Optimizing Bioenergetic Food Web Models of Ecosystems Using Gamification and Machine Learning

Ben Saylor
Master's Thesis Defense
San Francisco State University
May 15, 2017

Introduction

- Ecosystems are complex systems; ecological modeling is complex
- *Allometric trophic network (ATN)* models
    - Model population dynamics in terms of energy flow through *food webs*
    - Difficult to parameterize due to highly complex and nonlinear behavior
- Our objective: find ways of parameterizing ATN models to replicate sustaining ecosystems
- Our approaches: based on machine learning and gamification

Contributions

- Graph sampling algorithm for food webs
- New ATN simulator implementation
  - Improves correctness and performance over previous implementation
  - Adds steady state detection
- New environment score formula
- Showed that ML can predict ecosystem health
- Method of generating persistent simulated ecosystems
  - New ecosystems for Convergence game
- ML-based method for generating parameter hints
- ML-based method of searching ATN model parameter space

ATN models

- System of ordinary differential equations

$$B_i' = f(\mathbf{B})$$

ATN model equations

Producers:

$$B_i' = r_i B_i G_i(\mathbf{B}) - \sum_{j \in predators} (x_j y_{ji} \alpha_{ji} F_{ji} B_j / e_{ji})$$

Consumers:

$$B_i' = \sum_{j \in prey} (x_i y_{ij} \alpha_{ij} F_{ij} B_i) - \sum_{j \in predators} (x_j y_{ji} \alpha_{ji} F_{ji} B_j / e_{ji}) - x_i B_i$$

Growth function:

$$G_i(B) = 1 - \frac{B_i}{K_i}$$

Functional response:

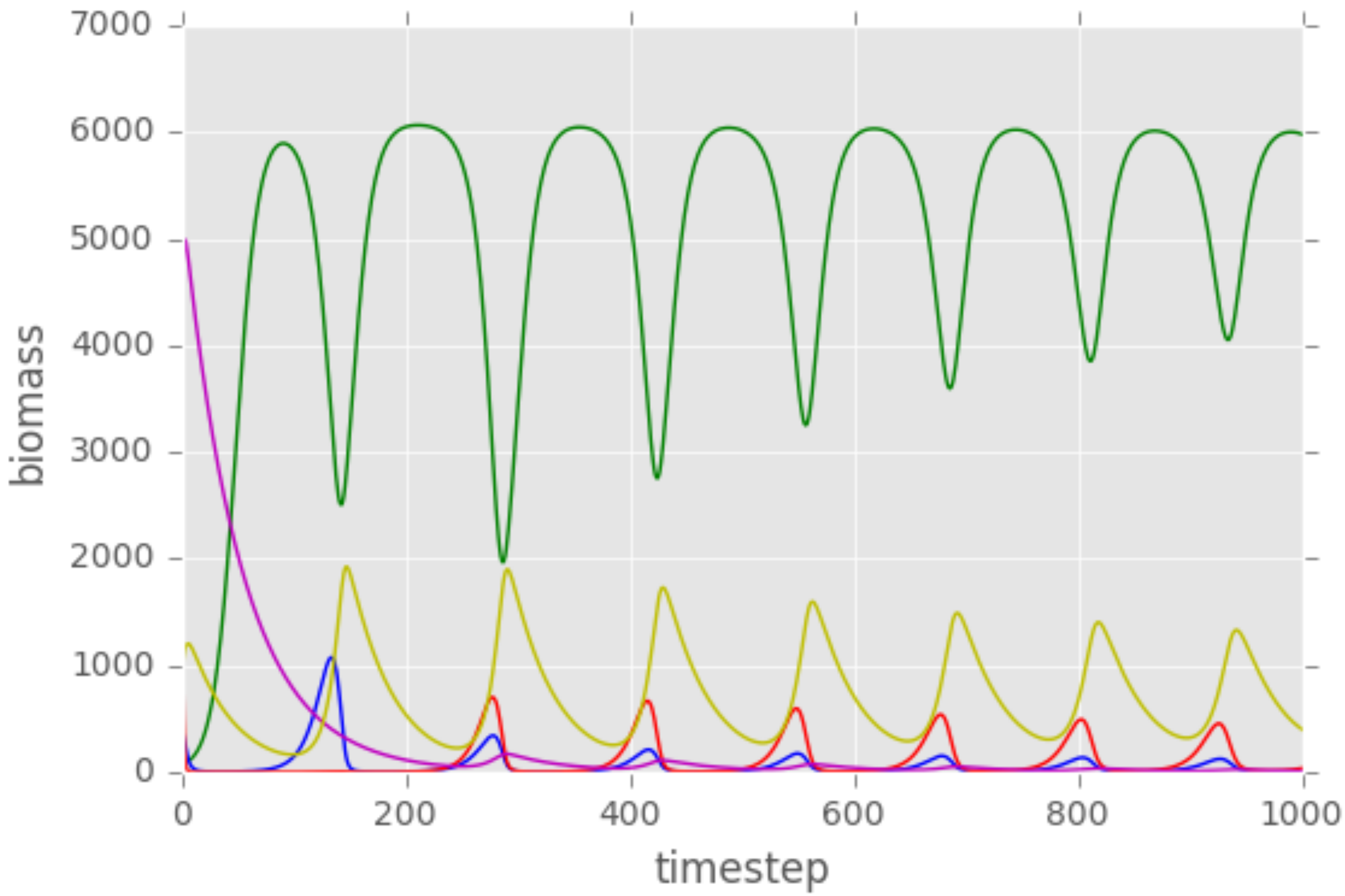
$$F_{ij} = \frac{B_j^{1+q_{ij}}}{\sum_{m \in prey} \alpha_{im} B_m^{1+q_{im}} + B_{0ij}^{1+q_{ij}}}$$

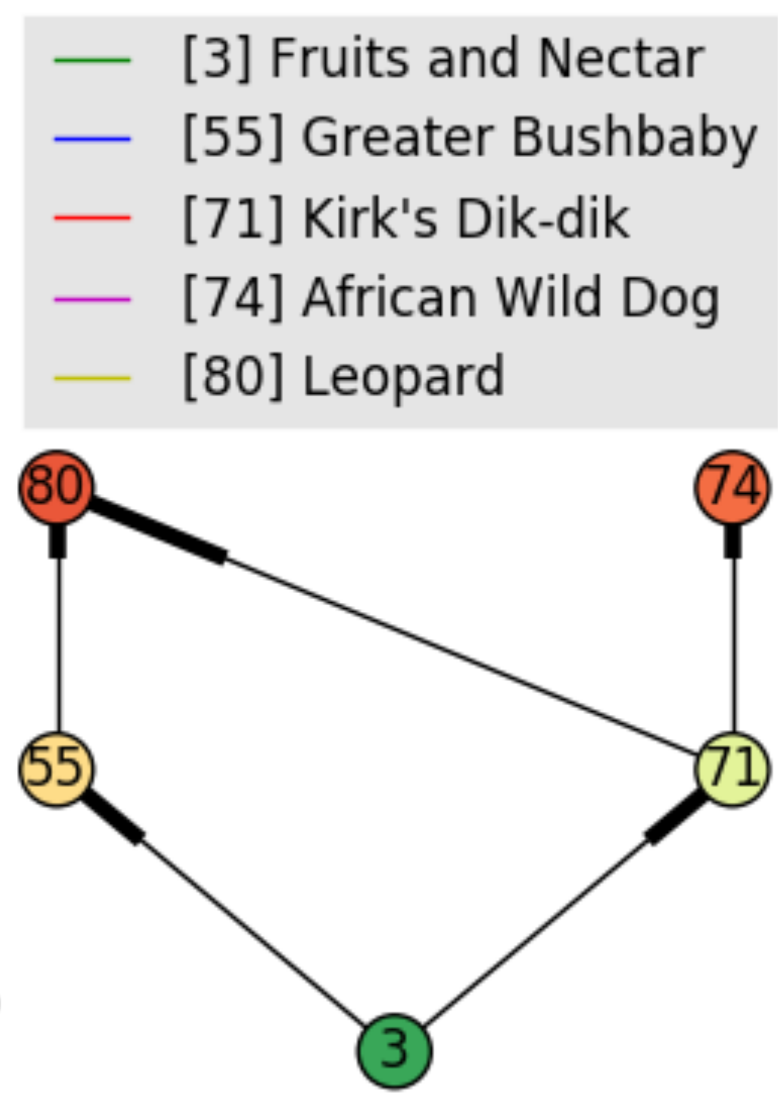
ATN model parameters

Parameter	Description
α_{ij}	Relative half-saturation density of predator i when consuming prey j
B_{0ij}	Half-saturation density of predator i when consuming prey j
c_{ij}	Competition between producers i and j for shared carrying capacity
e_{ij}	Assimilation efficiency of prey j by predator i
K_i	Carrying capacity of producer i
K_s	System-wide carrying capacity
q_{ij}	Functional response control parameter
r_i	Growth rate of producer i
x_i	Metabolic rate of consumer i
y_{ij}	Maximum ingestion rate of predator i when consuming prey j

ATN model simulation

- Integrate the equations, solving the *initial value problem*
- Result: biomass over time by species

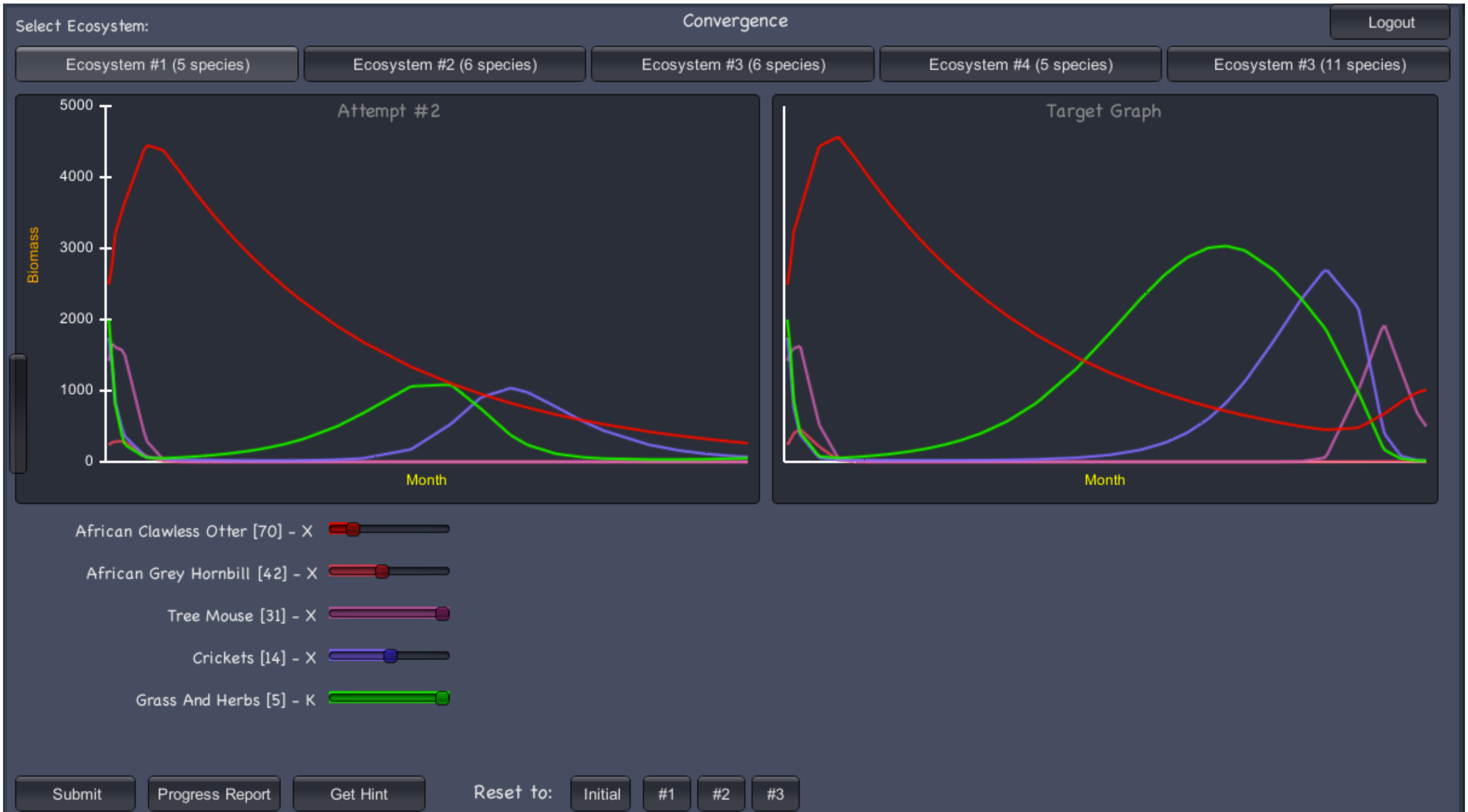




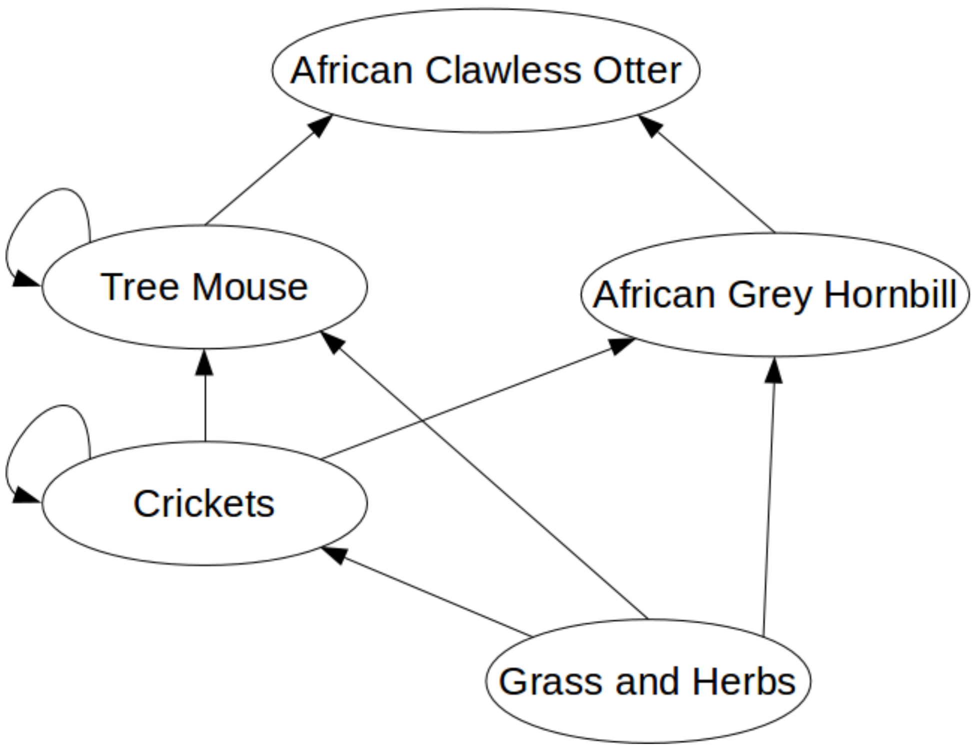
Gamification

- Crowdsourcing human problem-solving abilities to help solve problems that are hard using computation alone
- Successful example: FoldIt
- *World of Balance* by Dr. Yoon et al.: ecosystem nurturing game backed by ATN simulations
- Mini-games within WoB include *Convergence*

Convergence game



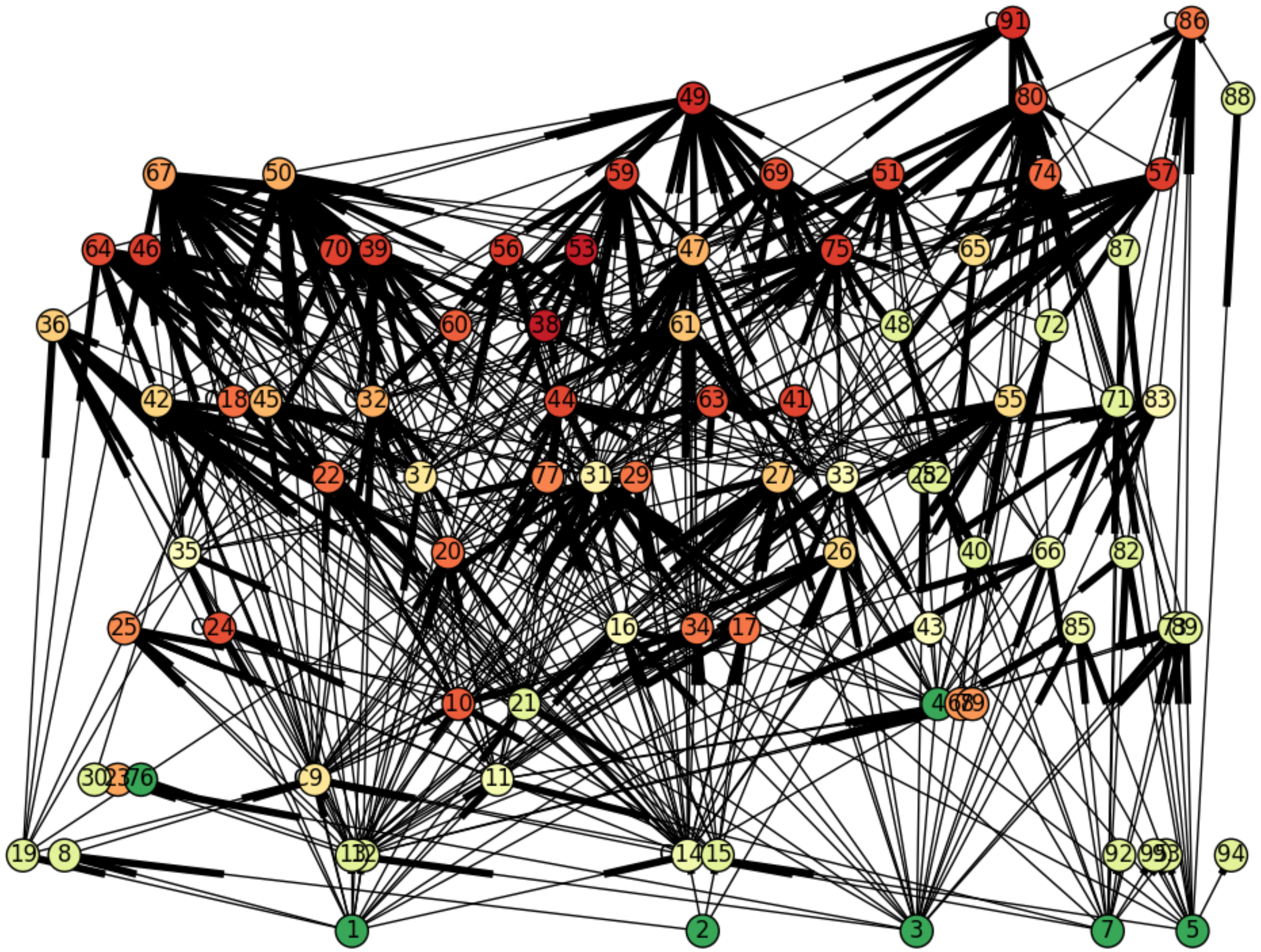
Food webs as graphs



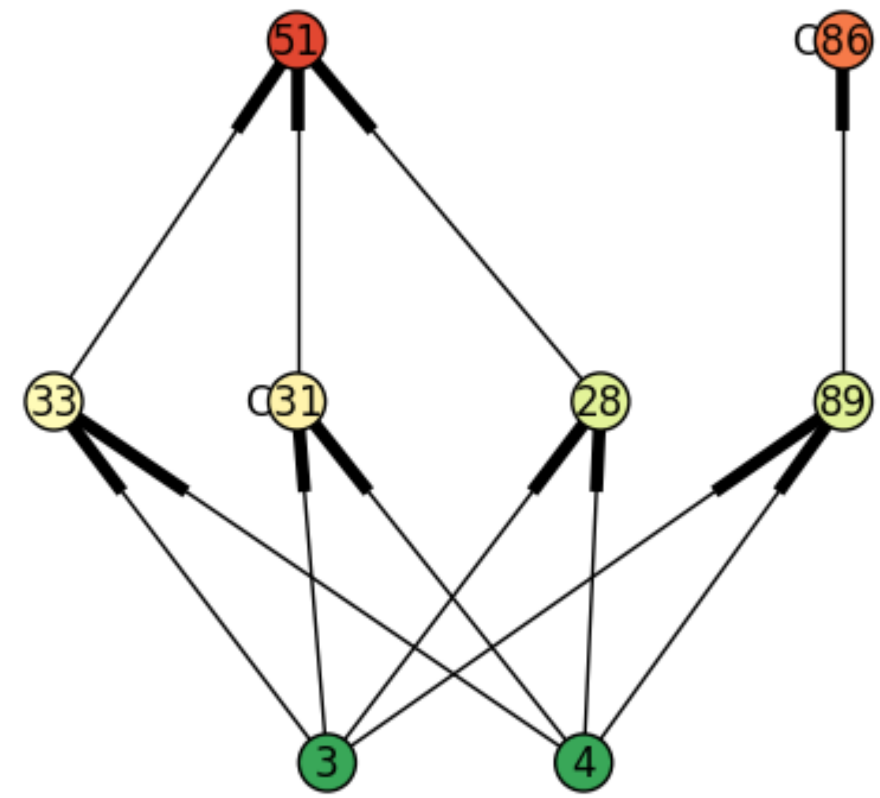
- Food webs are directed graphs
- Edges point from prey to predator

Graph concept	Food web concept
node	species, guild, resource
edge	feeding relationship
self-loop	cannibalism
source node	producer
non-source node	consumer
directed simple path from source node	food chain

Graph sampling and “subwebs”



87 Serengeti species



8 Serengeti species

Motivations: analysis, gameplay

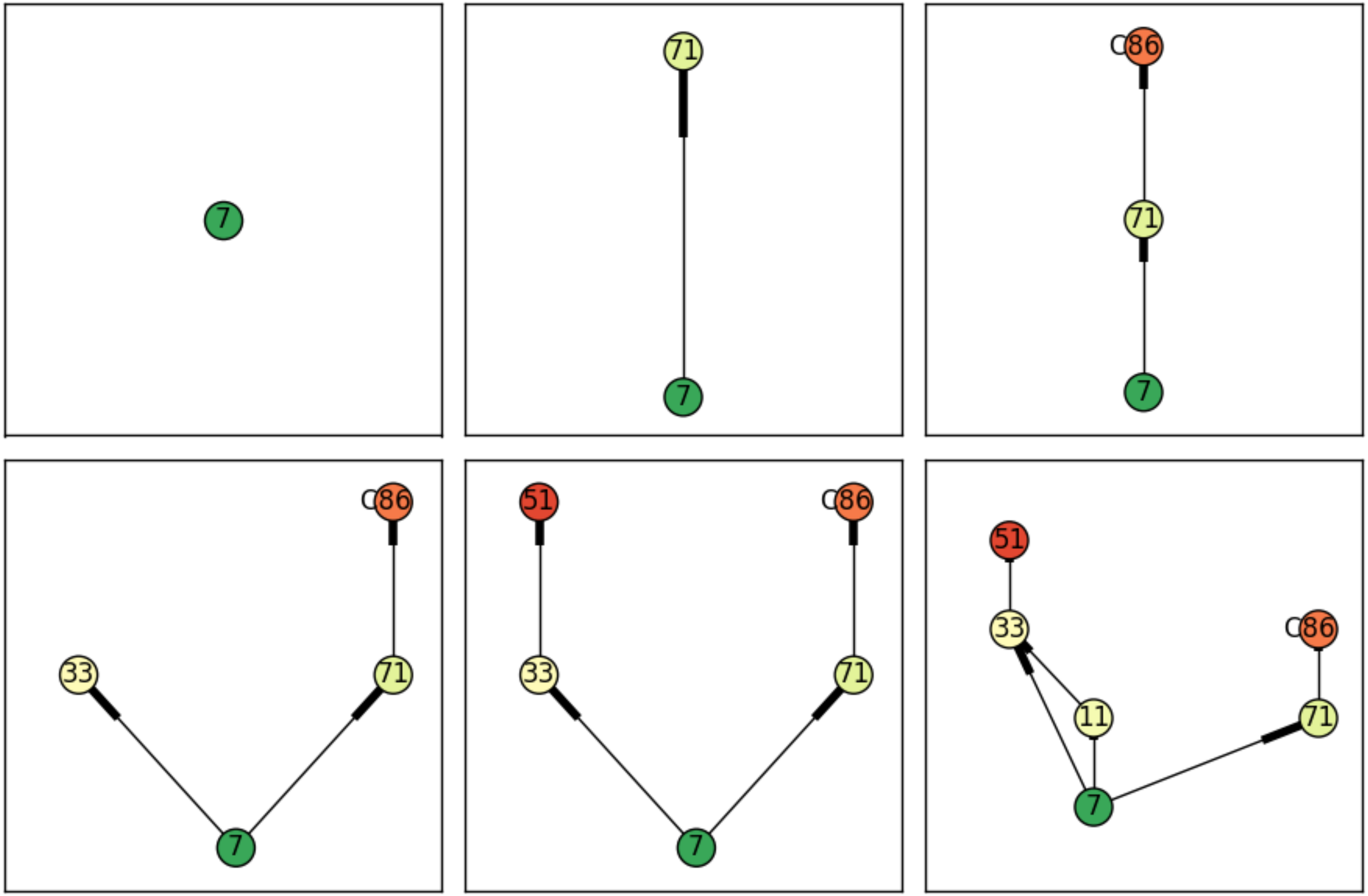
Subweb criteria

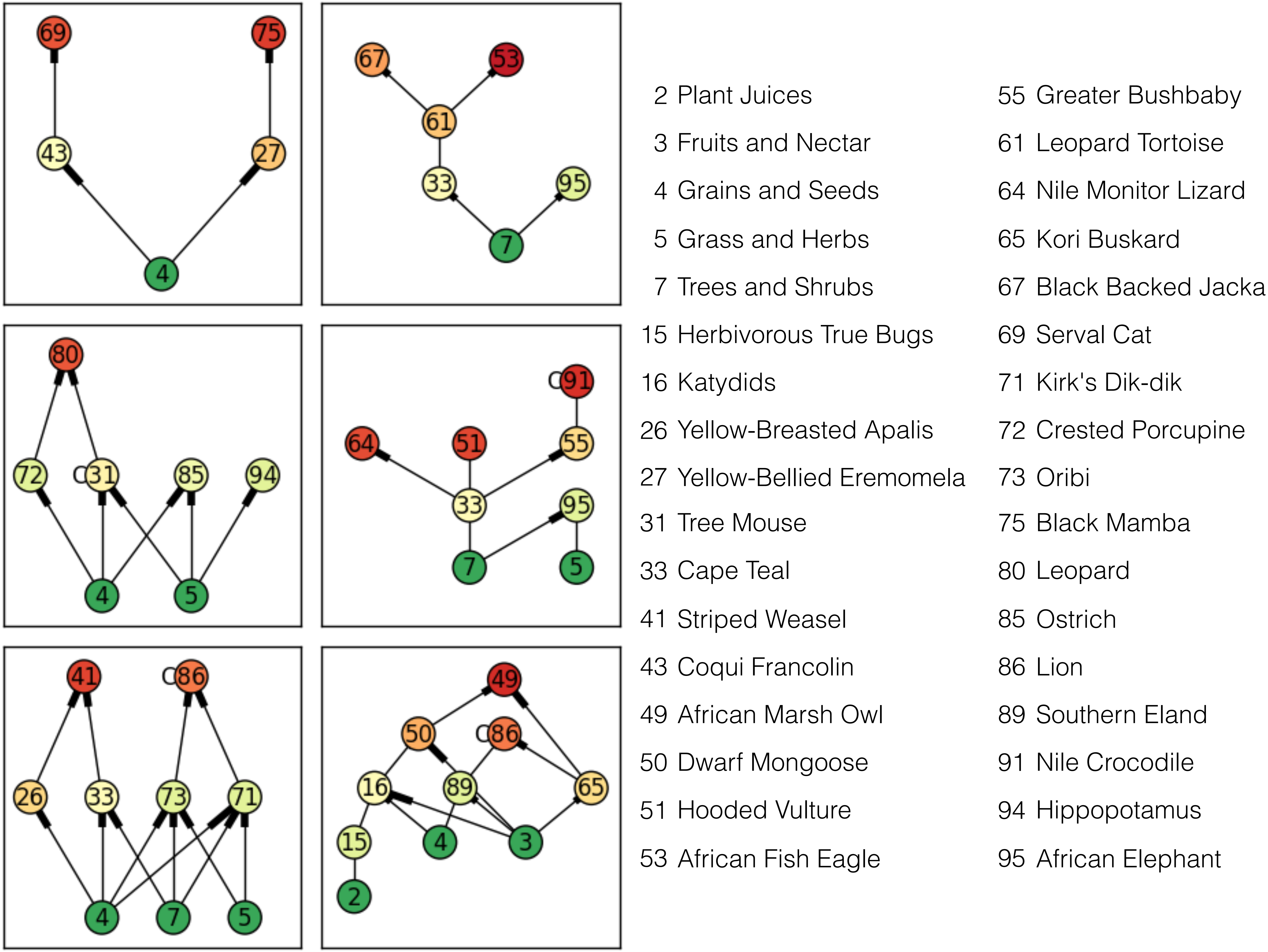
1. Single connected component
2. No incomplete food chains
3. Contains all edges between selected nodes

Random subweb algorithm

- Start with given number of producer nodes
- Each iteration, grow outward:
  - Try adding a predator of a “lonely plant eater”
  - If none, add any neighboring consumer
- Add all edges between selected nodes

Random subweb algorithm




2 Plant Juices
3 Fruits and Nectar
4 Grains and Seeds
5 Grass and Herbs
7 Trees and Shrubs
15 Herbivorous True Bugs
16 Katydids
26 Yellow-Breasted Apalis
27 Yellow-Bellied Eremomela
31 Tree Mouse
33 Cape Teal
41 Striped Weasel
43 Coqui Francolin
49 African Marsh Owl
50 Dwarf Mongoose
51 Hooded Vulture
53 African Fish Eagle
55 Greater Bushbaby
61 Leopard Tortoise
64 Nile Monitor Lizard
65 Kori Buskard
67 Black Backed Jackal
69 Serval Cat
71 Kirk's Dik-dik
72 Crested Porcupine
73 Oribi
75 Black Mamba
80 Leopard
85 Ostrich
86 Lion
89 Southern Eland
91 Nile Crocodile
94 Hippopotamus
95 African Elephant

Measuring ecosystem health

- Goals:
  - Provide WoB players with feedback via score
  - Provide target variables for machine learning
- WoB Environment Score
  - Rewards high biomass levels
  - Rewards presence of high trophic level species
  - Rewards species diversity

Original Environment Score

$$\left[\left[5\log_2\left(\sum_{i=1}^{N} b_i\left(\frac{B_i}{b_i}\right)^{T_i}\right)\right]^2 + N^2\right]$$

- Potential disadvantages:
    - Top predators and small animals can contribute orders of magnitude more weight
    - WoB body size data is estimated
    - N^2 does not scale with biomass - species diversity makes a very small contribution

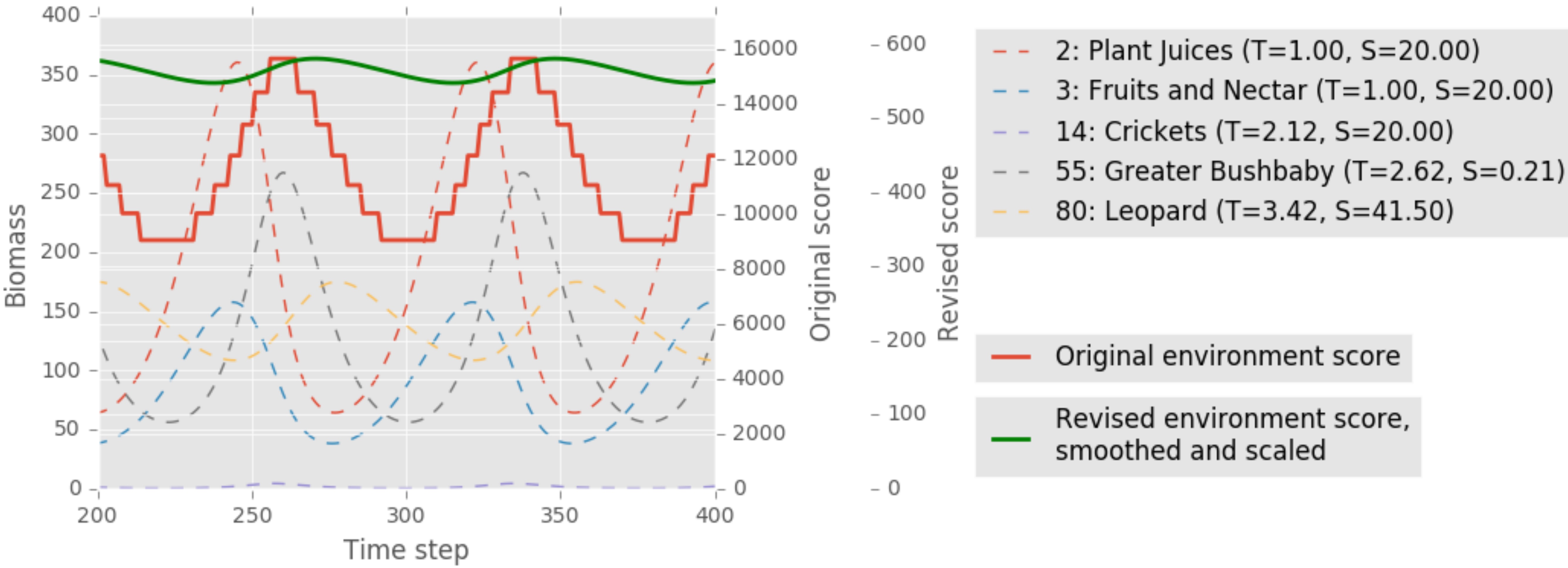
Revised Environment Score

$$BiomassScore = \sum_{i=1}^{N} T_i B_i$$

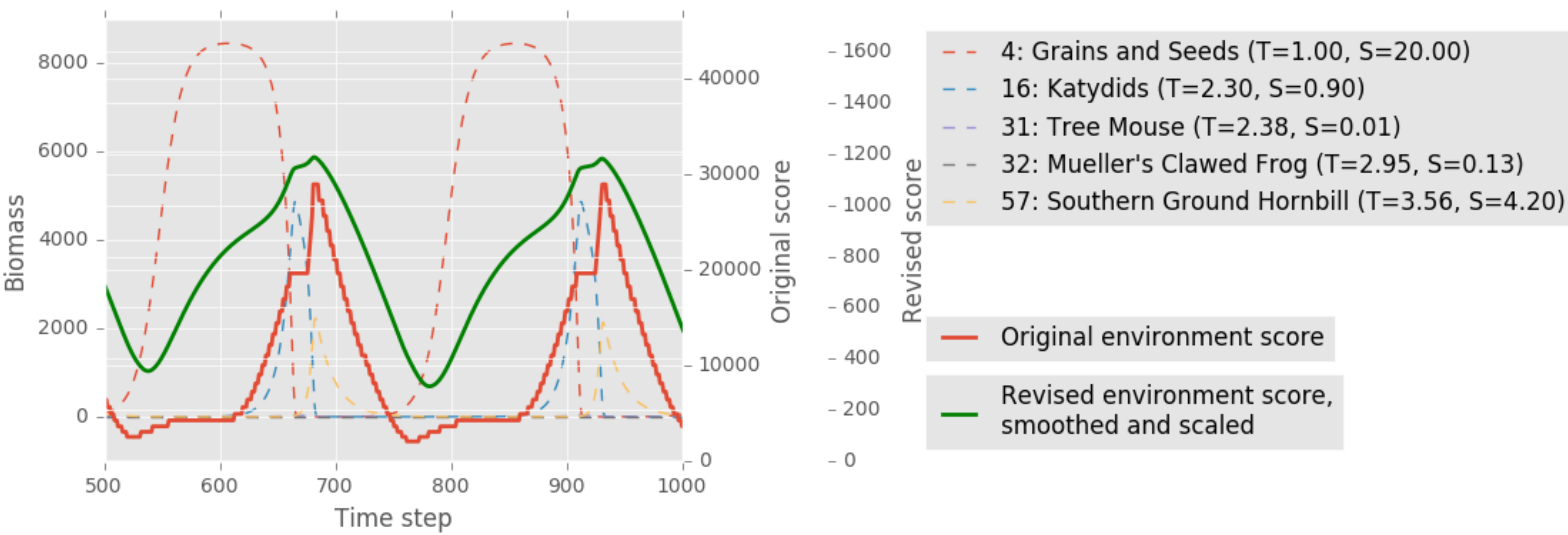
$$Shannon = -\sum_{i=1}^{N} p_i \log_2 p_i \quad \text{where } p_i = \frac{B_i}{\sum_{j=1}^{N} B_j}$$

$$RevisedEnvironmentScore = BiomassScore \times (1 + Shannon)$$

- Advantages:
  - Species contributions are more balanced
  - Species diversity makes a larger contribution



Biomass
Time step
Original score
Revised score
2: Plant Juices (T=1.00, S=20.00)
3: Fruits and Nectar (T=1.00, S=20.00)
14: Crickets (T=2.12, S=20.00)
55: Greater Bushbaby (T=2.62, S=0.21)
80: Leopard (T=3.42, S=41.50)
Original environment score
Revised environment score, smoothed and scaled



Biomass
Time step
Original score
Revised score
4: Grains and Seeds (T=1.00, S=20.00)
16: Katydids (T=2.30, S=0.90)
31: Tree Mouse (T=2.38, S=0.01)
32: Mueller's Clawed Frog (T=2.95, S=0.13)
57: Southern Ground Hornbill (T=3.56, S=4.20)
Original environment score
Revised environment score, smoothed and scaled

Parameter range hints for Convergence

- Goal: guide Convergence players by showing promising parameter ranges

- 4-step approach:

   1. **Species selection**: use our graph sampling algorithm to generate food webs

   2. **Parameter space exploration and simulation**: Generate 1,000s of simulations with randomized parameters, evaluate ecosystem health using linear time trend of environment score

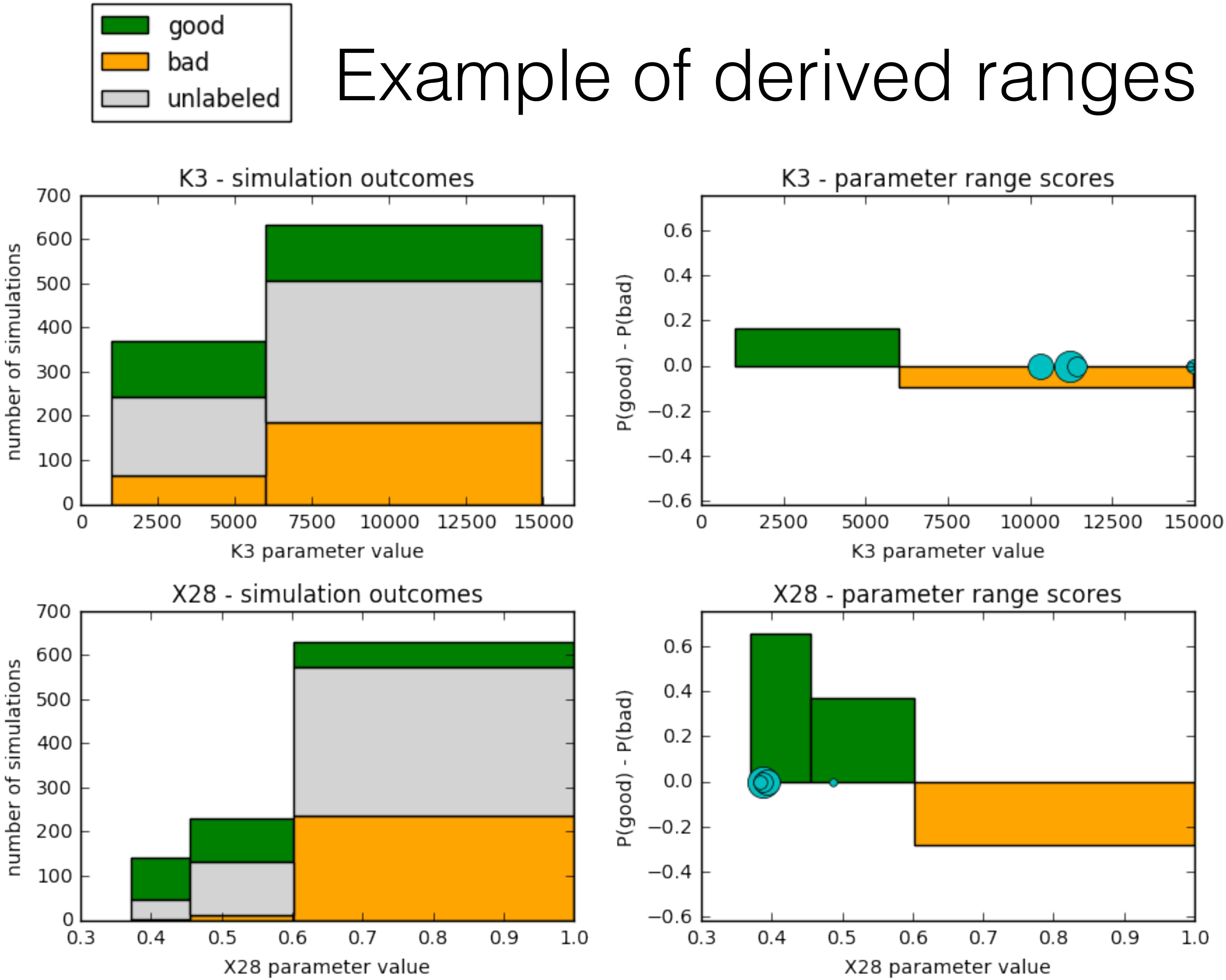
   3. **Machine-learning classification of simulation results**: Classify simulations as “good” or “bad” based on score trend values, train decision trees to predict label based on model parameters

   4. **Derivation of parameter ranges to display as game hints**: Use the decision tree structure to derive promising parameter ranges

Example of decision tree

```
X28 <= 0.601404
|   X51 <= 0.14091
|   |   X28 <= 0.455708: good (11.0)
|   |   X28 > 0.455708: bad (10.0/1.0)
|   X51 > 0.14091
|   |   X73 <= 0.079958
|   |   |   X86 <= 0.069814: good (22.0)
|   |   |   X86 > 0.069814: bad (5.0/1.0)
|   |   X73 > 0.079958: good (157.0)
X28 > 0.601404
|   X51 <= 0.18639: bad (194.0)
|   X51 > 0.18639
|   |   K3 <= 6003.88: good (56.0/1.0)
|   |   K3 > 6003.88
|   |   |   X73 <= 0.145747: bad (41.0)
|   |   |   X73 > 0.145747: good (4.0/1.0)
```

Example of derived ranges



Evaluation: classifier performance

- Evaluated approach on 3 food webs

Food web	Class	Precision	Recall	F1 score
5 species	good	0.951	0.951	0.951
	bad	0.959	0.959	0.959
10 species	good	1.000	0.996	0.998
	bad	0.996	1.000	0.998
15 species	good	0.996	1.000	0.998
	bad	1.000	0.996	0.998

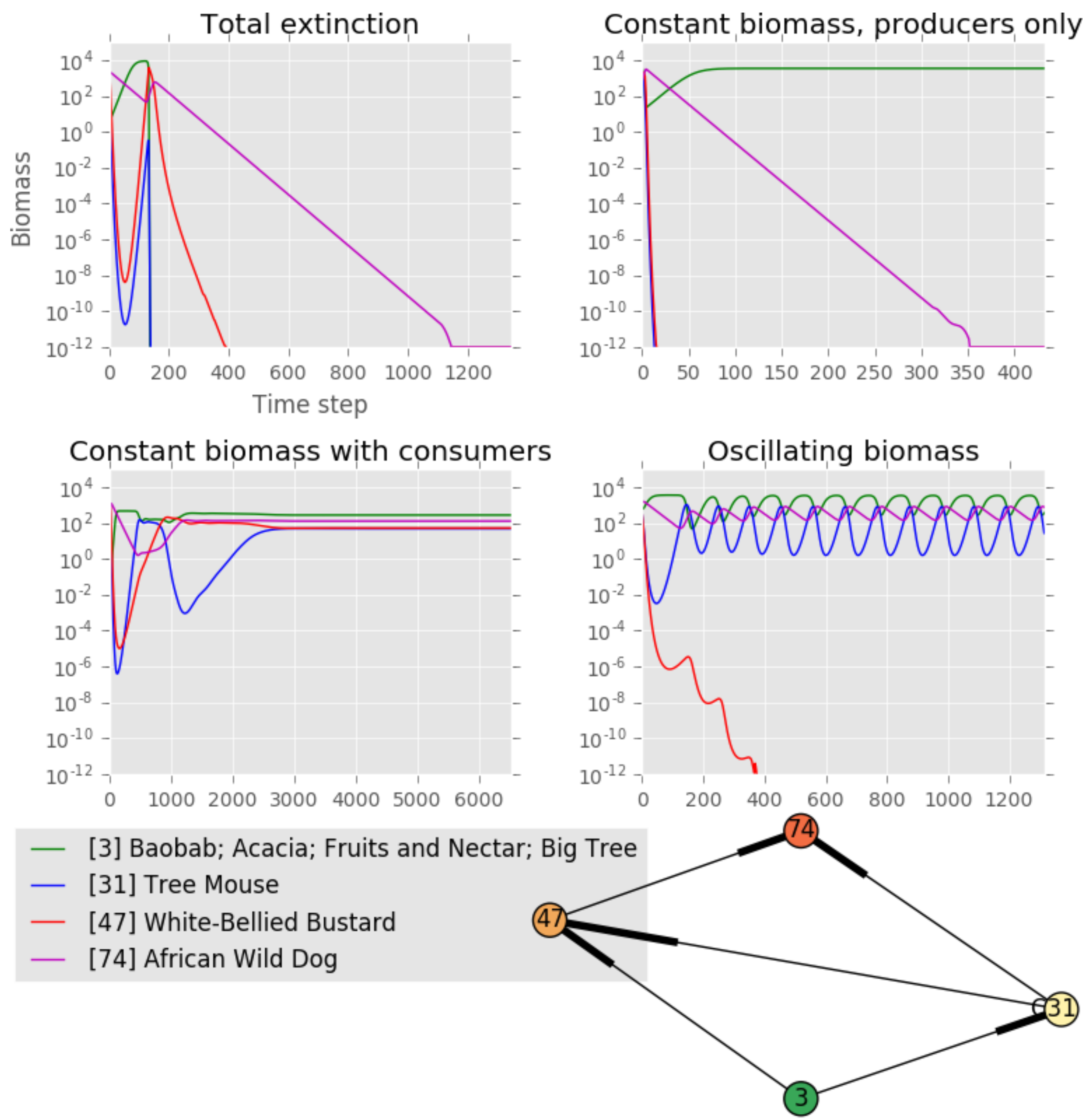
Evaluation: parameter ranges

- Conducted “simulated user study” with a test group given hints, a control group not given hints

Food web	Statistic	Control group	Test group
5 species	mean	-3.054789	5.358788
	std	9.410075	2.236706
10 species	mean	-5.187572	-6.016333
	std	8.226043	2.130220
15 species	mean	-2.250668	-0.975169
	std	5.101375	4.270391

Food web	t-statistic	p-value
5 species	27.508	1.772×10^{-127}
10 species	-3.0842	0.00209
15 species	6.0628	1.6038×10^{-9}

Steady states



Steady states: Implementation

- Basis: ATN model is memoryless: $\mathbf{B_t}' = f(\mathbf{B_t})$
  - Watch for zero derivatives: $\left|\frac{B'_i}{B_i}\right| \leq 10^{-10}$
  - Watch for return to previous state: $\left|\frac{B_i - B_{si}}{B_{si}}\right| \leq 0.01$
    - Wait for this to occur 3 times for confirmation

Steady states: Evaluation

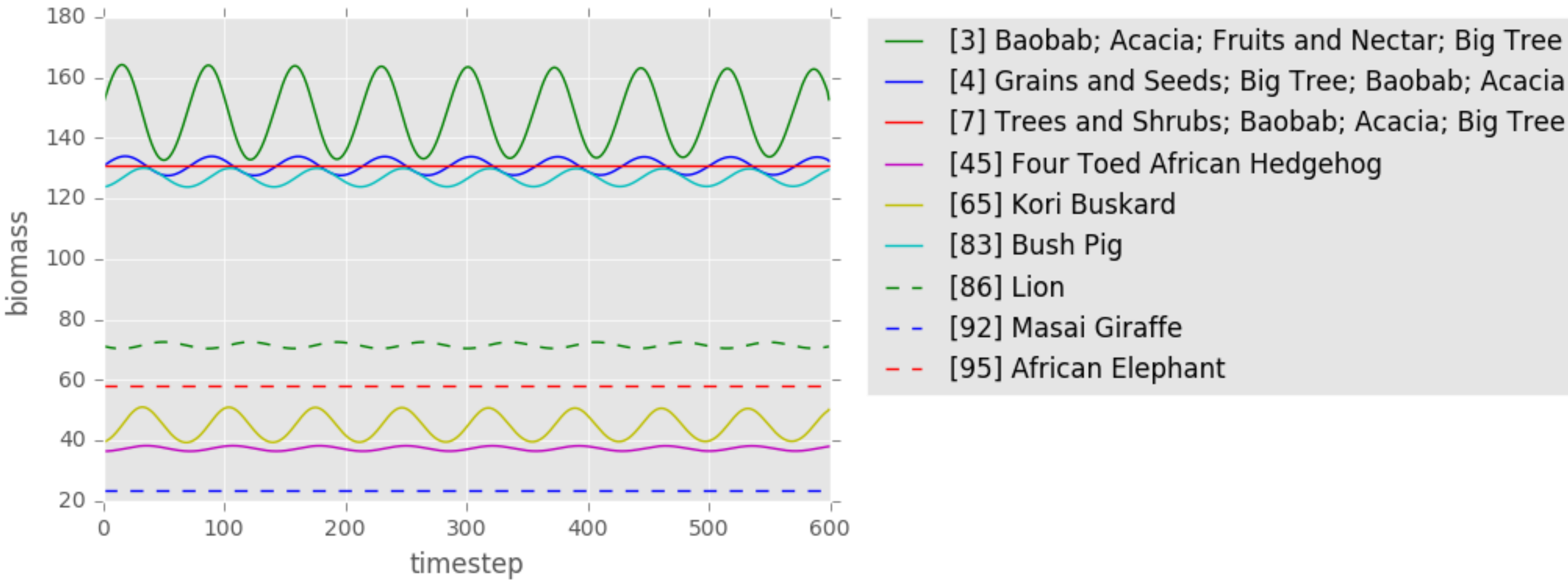
Food web	q	TP	FP	Precision
2-8-9-26-41	0.0	319	0	1.000
	0.2	954	0	1.000
3-21-55-80-85	0.0	29	1	0.967
	0.2	432	86	0.834
3-30-50-69-71	0.0	17	0	1.000
	0.2	929	50	0.949
2-3-5-8-9-21-22-69-71-94	0.0	718	1	0.999
	0.2	725	0	1.000
4-7-14-43-47-61-69-74-80-89	0.0	836	0	1.000
	0.2	850	0	1.000

Generating sustaining ecosystems for Convergence

- Objective: generate simulated ecosystems in which
  - all species survive, and
  - biomass graph is visually interpretable.
- Approach:
  1. Run many simulations to steady state including animals
  2. Drop species that went extinct, assemble food web from remaining species
  3. Generate new simulations *starting* with final biomass from previous simulations, pruned food webs, same parameters
  4. Automatically filter results for visual clarity

Results

- Preliminary experiment filtered 4,000 simulations down to 384 such as the one below



Using decision trees to narrow the parameter search space

- Objective: For a given food web, identify regions of the parameter space leading to sustaining ecosystems

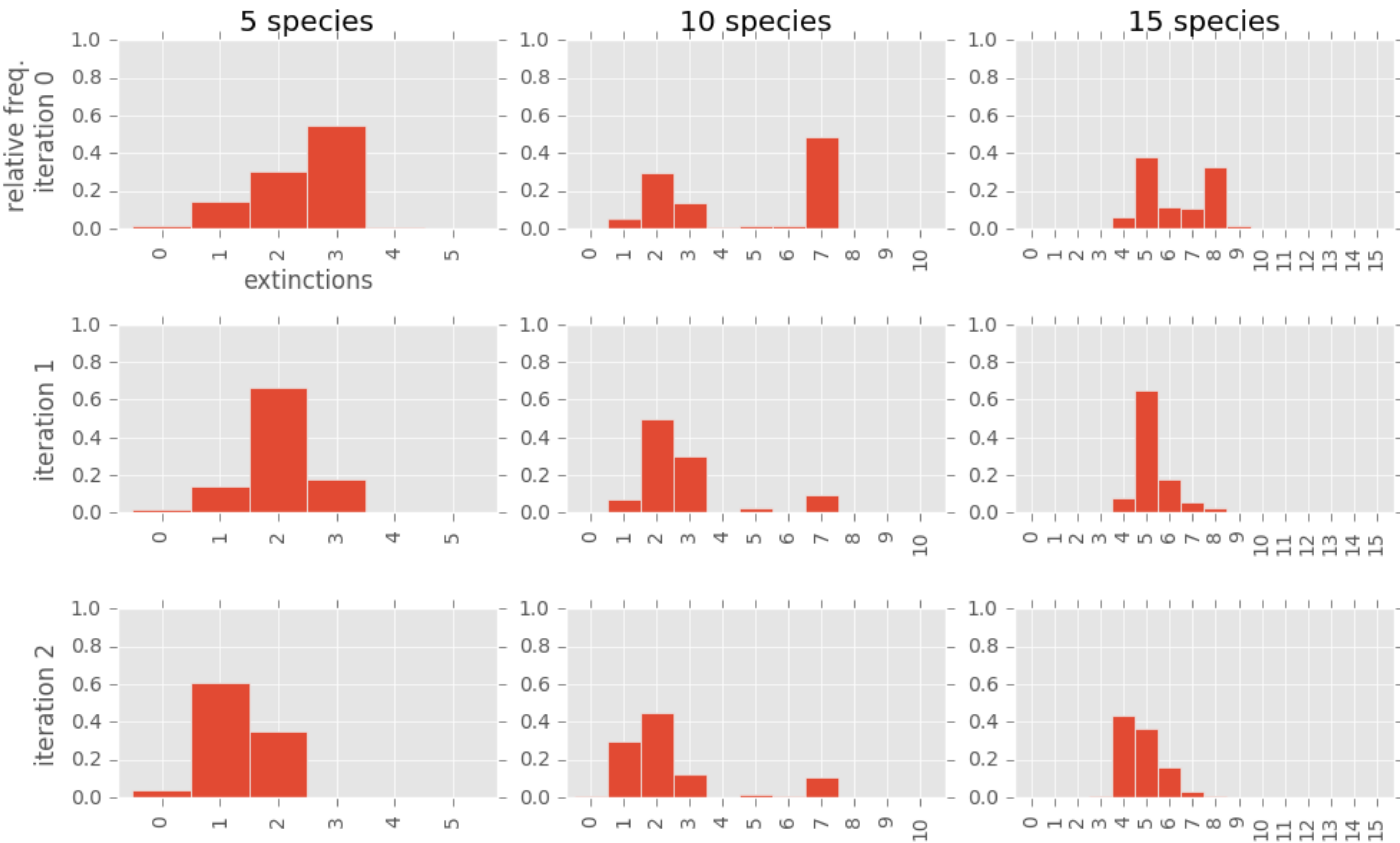
- Definition of “sustaining”: all species survive to steady state

- Approach: iteratively refine search space based on promising regions identified using decision trees

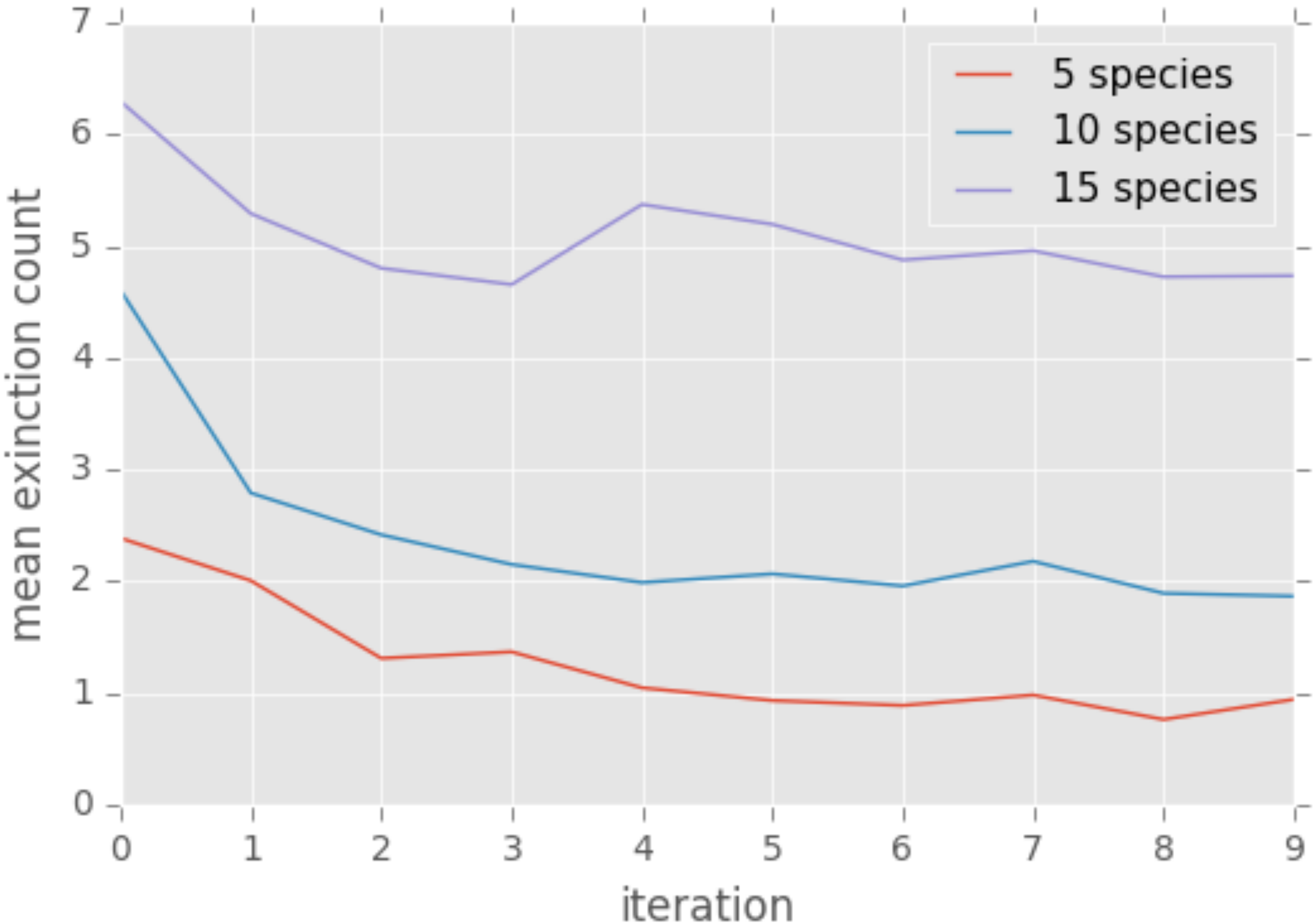
Decision tree search process

- Each iteration:
  - Generate 1,000 simulations with parameter values randomly drawn from the promising regions identified in the previous iteration
  - Train and test a decision tree to classify “good” vs. “bad” simulations based on median extinction count
  - Identify “promising” decision tree leaves that predict “good” simulations
  - Follow the path to each leaf to obtain promising region bounds

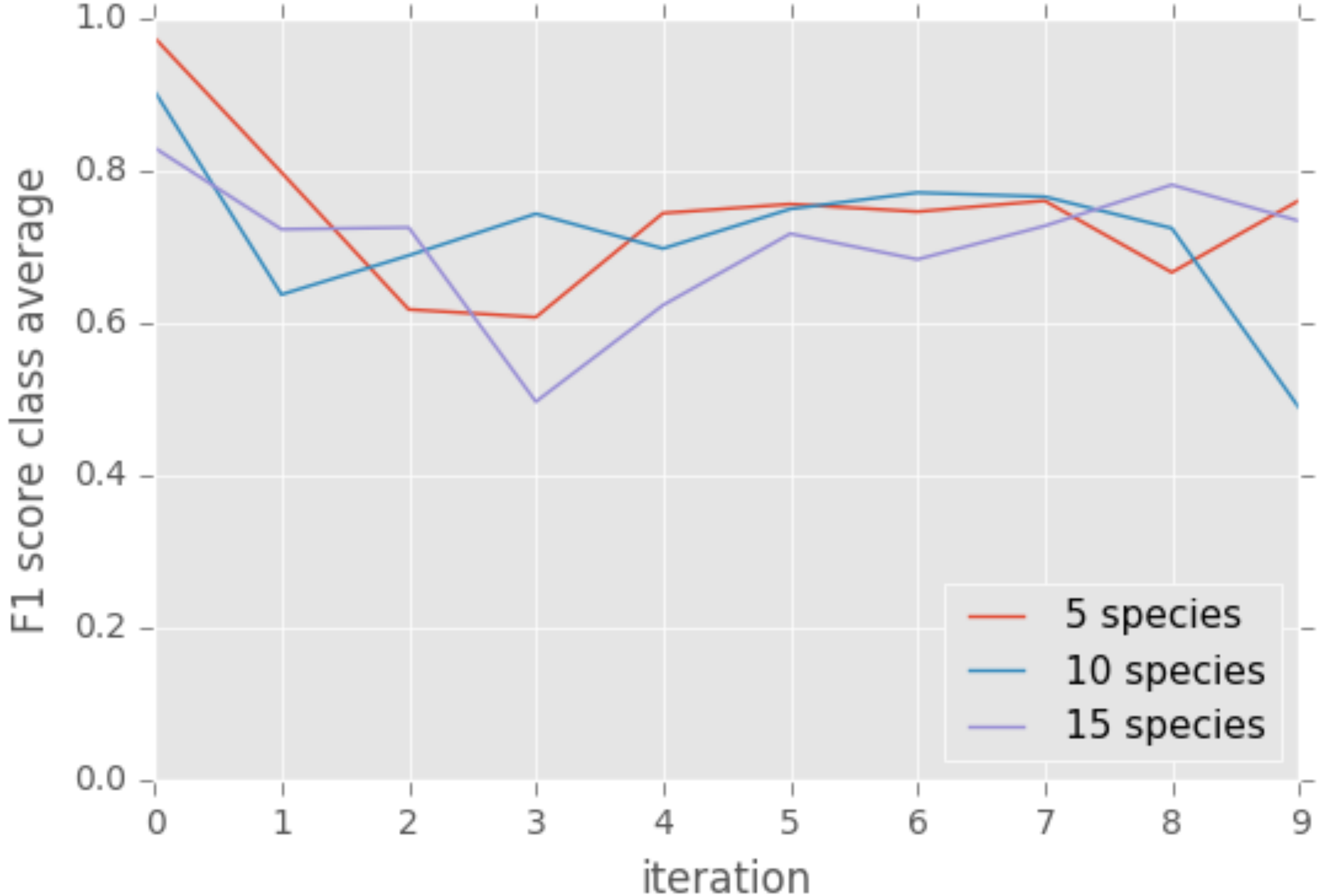
Decision tree search results

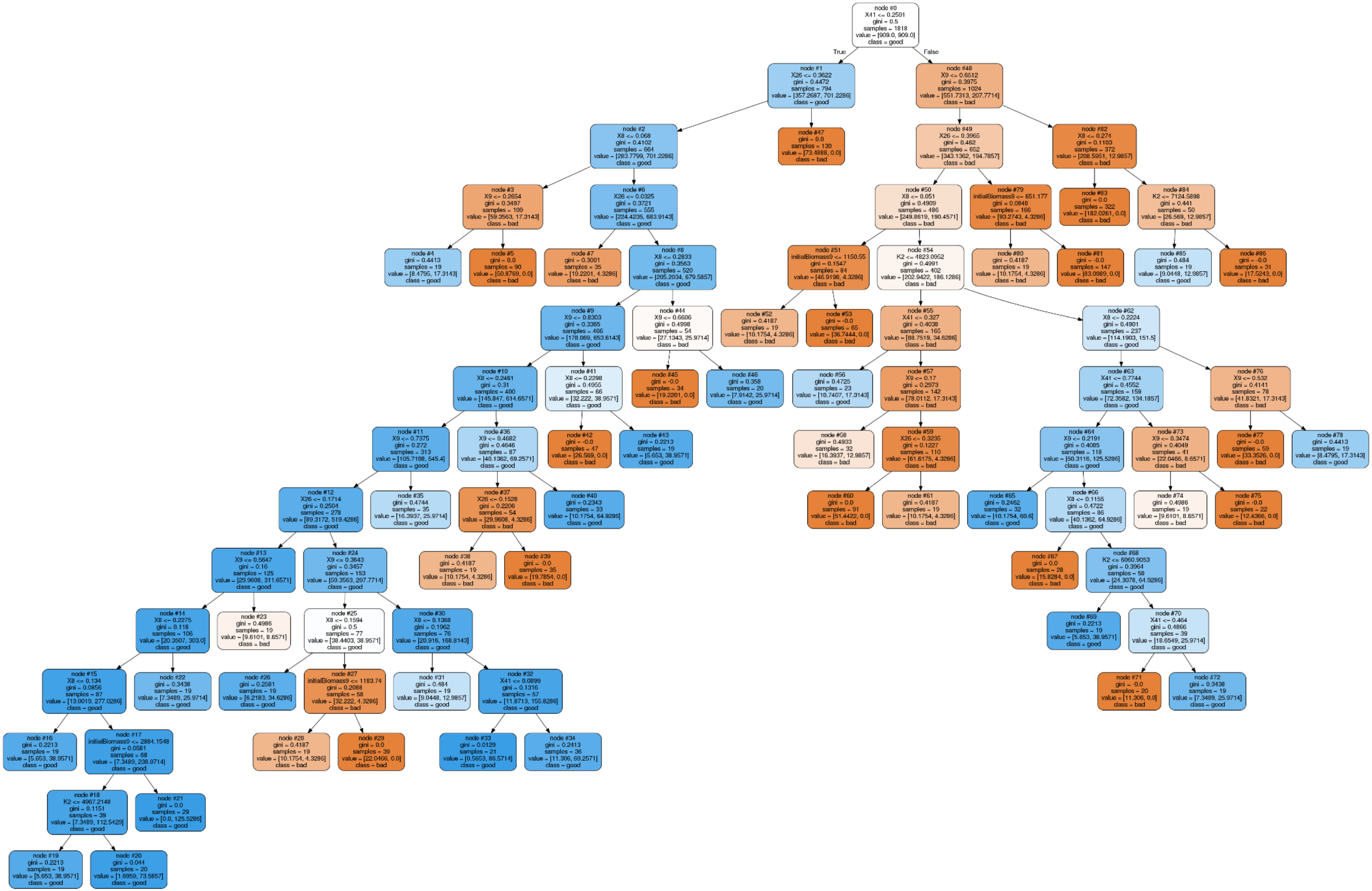


Progress in reducing extinctions



Classifier prediction performance





node #0 X41 <= 0.2591 gini = 0.5 samples = 1818 value = [909.0, 909.0] class = good
True
False
node #1 X26 <= 0.3622 gini = 0.4472 samples = 794 value = [357.2687, 701.2286] class = good
node #48 X9 <= 0.6512 gini = 0.3975 samples = 1024 value = [551.7313, 207.7714] class = bad
node #2 X8 <= 0.068 gini = 0.4102 samples = 664 value = [283.7799, 701.2286] class = good
node #47 gini = 0.0 samples = 130 value = [73.4888, 0.0] class = bad
node #49 X26 <= 0.3965 gini = 0.462 samples = 652 value = [343.1362, 194.7857] class = bad
node #82 X8 <= 0.274 gini = 0.1103 samples = 372 value = [208.5951, 12.9857] class = bad
node #3 X9 <= 0.2654 gini = 0.3497 samples = 109 value = [59.3563, 17.3143] class = bad
node #6 X26 <= 0.0325 gini = 0.3721 samples = 555 value = [224.4235, 683.9143] class = good
node #50 X8 <= 0.051 gini = 0.4909 samples = 486 value = [249.8619, 190.4571] class = bad
node #79 initialBiomass9 <= 651.177 gini = 0.0848 samples = 166 value = [93.2743, 4.3286] class = bad
node #83 gini = 0.0 samples = 322 value = [182.0261, 0.0] class = bad
node #84 K2 <= 7124.5898 gini = 0.441 samples = 50 value = [26.569, 12.9857] class = bad
node #4 gini = 0.4413 samples = 19 value = [8.4795, 17.3143] class = good
node #5 gini = 0.0 samples = 90 value = [50.8769, 0.0] class = bad
node #7 gini = 0.3001 samples = 35 value = [19.2201, 4.3286] class = bad
node #8 X8 <= 0.2833 gini = 0.3563 samples = 520 value = [205.2034, 679.5857] class = good
node #51 initialBiomass9 <= 1150.55 gini = 0.1547 samples = 84 value = [46.9198, 4.3286] class = bad
node #54 K2 <= 4823.0952 gini = 0.4991 samples = 402 value = [202.9422, 186.1286] class = bad
node #80 gini = 0.4187 samples = 19 value = [10.1754, 4.3286] class = bad
node #81 gini = -0.0 samples = 147 value = [83.0989, 0.0] class = bad
node #85 gini = 0.484 samples = 19 value = [9.0448, 12.9857] class = good
node #86 gini = -0.0 samples = 31 value = [17.5243, 0.0] class = bad
node #9 X9 <= 0.8303 gini = 0.3365 samples = 466 value = [178.069, 653.6143] class = good
node #44 X9 <= 0.6606 gini = 0.4998 samples = 54 value = [27.1343, 25.9714] class = bad
node #52 gini = 0.4187 samples = 19 value = [10.1754, 4.3286] class = bad
node #53 gini = -0.0 samples = 65 value = [36.7444, 0.0] class = bad
node #55 X41 <= 0.327 gini = 0.4038 samples = 165 value = [88.7519, 34.6286] class = bad
node #62 X8 <= 0.2224 gini = 0.4901 samples = 237 value = [114.1903, 151.5] class = good
node #10 X8 <= 0.2461 gini = 0.31 samples = 400 value = [145.847, 614.6571] class = good
node #41 X8 <= 0.2298 gini = 0.4955 samples = 66 value = [32.222, 38.9571] class = good
node #45 gini = -0.0 samples = 34 value = [19.2201, 0.0] class = bad
node #46 gini = 0.358 samples = 20 value = [7.9142, 25.9714] class = good
node #56 gini = 0.4725 samples = 23 value = [10.7407, 17.3143] class = good
node #57 X9 <= 0.17 gini = 0.2973 samples = 142 value = [78.0112, 17.3143] class = bad
node #63 X41 <= 0.7744 gini = 0.4552 samples = 159 value = [72.3582, 134.1857] class = good
node #76 X9 <= 0.532 gini = 0.4141 samples = 78 value = [41.8321, 17.3143] class = bad
node #11 X9 <= 0.7375 gini = 0.272 samples = 313 value = [105.7108, 545.4] class = good
node #36 X9 <= 0.4682 gini = 0.4646 samples = 87 value = [40.1362, 69.2571] class = good
node #42 gini = -0.0 samples = 47 value = [26.569, 0.0] class = bad
node #43 gini = 0.2213 samples = 19 value = [5.653, 38.9571] class = good
node #58 gini = 0.4933 samples = 32 value = [16.3937, 12.9857] class = bad
node #59 X26 <= 0.3235 gini = 0.1227 samples = 110 value = [61.6175, 4.3286] class = bad
node #64 X9 <= 0.2191 gini = 0.4085 samples = 118 value = [50.3116, 125.5286] class = good
node #73 X9 <= 0.3474 gini = 0.4049 samples = 41 value = [22.0466, 8.6571] class = bad
node #77 gini = -0.0 samples = 59 value = [33.3526, 0.0] class = bad
node #78 gini = 0.4413 samples = 19 value = [8.4795, 17.3143] class = good
node #12 X26 <= 0.1714 gini = 0.2504 samples = 278 value = [89.3172, 519.4286] class = good
node #35 gini = 0.4744 samples = 35 value = [16.3937, 25.9714] class = good
node #37 X26 <= 0.1528 gini = 0.2206 samples = 54 value = [29.9608, 4.3286] class = bad
node #40 gini = 0.2343 samples = 33 value = [10.1754, 64.9286] class = good
node #60 gini = 0.0 samples = 91 value = [51.4422, 0.0] class = bad
node #61 gini = 0.4187 samples = 19 value = [10.1754, 4.3286] class = bad
node #65 gini = 0.2462 samples = 32 value = [10.1754, 60.6] class = good
node #66 X8 <= 0.1155 gini = 0.4722 samples = 86 value = [40.1362, 64.9286] class = good
node #74 gini = 0.4986 samples = 19 value = [9.6101, 8.6571] class = bad
node #75 gini = -0.0 samples = 22 value = [12.4366, 0.0] class = bad
node #13 X9 <= 0.5647 gini = 0.16 samples = 125 value = [29.9608, 311.6571] class = good
node #24 X9 <= 0.3643 gini = 0.3457 samples = 153 value = [59.3563, 207.7714] class = good
node #38 gini = 0.4187 samples = 19 value = [10.1754, 4.3286] class = bad
node #39 gini = 0.0 samples = 35 value = [19.7854, 0.0] class = bad
node #67 gini = 0.0 samples = 28 value = [15.8284, 0.0] class = bad
node #68 K2 <= 6060.9053 gini = 0.3964 samples = 58 value = [24.3078, 64.9286] class = good
node #14 X8 <= 0.2275 gini = 0.118 samples = 106 value = [20.3507, 303.0] class = good
node #23 gini = 0.4986 samples = 19 value = [9.6101, 8.6571] class = bad
node #25 X8 <= 0.1594 gini = 0.5 samples = 77 value = [38.4403, 38.9571] class = good
node #30 X8 <= 0.1368 gini = 0.1962 samples = 76 value = [20.916, 168.8143] class = good
node #69 gini = 0.2213 samples = 19 value = [5.653, 38.9571] class = good
node #70 X41 <= 0.464 gini = 0.4866 samples = 39 value = [18.6549, 25.9714] class = good
node #15 X8 <= 0.134 gini = 0.0856 samples = 87 value = [13.0019, 277.0286] class = good
node #22 gini = 0.3438 samples = 19 value = [7.3489, 25.9714] class = good
node #26 gini = 0.2581 samples = 19 value = [6.2183, 34.6286] class = good
node #27 initialBiomass9 <= 1183.74 gini = 0.2088 samples = 58 value = [32.222, 4.3286] class = bad
node #31 gini = 0.484 samples = 19 value = [9.0448, 12.9857] class = good
node #32 X41 <= 0.0899 gini = 0.1316 samples = 57 value = [11.8713, 155.8286] class = good
node #71 gini = -0.0 samples = 20 value = [11.306, 0.0] class = bad
node #72 gini = 0.3438 samples = 19 value = [7.3489, 25.9714] class = good
node #16 gini = 0.2213 samples = 19 value = [5.653, 38.9571] class = good
node #17 initialBiomass9 <= 2884.1548 gini = 0.0581 samples = 68 value = [7.3489, 238.0714] class = good
node #28 gini = 0.4187 samples = 19 value = [10.1754, 4.3286] class = bad
node #29 gini = 0.0 samples = 39 value = [22.0466, 0.0] class = bad
node #33 gini = 0.0129 samples = 21 value = [0.5653, 86.5714] class = good
node #34 gini = 0.2413 samples = 36 value = [11.306, 69.2571] class = good
node #18 K2 <= 4967.2148 gini = 0.1151 samples = 39 value = [7.3489, 112.5429] class = good
node #21 gini = 0.0 samples = 29 value = [0.0, 125.5286] class = good
node #19 gini = 0.2213 samples = 19 value = [5.653, 38.9571] class = good
node #20 gini = 0.044 samples = 20 value = [1.6959, 73.5857] class = good

Future work

- Convergence hints: user study, dynamic version
- Steady state detection improvement
- Decision tree search improvements: class balance, classification accuracy, computational performance
- Study persistent chaotic dynamics
- Consider system-wide K for WoB and Convergence
- Machine learning approaches that generalize across different food webs

Immediate future work

- Evaluation of effects of an alternative producer growth function on steady state detection
- Evaluation of using promising regions from decision tree search to generate Convergence ecosystems