

Just Enough Python, I

What does Python have to do with the goal of this course?

A few words about modeling, and learning how to do it

Random numbers as the "life force" of simulation models

Using random numbers in Python

Python jargon 101: names, objects, methods, functions

Testing hypotheses by modeling: the MLB simulation

Anthro/Biol 5221
29 August 2008

What does Python have to do with the goal of this course?

Goal = *how to think* (more than *what to think*) about the process of evolution.

Thinking involves developing, applying, testing *theory*.

Theories are *models* that *represent* nature (simplified, abstracted).

For evolution, we have an excellent general model:
change in the relative frequencies of alleles at genetic loci.

So our specific models need to *represent*
(1) genes and their alleles, and
(2) the processes that change the frequencies of alleles
(mutation, genetic drift, natural selection, recombination).

Models *succeed* to the extent that their representations are "lifelike"
(i.e., "predict" facts about variation within and divergence between species).

Computing is essential *both* to developing models *and* to analyzing data.

What is computing?

At lowest level: flipping switches! (0,1)

At higher level: simple operations on numbers and letters

The secret: these can be used to represent almost anything!

A few words about modeling, and learning how to do it

All representations (of anything) are abstractions
(simplify reality, ignore many details)

Is this responsible? Why pretend we don't know a part of the *true story*?

Actually, it's *essential*, and our brains do it automatically.

But abstractions are of *many kinds*, and not all are equally helpful!

And different ones may be good for different purposes.

Example: the many ways to represent a "gene" (genome sequence to "A", "a")

Your goal: to learn the "arts of reductive abstraction" for evolutionary genetics.

But you can't learn it by listening to us talk about it!

You can only learn it by trying to do it. (Like riding a bicycle.)

Expect feelings of frustration and emptiness! But don't despair!

Random numbers as the "life force" of simulation models

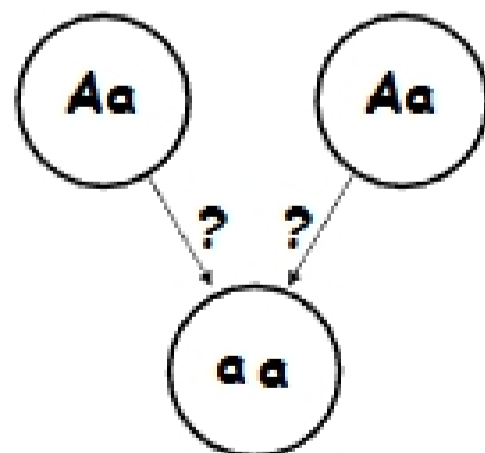
Evolution is all about *events* that happen to *genes* in *populations*:

- births
- deaths
- mutations
- recombinations

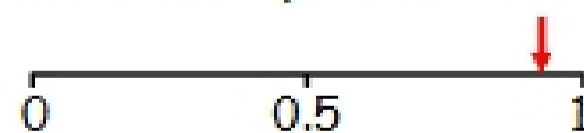
These events occur with *probabilities* that we can describe and estimate.

We can also *represent* them on the computer, using "random numbers".

Example: modeling a birth (Mendel's law of segregation)



Use a uniformly distributed random number to choose the left parent's allele



Use another to choose the right parent's allele



Python jargon 101: names, objects, methods, functions

(1) Names are born when you assign values to them.

```
x = 0
```

(2) They must begin with a letter (or underscore), and contain only letters, digits and underscores.

```
x_1 and _1x are legal, but 1_x and x-1 are not.
```

(3) Upper- and lower-case letters are different from each other!

```
"A" is not "a".
```

(4) You can't use the Python "reserved words":

and	del	for	is	raise
assert	elif	from	lambda	return
break	else	global	not	try
class	except	if	or	while
continue	exec	import	pass	yield
def	finally	in	print	

```
"And" would be legal, but why play with fire?
```

(5) Names do not have "types", but the *objects* they refer to do.

```
x = 0 means "x refers to the integer value 0".
```

```
x = 0.1 means "x refers to the floating-point number 0.1".
```

```
x = "Spam" means "x refers to the string of characters "Spam".
```

(6) The fundamental object types are numbers (integers, floats), strings of characters, lists (ordered arrays of other objects), dictionaries (arrays of paired objects), tuples (like lists, but simpler), and files.

(7) Each type of object has *methods* that it "knows" how to perform on itself.

```
For example, list.sort() sorts a list in some appropriate way.
```

(8) *Functions* are like methods, but they *return a value*.

```
sqrt(x) returns the square root of x
```

```
random() returns a random number
```