MABE information

11/11:

Hi Larissa,

It's not your fault; there were a few problems in the code. I fixed them and included a small world (TPMWorld) based on test world that only calls the tpm description function for each agent, and stores it in such a way that it will show up in the data csv file for the population. If you turn on population snapshots for each generation, and only run 1 generation, then you should see the data csv for the whole population being output to a new snapshot csv file. Alternatively, if you're running the whole experiment again, you can include the call to getTPMDescription() in that experiment world, and store the data in a similar fashion.

If you're integrating what I've attached into another project you already have going, the following files have been modified and will need to be copied over:

Brain/MarkovBrain/MarkovBrain.\*

Brain/MarkovBrain/Gate/AbstractGate.h

Brain/MarkovBrain/Gate/DeterministicGate.h

Brain/MarkovBrain/Gate/DecomposableGate.h

Brain/MarkovBrain/Gate/DecomposableFeedbackGate.h

World/TPMWorld/

Let me know how else we can help out,

 - Jory

11/16

You need these two files and then the TPM is included in the output.   
Note however that it is a state x state TPM which is really big for 2^14 states.   
For deterministic gates it should be exact, for decomposable gates you need to increase the

int maxReps=2; in MarkovBrain.h in the getTPMforTimepoint() function to probably at least 20.   
  
You can use this to compare the TPM you constructed in the deterministic case, but you might want to first collapse the stateXstate to stateXnode (as we just talked about).

12/7:

The code Jory gave me already is parallel to the deterministic, just took me forever to confirm that.   
So, ideally you should be able to recreate the TPMs of the direct decomposable gates in a straightforward manner. The only difference is the instead of   
mod(raw\_prob, 2) (which gives a 0 or 1)   
now it is:   
raw\_prob/255 (which gives a value between 0 and 1.   
Note: you have to recompile, and recreate the settings, so that you have the decomposable direct gates in the organism file.   
DecomposableDirectGate.\* goes into the Gate folder   
GateBuilder.\* into MarkovBrain   
and   
CircularGenome.cpp into CircularGenome   
I think that is it.   
  
L

12/11:

So I think what happens is that it doesn't actually recompile the header files if there are no changes elsewhere.   
It did apply my changes to the MarkovBrain.h header file when I added an output line to TPMWorld.cpp   
Not sure how to fix it. Just change some cpp file, recompile, and then it should do the 100 repeats. That will take ~30 s.

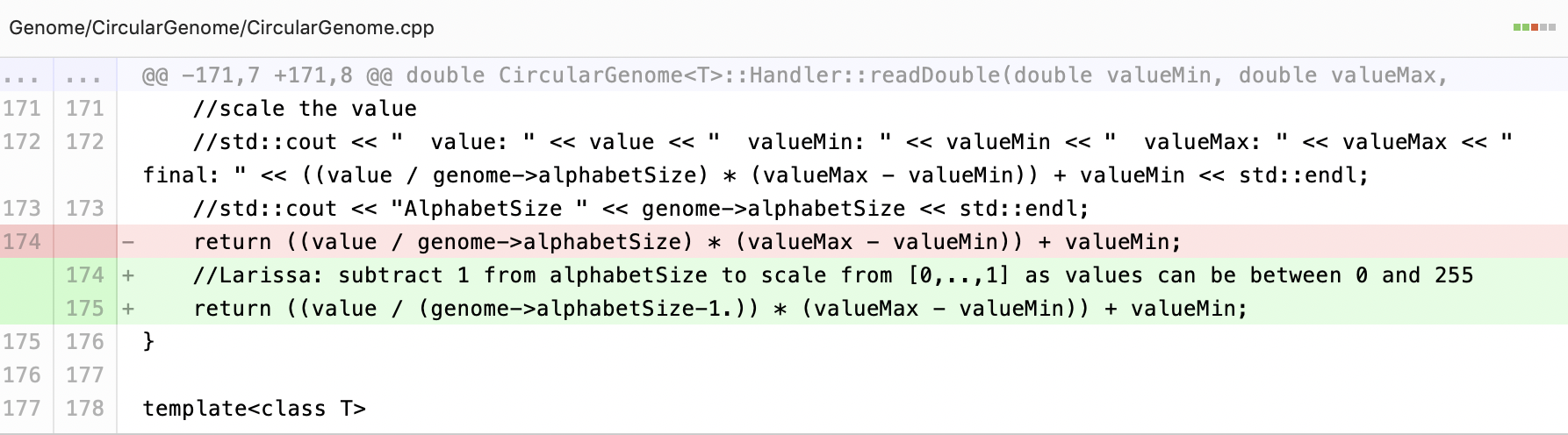
19/8/13:

GateBuilder.cpp

Larissa: I implemented a factorsList output to the DecomposableGates. However, we don’t actually care about those. Only the DecomposableDirectGates

Apart from the above and CircularGenome.cpp, none of the files I had sent previously changed in the updated developer version.

This is for CircularGenome<double>



TPM world: To make that run more changes to all the gates might be necessary. See “Record TPMs with TPMWorld” commit