | hill2 |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 10000 | 1000 | 1000 | 1000 | 0.52 | 0.36 | 449 | 41.95 | 4.02 |
| 2 | 10000 | 1000 | 1000 | 1000 | 0.18 | 0.50 | 298 | 23.92 | 3.31 |
| 3 | 10000 | 1000 | 1000 | 1000 | 0.52 | 0.15 | 379 | 25.86 | 3.15 |
| 4 | 10000 | 1000 | 1000 | 1000 | 0.39 | 0.59 | 408 | 41.43 | 4.22 |
| 5 | 10000 | 1000 | 1000 | 1000 | 0.02 | 0.31 | 163 | 6.36 | 1.82 |
| 6 | 10000 | 1000 | 1000 | 1000 | 0.31 | 0.18 | 290 | 14.82 | 2.48 |

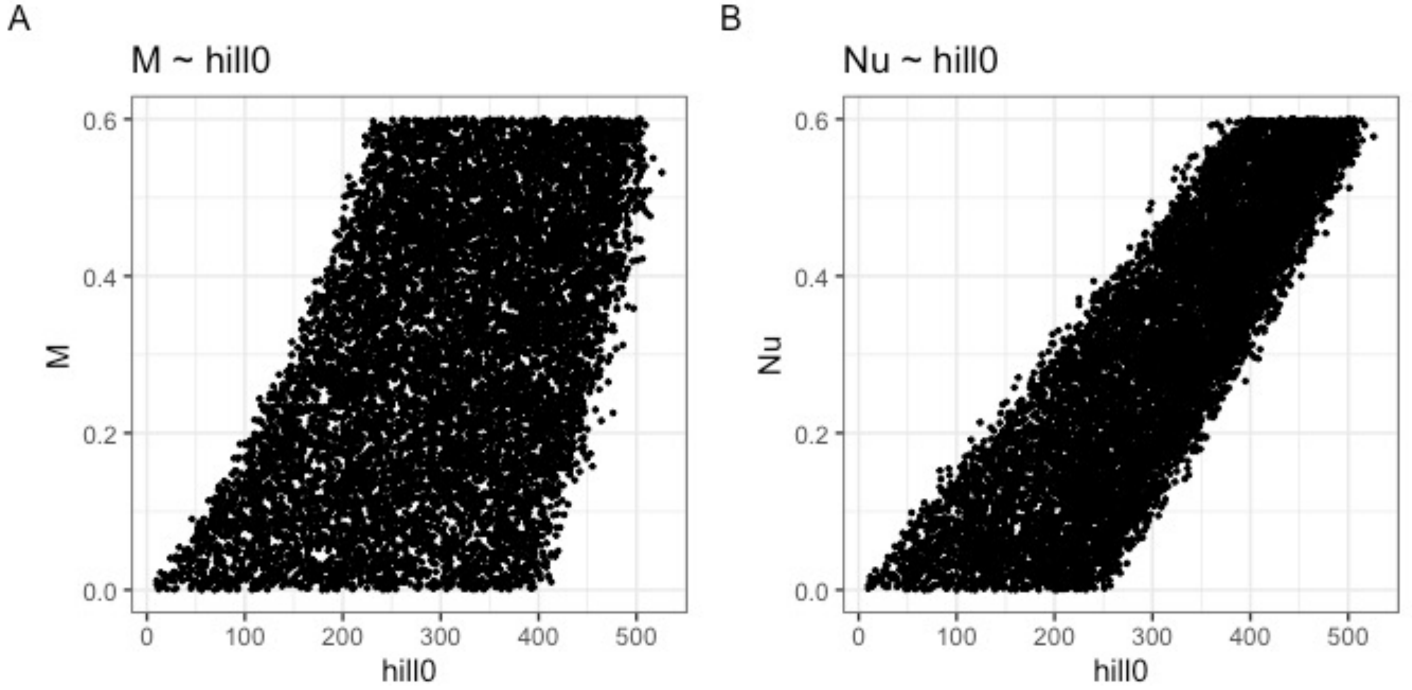
Parameters

Outcomes

# Visualize Hill numbers vs. M, Nu



# Relating outcomes to parameters



# Train a model

```
m_rf_model <- randomForest(M ~ hill0 + hill1 + hill2,
                           data = all_hills)
```

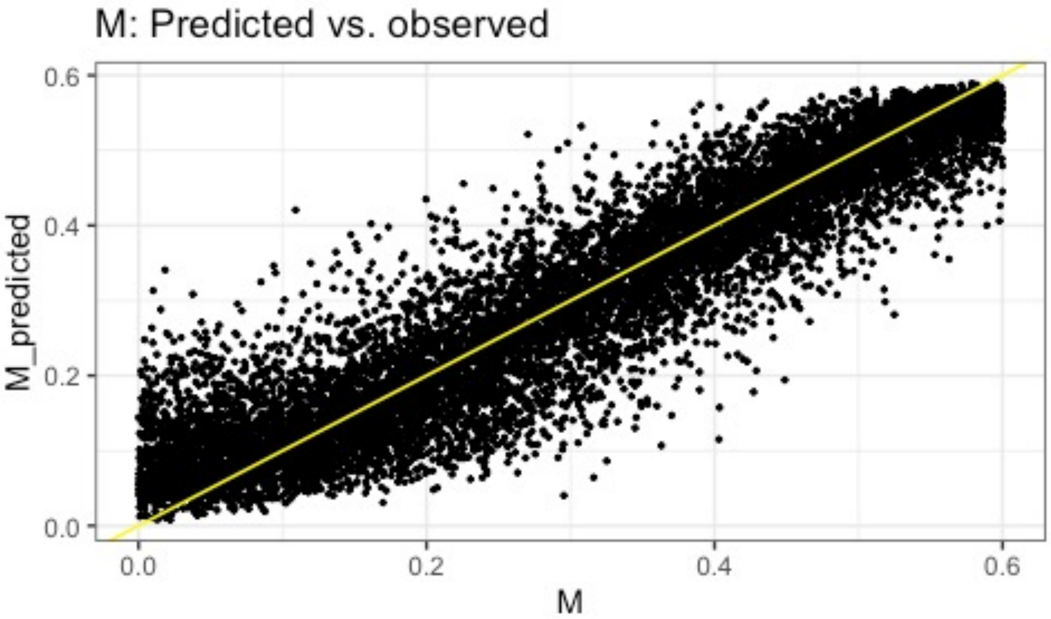
# Train a model

```
m_rf_model

Call:
 randomForest(formula = M ~ hill0 + hill1 + hill2, data
= all_hills)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 1

          Mean of squared residuals: 0.003975154
                    % Var explained: 86.76
```

# Explore model accuracy



# Train a model

```
nu_rf_model <- randomForest(Nu ~ hill0 + hill1 + hill2,
                                       data = all_hills)
```

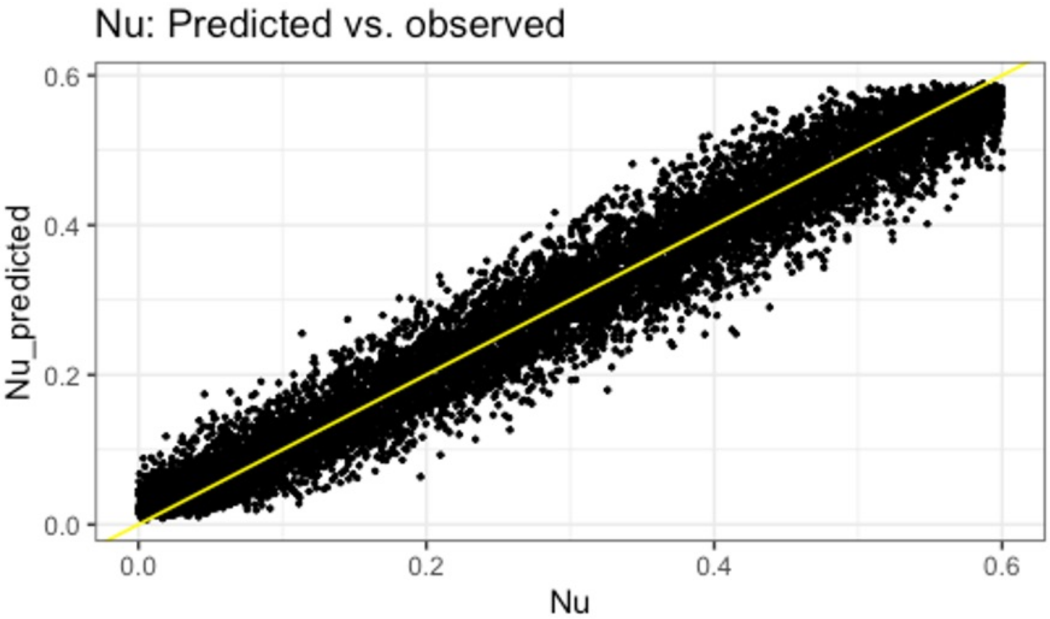
# Train a model

```
nu_rf_model

Call:
 randomForest(formula = Nu ~ hill0 + hill1 + hill2, data
= all_hills)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 1

 Mean of squared residuals: 0.001456496
                    % Var explained: 95.09
```

# Explore model accuracy



# Apply model to **new** (simulated) data

```
new_M <- runif(1, 0, 0.6)
new_Nu <- runif(1, 0, 0.6)

new_sim <- untb(Jm = 10000, Sm = 1000, J = 1000,
                m = new_M, nu = new_Nu, niter = 1000)
new_hills <- untb_hill(new_sim)

predicted_M <- predict(m_rf_model, newdata = new_hills)
predicted_Nu <- predict(nu_rf_model, newdata = new_hills)
```

# Apply model to **new** (simulated) data

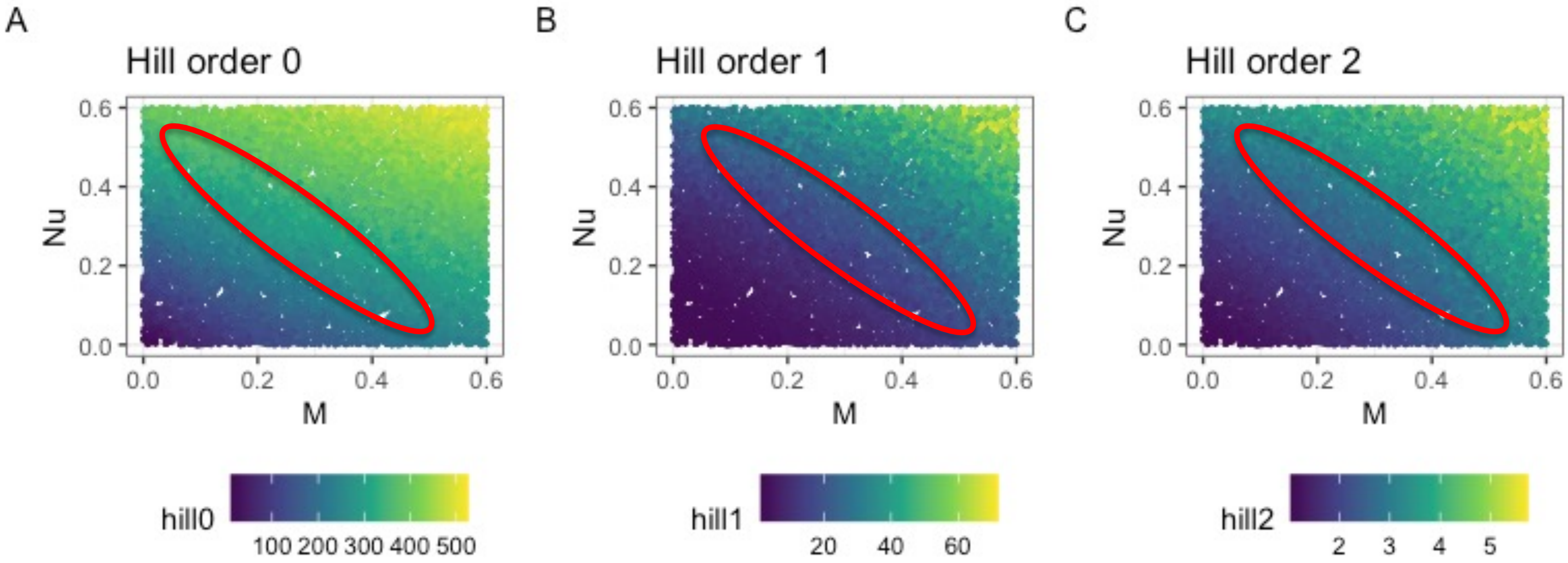
| | |
|---|---|
| `new_M` | `predicted_M` |
| 0.3258534 | 0.2825361 |
| `new_Nu` | `predicted_Nu` |
| 0.4382455 | 0.3794579 |

# Apply model to **new** (simulated) data

| | |
|---|---|
| `new_M` | `predicted_M` |
| `0.3258534` | `0.2825361` |
| `new_Nu` | `predicted_Nu` |
| `0.4382455` | `0.3794579` |

Estimation is good but not perfect!

# Challenges to estimation



# How could we improve?

# How could we improve?

- Different parameters, different rules
- More data dimensions
- Stay tuned!!!

# Recap

- In principle, we can use process models to infer the parameters that generate observed data
- This is complicated by:
  - Out-of-sample prediction
  - Model identifiability
  - Model run time
  - The underlying validity of the process model

# Looking ahead…

**Flexible**, **scalable**, **multidimensional**, and generally much **mess-**ier and more powerful models!