BioNetwork Research: WindowReps 2019 Spring Report

Henry Ye

During this quarter, I started testing the capabilities of windowReps to be the network alignment seeds and databases index.

Network Alignment Seeds

As network alignment seeds, the windowReps are expected to be as stable as possible. Therefore, I used the average life spans of each windowReps as the metrics to test whether the windowReps are suitable for being used as network alignment seeds or not. In the testing process, I started with SCerevisiae network, selected MCMC as the sampling method, and chose 30 as the window size (w) and 300,000 as the window sample sizes (n) since it can cover up to 95% of the SCerevisiae network based on the findings from last quarter. Eventually, I got the following average life span results by using different windowRep searching methods:

	MIN	MAX	DMIN	DMAX	LFMIN	LFMAX
3	12.028959	6.915173	12.037479	11.999905	6.937267	6.989285
4	8.927627	4.716332	8.951142	7.069791	4.656124	4.744285
5	6.968724	4.078217	5.078618	4.283661	4.055879	3.982502
6	NaN	3.644163	3.866616	3.577081	3.615804	3.618272

Table 1. SCerevisiae: Average life span of windowReps k=3-6 using different windowRep searching methods

(k=7/8 are not included here since it takes a very long time for the algorithm to get the results)

	MIN	MAX	DMIN	DMAX	LFMIN	LFMAX
3	12.128457	10.042469	12.085017	12.129659	10.092097	10.149731
4	8.961689	4.106403	8.947246	6.231512	3.611983	3.606889
5	NaN	3.137499	6.292135	4.224962	3.042856	2.914546

Table 2. IIDyeast: Average life span of windowReps k=3-5 using different windowRep searching methods

(k=6/7/8 are not included here since it takes a very long time for the algorithm to get the results)

From Tables 1&2 above, we can see that the average life spans of windowReps are all less than 15 steps when the window size is chosen to be 30. Besides, there's a noticeable decreasing trend of average life span as k increases, which is reasonable since it will become much harder to maintain a graphlet having more nodes. Also, by looking at the distributions of windowRep life spans (figure 1&2) below, we can clearly observe a decreasing trend again, and are able to fit an exponential distribution for most of the cases.

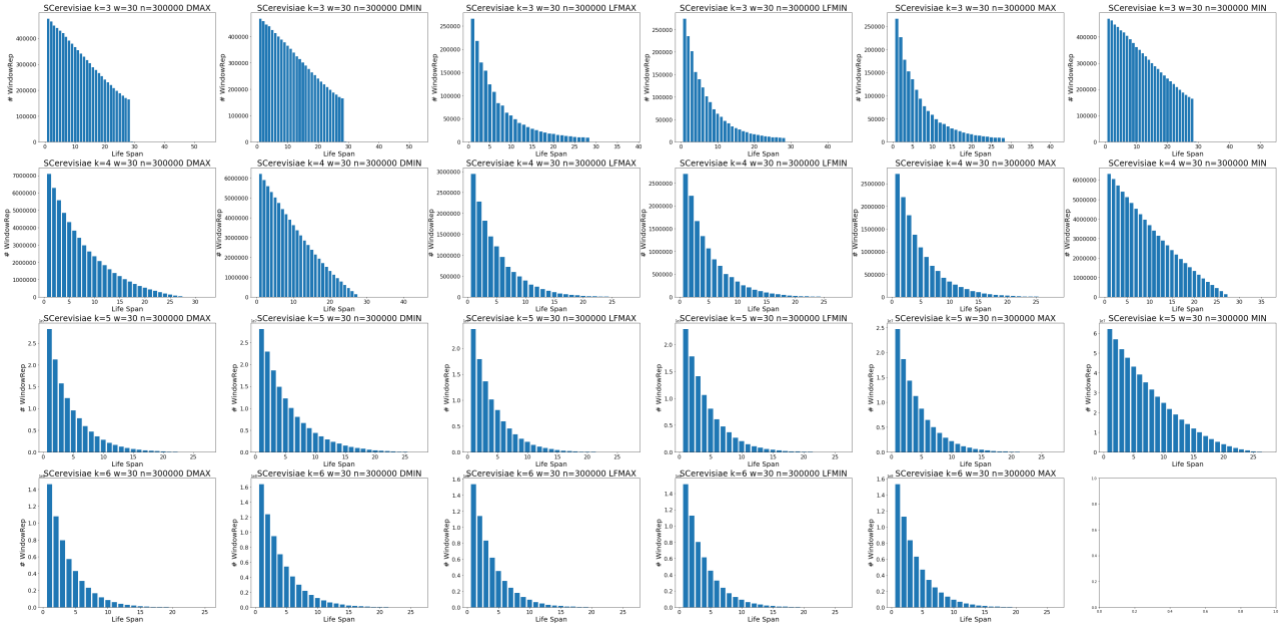

Figure 1: SCerevisiae: WindowReps life span distributions for K=3-6 using different windowRep searching methods

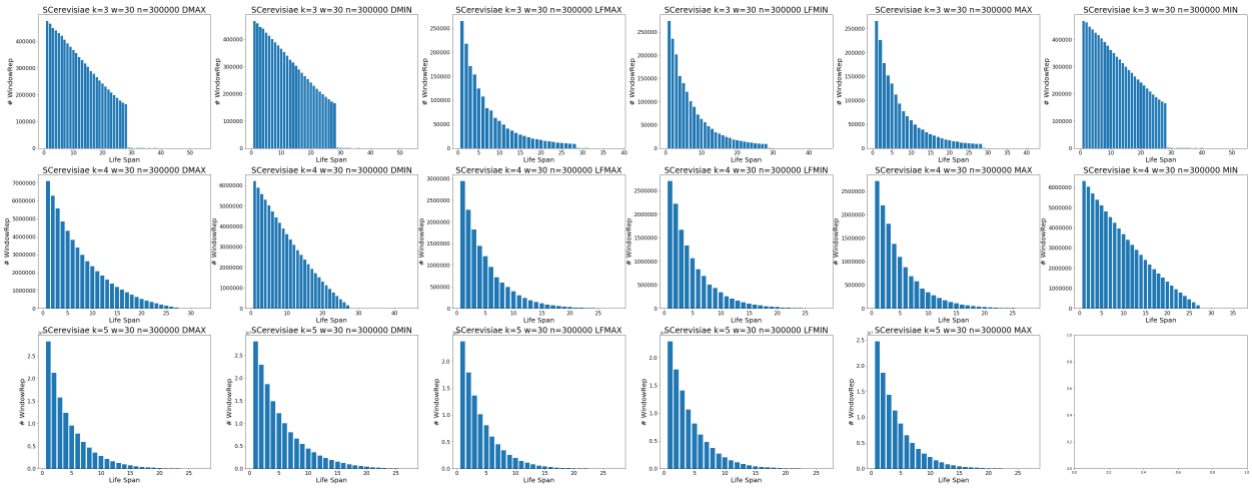

Figure 2: IIDyeast: WindowReps life span distributions for K=3-5 using different windowRep searching methods

According to the results above, we can conclude that windowReps are not ideal for being used as network alignment seeds when $k \geq 5$ since the windowRep will change within 5 steps for most of the time. However, when k=3/4 and windowRep searching method is chosen to be MIN or DMIN, there will be still quite a few windowReps having life span at around 10. Hence, for future work, k=3/4 graphlets can be investigated more as network alignment seeds.

Databases Keys

In order to test the potential of windowReps as network databases keys, we have to check whether the windowRep algorithm can produce characteristic k-graphlets in different networks. Therefore, we can firstly run the windowRep sampling algorithm twice on the same network and look for the windowRep overlapping percentage, as the higher of the overlapping percentage, the better of windowReps to be as database index.

During the experiment, various sampling methods, windowRep searching methods, windowRep sizes (k), window sizes (w), sample sizes (n) were used. I began with MCMC sampling algorithm again. However, I barely got any overlapped windowReps since MCMC is a walking algorithm, which requires much more window samples to cover a certain percentage of the network and make windows overlapped than the usual random sampling method. Thus, once I switched the sampling method to NBE, I was able to get a few overlapped windowReps at least.

MIN	MAX	DMIN	DMAX	LFMIN	LFMAX
27%	9.2%	27%	30.7%	6.7%	6.7%

Table 3: SCerevisiae: WindowRep overlapping percentage when k=4, w=30, n=2M using different windowRep searching methods

From Table 3 above, we can notice that none of the 6 windowRep searching methods is able to produce overlapping percentage greater than 50% or even 40%. Besides, since the results above are from k=4, and it's even harder to overlap graphlets having more than 4 nodes (requires much more window samples and searching time), we can expect lower overlapping percentage for k>4 (for reference, k=5 ≈ 0.3%, k=6 ≈ 1e-03%, k=7 ≈ 1e-07%).

Since windowRep overall overlapping percentage is not promising, we can possibly look at the overlapping results of a specific k-graphlet windowRep. In this case, I used maximizer as the windowRep searching method because it has the best performance of uniqueness.

Canon Ord	3	6	7	8	9	10
Percentage	0.4%	2%	4%	26%	36%	89.4%

Table 4: SCerevisiae: WindowRep overlapping percentage when k=4, w=30, n=2M using Maximizer

Using the same windowRep sizes (k), window sizes (w), sample sizes (n) but looking at the overlapping percentage of a specific canonical k-graphlet produced by maximizer searching method instead, we can reach almost 90% for graphlets having ordinal canonical equals to 10 according to Table 4. But if we change k=5 and keep everything else the same (Table 5), the largest percentage turns to 17% only. Consequently, the windowRep itself might not be suitable as network database index.

Canon Ord	4	10	11	14	16	17
Percentage	4.7e-03%	0.04%	3.6e-03%	0.03%	0.07%	2.24e-03%

18	19	22	23	24	25	26
0.09%	3.9e-03%	0.09%	0.08%	0.07%	0.12%	0.74%

27	28	29	30	31	32	33
1%	0.68%	1.78%	1.18%	2.8%	3%	17%

Table 5: SCerevisiae: WindowRep overlapping percentage when k=5, w=30, n=2M using Maximizer

Conclusion

- The windowRep is not suitable as network alignment seeds when $k \geq 5$, but we can investigate more for smaller graphlets ($k = 3$ or 4).
- The windowRep is not promising for being used as network database index.

To summarize, during the past quarter, I've:

- Fixed memory leak of windowRep algorithms.
- Added six windowRep searching methods in the BLANT repo
  (-p [MIN|MAX|DMIN|DMAX|LFMIN|LFMAX])
- Found life spans of windowReps by using different windowRep searching methods to test windowReps as network alignment seeds.
- Obtained overlapping results by choosing different k, w, n, and windowRep searching methods to test windowReps as network database keys.

Future work

- Investigate more for small graphlets as network alignment seeds.
- Focus on 8-graphlets, found its sub-cannon graphlets and sup-cannon graphlets, and test their potential of being network database index.