

Scripting, take 2

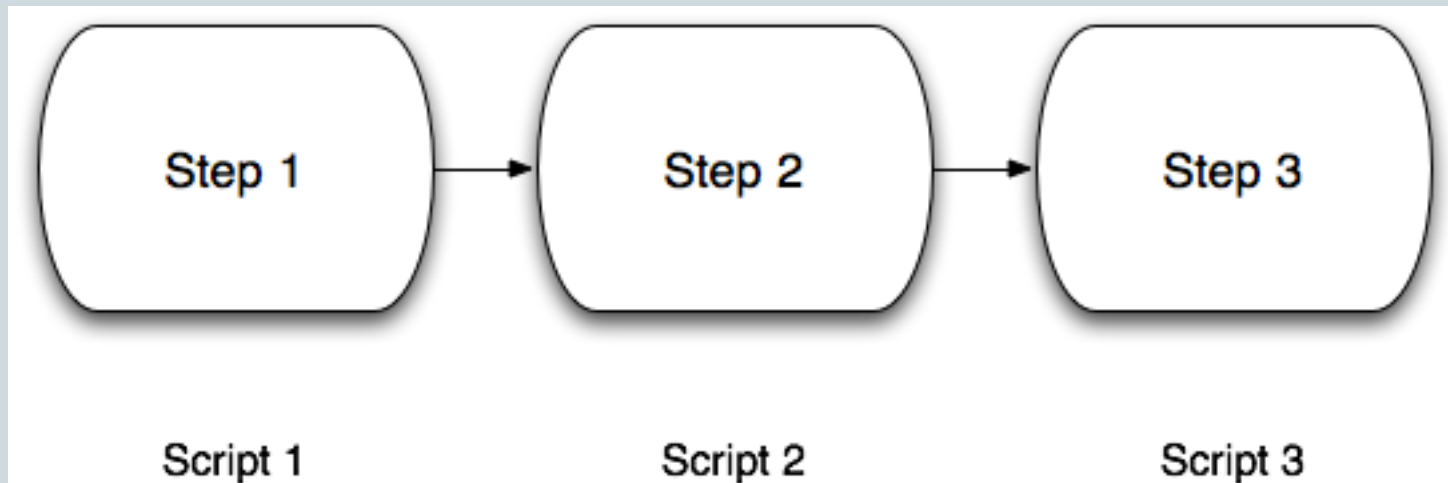


Outline



- More scripting
- Pipelining scripts
- Proving the scripts (and me) right (or wrong)

Pipelining scripts

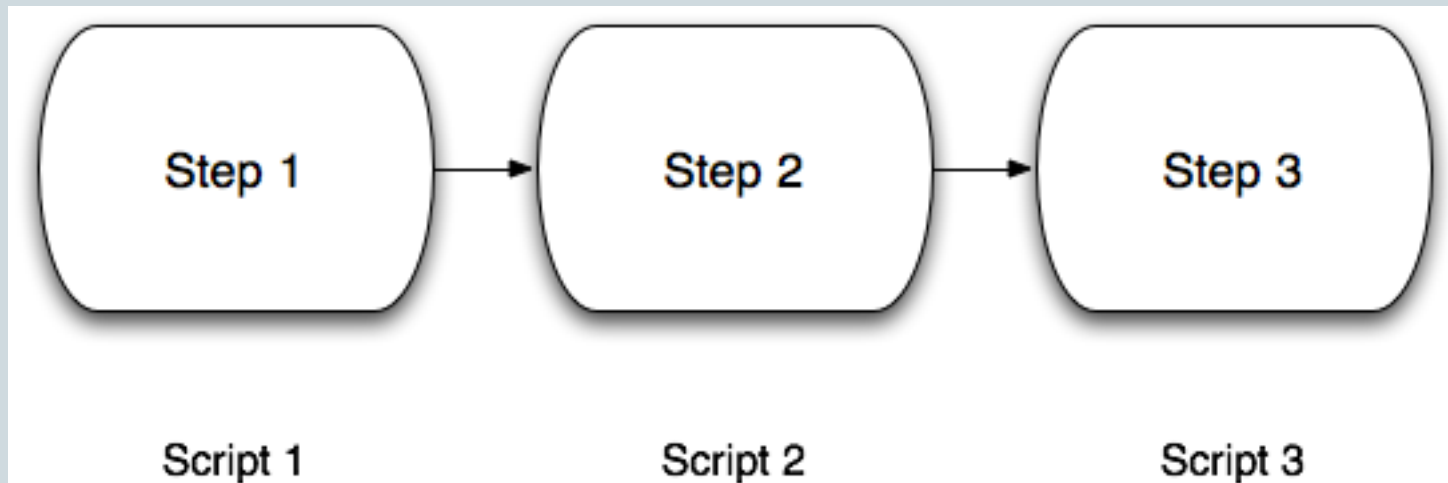


Why pipeline scripts?

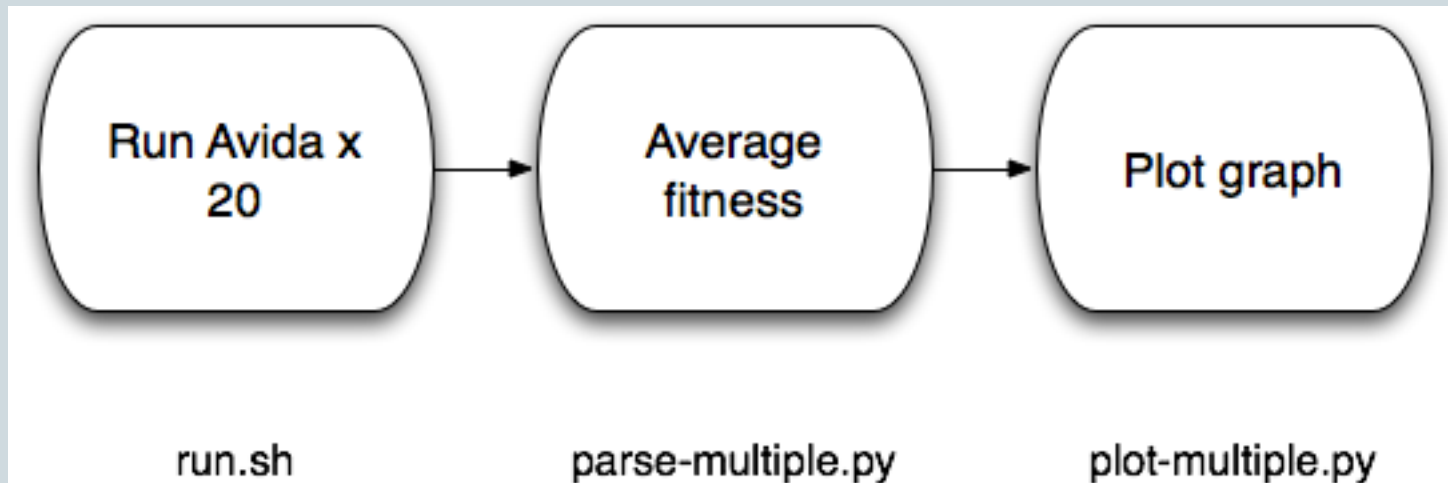


- Each script is easily reusable, by you and others;
- Each script can be developed in isolation;
- Each script can be tested in isolation;
- Each script is easier to understand;
- If you plug & play properly, you can use one script in multiple pipelines.

Pipelining scripts



Pipelining scripts



Analyzing multiple runs



- Say we want to average average fitness, per update, across all the runs.
- To do this, we have to have the following information:
 - For each run, for each update, the average fitness

Analyzing multiple runs



- Say we want to average average fitness, per update, across all the runs.
- To do this, we have to have the following information in one place:
 - For each run, for each update, the average fitness
- In what form is this information now?
 - Each update / average fitness
 - In separate files, one per run! (data/average.dat)

Current organization of data



run 1

u1: fitness
u2: fitness
u3: fitness
...

run 2

u1: fitness
u2: fitness
u3: fitness
...

run 3

u1: fitness
u2: fitness
u3: fitness
...

run 4

u1: fitness
u2: fitness
u3: fitness
...

Desired organization of data



update 1

run1/u1/fitness - run2/u1/fitness - run3/u1/fitness ...

update 2

run1/u2/fitness - run2/u2/fitness - run3/u2/fitness ...

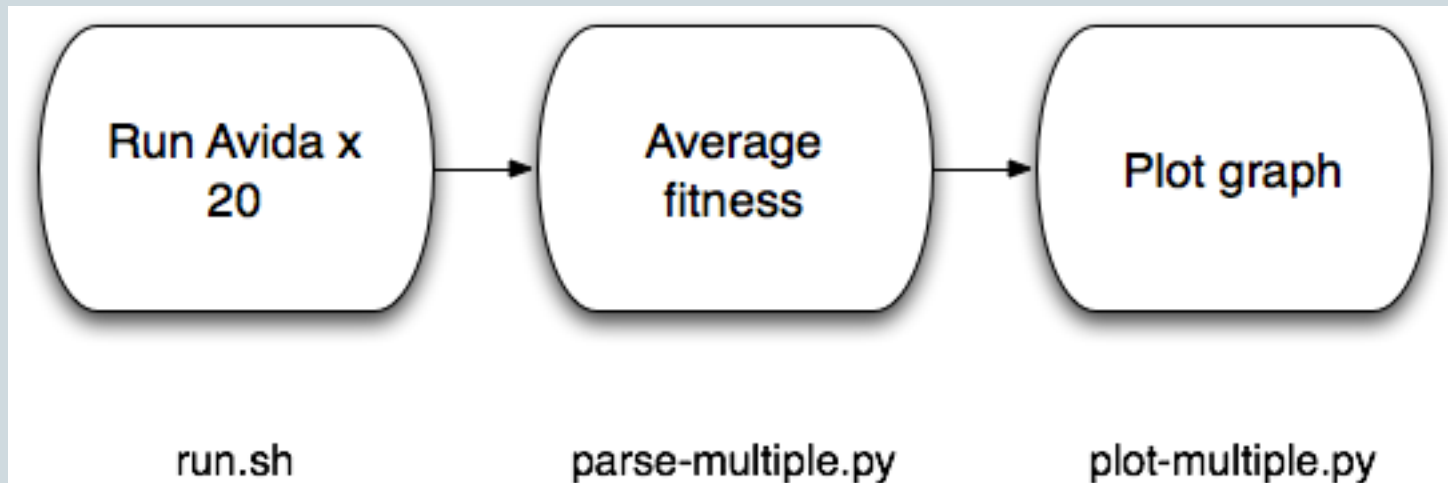
update 3

run1/u3/fitness - run2/u3/fitness - run3/u3/fitness ...

update 4

run1/u4/fitness - run2/u4/fitness - run3/u4/fitness ...

Pipelining scripts



Scripting!



How to test?



HW for next week:



- Read paper on Lenski LTEE
- Try running the scripts yourself!