Python for Ecologists

- Assuming not much programming experience
- Immersion approach
  - Short lecture on Python topic
  - Hands-on Python exercises
  - Rinse & repeat
- Will use ecological examples as much as possible

Your presenters

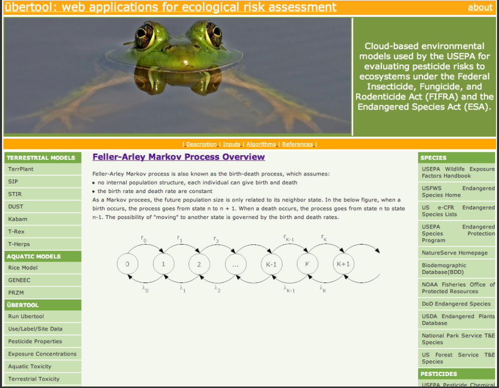
- Tom Purucker
- Tao Hong
- Chance Pascale

Why bother with Python when I have R?

- A scripting language (like R) but also,
- A high level programming language
- Designed to produce readable code
- Cross-platform

übertool Python project

- http://www.ubertool.org
- Created with Python as the science engine
- Integrates easily with web technologies such as HTML, JavaScript, JQuery



übertool: web applications for ecological risk assessment

about

Cloud-based environmental models used by the USEPA for evaluating pesticide risks to ecosystems under the Federal Insecticide, Fungicide, and Rodenticide Act (FIFRA) and the Endangered Species Act (ESA).

| Description | Inputs | Algorithms | References |

TERRESTRIAL MODELS
- TerrPlant
- SIP
- STIR
- DUST
- Kabam
- T-Rex
- T-Herps

AQUATIC MODELS
- Rice Model
- GENEEC
- PRZM

ÜBERTOOL
- Run Ubertool
- Use/Label/Site Data
- Pesticide Properties
- Exposure Concentrations
- Aquatic Toxicity
- Terrestrial Toxicity

Feller-Arley Markov Process Overview

Feller-Arley Markov process is also known as the birth-death process, which assumes:

- no internal population structure, each individual can give birth and death
- the birth rate and death rate are constant

As a Markov procees, the future population size is only related to its neighbor state. In the below figure, when a birth occurs, the process goes from state n to n + 1. When a death occurs, the process goes from state n to state n-1. The possibility of "moving" to another state is governed by the birth and death rates.

SPECIES
- USEPA Wildlife Exposure Factors Handbook
- USFWS Endangered Species Home
- US e-CFR Endangered Species Lists
- USEPA Endangered Species Protection Program
- NatureServe Homepage
- Biodemographic Database(BDD)
- NOAA Fisheries Office of Protected Resources
- DoD Endangered Species
- USDA Endangered Plants Database
- National Park Service T&E Species
- US Forest Service T&E Species

PESTICIDES
- USEPA Pesticide Chemical

Figure: übertool ecological risk web application

Getting setup

- We will use Python 2.7 (not 3)
  - http://www.python.org/getit/
- For Windows users
  - http://portablepython.com/wiki/Download

Test a script to see if python is working

```
# save this in a text file as hello.py
print "Hello␣world!"
# then run at the command prompt with
# python hello.py
```

Some extra libraries to install

- numpy- http://sourceforge.net/projects/numpy/
- scipy- http://sourceforge.net/projects/scipy/files/

Need a text editor

- Linux
- Mac
  - TextWrangler
  - Smultron
  - TextEdit (already installed)
- Windows
  - Notepad (already installed)
  - Notepad++
  - TextPad

Python IDLE IDE

- http://www.ubertool.org
- Created with Python as the science engine

Download the exercises for this class

- http://www.ubertool.org
- Created with Python as the science engine

Python objects

- Everything in Python is an object with these properties
  1. an identity (id)
  2. a type (type)
  3. a value (mutable or immutable)

Each Python object has an id

```
>>> n_predators = 12
>>> id(n_predators)
4298191056
```

Each Python object has a type

```
>>> n_predators = 12
>>> type(n_predators)
<type ’int’>
```

Each Python object has a value

- String, integer, and tuple object values are *immutable*

```
>>> n_prey = 88
>>> id(n_prey)
4298193184
>>> n_prey = 96
>>> id(n_prey)
4298192992 # id for n_prey has changed
```

- Dictionary and list items are *mutable*

```
>>> birds = ["cardinal", "oriole"]
>>> id(birds)
4332756000
>>> birds.append("gnatcatcher")
>>> id(birds)
4332756000 # id is still the same
```

Variables

```
pop_size = 112 # integer
pop_density2 = 4 # still an integer
pop_density = 4. # float
species_name = "Oedipina␣complex" # string
species_name = "4" # still a string
```

Python variable naming conventions

- all lowercase
- cannot start with numbers
- separate_words_with_underscores
- Style Guide for Python:
    - http://www.python.org/dev/peps/pep-0008/

Exercise 1- exer01_variables.py

```
import unittest

class TestVariables(unittest.TestCase):
    def test_variables(self):
        # create the variable ‘diffusion_rate‘,
        # and assign it a float value
        # ****************************************

        self.assert_(isinstance(diffusion_rate, float))

        # re−assign ‘diffusion_rate‘ to an integer value
        # ****************************************

        self.assertEqual(diffusion_rate, 6)
        self.assert_(isinstance(diffusion_rate, int))

        # create a variable ‘species_name‘,
        # and assign it to ’Pieza kake’
        # ****************************************

        self.assertEqual(species_name, "Pieza␣kake")
        self.assertTrue(isinstance(b, str))

if __name__ == ’__main__’:
    unittest.main()
```