HMC Summer Research Celebration

Curious about research opportunities at HMC?

Want to learn more about your friend's summer project?

Come to the poster session to learn about projects happening across campus!

Thursday Sept 23
Drop by anytime between
4:30 - 6:30 pm

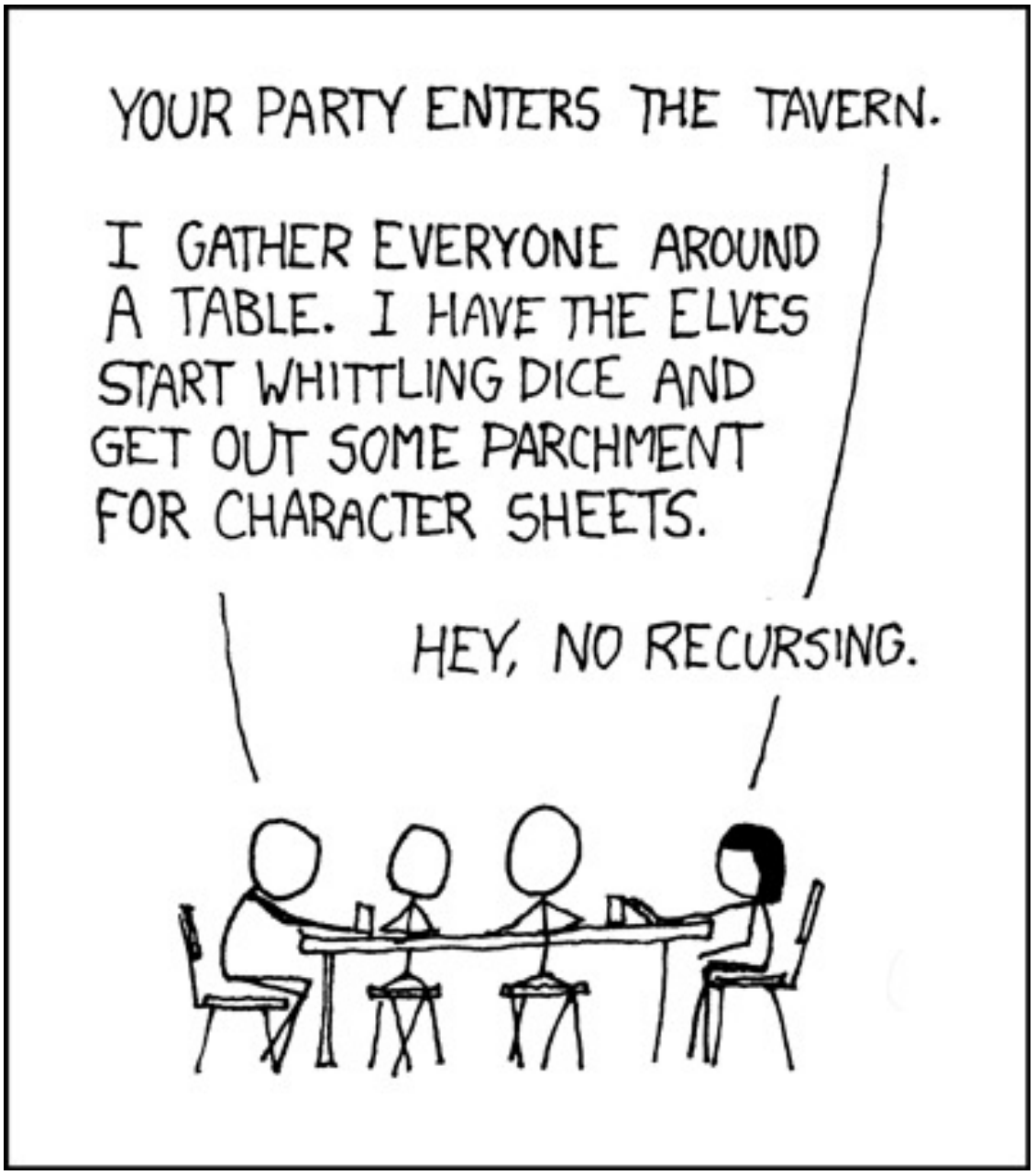
Attendees are eligible to win raffle prizes

Zoom Meeting 868 2909 2950, Passcode D5sDRH
Make sure you have the latest version of Zoom

CS 5 Green

Learning Goals

- Biology: Provide background for homework
- CS: Practice recursion



https://xkcd.com/244/

Recursion Recap

Step 1: Define the base case(s)

- Identify the simplest input(s) possible.
- Figure out how to handle the simplest input(s)

Step2: Handle the general case(s)

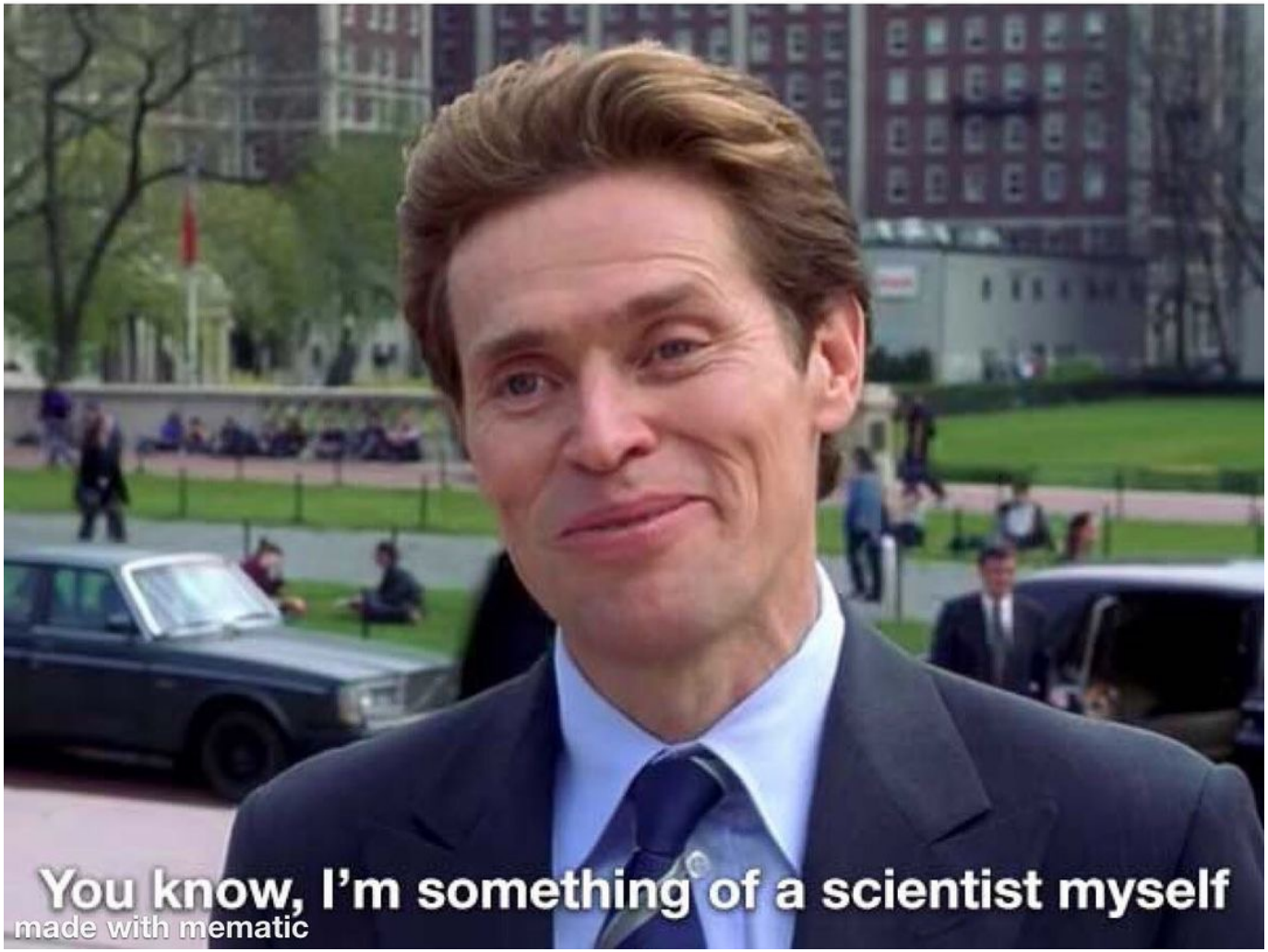
- Each recursive call should get closer to the base
- Begin building towards the desired output

The Power of Recursion…


```
>>> pow(10, 3)
1000

>>> pow(10, 0)
1

>>> pow(2, 5)
32

def pow(base, exp):    # assume exp is ≥ 0
```

Mitochondria

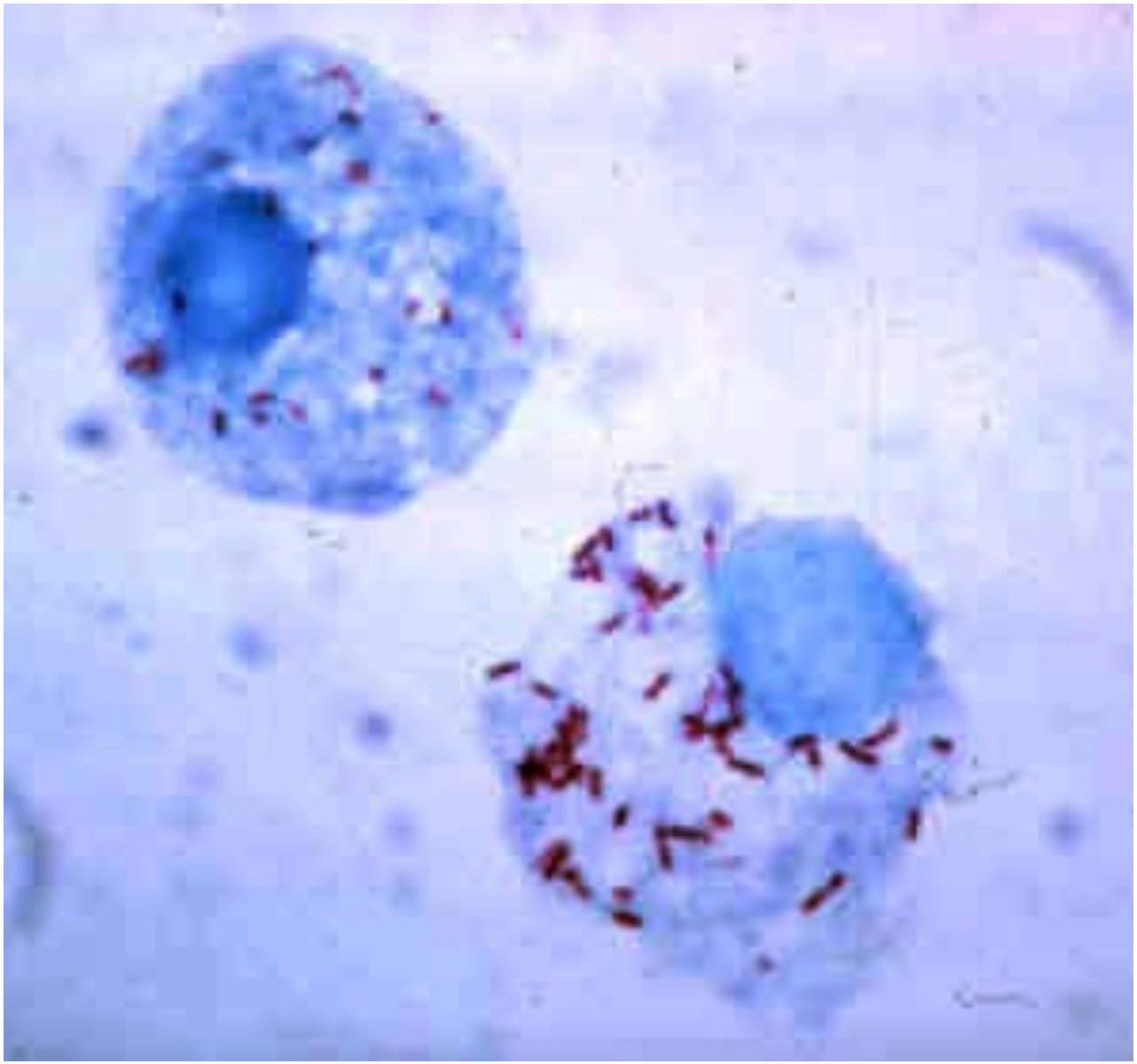
When you remember that the mitochondria is the powerhouse of the cell



Mitochondria are bacteria

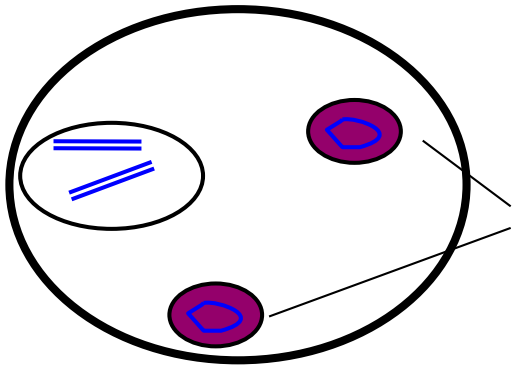
Rickettsia ricketsii

Rocky Mountain Spotted Fever

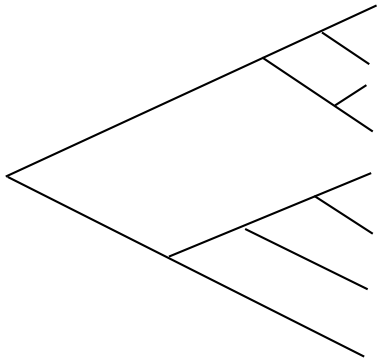


https://commons.wikimedia.org/w/index.php?curid=1378899

Mitochondrial eve and out of Africa



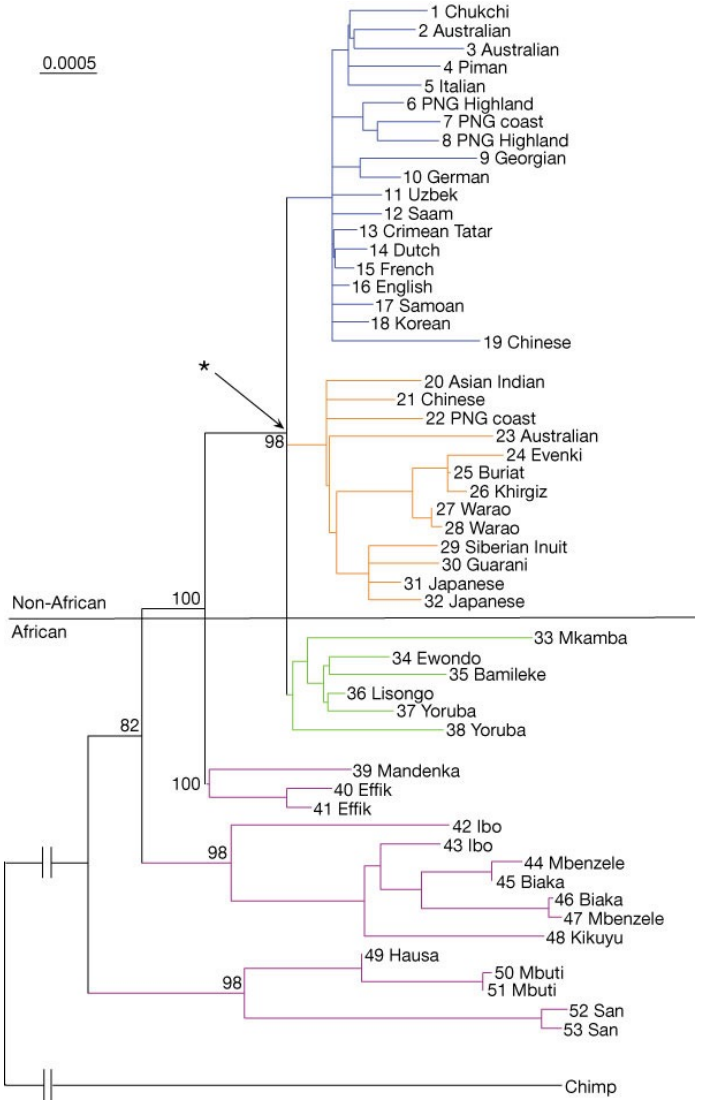
mitochondria



tree of mitochondrial inheritance

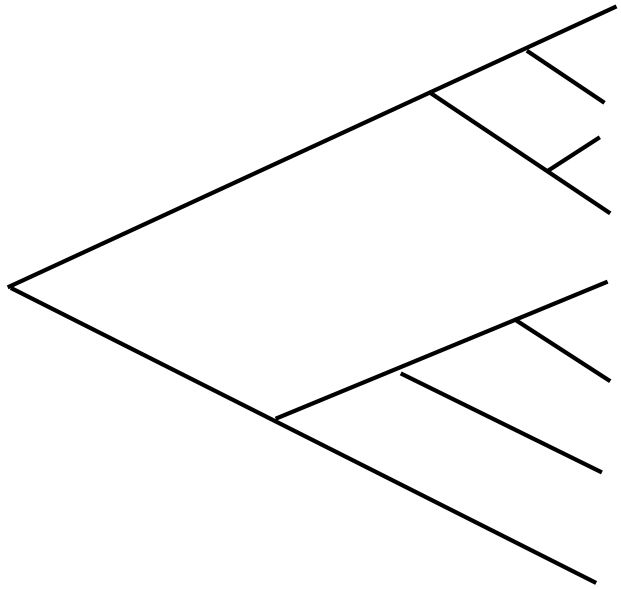
- Mitochondrial genome has maternal inheritance
- Tree leads back to a single woman: “mitochondrial eve”

Mitochondrial eve and out of Africa

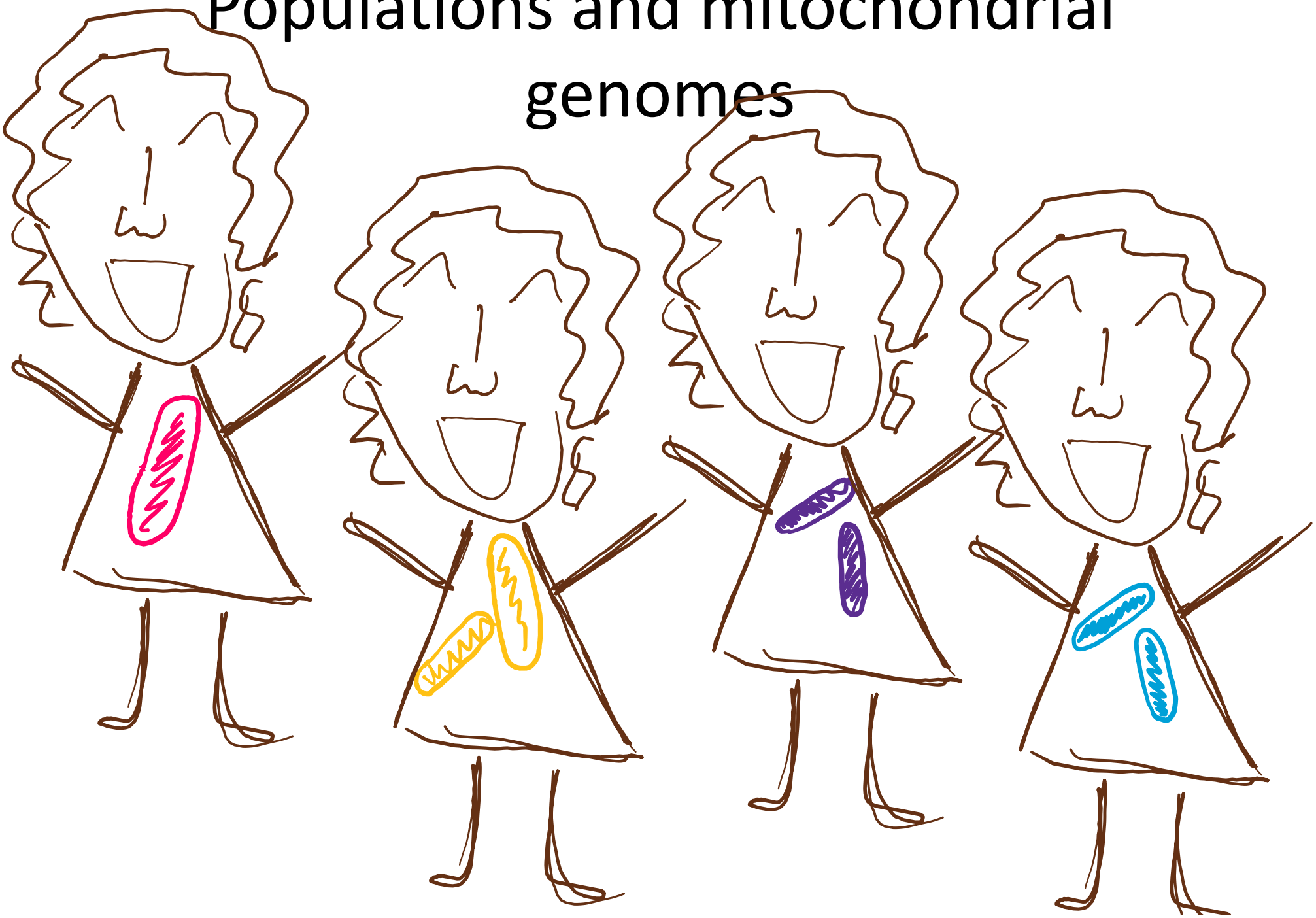


http://www.nature.com/nature/journal/v408/n6813/fig_tab/408708a0_F2.html
http://en.wikipedia.org/wiki/File:Migration_map4.png

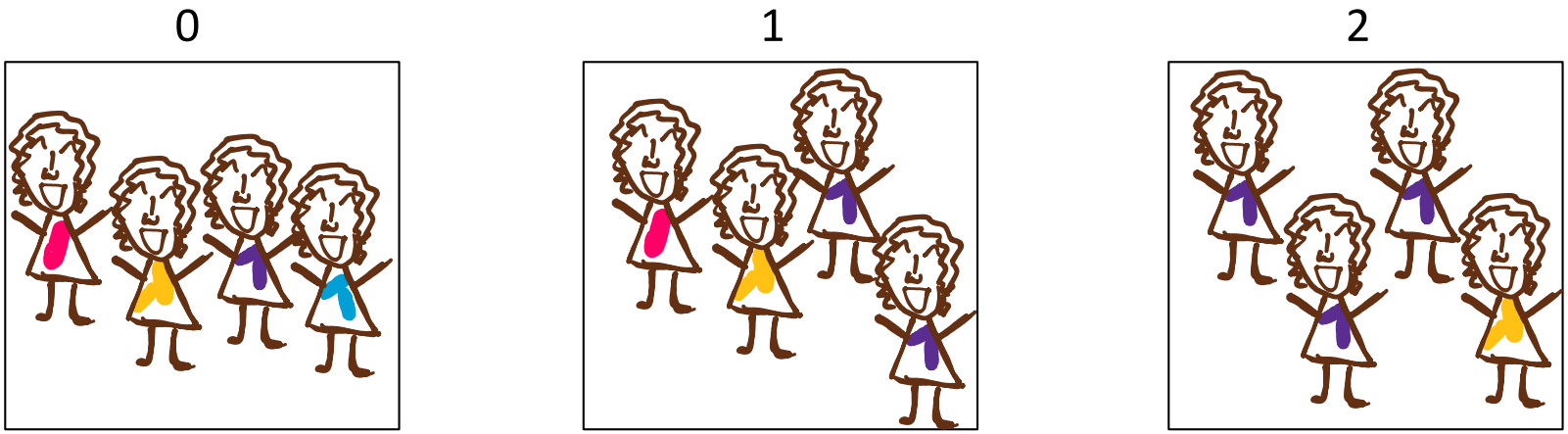
Why do all mitochondrial genomes today descend from a single woman?



Populations and mitochondrial genomes



A simple model of a population over time



- Fixed population size
- Haploid: have one copy of each sequence
- Asexual reproduction
- Make next generation by sampling with replacement

Representing a population on a computer

```
>>> popL = list(range(5))
>>> popL
[0,1,2,3,4]
```

We can sample the list using the `random` module

```
>>> import random
>>> random.choice(popL)
3  # picked a random number from the list
>>> popL
[0,1,2,3,4]  # original list remains unchanged
```

No recursion here!


```
import random

def nextGen(popL):
    """Given the population of the current generation, obtains
    next generation by sampling with replacement."""
```

`uniquify`, version 1 (iteration)

Worksheet

The **in** syntax might be useful:

```
>>> 9 in [4,2,7]
False
```

```
>>> uniquify1([9,9,7,5,3,5,7])
[9, 3, 5, 7]
```

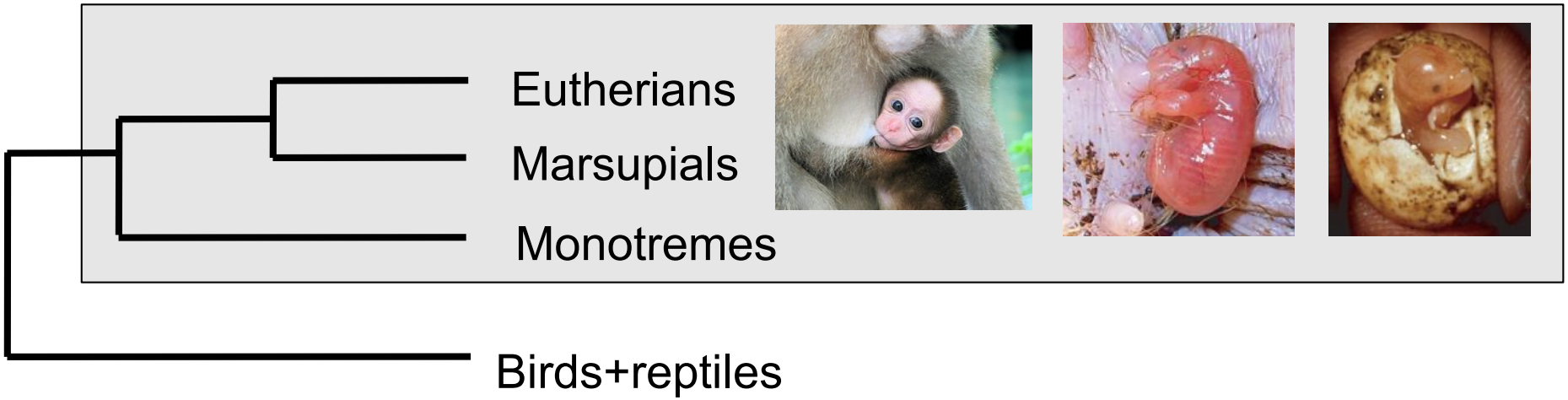
```
def uniquify1(inputL):
    '''Returns a list of the unique elements from L.
       Uses iteration and does not use recursion'''
```

`uniquify`, version 2 (recursive)


```
>>> uniquify2([9,9,7,5,3,5,7])
[9, 3, 5, 7]
```

```
def uniquify2(inputL):
    '''Returns a list of the unique elements from inputL.
       Uses recursion and does not use iteration'''
```

Evolutionary history of milk



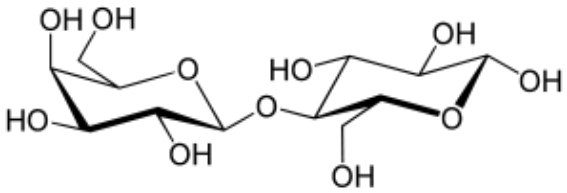
- All mammals have it, other vertebrates don't
- More than 200 million years old
- Most mammals can only consume milk as infants!

10,000 year old dirty dishes

Evershed et al. 2008; Nature
https://openclipart.org/detail/17676/cartoon-cow

Got milk?

The NIH estimates only 32% of human **adults** can digest lactose

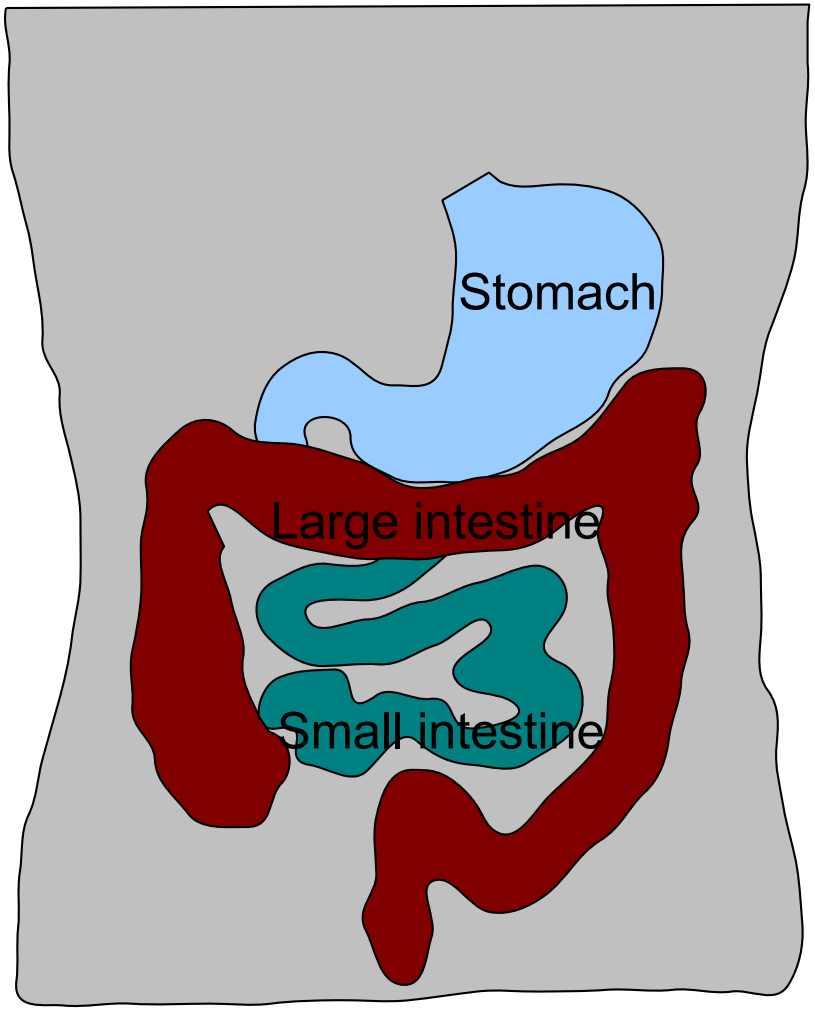


Lactose

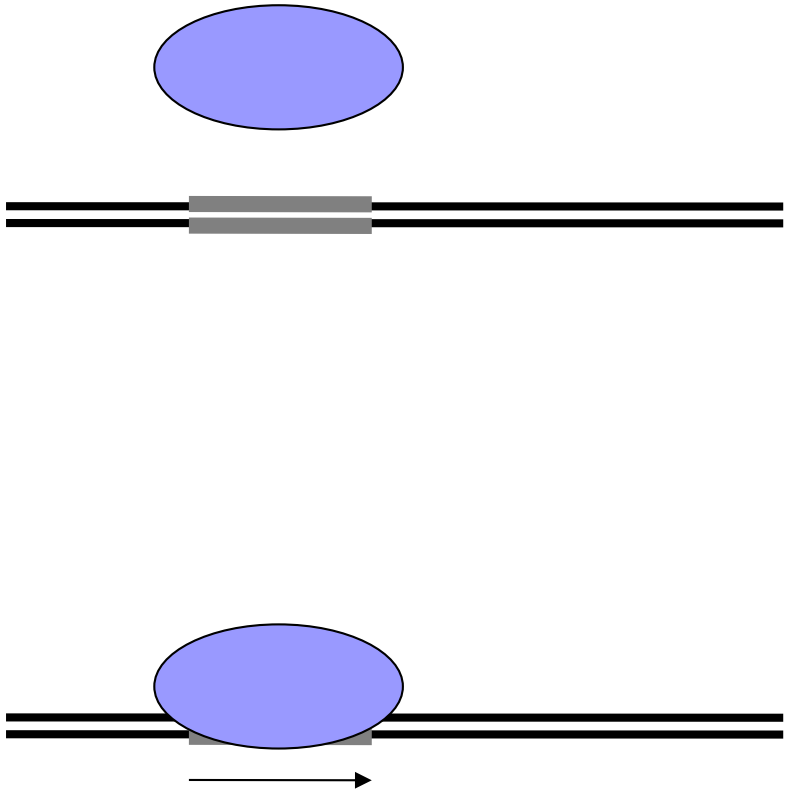
http://dx.doi.org/10.1038/4371241a
http://en.wikipedia.org/wiki/File:Beta-D-Lactose.svg

Got milk?

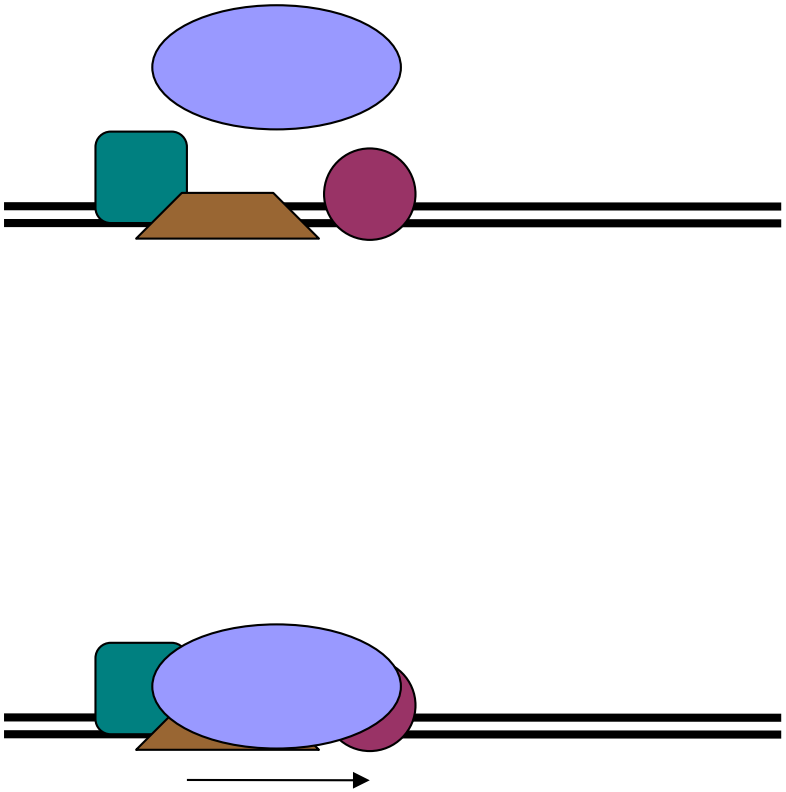
- Lactase breaks down lactose in small intestine
- Most children can digest lactose
- A few adults can and most can't



Gene regulation in five minutes

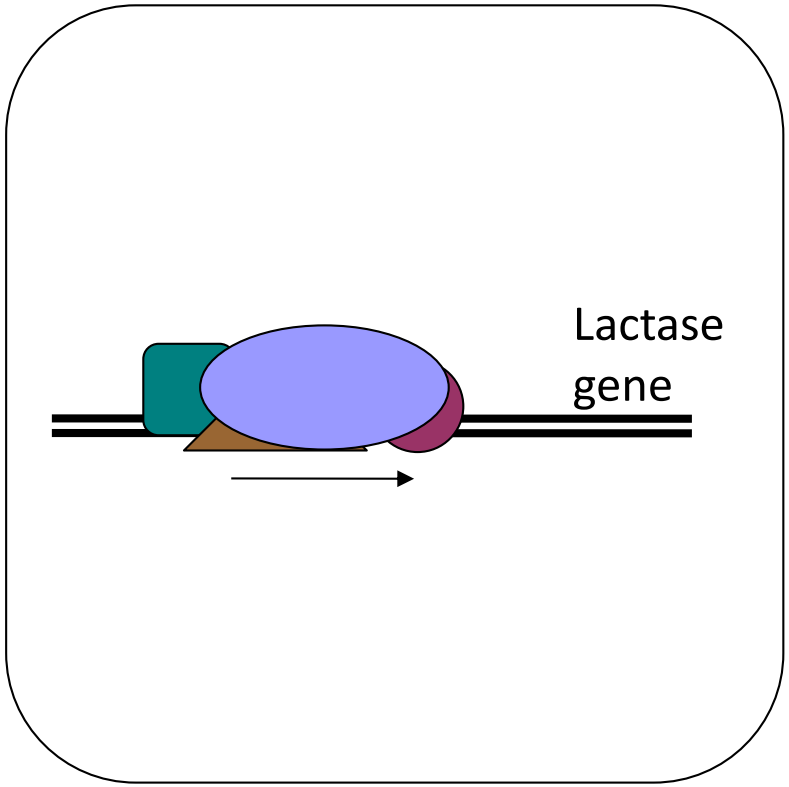


Gene regulation in five minutes

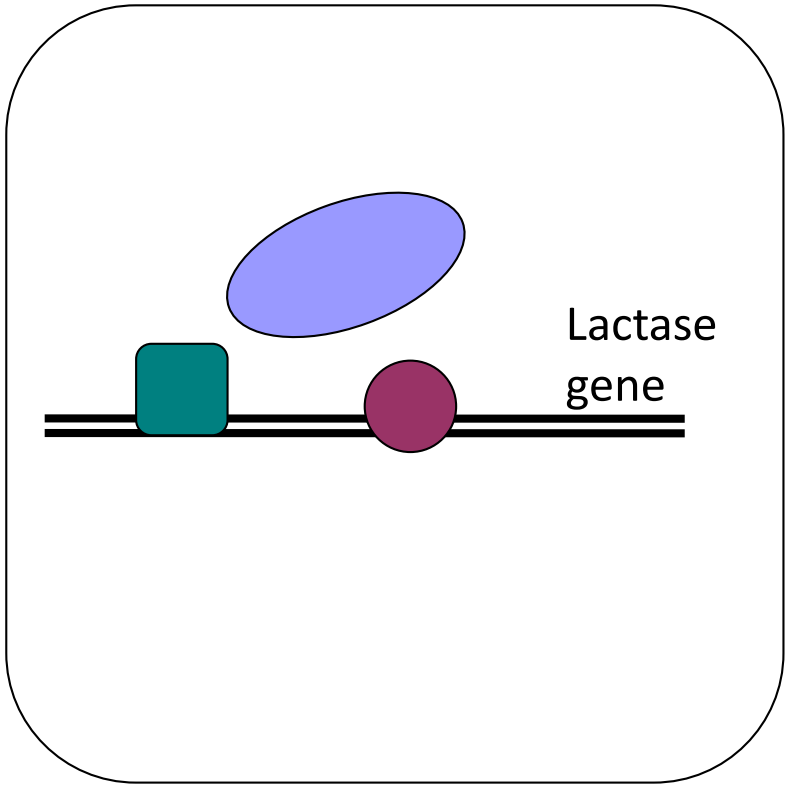


Gene regulation: cell type specific expression

Enterocyte cell of small intestine in infant



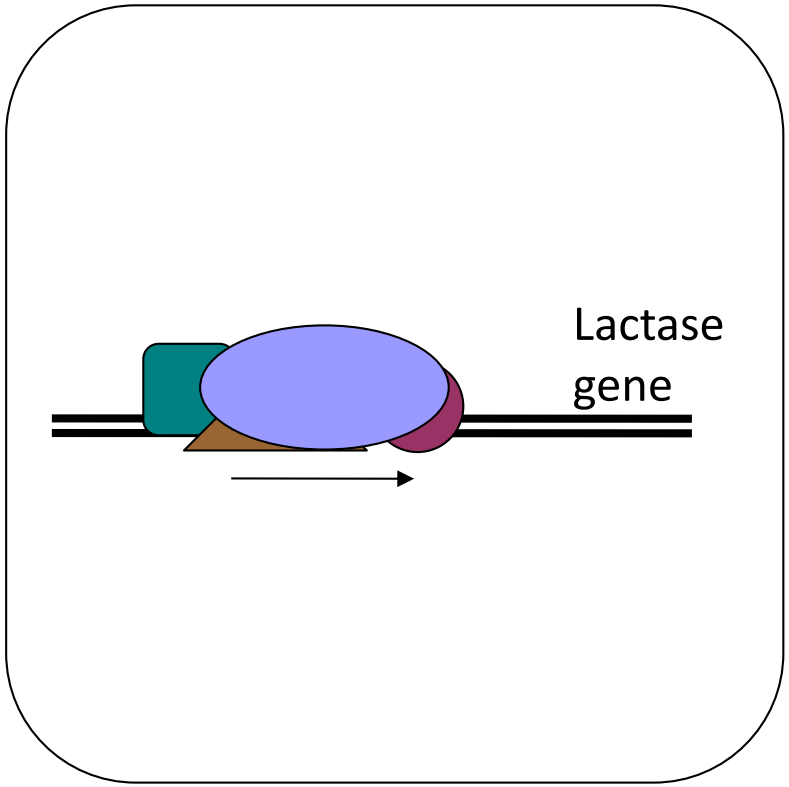
Other cell type in infant



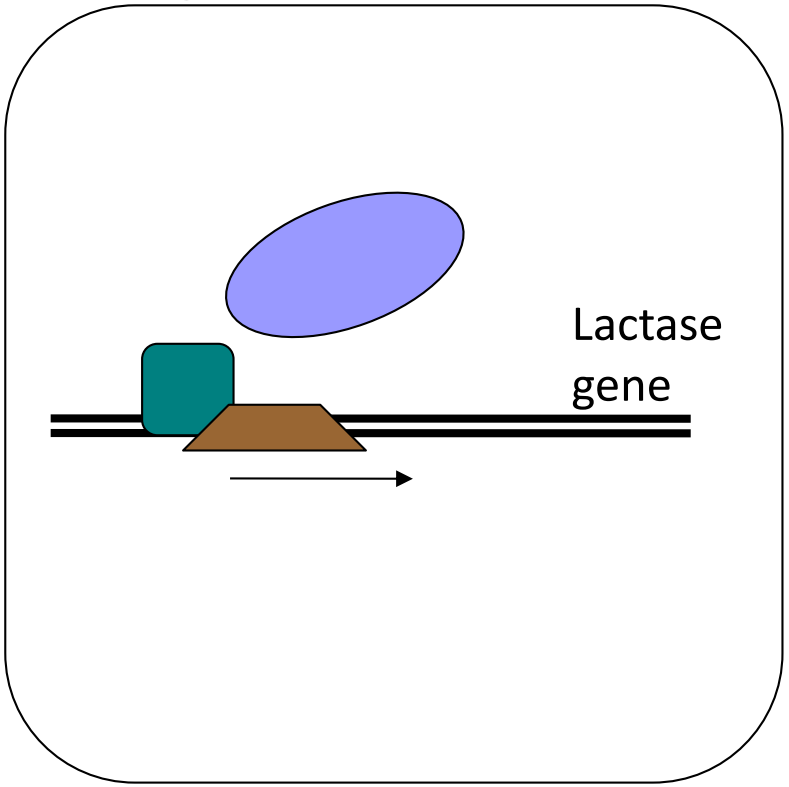
missing in this cell

Gene regulation: age specific expression

Enterocyte cell of small intestine in infant

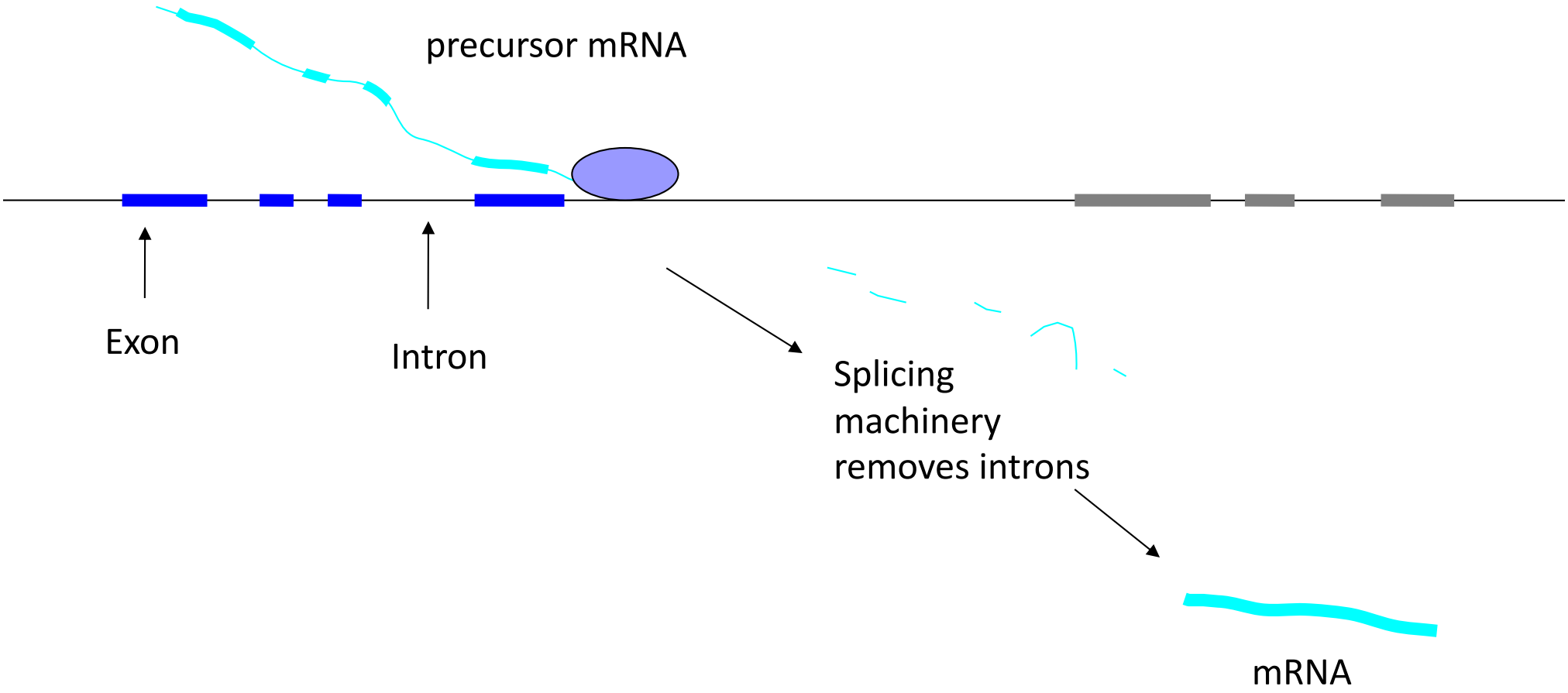


Enterocyte cell of small intestine of adult who can't digest milk.



missing in this cell

More on genes: a human gene



Using sequencing to see if someone expresses lactase

Obtain RNA from enterocyte cells

Convert to DNA and sequence

AATCCAAAGACATATACATCCCATAAGAACTAGGCCAGGCACGGTGGCTC

GATTGGGAGTTTGAGACCAGCCTGACCAACATGGAGAAACCCCATCTCTA

AGCTACTCAGGAGGCTGAGAAAGAAGAATCGCTTGAACCTGGGAGGCAGA

GTGAGACTCCATCTCAAAAACAAAACAAAACAAAAAAACAACCAGCAATG

AACATGATAAAATATCAAGAAATTCAACAAGAAACACTGAAAAACATATG

ATAACGGTATTTGTGGTACTCAGAACTGTTCACAAATGTTTGCTCTTCTA

Millions of short reads

Map onto reference human genome

The following pages have a number of exercises for you to do (in your notes). You're welcome to work at your own pace.

```
min
member
pal
```

[ ,  , ]

Minimum!


```
>>> min([372, 112, 42, 451])
42
>>> min([16])
16
```

Assume that the input list will never be empty! Use len as a helper function!

```
def min(inputL):
    '''Returns smallest value in a list'''
```

member


```
>>> member(42, [1, 3, 5, 42, 7])
True
>>> member(42, ['spam', 'is', 'yummy'])
False
```

This is sort of like the “in” thing in Python, but don’t use “in” here. Just list indexing, slicing, and recursion!

```
def member(thing, inputL):
    '''Return True if thing in inputL
    and False otherwise.'''
```

Palindrome?


```
>>> pal('radar')
True
>>> pal('amanaplanacanalpanama')
True
>>> pal('spam')
False


def pal(s):
    '''Returns True if s is a palindrome
    and False otherwise'''
```

Insertion Sorting


```
>>> sort([42, 57, 1, 3])
[1, 3, 42, 57]
```

The idea… Given a list like L = `[42, 57, 1, 3]`

- Slice off the first element. Now we have a shorter list… `[57, 1, 3]`
- Use recursion to sort that list. Now we have… `[1, 3, 57]`
- Now, insert L[0] (which is 42)into the right place in `[1, 3, 57]`…
  `[1, 3, 42, 57]`

```
def insert(x, sorted_list):
    '''Takes a number and sorted list as input and returns a new list
    that has x inserted into the right place in the sorted list'''

def sort(my_list):
   '''Sorts a list'''
```