While you are waiting...

- **socrative.com,** room number **1f2864a3**

First, an apology for Lab 4

- I said: “you can change the values of parameters (e.g. send_weighting, receive_weighting, mutation_rate) by changing them in the script and then re-running the script, **or by typing new values at the prompt**”

- That was true under the python environment we used to use

- But NOT under Canopy!!!

- You can type in a new value at the prompt, but when you call a function from the script, it uses the value defined in the script.

- Sorry! I’ll figure out how to work around this.

Simulating Language
Lecture 4: When will optimal signalling evolve?

Kenny Smith

kenny.smith@ed.ac.uk

Lab 3 worksheet

---

1. The two ways of scoring an agent's success depend on being understood (the first number), and understanding (the third number). What are the ecological interpretations of these scores? Which do you think are evolutionarily significant, and why?

- **Fitness for a signaller will be determined by ...**

  A: success as a sender only

  B: success as a receiver only

  C: both

  D: neither

socrative.com, 1f2864a3

Lab 3 worksheet

---

2. Can you construct a population where every agent gets approximately the same score for being understood, but different scores for understanding? What about the other way round?

A: I managed to do this

B: I didn’t manage to do this

C: I didn’t try this yet

socrative.com, 1f2864a3

Key point here for today’s lecture: sending and receiving are decoupled in our model

Lab 3 worksheet

4. Who communicates with who in a population? What other ways could you model this, and how would you start adjusting the code to implement your model? Hint: what if people only talked to people who were ‘near’ them?

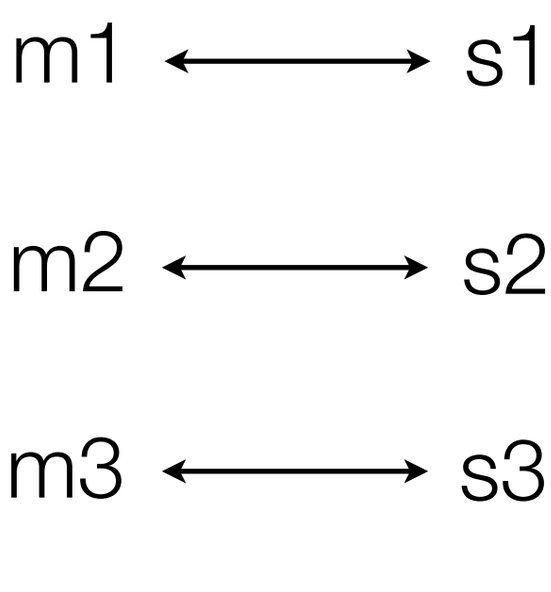
Optimal communication

- Oliphant (1996) talks about “Saussurean” signalling as the ideal communication system:

  - “What is important is that each signal ‘means’ the same thing to both the individual sending it and the individual receiving it. It must be possible to map some concept onto a symbol and then map back from the symbol to get the original concept.” (Oliphant 1996)

	s1	s2	s3
m1	1	0	0
m2	0	1	0
m3	0	0	1

	m1	m2	m3
s1	1	0	0
s2	0	1	0
s3	0	0	1



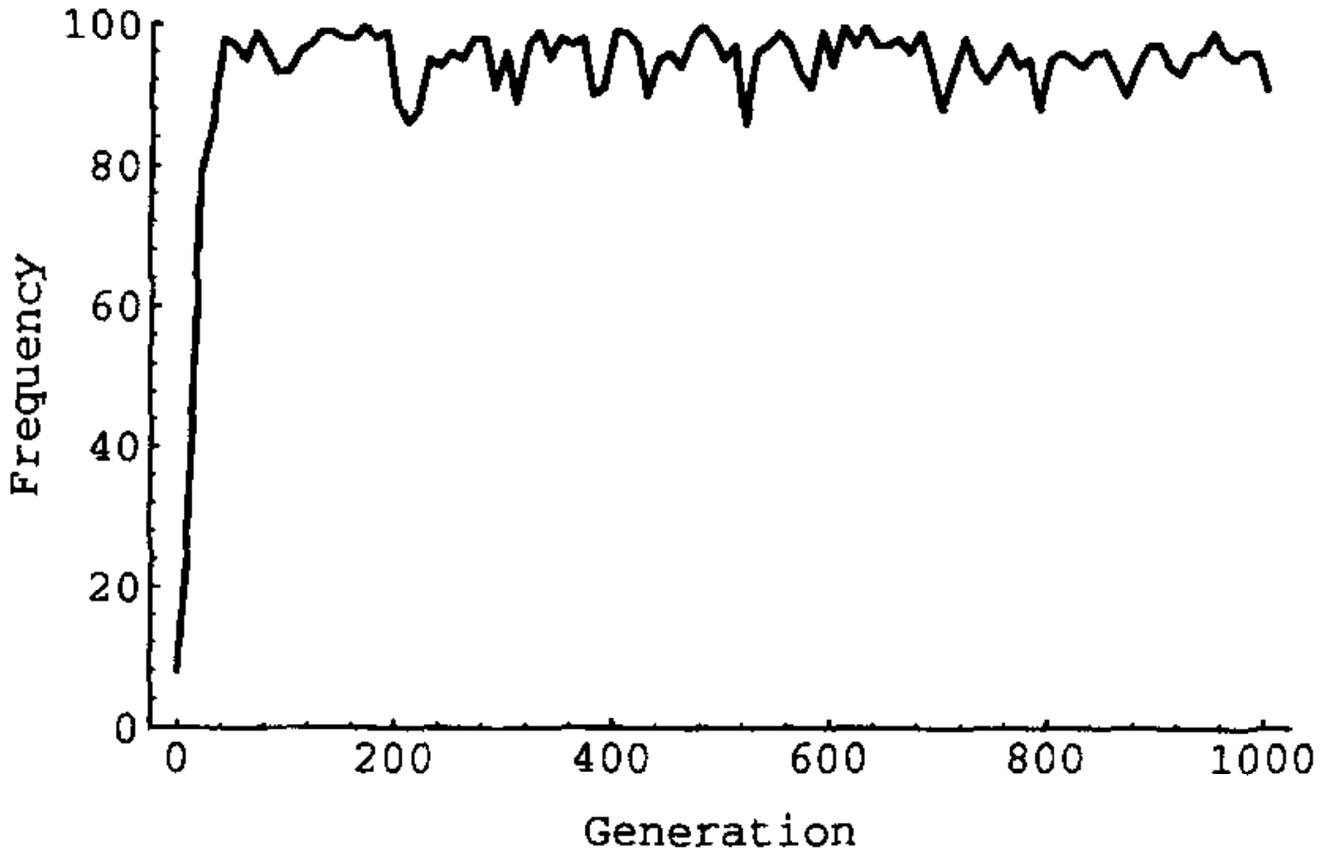
Optimal does not mean inevitable

- Natural selection does not necessarily create optimal solutions!
- Saussurean signalling is not the inevitable result of evolution
- Oliphant aims to show that it can only emerge given **specific conditions**

Oliphant's simulation 1

- Simplified variant of our model, with two signals, two meanings, and deterministic mappings between the two

- With fitness based on both sending success and receiving success, optimal communication evolves:



Note: not a fitness graph. Measures frequency of one of two optimal systems

m1 ⟷ s2

m2 ⟷ s1

How to get communication, solution 1:

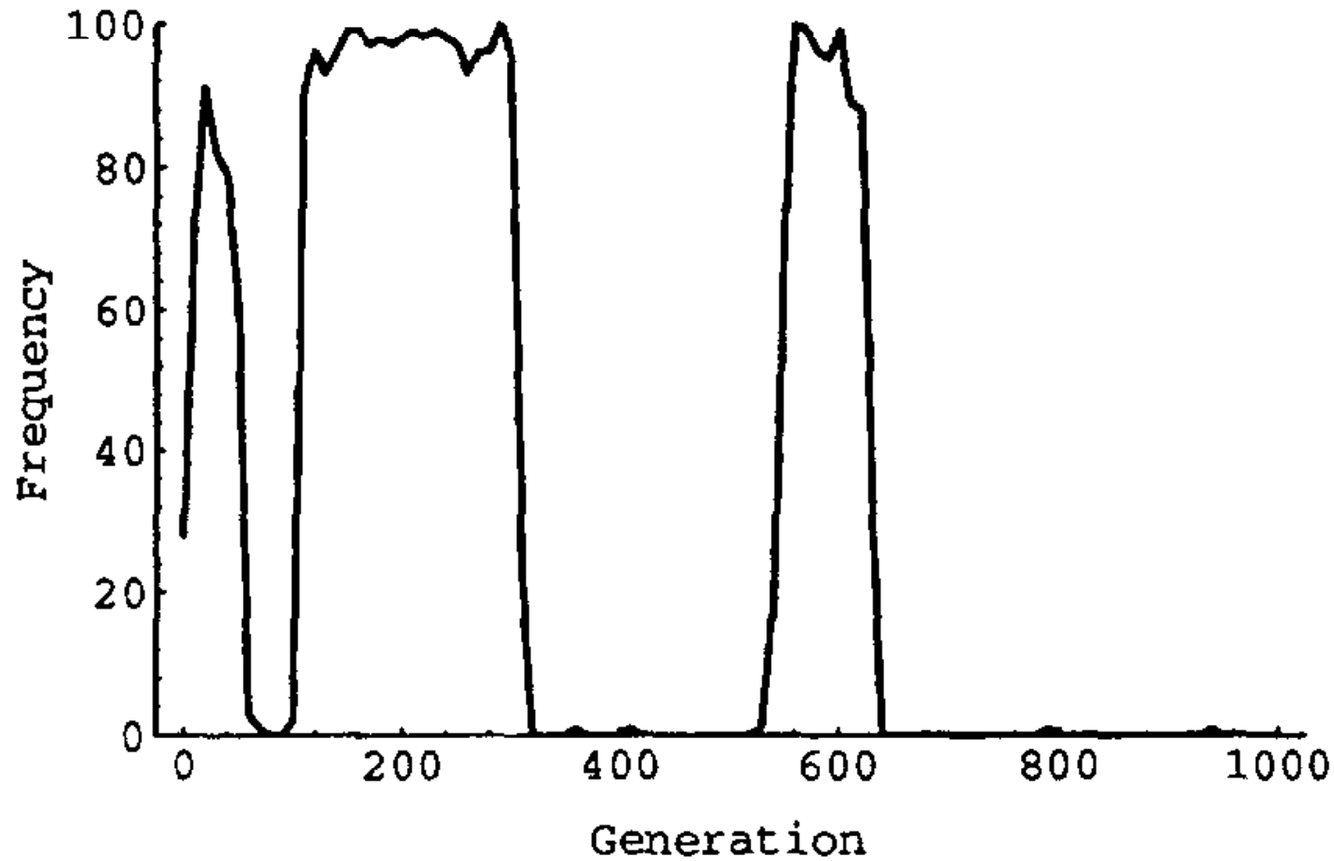
mutual benefit

Oliphant's simulation 2

- Is mutual benefit realistic? What benefit does a vervet monkey get for producing an alarm? Is there a cost?

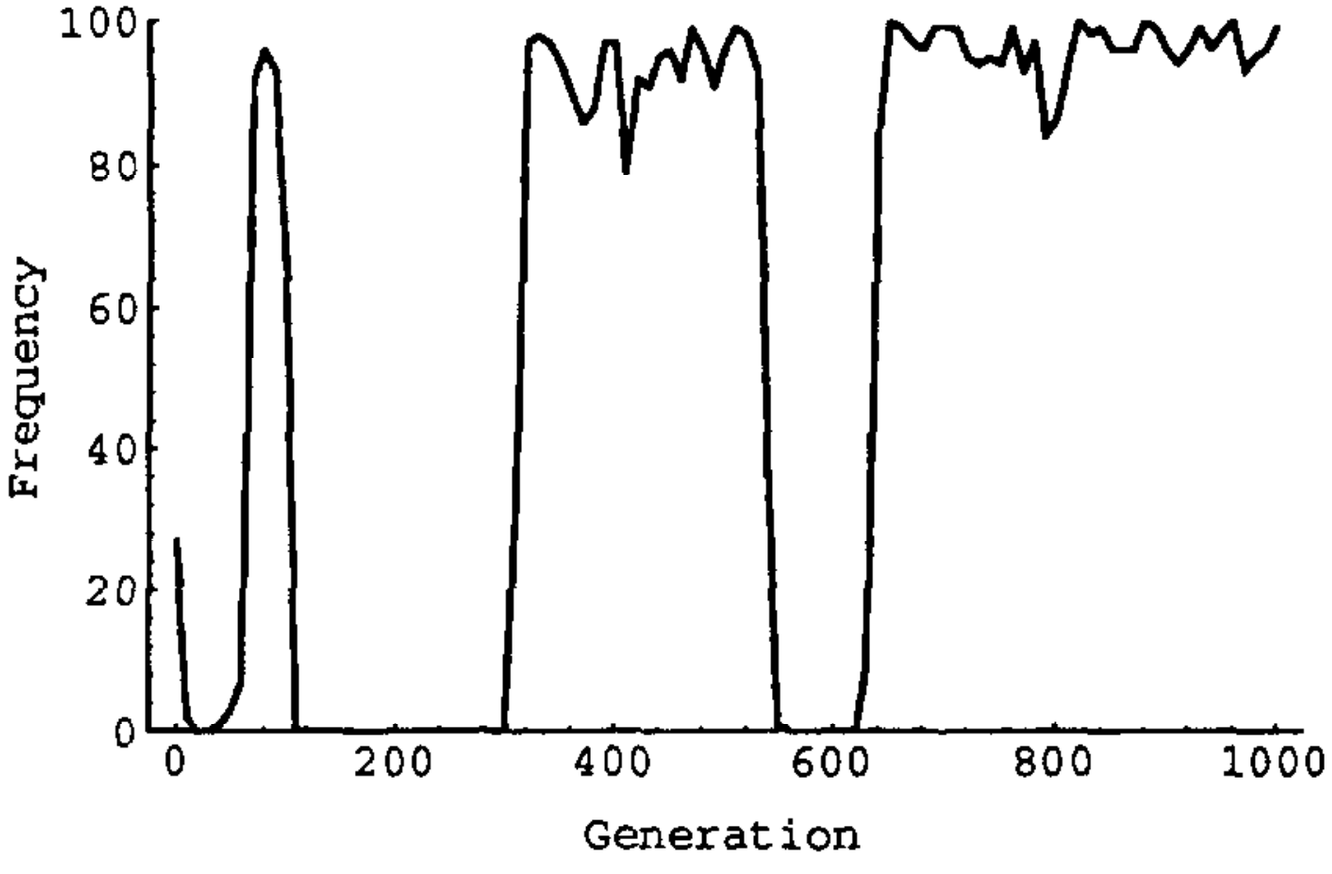
- Oliphant reruns the simulation with only receivers benefiting from successful signalling

- Population does not converge on optimal signalling
    - Reception behaviour looks optimal (but unstable)
    - Transmission behaviour wanders about at random
        - And these random fluctuations drive switches between reception systems



s1 ⟶ m1

s2 ⟶ m2



s1 ⟶ m2

s2 ⟶ m1

Oliphant's simulation 3

- Why do species behave **altruistically** to others when genes evolve selfishly?
- One answer: reciprocal altruism
- I'll scratch your back if you scratch mine
  I'll send optimally to you if you send optimally to me
- Oliphant uses agents with two signalling systems:
  - One to use if previous interaction with this partner was successful
  - One to use if previous interaction was unsuccesful
- Optimal signalling evolves (initially along with a deliberately unhelpful "punishment" system).

How to get communication, solution 2:

reciprocity

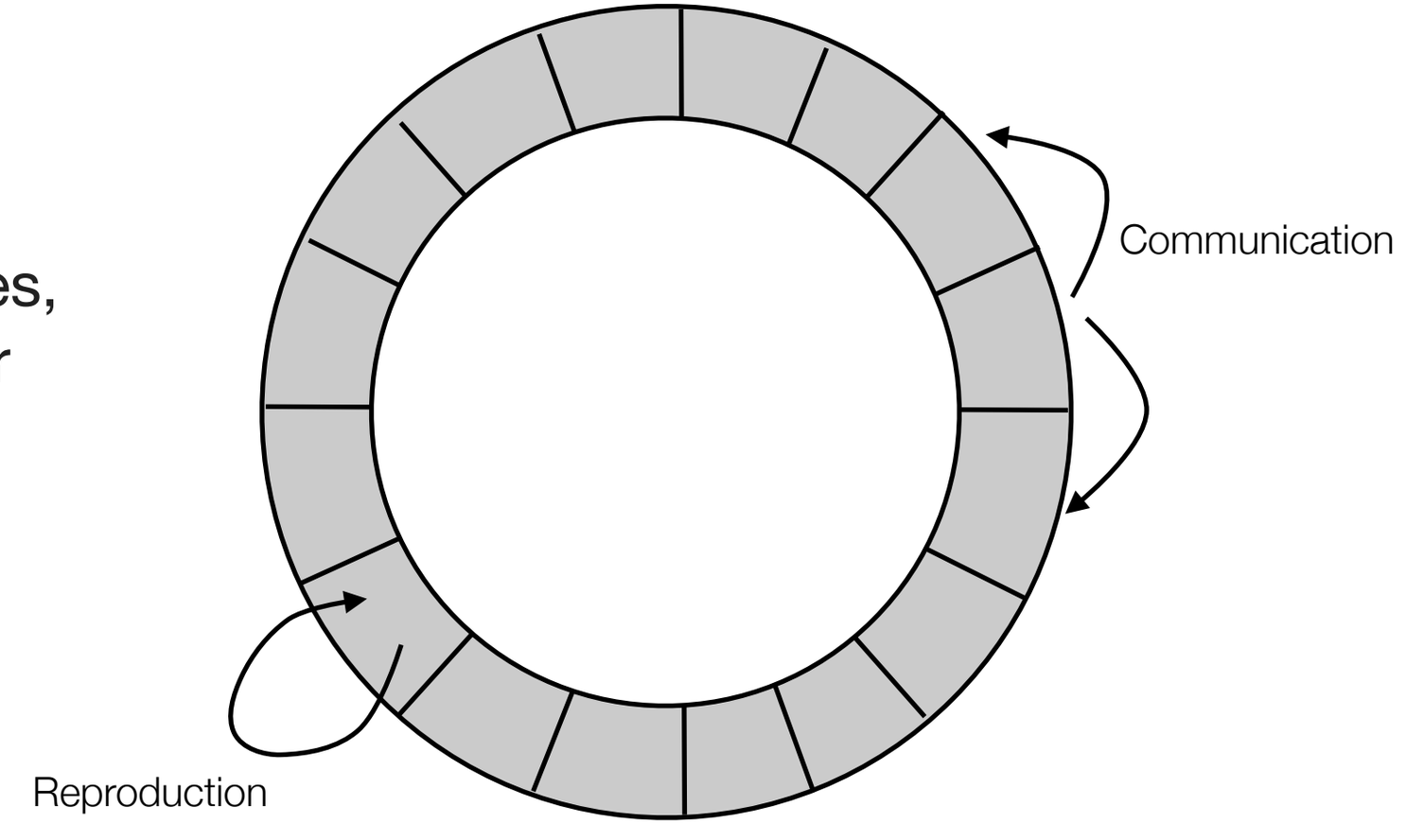
Oliphant's simulation 4

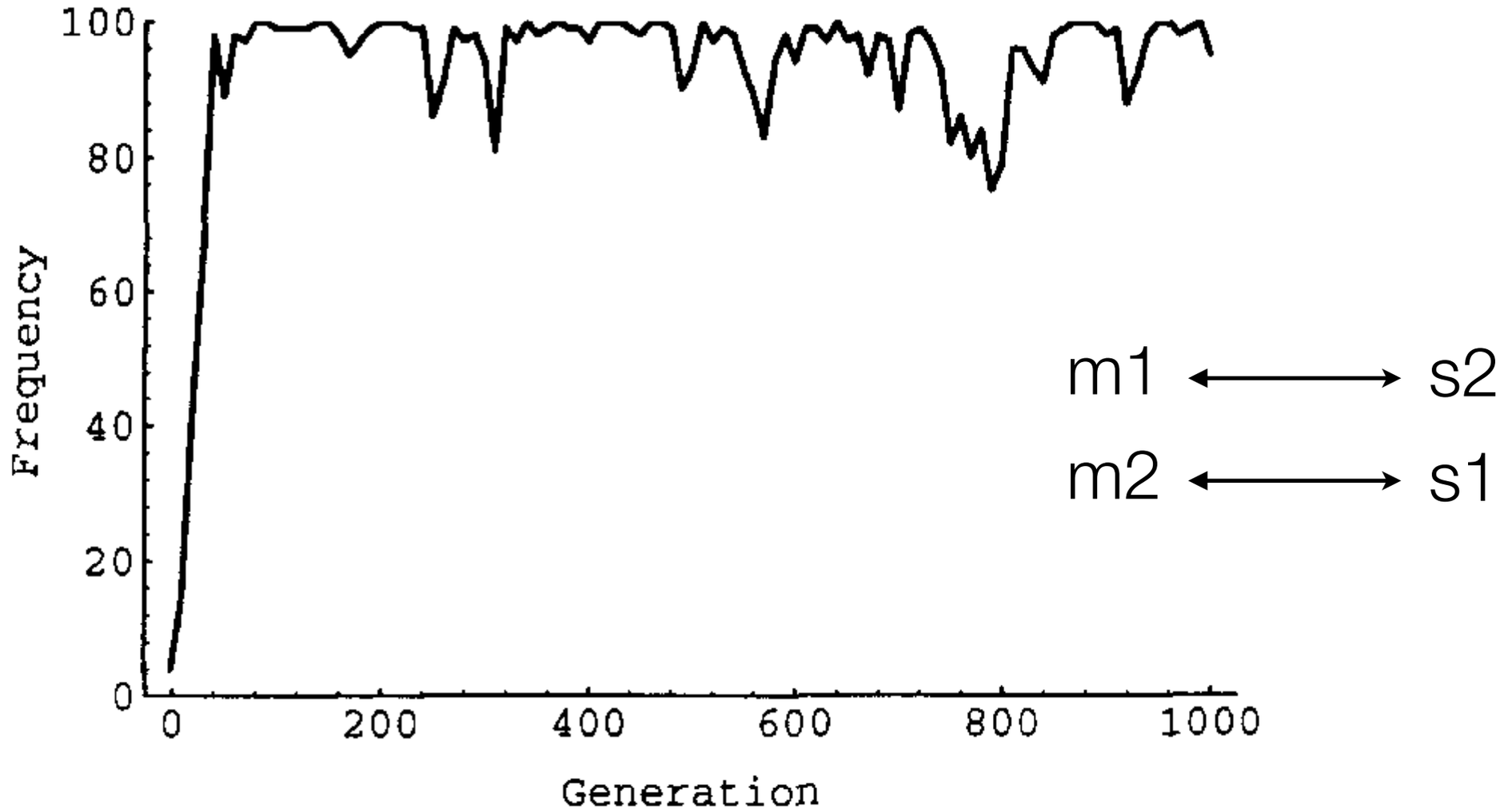
- Previous simulations have picked partners to communicate with at random.

- What if you talked more to people near you, and people near you were more likely to be related to you (e.g. have the same parent)?

- Organise agents in a ring:

- Optimal communication evolves, even without mutual benefit, or reciprocity!




100
80
60
40
20
0
Frequency
0
200
400
600
800
1000
Generation
m1 ⟷ s2
m2 ⟷ s1

Oliphant's simulation 4

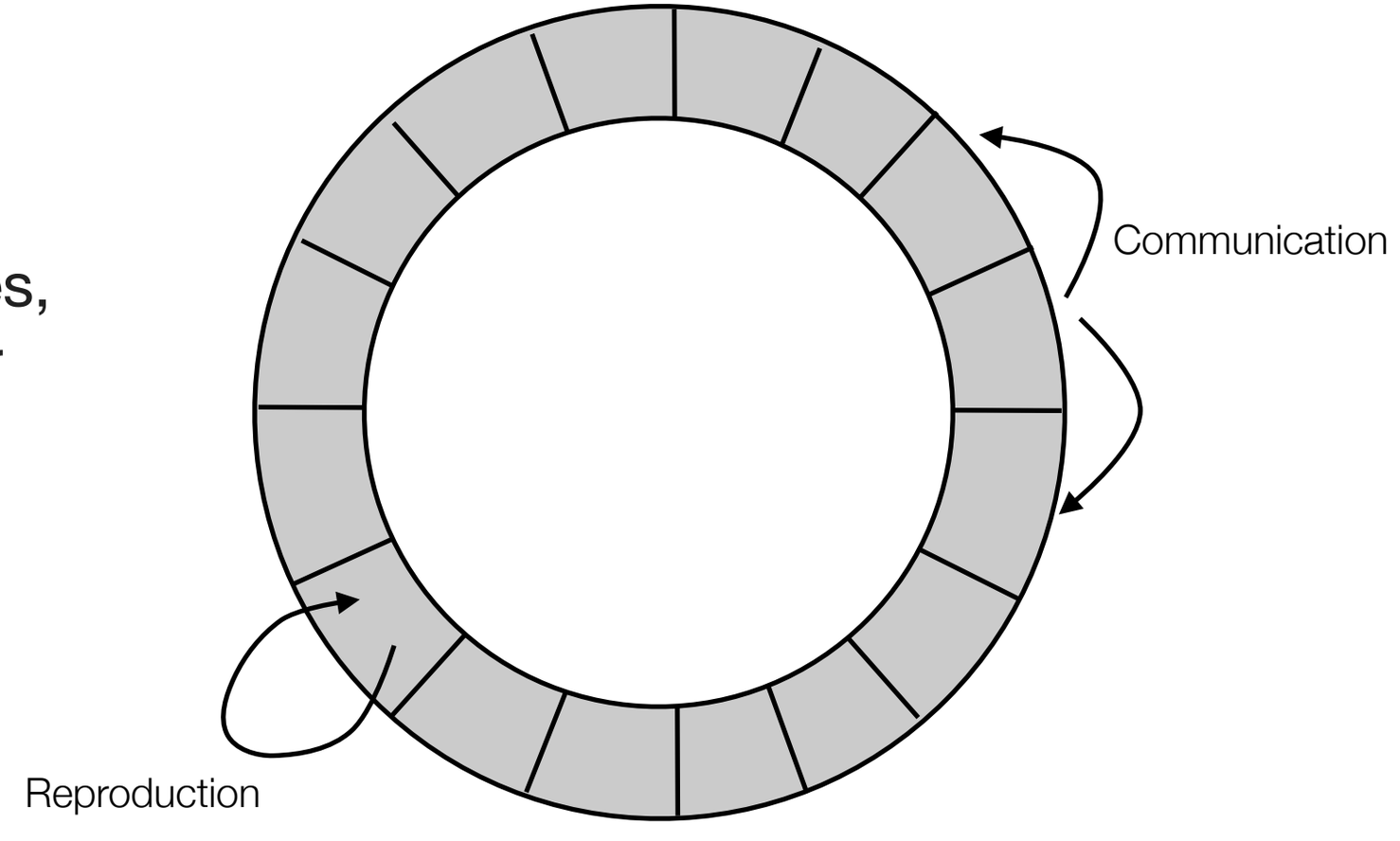
- Previous simulations have picked partners to communicate with at random.

- What if you talked more to people near you, and people near you were more likely to be related to you (e.g. have the same parent)?

- Organise agents in a ring:

- Optimal communication evolves, even without mutual benefit, or reciprocal altruism!

- Why?



How to get communication, solution 3:

spatial organisation

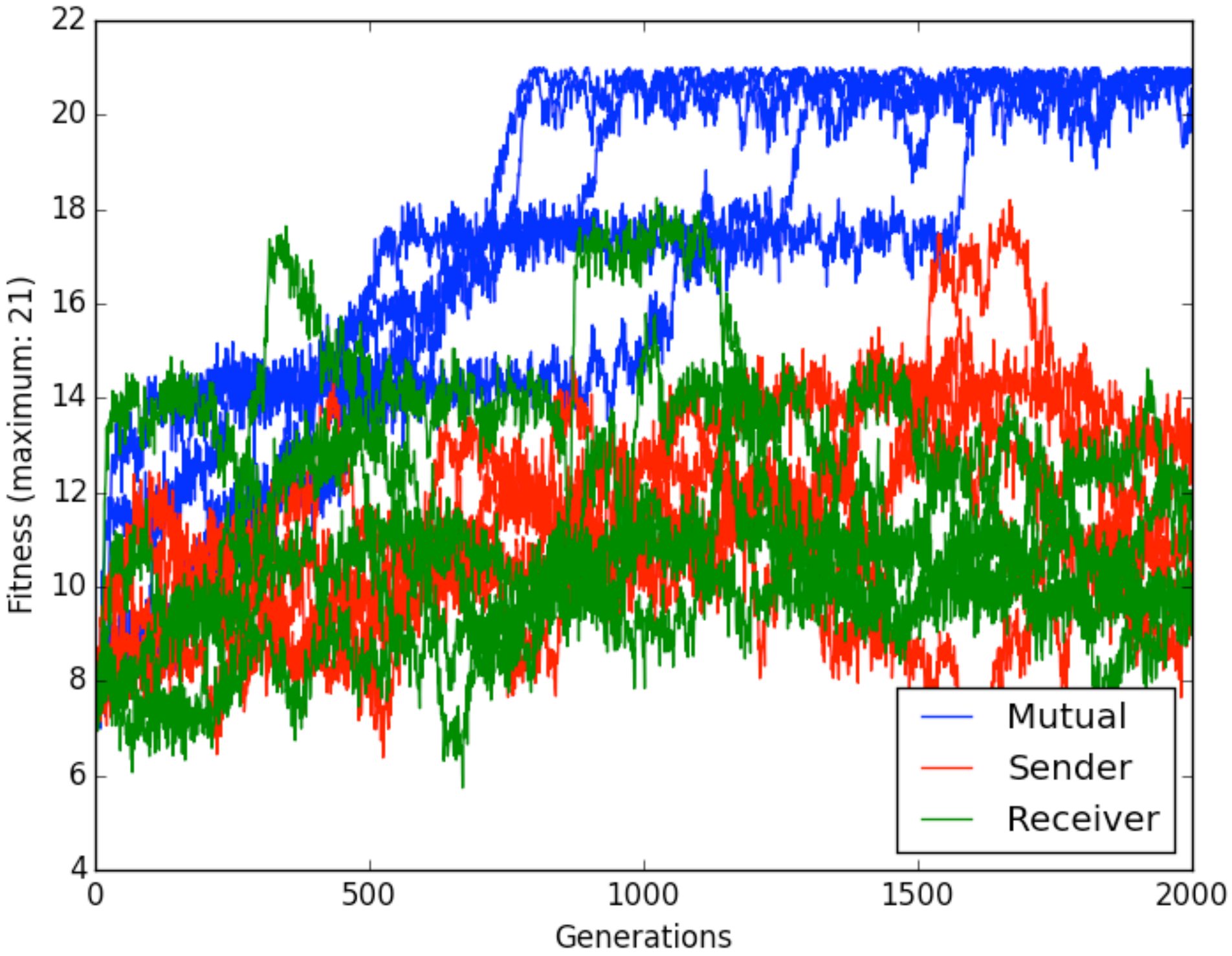
Summary

- Optimal “Saussurean” signalling does not automatically evolve by natural selection

- Needs either:
  - mutual benefit
  - reciprocity
  - spatial organisation

- Can we replicate the first of these results in our simulation model?

1.Under what conditions does stable, successful communication evolve? (Note that it is a very good idea to run the simulation a few times, and plot the results).



How to get that graph, method 1: typing at the prompt, do the following sequence of things:

```
%pylab
%run /Users/kennysmith/Dropbox/SimulatingLanguageCode/evolution1_KSedit.py (or whatever your file is called!)
n_runs_per_condition = 5
#now do the runs with mutual benefit
for r in range(n_runs_per_condition):
    this_pop,this_pop_fitness = simulation(1000)
    plot(this_pop_fitness,color='b',label="Mutual" if r == 0 else "")
```

THEN go into the script, chage send_weighting to 20 and receive_weighting to 0, then re-run the script

```
%run /Users/kennysmith/Dropbox/SimulatingLanguageCode/evolution1_KSedit.py (or whatever your file is called!)
for r in range(n_runs_per_condition):
    this_pop,this_pop_fitness = simulation(1000)
    plot(this_pop_fitness,color='r',label="Sender" if r == 0 else "")
```

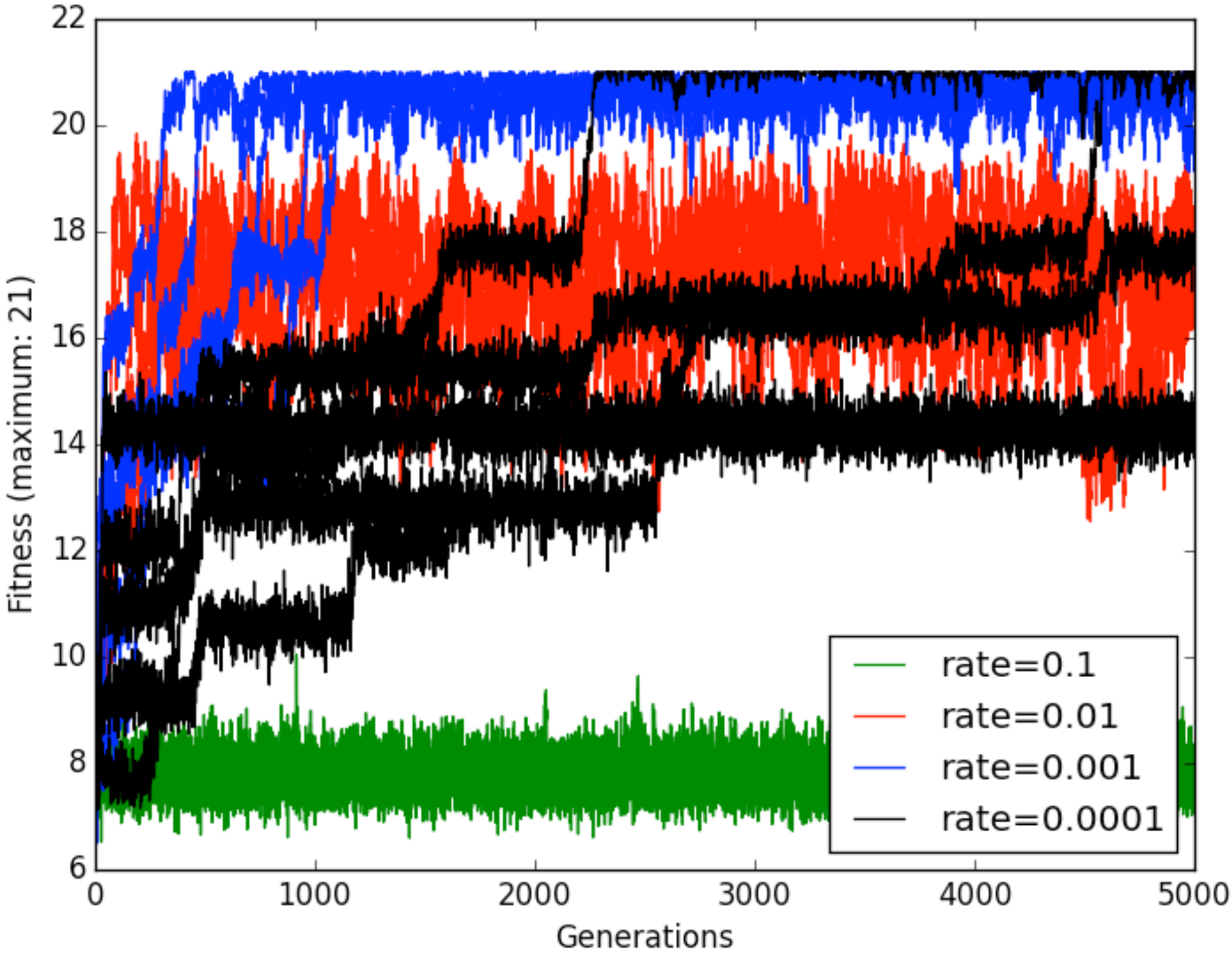
THEN go into the script, chage send_weighting to 0, receive_weighting to 20

```
%run /Users/kennysmith/Dropbox/SimulatingLanguageCode/evolution1_KSedit.py (or whatever your file is called!)
for r in range(n_runs_per_condition):
    this_pop,this_pop_fitness = simulation(1000)
    plot(this_pop_fitness,color='g',label="Receiver" if r == 0 else "")

xlabel("Generations")
ylabel("Fitness (maximum: 21)")
legend(loc=4)
```

How to get that graph, method 2: add the following to the end of evolution1.py

```
#because I am generating this figure from a script, rather than using the interactive
#prompt, plotting works a little differently:
#I import a module to do plotting, called matplotlib.pyplot
#then I create a new figure, using plt.figure()
#then at the end I add axis labels etc, and then crucially I have to tell it to show the figure

import matplotlib.pyplot as plt

send_weighting = 10    # weighting factor for send score
receive_weighting = 10 # weighting factor for receive score
n_runs_per_condition = 5

plt.figure()

#runs with mutual fitness
for r in range(n_runs_per_condition):
    this_pop,this_pop_fitness = simulation(2000)
    plt.plot(this_pop_fitness,color='b',label="Mutual" if r == 0 else "")

send_weighting = 20    # sender only benefit
receive_weighting = 0
for r in range(n_runs_per_condition):
    this_pop,this_pop_fitness = simulation(2000)
    plt.plot(this_pop_fitness,color='r',label="Sender" if r == 0 else "")

send_weighting = 0    # receiver only benefit
receive_weighting = 20
for r in range(n_runs_per_condition):
    this_pop,this_pop_fitness = simulation(2000)
    plt.plot(this_pop_fitness,color='g',label="Receiver" if r == 0 else "")

plt.xlabel("Generations")
plt.ylabel("Fitness (maximum: 21)")
plt.legend(loc=4)


plt.show()
```

2. Can you speed up evolution (or slow it down)? How? Is there a limit to how fast evolution can happen in the model?



Lab 4 worksheet

3. In earlier worksheets we gave you the option of modelling production and reception using a single matrix of weights, or of modelling populations in a more structured way (e.g. where each individual communicated with their neighbours). What difference do you think these factors will make to the evolution of communication? Make the necessary adjustments to the code and find out.

Lab 4 worksheet

4. In this model a parent's signalling system is transmitted directly to their offspring - this is our model of the genetic transmission of an innate signalling system. How else might a signalling system be transmitted from parent to offspring, and how might you model that process?

We'll explore this in the next **lecture**