Contents

1 Setup

This experiment tests the performance improvement of running function prediction on the DREAM networks after denoising the networks in various ways.

We look at class of denoising methods that all work by adding new edges to the network based off of some link predictor. In our case, the methods add the top 10% of edges from the link predictor to the network with a constant weight of 1.

There are 3 link prediction algorithms used:

- Pairwise distance predictor under normalized DSD
- GLIDE predictor under the L3 metric
- GLIDE predictor under the commone weighted metric

The parameters for the GLIDE predictor used in the experiment are as follows:

```
params = {"alpha" : 1, "beta" : 1000, "delta" : 0.001, "loc" : "l3"}
params = {"alpha" : 1, "beta" : 1000, "delta" : 0.001, "loc" : "cw"}
```

Performance was evaluated using 5-fold CV with the GO label set from 2019 where all GO labels annotating less than 50 or greater than 1000 proteins are discared (I use Lily's code to do this).

2 Results

The accuracy for each method is listed below:

	Normalized DSD	GLIDE CW	Glide L3	Network Enhancement	No Denoising
Dream 1	0.197	0.161	0.163	0.239	0.246
Dream 2	0.133	0.132	0.123	0.145	0.132
Dream 3	0.143	0.152	0.139	0.180	0.125
Dream 4	0.114	0.114	0.116	0.120	0.138
Dream 5	0.082	0.074	0.075	0.078	0.071
Dream 6	0.190	0.192	0.192	0.202	0.166
Average	0.143	0.137	0.135	0.161	0.146

3 Conclusion

Immediately we see that in every case, our algorithm performs worse than the network enhancement method. This should not surprise us. NE is tailored to the task of denoising networks, whereas our algorithm is not intelligent about how it adds new edges to the network. That is, we do a dumb thing and still get a good result in many cases.

In the case of DREAM networks 3, 5, and 6 we improve the performance of FP against that of the raw network, whereas in the remaining networks the performance degrades. I believe this is mainly due to the stupid weighting scheme we employ. In DREAM 1, edges have weights $w \in [0, 1]$, so when our denoising algorithm adds edges with weight 1, it is making too bold of claim about those edges importance. This is similarly the case with DREAM networks 2 and 4 and is what I think results in poor performance.

In contrast, the weights in DREAM 3 are positive integers and in DREAM 6 are in the set $[1, 100]$. So it make sense that predicting many low confidence edges would improve performance in these networks. The one outlier is DREAM 5. Performance does slightly improve, but weights are in the set $[0, 1]$. It seems strange, but upon looking closely at the network one sees that most weights are close to 1 and that performance barely improves.

This is a hopeful result to see so early on in experimentation. Even with an extremely stupid weighting denoising algorithm, performance is greatly improved in some cases. I think that if we employ something even slightly more intelligent such as average or min-edge weighting, performance will increase significantly.