

2# lab notes

First attempt at keeping some sort of a lab notebook, with an emphasis on experiments for my dissertation. Accordingly, let's start with a quick summary of what I said I'd do for my dissertation.

---

Sent to my committee on 19 April 2016:

## **changes to dissertation as originally proposed**

**overall:** Attempt to identify more refined proxies of female mating preferences. For example, in addition to recording association time, also track a female's orientation, overall activity levels, and possibly programmatically identify receptivity behaviors (e.g. glide responses).

**Chapter 1: transitivity and complexity of visual signal** *originally:* Test transitivity for female mating preferences for male body size only, then test for transitivity by varying both male size and courtship effort at the same time. *revised:* Test for transitivity for male body size only; these males will exhibit an intermediate amount of courtship. Next test transitivity for courtship effort using an intermediate-sized male. Finally, vary both male size and courtship effort at the same time as originally proposed.

**Chapter 2: decoy effects in the context of alternative mating behaviors** *originally:* Test decoy effect by varying two ways in which males differ—male courtship effort and body size—using two types of decoys, one meant to increase preference for the large male, a second to increase preference for the small male. *revised:* First, test for the decoy effect by only varying male courtship effort and keeping male body size constant. Second, test for the decoy effect by only varying male body size and keeping courtship effort constant. Third, test for decoy effect by varying both size and courtship effort as originally proposed.

**Chapter 3: decoy effects in the context of shoaling behavior** *originally:* Test for decoy effect manipulating the size of the shoal and the size of the individual schoolmates comprising each shoal. *revised:* As in Chapter 2, first test for the decoy effect only by varying the size of the shoal, keeping the size of the individuals that comprise each shoal the same. Next test for the decoy effect by varying only the size of the individuals comprising each shoal, keeping the number of individuals in each shoal constant. Last, test for the decoy effect in by varying both size of shoal and size of shoalmates simultaneously as originally proposed.

---

## 20 April 2016

- Made a schedule for chapter 1 experiments. Will start on 25 April, end on 8 June. Fish will have at least five days in between tests. n = 21 for each of the univariate transitivity tests; n = 43 for transitivity varying both male size and courtship effort (11 females that were tested for body size transitivity, 11 females that were tested for courtship transitivity, and 21 naive females that were never tested before).
- Made plans to build a second tank identical to the I have now. Cost breakdown: *2 s: \$1000* webcam: \$100 *cables and adapters: \$100* Plexiglas: \$200 *screens: \$220* **total: ~\$1600**

That should be pretty do-able given that I got the EEB DDIG for ~\$8k.

- Fish for these experiments will be housed in *pairs*. This cuts down on the number of tanks needed and ensures the females are still interacting socially with other fish. I set up seven new ~2.5 gallon tanks. In addition to the 14 individual tanks I already have, this is enough room for  $7 + 14 = 21*2 = 42$  fish, which should be what I need.
- n = 11 new females were isolated from males. To be used to the experiment starting 2 May.

## 21 April 2016

On the to-do list today:

- checking the animation builds. See `~/Desktop/chapter_1/vary_size_only/building_videos/`.
- testing `run_transitivity.sh` for bugs and errors.
- figuring out how to take videos with **new logitech webcam** and have it work with opencv immediately.

Using the lifecam webcams, I can do `ffmpeg -f avfoundation -video_size 1280x720 -framerate "$FPS" -i "Micro:none" -crf 28 -vcodec libx264 -y -t "$LENGTH" "$NAME"".avi"`

and opencv can read the resulting video fine. Trying:

```
ffmpeg -f avfoundation -video_size 1920x1080 -framerate 10 -i "HD:none" -crf 28 -vcodec libx264 -y -q:v 1 -t 60 recording3.avi
```

So opencv doesn't like the **avi** file this results in. If you convert it to a mp4 container using `ffmpeg`, it reads it in, but the quality is terrible.

What about if you record it as a mp4?

Problem loading the video file.

Another weird thing is that when I save it as a avi file, it says the resulting video is 600 fps (via ffmpeg or `file recording3.avi`), when though I'm specifying the frame rate when I take the video.

What if I use a different codec?

```
ffmpeg -f avfoundation -video_size 1920x1080 -framerate 10 -i "HD:none" -crf 28 -vcodec mpeg4 -y -q:v 1 -t 60 recording5.avi # same error with opencv, 600 fps
```

```
ffmpeg -f avfoundation -video_size 1920x1080 -framerate 10 -i "HD:none" -crf 28 -vcodec mpeg4 -y -q:v 1 -t 60 recording6.mp4 # doesn't record
```

```
ffmpeg -f avfoundation -video_size 1920x1080 -r 10 -i "HD:none" -crf 28 -vcodec libx264 -y -q:v 1 -t 60 recording7.mp4 # same error with opencv
```

```
ffmpeg -f avfoundation -video_size 1920x1080 -framerate 10 -i "HD:none" -crf 28 -vcodec mpeg4 -y -q:v 1 -t 60 recording6.mp4
```

For all these videos, note that I passed ffmpeg a framerate of 10. All are 1920x1080:

| container | codec | fps             | result with OpenCV              |
|-----------|-------|-----------------|---------------------------------|
| H.264     | 10    | fails           | mp4                             |
| H.264     | >30   | fails           | mpeg                            |
| MPEG4     | ?     | fails recording | mpg                             |
| MPEG4     | ?     | fails recording | avi                             |
| MPEG4     | >30   | fails           | mp4 (recorded initially as avi) |
| MPEG4     | 28.50 | avi             | works                           |

But this last video (1) is poor quality and (2) requires as additional step. Also, ffmpeg refuses to record a video with a mp4 container and mpeg codec.

Because this is taking too long, I'm going to switch back to using the MS lifecam video. As an additional plus, this camera has better white balancing.

The same code I used to use with the LifeCam isn't working for unknown reasons. Further testing revealed that running a line like this one

```
ffmpeg -f avfoundation -video_size 1280x720 -framerate 10 -i "Micro:none" -crf 28 -q:v 1 -vcodec mpeg4 -y -t 300 recording17.mp4
```

is exactly what I want: records a high quality video that OpenCV can use immediately. For whatever reason (I probably discovered this before but have

since forgotten it, hence this lab notebook), it seems like using a video size of 1920x1080 (full HD) may be causing some of the problems.

- I also worked a bit on debugging the shell script that will run the transitivity trials.

**for tomorrow:**

- calculate angle of fish in script
- calculate distance fish moved in script (as running average, or what? probably do this in R)
- add argument for whether you're running the script on a laptop, iMac, or don't want to see the script running
- change association zones and print them to the background image for visual inspection

**for the future:**

- make animations where courtship effort differs
  - create `run_transitivity_courtship.sh`
- 

## 22 April 2016

- make prelim questions for Zach for our meeting @ 4
- set up to record several test videos to test out tracking program on three-screen square tank.
- read lab meeting paper on autism in zebrafish

*note:* I talked to Ian and tried to record from the lifecam webcam at a video size of 1920x1080 (full HD); this does not work. FFMPEG doesn't throw an error but simply won't start taking the video.

- changes to `tracker.py`

Got rid of global variables for the tank bounds.

Added an argument for the number of frames to use when determining the background.

I removed the name argument because the necessary information is already in the path to the video (the name is always just whatever the video is called).

Changed upper variable to 255,255,10 to make it a little more lenient.

*major problems:* I'm trying to get rid of reflection tracking by selecting the closest contour to the center of the tank when there are 2+ contours that meet my criteria. As written, this is not working.

I also neglected to test the `run_transitivity.sh` script; perhaps I should do this tomorrow.

**for future luke:**

- Think about the graphs you want to output from each trials. E.g. tank with tracks overlayed + tank with graph of colored density overlayed + angles in each association zone.
- Also measure latency to enter each association zone.

## **weekend of 23 April 2015**

Worked on incorporating angle measurements of the fish into the tracking. I'm using a example idealized fish-from-above with known angles to test my method. It's working pretty well but I'm facing some issues given that things are a little difficult to interpret in a photo where numpy opencv represents the upper-left-hand corner of the photo as the origin, i.e., the entire photo is in the 4th quadrant of a Cartesian graph.

I also faced some issues with the transitivity shell script. Two main issues:

- creating the `.asc` files that record what types of trials a female hasn't had yet. You can't use `touch` to create a file in a non-existent directory; creating the directory fixed the problem.
- recording the videos. I'm now convinced that I'm specifying the path to the video file incorrectly; I'm going to try saving the trial videos to the `displaying_videos` sub-directory and see if that works. Otherwise I'll have to revert to using a `.avi` container with a `libx264` codec as normal (but recall that opencv can't read this type of video).

## **25 April 2015**

- started acclimation trials @ 2 hours each. I'm going to do three acclimation trials, each with 7 fish, for a total of 21 fish over 6 hours.

- I also verified and incorporated code for measuring the angle a fish is orientated. See the code and examples in the `testing_angle_detection` folder. I show that when used what a fish ideally looks like from above, this method does an awesome job of realizing the actual angle. The fish in the videos, however, don't appear this ideally, and frequently the angle measurements are clearly wrong. Part of the issue is that the number of pixels the fish occupies is really small. The issue may be the poor resolution of the videos.

After talking with Ian, we found that running

```
ffmpeg -f avfoundation -video_size 1920x1080 -r 20 -i "Micro:none"
-q 0 -t 300 -y out.mp4
```

recorded a video in full HD. However, you do have to run something like

```
ffmpeg -i out_again.mp4 -vcodec mpeg4 -q 0 out_again.mp4
```

to get it to work with OpenCV.

---

## 26 April 2016

- made cage cards for tanks holding the pairs of fish.
  - completed the second round of acclimation trials. Some of the fish seemed a bit more skittish today...the bubbler was left out of the water overnight, maybe the tank had low oxygen?
  - met with Michael Domjan and had him sign off on my proposal.
  - restricted angle tracking only when the difference between the center of mass and the center of the bounding rectangle are greater than 3 pixels apart.
  - read various papers.
  - Thinking about journals to submit slime mold papers to (experimental economics, journal of comparative psychology)
  - Created my own color palette that is perceptually uniform. It is similar to viridis but less bright and with less yellow at the high end. Find it in `plotting_function.R` where you can run `blues(5)` to return 5 colors along this spectrum.
  - Started more slime molds cultures. Trying culturing on filter paper.
-

## 27 April 2016

- Came up with names that start with 's' (for varying *size* only) to use for these females in R: `require(redingPlot); require(babynames)`  
`names <- babynames %>% filter(sex == "F") %>% select(name,n)`  
`%>% group_by(name) %>% tally %>% arrange(desc(n))` Now only take the 70 most popular names starting with S: `names[grep("^S", as.character(names$name)),] %>% .$name %>% .[1:42]`
  - changed the `run_transitivity.sh` script so that it records an HD video.
  - Culturing the slime mold on filter paper results in drying out. I'll have to try having the filter paper connected to some water source (?)
  - I tested the transitivity shell script; it appears to be working as planned. The `.asc` files are being written and lines deleted from them properly.
  - I created a log sheet where the person running the trials can copy down some of the information that the computer records anyway; it'll just be nice to have some sort of a backup.
  - I created a protocol and running the transitivity trials.
  - named eight females and classified found some characteristic to differentiate each female from their tank-mate
  - created new twilio account for texting when the end of a trial is.
  - ran all the day's trials.
  - made an automator script that saves my cv as a pdf and pushes it to my github website every weekend
  - made a cron script that saves this document as a pdf and posts it to my github every day at 5
  - set up two more slime mold plates; Joseph actually set these up, I just put the slime molds in them and put them under the camera.
- 

## 28 April 2016

- labeled eight more tanks with the females of females; came up with identifying characteristics of each
- reset my github RSA key so that it's working now
- fixed issue with the shell program texting me when a trial is over
- ran today's trials (n=7)

## slime molds and prospect theory

I did some further thinking and reading today about prospect theory and how it could be tested in non-humans systems. Part of what I wasn't grasping before is that prospect theory combines *loss aversion* with *risk aversion* to recapitulate the four-fold pattern of risk attitudes:

**TABLE A.2A** The Fourfold Pattern of Risk Attitudes

|                  | Gains                  | Losses                   |
|------------------|------------------------|--------------------------|
| Low probability  | $c(\$100, .05) = \$14$ | $c(-\$100, .05) = -\$8$  |
|                  | <i>Risk seeking</i>    | <i>Risk aversion</i>     |
| High probability | $c(\$100, .95) = \$78$ | $c(-\$100, .95) = -\$84$ |
|                  | <i>Risk aversion</i>   | <i>Risk seeking</i>      |

$c(x, p)$  is the median certainty equivalent of the prospect that pays \$x with probability  $p$ .

Table A.2A adapted from Tversky and Kahneman (1992).

Figure 1: Four-fold pattern of risk attitudes.

These two ideas are encompassed in both the value function (Fig. 1 of Tversky and Kahneman's classic science paper) **and** a probability weighing function (Fig. 2 of the same). While I think it's possible to test whether something like a slime mold is loss aversive, testing whether they are also *risk* aversive is more difficult because it involves representing options in a probabilistic way. This is why virtually all studies of prospect theory use *gambles* because gambles are inherently probabilistic. I think presenting options to slime molds in a probabilistic way is perhaps not realistic or even possible. I think that representing something as a probability requires learning. For example, you might train a bird to associate a 50% chance of a good food reward with one color button and a 70% chance of a poor quality food with a different color button. In this case, I think you might be able to test for evidence of prospect theory, but I think it would still be difficult; many studies that seek to estimate key parameters of prospect theory require questions like, 'How much money would have to be on the line for you to accept [some gamble]...' that would be extremely tedious to determine in an animal lacking language.

I think the next thing to tackle is how people have tested for loss aversion in humans.

## 29 April 2016

- ran transitivity trials (n=7)
- worked on sending text messages through python for the end of each trial using plivo
- labeled the rest of the fish; came up with characteristics that can be used to distinguish between two fish in a pairs
- wrote thank-you letter for EEB DDIG - like grant. The \$8k will be deposited into my bank account, not in an account in IB unfortunately.

### slime molds

Recall that Joseph and I recently decided that we need to be more precise about how we're setting up each the choice experiments for the slime mold.

Things we've settled on:

- oat disks should face up (i.e. the side of the disk containing the majority of the oats should face towards the camera)
- the slime molds should also face up so that we can photograph them better
- we should not use particularly old or young slime molds; preferably use slime molds that are actively foraging
- the piece of agar with the slime mold and the oat disks should be in contact from the beginning of the trial.

All the results / information / protocols etc. w/r/t the slime molds are in `~/Desktop/slime_mold/`. The experiments are in the `experiments` subfolder. The following experiments were all done according to the principles above:

- 22 April (1 2-way, 1 3-way)
- 25 April (2 2-way, 2 3-way)
- 27 April (1 2-way, 1 3-way)
- I started these on a Wednesday from plates that were already set up in the fridge. We aren't sure who set these up. These must be for the same experiment, but they were cold when the experiment started, plus the results are a bit weird, so I think they should be excluded.

Joseph set up 2 4-way plates today.

So by Monday, we should have:

| 2-way replicates | 3-way replicates | 4-way replicates |
|------------------|------------------|------------------|
| 33               | 30               | 16               |

On Monday, we'll set up two more 4-way plates to up the sample size to 32.

### To do:

- autoclave slime molds
- determine what to measure / how to measure it for the slime molds
- modify animations for courtship experiments next weekend

### **weekend of 30 April**

- autoclaved slime molds. Note that you should record the amount (roughly) of autoclaved material in the lab safety binder, put it in a biohazard bag, and throw it in the dumpster, noting with a sticker that the bag has been autoclaved.

### **2 May 2016**

Wasn't feeling too well today so didn't get a whole lot done.

- worked on animations for this week's trials. Having trouble deciding how often the non-courtship male should turn.
- Joseph set up two more 4-way choice tests.
- read some papers on models in decision-making
- worked on Python script that allows the user to crop a video. Stored in `~/Documents/random_scripts/crop_video.py`.
- ran acclimation trials.

## 3 May 2016

- ran acclimation trials
- fixed `~/Documents/random_scripts/crop_video.py`. Note the problem was a disconnect between the dimensions of the video as read by opencv and by ffmpeg (ffmpeg was correct). I fixed the problem by scaling the video to a common size (1920x1080) and then everything worked perfectly.
- worked on animating.
- created a prototype python program for mary. Shows the first frame a video and prompts the user to click where there are males, females, ... It's called `locate_fish.py`. Need to add option for the user to see the video looped at any time.
- wrote a python program called `increase_blue.py` to try and tweak some of the slime mold videos to pull out the yellow relative to other colors. So far unsuccessful.
- put females in pairs, labeled tanks

## 4 May 2016

- tested out transitivity script for transitivity trials varying courtship effort only.
- for the animations for these trials, I decided to make the difference between the high- and low-courtship males as dramatic as possible. The high courtship male courts aggressively and makes six turns before swimming offscreen; the low courtship male swims on-screen, sits there, then swims off screen. The intermediate courtship male courts a little, but the turns are slower and less aggressive.
- I checked and fixed errors with the video files used in the trials today. In some parts of some of the videos, the background was flipped.

As a reminder, for this week:

| monday                 | tuesday                | wednesday        | thursday         | friday                  |
|------------------------|------------------------|------------------|------------------|-------------------------|
| acclimate group 2 fish | acclimate group 2 fish | test $n=11$ fish | test $n=10$ fish | leave town for Richmond |

- Did water changes on tanks housing fish for this experiment.
- I'm also in the process of moving all my lab back up'ed data to a new 4 TB

hard drive from a smaller 2 TB hard drive, which I will wipe afterwards.

- If the data from today are shit, it might have something to do with the constant banging from the renovation at the end of the hallway.
- From looking at the slime mold videos from Monday, it's clear that there wasn't enough slime mold for each experiment (the mass of the slime mold was too low), which is on me for not making more plates.
- Weighed and measured SL of 14 fish I tested last week. I recorded the results in `weights_and_lengths.csv` in `/Users/lukereding/Desktop/chapter1/vary_size_only`.
- Throughout the day, I tested fish (total n = 12), I only have 9 trials to do tomorrow.

## 5 May 2016

- ran the rest of the trials for the week (n = 9)
- weighed and took photos of more females for SL measurement
- make some keynote presentations with cool slime mold / swordtail videos and photos
- attempted for a while to get some slow motion videos of swordtails courting
- set up slime mold plates
- set up time lapse for slime molds

## 11 May 2016

- registered for Evolution 2016 meeting
- ran 14 trials. Group 1A and 1B. Note that there was considerable noise from ~1:30 - 3.
- set up two four-way slime mold plates. Make two new 5% plates. Make new protocol sheet for the slime molds to hang in the slime molds part of the lab.
- Improved the style of my CV
- Started a markdown document that will provide a cheat sheet for the linux command line book I'm reading.
- Now that I'm back I need to think about creating a second mate choice tank. Using raspberry pi's would decrease my cost by a lot but possibly make things harder.

## 12 May 2016

- ran 14 more trials. Groups 1C and 2B.
- updated the website: [lukereding.github.io](http://lukereding.github.io)

## 13 May 2016

- ran 14 more trials.
- wrote a couple of python scripts to do computer vision stuff. Both are in `/Documents/random_scripts/`. `k_means_color.py` takes a photo as input and clusters similar colors into groups. The number of groups is determined by the `-n` parameter. It then finds the average color of each group and outputs a new photo with these new colors. `k_means_video.py` does the same thing, but accepts and outputs a video file instead of a single image.



Example here:

> these might come in handy for tracking

- slime molds. I'm constantly dissatisfied with how variable the slime molds from my 'culture' plates are when I go to set up an experiment. I think it's important to take slime molds from the same stage of growth (and also the same amount of slime mold each time); therefore I need to determine the amount of oats I should plate and how long I should wait before the

slime molds achieve the desired density. So I set up three plates today. One contained 0.25g oats, the second 0.5g, the third 0.75g oats. I made sure the oats were randomly distributed across the agar by shaking the agar plates after putting the oats on. I then added two slime-covered oats to each of these plates on opposite sides of the plate. I put these slime-covered oats near fresh oats to ensure colonization of the fresh oats. I am taking photos every 10 minutes and on Monday I'll be able to assess (1) the amount of oats I should be putting on my 'culture' plates and (2) how long I need to wait after an initial seeding before the slime molds are ready to be used in an experiment.

### Things to think about:

- Number of agar plates to make.
- You need to make a lot of oat plates. Grind a lot of oats and think about the number of each type of plate you need.

## 16 May 2016

- ran 7 trials. It was pretty noisy due to the construction today.
- made a cron job to ensure my slime mold data is backed up to my lab bkp drive daily.
- Use this line to back up the trial videos from mini1:

```
rsync -avt --delete --rsh=ssh $mini1:/Users/lukereding/Documents/chapter1  
/Volumes/lpr_lab_bkp/
```

- Made new templates for the slime mold. I can now fit 13 binary choices on one plate and 8 3-way choices.
- Talked with Joseph about slime molds for a while. For our project dealing with the number of choice options:
  - record split decisions. Sometimes the slime mold doesn't make a clear choice, but grows to more than one oat disk. These should be recorded by excluded (following Latty and Beekamn 2011).
  - consider a choice made when the slime mold covers the entire top surface of the oat disk.
  - record the number of options explored. Sometimes the slime mold goes to one of the options and sticks with it; something it explores all the options then clearly chooses one of them. We should record the number of different options the slime originally explored.

- record accuracy: did the slime mold go the the best option?
- time to decision. Note that there are probably differences in terms of the amount of slime mold on the agar on different days. To accord for this, and to make the *time to decision* meaningful, when analyzing the data run a model like this: `time to decision ~ # choices + day` to account for any day effect.
- In the future, to make the amount of slime consistent between all trials, I'm going to try and see what happens if I seed a 2% oat plate with slime mold and let grow 2-3 days.

## 18 May 2016

- ran trials
- added change to `run_transitivity.sh` because I was noticing that the trials were taking slightly longer than expected. It turns out they were taking longer than expected. The problem is that for some reason the computer has gotten slower at encoding the video while taking the video. Proposed solutions to the problem can be found [here](#) and [here](#). I changed the call to ffmpeg so that the mini is no longer encoding the video while it takes the video. Note that the videos are now enormous (16GB) and will need to be encoded to a lower size like this: `ffmpeg -i temp.mp4 -vcodec libx264 -crf 23 -preset medium -pix_fmt yuv420p output.mp4` (this works, but verify this works with opencyc).

This is a huge bummer and very problematic. All hope may not be lost though. Let's assume that the video took at a constant, slower rate. Then:

| display video time (actual time elapsed) | ffmpeg video time |
|--|-------------------|
| 0:00                                     | 0:00              |
| 5:00                                     | 4:12              |
| 10:00                                    | 8:24              |
| 11:00                                    | 9:14              |
| 16:00                                    | 13:27             |

This is approximately right, ideally I want to determine the time between the when the file was created and when it was last modified. I measured the actual time and 'ffmpeg' time for one trial and found that the ratio of ffmpeg to actual time didn't change much (0.86, 0.856, 0.887, 0.885).

- attempted to help Kelly with her tracking.

- reinstalled opencv. This took a long time and a lot of troubleshooting but I'm not sure what I did to make it work. In the future always install --HEAD verison.
- made several slime molds 2% plates for culturing slime molds. Revised my slime mold protocols to reflect a better way of getting repeatable plates.

Tomorrow: - make more slime mold plates - figure out how to get the difference between the data created and the date modified on a mac. This will tell us, for each video, the ratio between the actual time and the 'ffmpeg' time and so will allow us to determine how we need to analyze the videos. - run trials!

## 19 May 2016

- made more 2%, 9%, and 10% plates for the slime molds.
- It looks like using a 2% plate to grow the slime mold for two days prior to coring and using in an experiment is no good—not nearly robust enough growth. Try maybe a 3% or 4% plate. The 3% plate I used earlier this week seemed to work really well.
- Helped Kelly with tracking her fish. Scrapped HSV tracking for good-ol' RGB tracking based on a difference image and it works great for her. For some reason it's not working so well for me in my videos.
- ran 13 trials.
- built the second tank. I had to use some stuff typically used to bind things in aquariums to keep in a couple of cracks. I will test and make sure it's a watertight solution. If not, I'll have to put plexiglass shavings into the cracks and cement with the normal acrlyic cement. The tank already seems bounded quite well but I'll let it dry until Monday before testing it's watertight abilities.
- went to Mystee's final presentation

## weekend

Ran acclimation trials in groups of 10 and 11.

## 23 May 2016

- named group 3 fish.

```

require(redingPlot); require(babynames) names <- babynames %>%
filter(sex == "F") %>% select(name,n) %>% group_by(name) %>%
tally %>% arrange(desc(n)) names[grep("^[^SC].*[cs]$",
as.character(names$name)),] %>% .$name %>% unique %>% .[1:21]

```

## 23, 24 May

- went to data science in R course
- ran trials.
- worked on `locate_fish.py` for analyzing common garden trials.
- working on packaging `locate_fish.py` as an app and an `.exe` file
- neither py2app is working (getting an error even when I use development version) or py2exe (thinks python isn't installed). Perhaps try installing python 2.7.x onto mr freeze

## 25 May

- on mini1, I added a shell script called `encode` and install it to `/usr/local/bin`, that way I can simply call `encode` from anywhere on the command line and the script will execute because the location in my `$PATH`. `encode` is in `~/Documents/random_scripts/` on my iMac.
- looked into `Rcpp`. Takehome: only use for speed bottlenecks in your code. Make sure to make use of the vectorized C++ functions provided by `Rcpp`.
- ran trials
- Returned a bunch of females from the experiment back to home tank (community tank) this week

## 26, 27 May

- tried over and over to get `common_garden` python file as a standalone executable binary. Works on Mary and Ian's computers (Macs) but not on Kelly's Window machine. It works using `pyinstaller`. Py2Exe requires moving around some weird file that I didn't understand. Py2App doesn't work between my mac computers. Nuitka gives a `Illegal operation: 4` error whenever I run the resulting `.exe` on my mac.
- I ordered a refurbished, bottom of the line mini2.
- I wrote `bkp`, a script stored in my `usr/local/bin` that writes a tar bzip2'ed file. Works like `bkp dir_you_want_to_store dir_you_want_to_store_it_to`
- ran ~10 trials.
- make a cron job to send these notes to github every Friday

## **30 May**

- spend most of the day looking into Docker. Could be useful for transferring software (e.g. the tracking code)
- ran 12 trials

## **31 May**

- ran 9 trials
- ordered another mac mini; ordered screens from adafruit; ordered better airstones; cut felt for tank (but need to measure the screens from adafruit before gluing)
- made new 2% and 3% oat plates to be tested with the slime molds using the oat powder from the coffee grinder for the first time. It looks like it works pretty well except I need to find some way to stop clumping.

## **1-2 June 2016**

- ran trials.
- worked on the common garden GUI. Version 0.0.1 is operational.
- set up a new mac mini with the slime molds. This computer can also be used to run the common garden program and can be used for ImageJ and other lab applications.
- got a warning from IT. Note: Do not attempt to use GUI programs within a Docker container by porting the display via an X server to your machine. This leaves the machine open to attacks from hackers.

## **6 June 2016**

- Tried to change the common\_garden python script so that the user can see the video looped AND enter in behaviors at the same time. I can make a minor modification to the script as is to get both the opencv window showing the video and the Tkinter window open at the same time, but the user can't click inside the Tkinter window. Turns out solving this is not trivial.
- I've verified that I can have two Tkinter windows open at once.
- I've been able to port images taken through opencv into a Tkinter window.

- When I try to do both, I get an error on the opencv side of things saything that there is no default Tkinter window, which there isn't, because there can't be two windows at once. Catch 22.
- I've seen unanswered questions on StackOverflow similar to the issue I'm facing.
- Ran trials.

## 7 June 2016

- attempted to make new slime mold plates by smoothing out the lumpiness of the oats with a tissue homogenizer. Worked pretty well for on 25 ml.
- made a 'worksheet' for the students scoring the common garden videos to use.
- updated `common_garden.py` and the resulting exe file. Note that pyinstaller can create an .app, but that for some reason the app never works quite right. A workaround is to make an Automator script that simply calls the unix executable; this automator task can then be saved as an application you can drag into the dock.
- updated ImageJ protocol.
- Put several more documents related to the common garden online.
- Ran 14 trials.

## 8 June 2016

- ran 14 trials
- attempted to make new slime mold plates by smoothing out the lumpiness of the oats with a tissue homogenizer with 100 ml (4 plates) at a time. The homogenizer is too small, I think, to be effective with this much liquid. The last plate I poured seemed heavy on oats. There were still some large oat clumps in the resulting plates.
- continued to make changes to the tracker to try to find the best way to track the fish. Also made changes to `show_results.R`, an R script that draws tracking coordinates on a photo of the tank. Useful for validating the tracker and looking for spurious tracks.
- backed up mr hellfire and renamed videos.
- made short videos from these new common garden videos with `make_videos.sh`.
- formatted new external HD.

## 14 June 2016

Back from Virginia / Maryland.

- [This](#) webpage contains super useful stats explanations with applications in R. I'm printing out some of the more useful pdfs and putting them on the book shelf on my office. Also note that [this](#) book is also a super useful resource and is available as a free pdf at the link.
- I spent most of the day working on improvements and modifications to the tracking program. I'm trying to write things as a feather file as well as a csv. I altered my code so that it was more in line with PEP standards. I also added a couple of functions to replace bulky parts of the code, like parsing the arguments. I also added `distance_to_edge` which returns the distance between the edge of the tank and the fish. I hard-coded values that define the bounds of the tank and the left and right association zones. The result looks like this, with the tank bounds in green and the association zones in yellow:



Figure 2: bounds

I'm currently diagnosing a problem where the `angles` and `zones` list are 3x as long as they should be. Using the simple `cv2.blur`, tracking takes ~11 min for a 21 min video.

- I also experimented with the best way to create an agar oat plate. What I've found:

### **new protocol for making oat plates**

- Heat up 50 ml of agar (50 ml water with 2.2g agar) in microwave until dissolved.
- Meanwhile, measure the appropriate amount of oat for two plates. For 2% oat plates, this is 0.5g. Do this on the sensitive scale in the molecular lab.
- Add the oats to a 50ml Falcon tube. The tune should be completely dry, otherwise clumps will form in the next step.
- Pour in ~5 ml of the hot agar solution. Use the homogenizer to homogenize the mixture.
- Keep adding agar, a little at a time, to the Falcon tube.
- When all the agar is added, shape up the falcon tube.
- Pour 25 ml into a graduated cylinder and the rest into a plate. Pour the reserved 25 ml into a second plate.

## **15 June 2016**

- made some changes to the tracker, fixed some issues with creating the dataframe in pandas. Added ability to record the tracking output.
- Talked with Molly about the tracking program. Her suggestions:

### **suggestions for the tracking program:**

- Add a ‘plant zone’—a rectangle that includes the plant to see how much time the fish spends there.
- Add additional thigmotaxis measures. For example, in my current measure (pixel distance from the edge of the tank to the fish), a fish at the bottom of the tank but against the wall of the tank would have a greater thigmotaxis score than a fish that was also against the side of the tank but higher up in the water column. Figure out a way so that both would have the same score.
- Record latency to enter each zone.
- Record first zone entered.
- Add activity score (need to interpolate points; see below)
- Possibly add a Kalman filter
- Add three course zones (left, middle, right) ##### my additional thoughts:
- A lot of these things are incompatible with a dataframe, so I should write them as a json file to incorporate different data types

- You have R code that interpolates missing coordinates assuming the fish moves in a straight line. Code this in python and add to script OR use a Kalman filter in Python to do it.
- made sure the association zones were of equal size

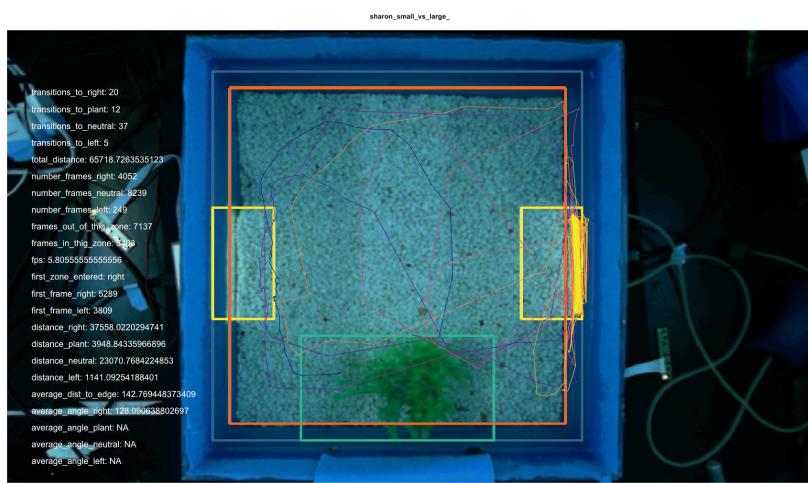
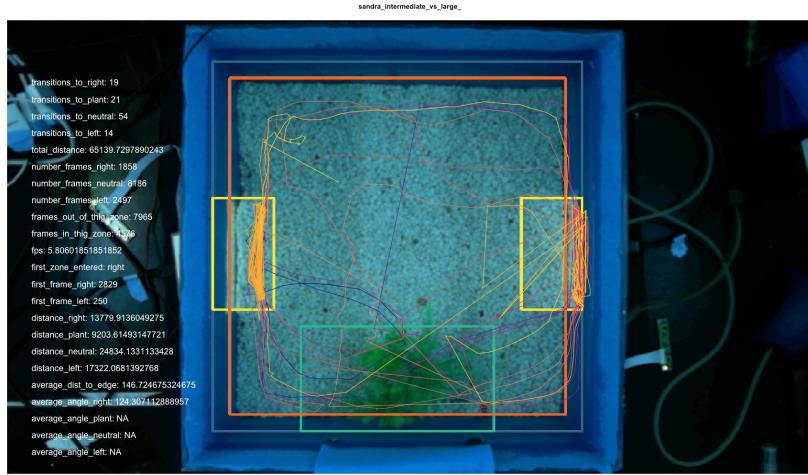
## 21 June 2016

I've been at the Evolution meetings for the past few days.

- I had Rachel and Caleb start to move tanks from 512 downstairs. Rachel started to learn to analyze videos from the common garden experiment.
- Read some papers on social networks.
- Small tweaks to the tracker.
- Added `parallel` to `/usr/local/bin`, use to run jobs on your machine in parallel. Used like `parallel -j 4 "time python tracker.py -t -i" /path/to/video.mp4`.
- slime mold growth is still slow.
- talked to Ian and ordered a go pro capable of 4k video to try to record trials.
- Reminder: add Mary's email to the website.
- The tracker seems to be working pretty well. Examples (these are created by `show_result.R` which is called from the python script):









## 22 June 2016

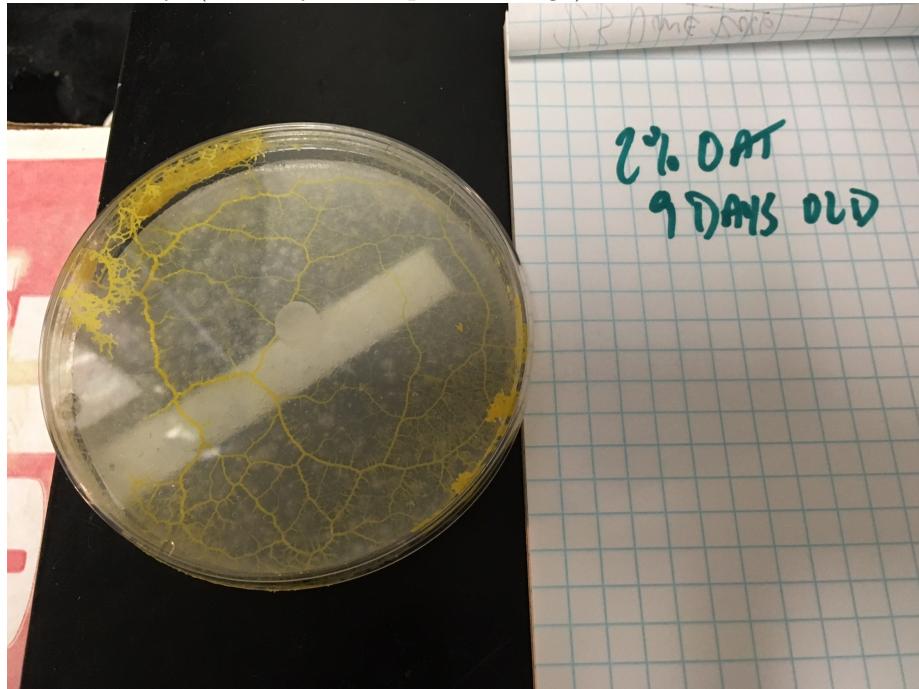
- Downloaded `gtime` (`brew install gnu-time`), the GNU flavor of the time utility that is much more informative than the UNIX `time`.
- Developed sheet that Quin can use to start scoring the slime mold videos.
- Got rid of the parallel script from yesterday, which wasn't working as desired, in flavor of GNU parallel. Works like this. `ls *.gz | gtime parallel -j+0 --eta 'zcat {} | bzip2 -9 >{.}.bz2'`. `-j+0` means *use all available cores*. `--eta` gives useful information about how long it will take. `{}` is the filename from `ls`. `{.}` strips the extension from the filename, so it'll make `star.gz` to `star`. `parallel` can also send jobs to several machines. See <https://www.youtube.com/watch?v=OpaiGYxkSuQ&list=PL284C9FF2488BC6D1> starting around minute 6. Another example: `ls *.mp4 | gtime parallel -j+0 --eta 'python tracker_no_hsv.py -i {} -t 2>&1 {.}.log'`
- spent awhile trying to write a python script that would do what `show_result.R` does now. Not straight-forward. This was motivated in large part because of the slight weird offset when plotting the tracks on the image.
- I also tried to use the gopro to take video. I think the low light conditions are going to make it hard. The resulting videos have a lot of noise and look grainy, even after manually setting a bunch of the setting for the camera.
- Quin scored 6 slime mold plates.

## 23 June 2016

- Continued to trouble shoot tracker for various issues, mainly that it looks like the tracker will often pick non-fish things up.
- learned basics of using ggplot2.
- Entered in slime mold data I have so far to do analysis. The R Markdown file is in my slime mold experiments folder. I added a choice overload private git repository.
- Decided to begin experiments for ‘irrational time allocation’ experiments. Set up 5vs10% and 9vs10% plates, each with 13 replicates, total  $n = 52$ . Tomorrow I need to make more 4% plates (see below).

From now on, grow slime molds on 4% plates. Seed with a single core cut with the largest cork borer. Wait 5 days before using in experiments. See photos below:

2% after 9 days (too veiny, not dispersed enough)



4% after 5 days



**Also:** Name your replicates. All it takes is writing a number like this:



## 27 June 2016

- Quin scored more slime mold videos. She also read several chapters in the poeciliid ecology and evolution book. Quin expressed interest in understanding the condition-dependent behavior of the intermediate male.
- Rachel took videos of the round 2 tanks and verified that all the round 1 tanks have the adults they should. She also measured SL of some of my transitivity fish.
- I emailed Molly about the changes I made to tracker for her approval.
- Figure club with Rong. We discussed [this paper](#). For next time:
  - [this paper](#)
  - [this paper](#) and [this paper](#)
- Made new 4% plates. Problems with clumping persist. I ordered a sifter.
- Transferred slime mold to 4% plate to grow up for a few days.
- Did choice overload data analysis, as Quin finished scoring the videos.

## 28 June 2016

- Organized my stats binder. Whenever I find a helpful online tutorial or stats resource, I print it out and put it in my stats binder.
- tweaks to tracker.
- Wrote code to help Emma measure distances between her spiders. See [github.com/lukereding/emma\\_spiders](https://github.com/lukereding/emma_spiders)
- Water changes to common garden tanks.
- Other general common garden tasks.

## 29 June 2016

- Quin scored the first video from the my time allocation in choice experiment.
- Quin figured out that the weird species that came with the swordtails from Mexico is *Gambusia vittata*.
- 

## 30 June 2016

### aleatorio

- I emailed with A. Dussutour and she is going to send me the Japan and Australian slime mold strains.

- Started writing the methods for the slime mold paper.

### **analyzing the transitivity videos**

The closest I've gotten to get parallel to work over multiple servers:   
`ls /Users/lukereding/Documents/chapter1/vary_size_only/trial_recordings/*.mp4 | head -n 4 | gtime --verbose parallel --eta -S 4/$mini2,4/$mini5,: -j4 --trc {} 'echopwd&& python tracker.py -i {} -t > {.}.log'`

To run just on mini1:   
`ls /Users/lukereding/Documents/chapter1/vary_size_only/trial_recordings/*.* | gtime --verbose parallel --eta -j4 'echopwd&& python tracker.py -i {} -t > {.}.log'`

That's what I ended up doing, and it's running pretty fast (~20 min for 4 videos). At 5:30 it's done 36 videos (I started it around 2).

In the future, when putting opencv on a mac:   
`brew install ffmpeg  
 brew tap homebrew/science  
 brew install --HEAD opencv --with-ffmpeg  
 from here`

### **making oat plates: a new direction**

I tried a new technique to make oat plates, and I think this one's a winner.

the “protocol” – - Warm up your agar in the microwave until it’s all dissolved.

- Meanwhile, sift your oat power twice.
- Then weight the right amount of your oat powder.
- Place a 200 ml beaker on a hot plate / stir plate with a stir bar. Add the oat powder and start stirring.
- (optional: turn up the heat on the hot plate.)
- Slowly add the hot agar over the oats. Add only a little at a time, and move the beaker around so that it dislodges any oat stuff to the sides of the beaker.
- Keep pouring.
- When done pouring, crank up the speed and chill out for a bit.
- Quickly measure 25 ml in a graduated cylinder and pour into a plate.

Reminder: Quin is gone 6 July - 3 August.

## **1 July 2016**

### **analyzing transitivity videos:**

#### **courtship only**

For the courtship varying trials:

```
ls /Users/lukereding/Documents/chapter1/vary_courtship_only/trial_recordings/*.mp4  
| gtime --verbose parallel --eta -j4 'python tracker.py -i {} -t >  
{.}.log'
```

Output from parallel:

```
Command being timed: “parallel --eta -j4 python tracker.py -i {}  
-t > {.}.log” User time (seconds): 70841.25 System time (seconds):  
2759.03 Percent of CPU this job got: 380% Elapsed (wall clock) time  
(h:mm:ss or m:ss): 5:22:36 Average shared text size (kbytes): 0 Av-  
erage unshared data size (kbytes): 0 Average stack size (kbytes): 0  
Average total size (kbytes): 0 Maximum resident set size (kbytes):  
1272463360 Average resident set size (kbytes): 0 Major (requiring  
I/O) page faults: 56180 Minor (reclaiming a frame) page faults:  
29185699 Voluntary context switches: 148273 Involuntary context  
switches: 24863225 Swaps: 0 File system inputs: 9826 File system  
outputs: 3942 Socket messages sent: 0 Socket messages received: 0  
Signals delivered: 119 Page size (bytes): 4096 Exit status: 0
```

Note that the log files save (annoying) to the location of the videos.

This took 5:22:36 for 60 videos, or 322.6s per video.

### courtship and size

```
ls /Users/lukereding/Documents/chapter1/vary_size_and_courtship/trial_recordings/*.mp4  
| gtime --verbose parallel --eta -j4 'python tracker.py -i {} -t >  
{.}.log'
```

### slime molds

Tried taking photos of slime molds. The results weren't great.

I should consider 3d printing the ‘troughs’ I was thinking about using for the choice order experiment.

### loss aversion

While I'm waiting for other slime mold strains to get here, I might as well do some pilots. The thing I'm most excited about is loss aversion.

**raising:** I'm going to try and raise slime molds on either 4% oat or 8% oat with 0.5M NaCl. They will then be tested (after a week) on 2% oat vs. 8% oat 0.75M NaCl. I made three 4% and 8% oat 0.5M NaCl plates today. The plate-making method of two days ago worked pretty well, although there were still some sizeable clumps in the 8% batch (I only sifted the oat flour once, but I don't think that can make that much of a difference).

## other

Rachel measured SL of my female transitivity fish and took videos of the round 1 tanks

Quin checked and made sure she saw all the adult fish in the round one tanks that should be there. She analyzed ~8 common garden videos after Mary and I went over how they should be scored.

## 2 July 2016

Tracker finished running on `vary_size_and_courtship` videos. I involved the script without the `-t` flag. The result:

```
Command exited with non-zero status 1 Command being timed:  
"parallel -eta -j4 python tracker.py -i {} > {}.log" User time (seconds): 108866.62 System time (seconds): 4134.54 Percent of CPU  
this job got: 363% Elapsed (wall clock) time (h:mm:ss or m:ss):  
8:37:47 Average shared text size (kbytes): 0 Average unshared data  
size (kbytes): 0 Average stack size (kbytes): 0 Average total size  
(kbytes): 0 Maximum resident set size (kbytes): 1275625472 Average  
resident set size (kbytes): 0 Major (requiring I/O) page faults:  
78918 Minor (reclaiming a frame) page faults: 52048249 Voluntary  
context switches: 245477 Involuntary context switches: 43846964  
Swaps: 0 File system inputs: 20900 File system outputs: 6799 Socket  
messages sent: 0 Socket messages received: 0 Signals delivered: 260  
Page size (bytes): 4096 Exit status: 1
```

For future reference:

```
python mod_time.py -i ..../trial_recordings/ [('sabrina_intermediate_vs_large_.log',  
168346.0), ('sabrina_intermediate_vs_large_.mp4', 23.0), ('sabrina_small_vs_intermediate_.log', 168812.0), ('sabrina_small_vs_intermediate_.mp4',  
26.0), ('sabrina_small_vs_large_.log', 158264.0), ('sabrina_small_vs_large_.mp4',  
23.0), ('sally_intermediate_vs_large_.log', 157148.0), ('sally_intermediate_vs_large_.mp4',  
13.0), ('sally_small_vs_intermediate_.log', 1230.0), ('sally_small_vs_intermediate_.mp4',  
1504.0), ('sally_small_vs_large_.log', 1226.0), ('sally_small_vs_large_.mp4',  
1492.0), ('samantha_intermediate_vs_large_.log', 1209.0), ('samantha_intermediate_vs_large_.mp4', 1507.0), ('samantha_small_vs_intermediate_.log',  
1240.0), ('samantha_small_vs_intermediate_.mp4', 1489.0), ('samantha_small_vs_large_.log', 1201.0), ('samantha_small_vs_large_.mp4', 3174.0), ('sandra_intermediate_vs_large_.log',  
1285.0), ('sandra_intermediate_vs_large_.mp4', 1478.0), ('sandra_small_vs_intermediate_.log', 1261.0), ('sandra_small_vs_intermediate_.mp4',
```

1520.0), ('sandra\_small\_vs\_large\_.log', 1172.0), ('sandra\_small\_vs\_large\_.mp4', 1483.0), ('sara\_intermediate\_vs\_large\_.log', 1265.0), ('sara\_intermediate\_vs\_large\_.mp4', 1490.0), ('sara\_small\_vs\_intermediate\_.log', 1221.0), ('sara\_small\_vs\_intermediate\_.mp4', 1499.0), ('sara\_small\_vs\_large\_.log', 1208.0), ('sara\_small\_vs\_large\_.mp4', 1490.0), ('sarah\_intermediate\_vs\_large\_.log', 1227.0), ('sarah\_intermediate\_vs\_large\_.mp4', 1497.0), ('sarah\_small\_vs\_intermediate\_.log', 1233.0), ('sarah\_small\_vs\_intermediate\_.mp4', 1464.0), ('sarah\_small\_vs\_large\_.log', 1308.0), ('sarah\_small\_vs\_large\_.mp4', 1535.0), ('savannah\_intermediate\_vs\_large\_.log', 1269.0), ('savannah\_intermediate\_vs\_large\_.mp4', 1487.0), ('savannah\_small\_vs\_intermediate\_.log', 1274.0), ('savannah\_small\_vs\_intermediate\_.mp4', 1260.0), ('savannah\_small\_vs\_large\_.log', 1303.0), ('savannah\_small\_vs\_large\_.mp4', 1491.0), ('shannon\_intermediate\_vs\_large\_.log', 1225.0), ('shannon\_intermediate\_vs\_large\_.mp4', 1503.0), ('shannon\_small\_vs\_intermediate\_.log', 1197.0), ('shannon\_small\_vs\_intermediate\_.mp4', 1486.0), ('shannon\_small\_vs\_large\_.log', 1206.0), ('shannon\_small\_vs\_large\_.mp4', 1482.0), ('sharon\_intermediate\_vs\_large\_.log', 1227.0), ('sharon\_intermediate\_vs\_large\_.mp4', 1482.0), ('sharon\_small\_vs\_intermediate\_.log', 1220.0), ('sharon\_small\_vs\_intermediate\_.mp4', 1529.0), ('sharon\_small\_vs\_large\_.log', 1217.0), ('sharon\_small\_vs\_large\_.mp4', 1471.0), ('sheila\_intermediate\_vs\_large\_.log', 1199.0), ('sheila\_intermediate\_vs\_large\_.mp4', 1498.0), ('sheila\_small\_vs\_intermediate\_.log', 1299.0), ('sheila\_small\_vs\_intermediate\_.mp4', 1480.0), ('sheila\_small\_vs\_large\_.log', 1178.0), ('sheila\_small\_vs\_large\_.mp4', 1485.0), ('sherry\_intermediate\_vs\_large\_.log', 1222.0), ('sherry\_intermediate\_vs\_large\_.mp4', 1498.0), ('sherry\_small\_vs\_intermediate\_.log', 1217.0), ('sherry\_small\_vs\_intermediate\_.mp4', 1486.0), ('sherry\_small\_vs\_large\_.log', 1255.0), ('sherry\_small\_vs\_large\_.mp4', 1489.0), ('shirley\_intermediate\_vs\_large\_.log', 1211.0), ('shirley\_intermediate\_vs\_large\_.mp4', 1472.0), ('shirley\_small\_vs\_intermediate\_.log', 1201.0), ('shirley\_small\_vs\_intermediate\_.mp4', 1494.0), ('shirley\_small\_vs\_large\_.log', 1191.0), ('shirley\_small\_vs\_large\_.mp4', 1486.0), ('sofia\_small\_vs\_large\_.log', 1229.0), ('sofia\_small\_vs\_large\_.mp4', 1486.0), ('sophia\_small\_vs\_intermediate\_.log', 1200.0), ('sophia\_small\_vs\_intermediate\_.mp4', 1496.0), ('sophia\_small\_vs\_large\_.log', 1214.0), ('sophia\_small\_vs\_large\_.mp4', 1516.0), ('stacy\_intermediate\_vs\_large\_.log', 1272.0), ('stacy\_intermediate\_vs\_large\_.mp4', 1488.0), ('stacy\_small\_vs\_intermediate\_.log', 1261.0), ('stacy\_small\_vs\_intermediate\_.mp4', 1495.0), ('stacy\_small\_vs\_large\_.log', 1305.0), ('stacy\_small\_vs\_large\_.mp4', 1480.0), ('stella\_intermediate\_vs\_large\_.log', 1238.0), ('stella\_intermediate\_vs\_large\_.mp4', 1478.0), ('stella\_small\_vs\_intermediate\_.log', 1238.0), ('stella\_small\_vs\_intermediate\_.mp4', 1486.0), ('stella\_small\_vs\_large\_.log', 1260.0), ('stella\_small\_vs\_large\_.mp4', 1491.0), ('stephanie\_intermediate\_vs\_large\_.log', 1272.0), ('stephanie\_intermediate\_vs\_large\_.mp4', 1524.0), ('stephanie\_small\_vs\_intermediate\_.log', 1226.0), ('stephanie\_small\_vs\_intermediate\_.mp4', 1498.0), ('stephanie\_small\_vs\_large\_.log', 1235.0), ('stephanie\_small\_vs\_large\_.mp4', 1479.0), ('sue\_intermediate\_vs\_large\_.log', 1292.0), ('sue\_intermediate\_vs\_large\_.mp4', 1267.0), ('sue\_small\_vs\_intermediate\_.log', 1291.0), ('sue\_small\_vs\_intermediate\_.mp4', 1494.0), ('sue\_small\_vs\_large\_.log', 1319.0), ('sue\_small\_vs\_large\_.mp4', 1536.0), ('susan\_intermediate\_vs\_large\_.log', 1261.0), ('susan\_intermediate\_vs\_large\_.mp4', 1494.0), ('susan\_small\_vs\_intermediate\_.log',

```

1215.0), ('susan_small_vs_intermediate_.mp4', 1471.0), ('susan_small_vs_large_.log', 1208.0), ('susan_small_vs_large_.mp4', 1531.0), ('suzanne_intermediate_vs_large_.log', 1260.0), ('suzanne_intermediate_vs_large_.mp4', 1507.0), ('suzanne_small_vs_intermediate_.log', 1266.0), ('suzanne_small_vs_intermediate_.mp4', 1484.0), ('suzanne_small_vs_large_.log', 1256.0), ('suzanne_small_vs_large_.mp4', 1474.0), ('sydney_intermediate_vs_large_.log', 1255.0), ('sydney_intermediate_vs_large_.mp4', 1477.0), ('sydney_small_vs_intermediate_.log', 1250.0), ('sydney_small_vs_intermediate_.mp4', 1493.0), ('sydney_small_vs_large_.log', 1245.0), ('sydney_small_vs_large_.mp4', 1485.0), ('sylvia_intermediate_vs_large_.log', 1209.0), ('sylvia_intermediate_vs_large_.mp4', 1491.0), ('sylvia_small_vs_intermediate_.log', 1219.0), ('sylvia_small_vs_intermediate_.mp4', 1488.0), ('sylvia_small_vs_large_.log', 1243.0), ('sylvia_small_vs_large_.mp4', 1500.0), ('test_small_vs_intermediate_.log', 1249.0), ('test_small_vs_intermediate_.mp4', 1661.0), ('whatever', 0.0)]

```

## 2 July 2016

I re-ran the tracking code for the fish in the `vary_size_only` experiment because of issues with the `mod_time.py` script. Actually, weirdly for 4 of the videos it said the time between creation adn modification was ~13 seconds, which is bizarre; the other videos were fine. Everything looked fine after re-running with a modified verion of `tracker.py` that said, “if the proportion of frames to keep is super low, use 0.7”. The output:

```

ETA: 0s Left: 0 AVG: 290.55s local:0/64/100%/309.5s Command being timed: "parallel -eta -j4 python tracker.py -i {} -t > {}.log" User time (seconds): 72527.50 System time (seconds): 2679.01 Percent of CPU this job got: 379% Elapsed (wall clock) time (h:mm:ss or m:ss): 5:30:06 Average shared text size (kbytes): 0 Average unshared data size (kbytes): 0 Average stack size (kbytes): 0 Average total size (kbytes): 0 Maximum resident set size (kbytes): 1273659392 Average resident set size (kbytes): 0 Major (requiring I/O) page faults: 45383 Minor (reclaiming a frame) page faults: 30109725 Voluntary context switches: 136158 Involuntary context switches: 24788670 Swaps: 0 File system inputs: 9115 File system outputs: 4478 Socket messages sent: 0 Socket messages received: 0 Signals delivered: 137 Page size (bytes): 4096 Exit status: 0

```

## 4 July 2016

```

ls /Users/lukereding/Documents/chapter1/vary_courtship_only/trial_recordings/*.mp4
| gtime --verbose parallel --eta -j4 'python tracker.py -i {} -t >

```

```
{.}.log'
```

with result:

```
Command being timed: "parallel -eta -j4 python tracker.py -i {} -t > {.}.log" User time (seconds): 70499.01 System time (seconds): 2833.08 Percent of CPU this job got: 378% Elapsed (wall clock) time (h:mm:ss or m:ss): 5:22:29 Average shared text size (kbytes): 0 Average unshared data size (kbytes): 0 Average stack size (kbytes): 0 Average total size (kbytes): 0 Maximum resident set size (kbytes): 1275887616 Average resident set size (kbytes): 0 Major (requiring I/O) page faults: 64824 Minor (reclaiming a frame) page faults: 28538529 Voluntary context switches: 152177 Involuntary context switches: 25153071 Swaps: 0 File system inputs: 12755 File system outputs: 4224 Socket messages sent: 0 Socket messages received: 0 Signals delivered: 126 Page size (bytes): 4096 Exit status: 0
```

## 5 July 2016

### slime molds

#### risk sensitive foraging

Chris Reid has a really interesting paper in which he presents slime molds with two arms that differ in the number, quality, and regularity of their food rewards. There are some interesting results, but he comes really close to doing something that would be really cool: testing for risk sensitivity for foraging in a slime mold, further connecting classic behavioral ecology with decision-making in the slime mold. I'm reading papers on risk-sensitive foraging now.

I set up a small trial where I present a slime mold with two arms. One of the arms has variable rewards, the other has even rewards. The number of rewards on each arm is the same. I set up 5 replicates but it looks like due to an error an will equal4.

### plates

The 0.5 M salt 8% oat plates I made are not going to work. The slime molds plated on them have not grown and are maybe dead. I'm going to make 8% oat plates with

- 0.1 M NaCl
- 0.05 M NaCl

- 0.025 M NaCl

and plate slime molds on each of those.

| plate       | salt amount | oat amount | agar amount | total volume |
|-------------|-------------|------------|-------------|--------------|
| 0.1M NaCl   | 0.32g NaCl  | 4.4g       | 1.1 g       | 55 ml        |
| 0.05M NaCl  | 0.16g NaCl  | 4.4g       | 1.1 g       | 55 ml        |
| 0.025M NaCl | 0.08g NaCl  | 4.4g       | 1.1 g       | 55 ml        |

I plated the molds (one plate of each of the above) around 6. .

### 3d printing

I also 3d printed a group of 4 y mazes and a group of 5 ‘troughs’ from each of two companies (3D hubs and make xyz). The idea would be to pour hot agar into the mold, let harden, then set up choice experiments. I plan on experimenting to see whether a y maze yeilds clearer results than a simple two-armed ‘trough’.

In the future when 3d printing, always use 3d Hubs.

### fish care

Rust abatement continues. Should be done by the end of today. Plan to move fish back into the fish room tomorrow if possible.

## 6 July 2016

### transitivity trials

Now that I’ve run all the tracking code on the videos, I’m rsync’ing the `chapter1` folder on `mini1` with the chapter 1 folder on `lpr_bkp_2` (my second backup hard drive). I’m syncing with the backup hard drive to avoid any conflicts I have between the `mini1` version of `chapter1` and the version on my desktop (although the one on `mini1` should be more up to date).

```
rsync -avt --delete --rsh=ssh $mini1:/Users/lukereding/Documents/chapter1
/Volumes/lpr_bkp_2/chapter1
```

### fish care

Spent all day moving tanks back into 512 and rearranging 512.

## 7 July 2016

### transitivity trials

Began to verify tracking worked for the transitivity trials. Plotted some basic graphs.

#### tracks

```
library(grid)
require(readr)
require(ggplot2)
require(viridis)

df <- read.csv("/Volumes/lpr_bkp_2/chapter1/chapter1/vary_size_only/coding/susan_intermediat

# get rid of places where interpolat_x_linear is NA
require(dplyr)
df %<% filter(!is.na(x_interpolated_linear))

utate(dy = y_interpolated_linear - lag(y_interpolated_linear, 1))

df %<%
  mutate(dx = x_interpolated_spline - lag(x_interpolated_spline, 1)) %>%
  mutate(dy = y_interpolated_spline - lag(y_interpolated_spline, 1))

ggplot(data=df, aes(x=x_interpolated_linear, y=y_interpolated_linear)) +
  geom_segment(aes(xend=x+dx, yend=y+dy , color = zone), arrow = arrow(length = unit(0.3,"cm")))
  scale_color_manual(values= viridis(5)[1:4]) +
  coord_flip() +
  theme_new()
```

And:

```
df <- read.csv("/Volumes/lpr_bkp_2/chapter1/chapter1/vary_size_only/coding/susan_intermediat

dt <- df %>% filter(!is.na(angle))
dt$cut_angle <- cut_interval(dt$angle,10)

ggplot(dt, aes(x = cut_angle)) +
  geom_bar(width = 1) +
  coord_polar() +
  scale_x_discrete(breaks=NULL) +
  facet_wrap(~zone) +
```



Figure 3: tracks

```
xlab("angle of fish") +
ylab("# of frames") +
theme_new()
```

## common garden

I also began to analyze the data taken from videos of the common garden experiment. I make a data frame from all the json files and plot using that data frame. The code is at `~/Documents/common_garden/common_garden_json_analysis.R`

## slime molds

The 0.1M NaCl has displayed growth but it's weak and it's starting to sporulate in the middle. Unsurprisingly, the 0.025M plate looks the best and the 0.05M plate is intermediate.

The 3d printed molds should be here on Friday and Saturday.

The pilot ‘change variance of food reward’ experiments were largely unsuccessful. I think the slime mold was too small to start out with, and the lack of barriers between the different replicates meant that the slime molds typically cheated.



Figure 4: angles

## 8 July 2016

I think there has to be some connection between models of risk sensitive foraging, Weber's law (see Kacelnik paper) and the decoy effect. Compare Fig. 1 of Kacelnik's Weber law paper to the neuroeconomics papers on your desk.

Read papers.

Helped move more tanks. Water changes, etc.

Talked with Meghan.

Set up experimental slime mold plates with 3d printed stuff as a proof of concept.

## 11 June 2016

### slime molds

Doing some simple pilots with the 3d printed 'troughs' I had made. Similar to two times ago, I put the slime mold in the center of the trough and place 5 oat disks on either side of it. The oat disks are either distributed randomly

according to a uniform distribution or are evenly spaced. There is always an oat disk on either side of the slime mold when it starts. One problem with this setup—which I got from Reid’s recent paper—is that it probably makes sense for the slime mold to go to the random arm because it’ll probably have to move less to consume all 5 oat disks (unless the last oat disk on the random arm is in the final position).

One thing I’m doing differently is letting the oat ‘diffuse’ into the agar before adding the slime mold. Reid’s Behav. Ecol. paper suggests that long-distance food sources take a while to move through the agar.

### **figure club**

Did figure club with Rong.

## **12 July 2016**

### **transitivity trials**

#### **plotting**

Spent a while in R tweaking colors and plotting aggregate data from each experiment to make sure there are no overall patterns of bias, etc.

#### **tracker.py**

Decided to redo tracking of all videos. Need to add a check to make sure aberrant frames aren’t included in the analyses. In the `vary_size_only` trials, check out `samantha_small_vs_large_` and `stephanie_intermediate_vs_large_` to make sure the changes have had the desired effect.

- Added

```
if prop > 1.1:  
    prop = 1 / prop
```

to `tracker.py`. For some of the videos, was getting a `prop` value of 1.4 instead of 0.7, thus the change.

- tracker now looks like aberrant frames, defined by the fish moving more than 200 pixels in a single frame. It deletes a single frame at a time, recalculates distances, etc, then repeats. This gets rid of all aberrant frames.

---

*aside:*  
Upgraded  
all pip  
packages. In  
the future,  
if you get a  
permissions  
error, use:  
pip  
install  
--upgrade  
--user  
SomePackage

To upgrade  
all pip  
packages at  
once, use:  
pip freeze  
--local |  
grep -v  
'^-\-e' |  
cut -d =  
-f 1 |  
xargs -n1  
pip  
install  
--user -U

---

- transferred new `tracker.py` to `minil` and re-ran `vary_size_only` videos using

```
ls /Users/lukereding/Documents/chapter1/vary_size_only/trial_recordings/*.mp4
| gtime --verbose parallel --eta -j4 'python tracker.py -i {}'
-t > {.}.log'
```

Output was similar to that below for the varying courtship only trials.

- ```
ls /Users/lukereding/Documents/chapter1/vary_courtship_only/trial_recordings/*.mp4
| gtime --verbose parallel --eta -j4 'python tracker.py -i {}'
-t > {.}.log'
```

Output:

```
ETA: 0s Left: 0 AVG: 302.33s local:0/60/100%/332.2s Command  
exited with non-zero status 5 Command being timed: "parallel -  
eta -j4 python tracker.py -i {} -t > {.}.log" User time (seconds):  
70382.00 System time (seconds): 2725.38 Percent of CPU this job  
got: 366% Elapsed (wall clock) time (h:mm:ss or m:ss): 5:32:14 Aver-  
age shared text size (kbytes): 0 Average unshared data size (kbytes):  
0 Average stack size (kbytes): 0 Average total size (kbytes): 0 Max-  
imum resident set size (kbytes): 1271267328 Average resident set  
size (kbytes): 0 Major (requiring I/O) page faults: 56008 Minor (re-  
claiming a frame) page faults: 28851525 Voluntary context switches:  
163174 Involuntary context switches: 29891280 Swaps: 0 File sys-  
tem inputs: 9986 File system outputs: 4182 Socket messages sent: 0  
Socket messages received: 0 Signals delivered: 119 Page size (bytes):  
4096 Exit status: 5
```

## 14 July 2016

### transitivity analysis

Kept working on transitivity analysis automatic error-checking. Now, after the tracker runs, I find the frames in which it says the fish moved > 200 pixels. I set the x/y coordinates for that frame as NA, then I repeat until the dataframe doesn't contain any excessive distances. Then I save this as a csv file. The json file contains a variable called 'number\_abbarent\_frames'; this is the number of frames that were corrected.

### slime molds

Wrote intro to choice overload paper.

## 15 July 2016

Size only and courtship only trials were analyzed last night. The job to run vary size and courting did not fire though.

```
ls /Users/lukereding/Documents/chapter1/vary_size_and_courtship/trial_recordings/*.mp4  
| gtime --verbose parallel --eta --bar -j4 'python tracker.py -i  
{} > {.}.log'
```

Scratch that. The courtship only trials were supposed to run but didn't for some reason.

To sync with backup drive for local analyses:

```
rsync -avt --delete --rsh=ssh $mini1:/Users/lukereding/Documents/chapter1  
/Volumes/1pr_bkp_2/chapter1
```

## 18 July 2016

### transitivity analysis

To recap:

I've re-run the tracker.py script on all the video. The distance cut-off is now 100 pixels between two consecutive frames. Note that the `vary_size_and_courtship` videos are the right length so I don't need to run them with the `-t` flag. This also means that I can run them on my iMac since I don't need access to the file creation times, which is what I did. This morning I transferred the resulting files from my iMac to mini1, then synch'ed mini1 with my backup drive.

## 19 July 2016

### transitivity trials

I was having some issues with some of the transitivity trial results. In some cases, towards the end of the trial, the coordinates of the fish would appear to move in a straight line away from the tank. The reason for this is that I was using a spline for the interpolation, not the linear interpolation, when there were NAs for the coordinates at the end of the dataframe.

Internally, I've been using the (potentially incorrect) spline data to compute the number of frames in the zones, etc. I also put off correcting a problem computing the distance between the fish and the edge of the tank, which was incorrect as well. This means I will have to re-run the tracker yet again.

I also fixed a problem where I was unable to get the distance from edge of the tank (variable and function names were identical).

### other

Went to career education workshop.

Added 'notify' to `/usr/local/bin/`; when called it creates a notification and rings a bell. Meant to be used when a command finishes running; `echo foo && notify`. Esp. useful for long commands.

## 20 July 2016

### fish

Went to BFL this morning to try and catch some females. Very unsuccessful. Will try again this afternoon.

### transitivity

I've realized that the json files I have don't really do what I want them to do. They are useful for making an aggregate dataset about overall what happened in a given trial, but they aren't fine-scale enough to say what happens in a given trial.

I've thought out it and I think the solution is to make a single dataframe from each experiment. Each row will be a frame from a video. I need to add a column for 'period' or what time period the trial is (are the screens showing background or male fish?). So I'll import in each csv into R, add the period column, then concatenate it onto the next dataframe.

Then if I want to get summary statistics, I can do something like `df %>% group_by(period) %>% summarise(avg(angle))` or whatever. Keeping all the data for an experiment together gives me the flexibility I need to do any analysis.

## 9 August 2016

Back from vacation.

- Submitted procard receipts to Sylvia.
- Fed fish.
- Emailed Quin about coming in next week.
- 

## 10 August 2016

### slime molds

### choice overload

I've been plagued by the low sample sizes of the choice overload study, so I'm going to try to increase the sample sizes. I've made a schedule ending 24 August

that should increase the sample size in each experiment to ~100. Justification: The accuracy in the two-way choice is 93% right now. If I wanted to detect a 20% decrease in accuracy, I would need a sample size of 100 per group.

### loss aversion

In order to do the loss aversion experiment I want to do, I need to present slime molds with a pretty clear choice between two options (say, x and y). If  $p(x)$  is near 0.5, it's relatively more difficult to detect an effect than when  $p(x)$  is really low or really high. Right now I'm thinking:

|   | [oat] | [NaCl] |
|---|-------|--------|
| x | 8%    | 0.1 M  |
| y | 2%    | 0M     |

I need to test and see whether the slime molds display a clear preference though. I think that the slime mold will highly prefer x to y, but I need to test this.

---

*Side note:* to render this as a pdf use pandoc: `pandoc lab_notes.md --latex-engine=xelatex -f markdown -o lab_notes.pdf && open lab_notes.pdf` (or just type `render_lab_notes` in the terminal) or as html, `pandoc -s -S --toc -c ~/Documents/css/buttondown.css lab_notes.md -o lab_notes.html && open lab_notes.html`