```
Error: could not open file /Users/jbeasley/Documents/College/F19/CS191/src/
NetworkEpistemology.jl
```

Evaluation

As a part of this thesis, I wrote NetworkEpistemology.jl, a Julia package that makes it easy to write agent-based epistemology models and simulate them efficiently. I chose Julia over NetLogo, which is commonly used in agent-based modeling, because Julia runs faster on large graphs and makes it very easy to decompose a model into several interconnected pieces. This makes it easy to, for example, swap a model of beliefs without changing code that models social interactions.

This package includes an implementation of Zollman's "transient diversity" model, which successfully replicates the same results Zollman presented in his paper. Starting from this replication, I swap Zollman's ideal graph models for the social epistemology citation graph generated above while leaving the rest of the model unchanged.

In this model, I collect the following information at each step. The model is in agreement if every author in the model believes the action that actually has the highest probability of success has the highest probability of success. Zollman presents the metric of "Probability of Successful Learning" which is defined as the percentage of trials where the authors were in agreement after 10000 steps. I also collect the actual fraction of authors that are correct, and a measurement of how much their beliefs are changing from step to step. For this model, this is the sum of the change in expectations of each action from step to step. Smaller changes should mean the model is less likely to change.

```julia
struct TransientDiversityStepStats
    agree::Bool
    fractionCorrect::Rational{Int16}
    totalBeliefChange::Float64
end
```

```
totalBeliefChange_by_run_df (generic function with 1 method)
```

Scaling Up Zollman

First, I'll run the model on a cycle graph, but with many more nodes than those considered by Zollman. Whereas the largest such model considered by Zollman had eleven agents, this one uses 500 because that is closer to the size of a more realistic academic community.

```julia
cycle_spec = TransientDiversityExperimentSpec(
    TransientDiversityModelState(SimpleDiGraph(cycle_graph(500)), 1000, [0.5, 0.499], Unifor
    10000,
    100,
```

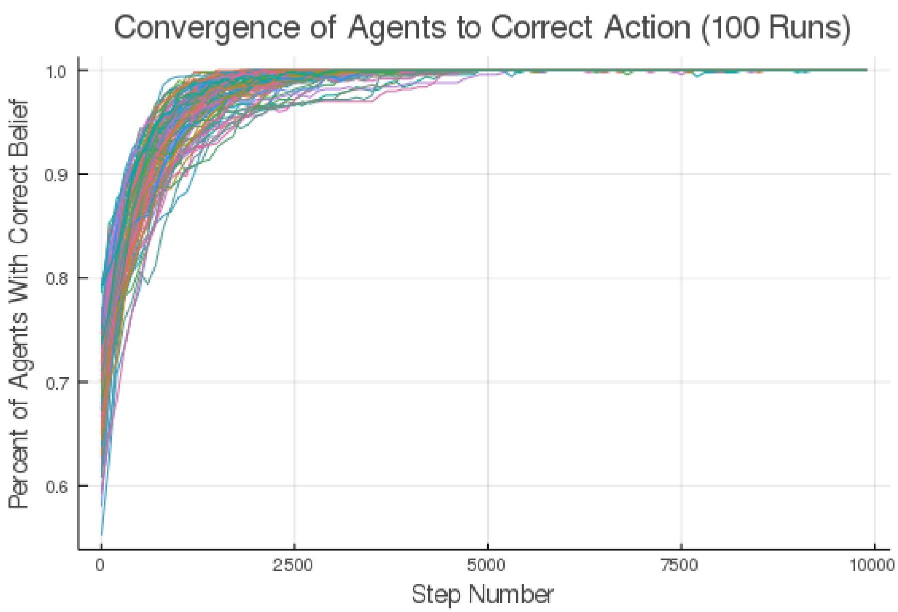
```
    100
)
cycle_results = run_experiment(cycle_spec)
mean([res.agree for res in cycle_results[:,end]])
```

```
1.0
```

In this case, we see that in every run, every agent believed the better action was better. This is in line with Zollman's results showing larger cycle graphs converged at high rates.

We can look in more detail if we see how many agents are converged at each step. In the following graph, each color of line corresponds to a different run and the graph shows that the network converges relatively quickly until every agent has the right answer. On some runs, a stray agent will flip back to the incorrect answer, then flip back again, which can be seen past the 5000th step.

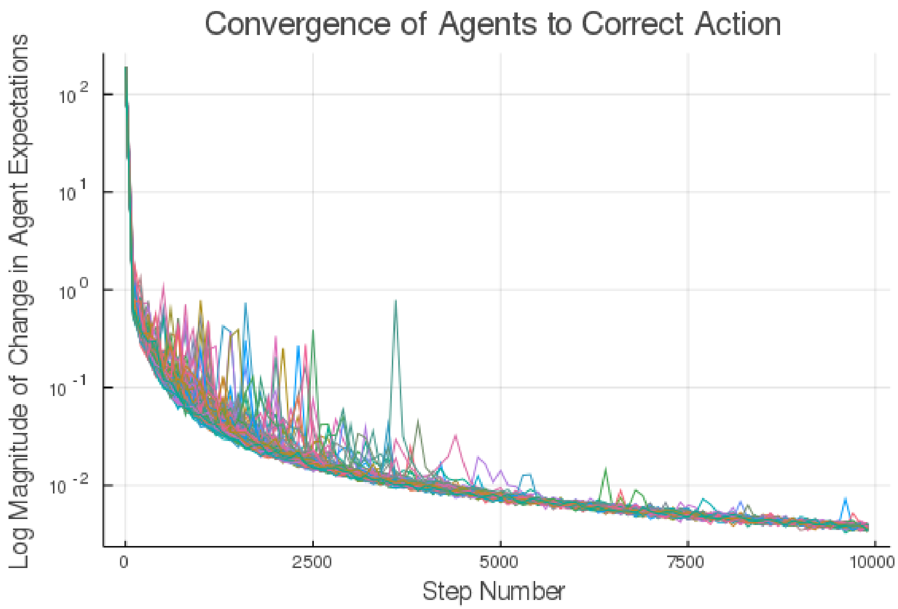
```
@df fractionCorrect_by_run_df(cycle_results, cycle_spec) plot(:StepNumber, cols(2:100),
    xlabel = "Step Number",
    ylabel = "Percent of Agents With Correct Belief",
    title = "Convergence of Agents to Correct Action (100 Runs)",
    label = "")
```



To get at this property of convergence a bit more rigorously, we can look at the magnitude of the changes in underlying beliefs. Graphing the magnitude of these changes on a log scale, we can see that the vast majority of change happens in the first few steps, with little change happening beyond the 5000th step.

```
@df totalBeliefChange_by_run_df(cycle_results, cycle_spec) plot(:StepNumber, cols(2:100),
```

```
    xlabel = "Step Number",
    ylabel = "Log Magnitude of Change in Agent Expectations",
    yscale = :log10,
    title = "Convergence of Agents to Correct Action",
    label = "")
```



Modeling Social Epistemology

If we load the subgraph of the MAG created in the previous section, we can directly run Zollman's model on the citation network of social epistemology. This graph is larger still, containing 799 nodes, however its pattern of edge connections is much more complicated than the simple cycle, wheel and complete structures that Zollman uses.

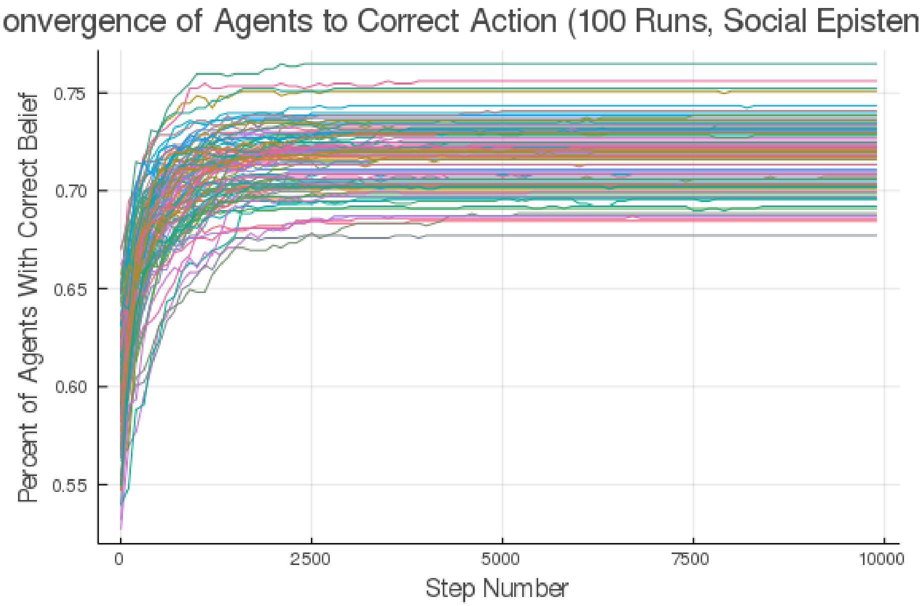
```
g = loadgraph("social_epistemology.graphml", "digraph", GraphIO.GraphML.GraphMLFormat())
social_spec = TransientDiversityExperimentSpec(
    TransientDiversityModelState(g, 1000, [0.5, 0.499], Uniform(0, 4), Uniform(0, 4)),
    10000,
    100,
    100
)
social_results = run_experiment(social_spec)
mean([res.agree for res in social_results[:,end]])
```

```
0.0
```

Notice that on none of the runs did all authors converge to the correct answer.

Now, we must check the more granular step-by-step results to see what happened.

```
@df fractionCorrect_by_run_df(social_results, social_spec) plot(:StepNumber, cols(2:100),
    xlabel = "Step Number",
    ylabel = "Percent of Agents With Correct Belief",
    title = "Convergence of Agents to Correct Action (100 Runs, Social Epistemology)",
    label = "")
```



In the above graph, we see not only that on none of the runs did all the authors converge to the right answer, but that the model converged at essentially the same rate, but hit a local maxima.

Now, one counterargument to this might be that because the graph is directed and has no large strongly connected component, it isn't truly a cohesive community and might be artificially so because citations almost certainly underestimate communication. So, to show that this is not the root of the issue, I run the same experiment on the same graph with every edge modified to be bidirectional. Because there was a large weakly connected component, the overall graph becomes much more connected and cohesive.

```
ug = union(reverse(g), g)

social_spec_ud = TransientDiversityExperimentSpec(
    TransientDiversityModelState(g, 1000, [0.5, 0.499], Uniform(0, 4), Uniform(0, 4)),
    10000,
    100,
    100
)
```

```
social_results_ud = run_experiment(social_spec_ud)
mean([res.agree for res in social_results_ud[:,end]])
```

```
0.0
```

After running this experiment, we see again that in no run does every agent converge to the correct answer and that the model converges much like the directed one did.

```
@df fractionCorrect_by_run_df(social_results_ud, social_spec_ud) plot(:StepNumber, cols(2:10
    xlabel = "Step Number",
    ylabel = "Percent of Agents With Correct Belief",
    title = "Convergence of Agents to Correct Action (100 Runs, Undirected Social Epistemolo
    label = "")
```