Growing Assembly Graphs: A Brief Literature Review

Camille Scott

May 5, 2016

Abstract

Significant prior work has been done on the statistics of the sequencing process itself, where the sequencing requirements for sampling a complete genome are explored; the construction of assembly graphs from sampled fragments; and the assembly process itself, which is mostly an exercise in the interrogation of these graphs. To better understand our problem, I will review several papers that span these topics. The first is one of the original papers describing the sequencing statistics; the second, a detailed treatment of the assembly problem and through the lens of different assembly graphs' potentials to fully represent the information in a genome; the third examines percolation theory in probabilistic de Bruijn graphs, which serves as a useful jumping-off point for our project.

1 Sequencing Statistics

Paper: J. C. Roach, "Random subcloning.," *Genome Research*, vol. 5, pp. 464–473, Dec. 1995

Roach's work extended a coverage model original proposed by Lander and Waterman [2]; the prior model is accurate at low redundancy, but not at the higher redundancies present in shotgun studies. This paper came from the days of "strategic sequencing," when sequencing technology was still in its infancy and optimizing cost versus information was extremely important for genomics projects. While we no longer have this concern due to the proliferation of extremely cheap shotgun sequencing, Roach's coverage equations are still important for understanding how genomic information increases as additional fragments are added, and as such can help us understand how assembly graphs grow as more fragments are added.

This model starts by assuming a linear discrete genome G, from which n fragments of length L are generated. Fragments cannot overlap the end of G, so there are $G_e = G - L + 1$ available start sites for fragments. Redundancy R is $\frac{nL}{G}$. We assume that T bases of a pair of fragments are necessary to detect an overlap. They then look at the domain space of spacings between start sites

D_k. They show that the density function for these lengths is:

$$f_{D_k}(x) = n(1 - \frac{1}{G_e})^{n-1}$$

From this, and the observation that a gap occurs at a site S_k iff $D_k > L - T$, they go on to calculate the probability of a gap following a fragment, where $f_G = fracL - TG_e$ is the effective fractional target coverage per fragment (in other words, the contribution of each fragment), as:

$$P_{gap} \approx \int_{L-T}^{G} f_{D_k}(x)dx = (1 - f_G)^n$$

From P_{gap}, they then go on to derive: the expected number of gaps $E(N_{gaps})$, the expected number of islands $E(N_{islands})$, the expected fragments per island, and several models of the distribution of lengths of the islands.

2 Assembly Feasibility

Paper: G. Bresler, M. Bresler, and D. Tse, "Optimal assembly for high throughput shotgun sequencing," *BMC Bioinformatics*, vol. 14, no. 5, pp. 1–13, 2013

3 Percolation in de Bruijn Graphs

Paper: J. Pell, A. Hintze, R. Canino-Koning, A. Howe, J. M. Tiedje, and C. T. Brown, "Scaling metagenome sequence assembly with probabilistic de Bruijn graphs," *Proceedings of the National Academy of Sciences*, vol. 109, no. 33, pp. 13272–13277, 2012

This work was primarily concerned with the employing a probabilistic data structure for the representation of a de Bruijn graph. Previous de Bruijn graph implementations stored k-mers exactly, requiring massive memory. Here, the authors use a Bloom filter [5] to only store the hashed k-mers; traversal is accomplished by exhaustively checking for the presence of neighbors. Because each node can have at most eight neighbors (four in either direction), the additional cost is minimal. This advancement radically reduced the memory required to store a de Bruijn graph, while introducing a new challenge to contend with: Bloom filters have a false positive rate, parameterized by the size of their underlying table and the number of elements they contain. In a network sense, this means that some nodes will gain neighbors corresponding to sequences which do not exist in the original set of fragments.

The practical goal of this work was to divide an extremely large set of fragments into approachable chunks. In metagenomics, the set of fragments contains genomes from potentially millions of species; in a "perfect" data set, where genome share no sequence content and there is no error, each of these genomes would be represented by its own connected component in the assembly graph. An effective solution to decomposing the metagenome assembly problem is then

to find all the disconnected components in the assembly graph and divide up the fragments based on the component to which they correspond. To use the probabilistic de Bruijn graph for this task, it was important to determine whether false positives in the data structure could cause otherwise disjoint components to become erroneously connected. It was observed that for increasing false positive rates, the effect on the connectivity was similar to a geometric phase transition: at false positives rates of about 0.18, the average component size begins to grow rapidly. This is first shown via simulation, and then studied with percolation theory. They used a site percolation model where the probability p of a site being active is equivalent to the false positive rate of the bloom filter.

References

[1] J. C. Roach, "Random subcloning.," *Genome Research*, vol. 5, pp. 464–473, Dec. 1995.

[2] E. S. Lander and M. S. Waterman, "Genomic mapping by fingerprinting random clones: a mathematical analysis," *Genomics*, vol. 2, pp. 231–239, Apr. 1988.

[3] G. Bresler, M. Bresler, and D. Tse, "Optimal assembly for high throughput shotgun sequencing," *BMC Bioinformatics*, vol. 14, no. 5, pp. 1–13, 2013.

[4] J. Pell, A. Hintze, R. Canino-Koning, A. Howe, J. M. Tiedje, and C. T. Brown, "Scaling metagenome sequence assembly with probabilistic de Bruijn graphs," *Proceedings of the National Academy of Sciences*, vol. 109, no. 33, pp. 13272–13277, 2012.

[5] A. Broder and M. Mitzenmacher, "Network applications of bloom filters: A survey," *Internet mathematics*, vol. 1, no. 4, pp. 485–509, 2004.