Scaffolding Large Genomes using Integer Linear Programming

James Lindsay[1], Hamed Salooti[2], Alex Zelikovsky[2], Ion Măndoiu[1]

[1]Computer Science & Engineering Department, University of Connecticut
371 Fairfield Way, Storrs, CT 06269
{james.lindsay,ion}@engr.uconn.edu
[2]Computer Science, Georgia State University
34 Peachtree Street, Atlanta, GA 30303
alexz@cs.gsu.edu, hsalooti1@student.cs.gsu.edu

Abstract. The precipitous drop in sequencing costs has generated much enthusiasm for very large scale genome sequencing initiatives, such as the *Genome 10K* (G10K) project. Despite renewed interest the assembly of large genomes from short reads is still an extremely resource intensive process. This work presents a scalable algorithms to create scaffolds, or ordered and oriented sets of assembled contigs, which is one part of a practical assembly. This is accomplished using integer linear programming (ILP). In order to process large mammalian genomes we employ non-serial dynamic programming (NSDP) and a hierarchical strategy. Finally novel quantitative metrics are introduced in order to compare scaffolding tools and gain deeper insight into the challenges of scaffolding.

1 Introduction

Short contig lengths, typically between 2-4Kb, are characteristic of draft genome assemblies generated from low-coverage (2x) Sanger, (10x HTS) reads over the past decade. To increase the utility of such fragmented assemblies, additional long-range linkage information is used to orient contigs relative to one another and order them in larger structures referred to as scaffolds. Unfortunately linkage information provided by HTS pairs is noisy due to both chimeric pairs resulting from library preparation artifacts and erroneous mapping of reads originating from repeats. These difficulties, along with the sheer number of HTS pairs and contigs that must be handled, render scaffolding methods developed for Sanger pairs such as[5, 9] ineffective on HTS data. While recent algorithmic advances [1, 2, 10], and [7] have led to improve scaffolding accuracy from such noisy HTS paired reads, scaling these methods to datasets consisting of hundreds of thousands to millions of contigs and hundreds of millions of read pairs, as expected for a vertebrate genome, remains a significant challenge.

Our approach utilizes a pure ILP to find the scaffolds which are most consistent with the supplied linkage information. The presented model is equally or more accurate than the leading methods, and is able to solve large instances

where all others fail. This is achieved through the use of non-serial dynamic programming (NSDP) paradigm which divides the problem into smaller subproblems that are solved and combined into an optimal solution. When genomes are very large, and the contig quality is low a hierarchical approach is employed to achieve scalability. This approach utilizes the highest quality data first, and scaffolds found in earlier rounds serve as a filter for later rounds.

2 Scaffolding

Our approach breaks the scaffolding problem into two steps; first a compatible orientation and pairwise order is found using a robust Integer Linear Program (ILP), then a complete "golden path" or scaffold is computed using weighted bipartite matching.

2.1 ILP Model

The scaffolding ILP model is best described using the scaffolding graph $G = (V, E)$, where vertex represent contigs and edges are derived from linkage information provided by paired HTS reads. Paired sequencing libraries are constructed such that the order and orientation of the reads relative to their originating fragment is known. For this work we will use the mate-pair style of reads where the pairs come from the same strand and are in the same orientation. In a pair each read can map on the 5' or 3' strand of each contig, there are 4 possible orientations of two contigs i, j connected by a paired read. We define several boolean variable to assist in creating the model. First $S_i = \{0, 1\}$ $\forall i \in N$ is used to indicate the orientation of each contig, $S_i = 0$ indicates the contigs orientation does not change. Then $S_{ij} = \{0, 1\}$ $\forall (i, j) \in E$ tells if the contigs have the same orientation ($S_{ij} = 0$) or ($S_{ij} = 1$) if one is flipped. Four state variables, $A_{ij}, B_{ij}, C_{ij}, D_{ij} = \{0, 1\}$ $\forall (i, j) \in E$ represent the 4 possible order and orientations of adjacent contigs, they are mutually exclusive. Each state has an associated weight $A^w_{ij}, B^w_{ij}, C^w_{ij}, D^w_{ij}$ which are the sum of the weight of corresponding pairs. The objective is to maximize the number of concordant edges.

$$\text{Max} \sum_{(i,j)\in E} A^w_{ij} \cdot A_{ij} + B^w_{ij} \cdot B_{ij} + C^w_{ij} \cdot C_{ij} + D^w_{ij} \cdot D_{ij}$$

The following constraints enforce the behavior of the orientation variables.

$$\begin{aligned} S_{ij} &\leq S_i + S_j \quad A_{ij} + D_{ij} \leq 1 - S_{ij} \\ S_{ij} &\geq S_j - S_i \quad B_{ij} + C_{ij} \leq S_{ij} \\ S_{ij} &\leq 2 - S_i - S_j \\ S_{ij} &\geq S_i - S_j \end{aligned}$$

An additional 8 constraints forbid two and three cycles in all instances where 3 contigs are connected. The solution to this ILP gives the orientation of every

contig, which induces a partial ordering of the scaffolding graph. We can construct a directed scaffold graph $DG = (V, E)$ where the vertex set is the same as the previous graph and directed edges are induced by adjacent vertices with compatible orientations. This graph still does not indicate the linear path that is the scaffold.

2.2 Path Finding

A complete ordering of the directed scaffold graph will yield the desired scaffold. To do this we use maximum weighted bipartite matching. We define a bipartite graph $B = (V^1 \cup V^2, E)$ where each vertex in V^1 corresponds to the 5' end of a contig, and each vertex in V^2 to the 3' end of a contig. Edges are defined by directed scaffolding graph. The Hungarian algorithm can be used to solve this in $O(V^3)$. The edges are weighted by the same weight as the chosen state indicator.

3 Scaling the Algorithm

It is impractical to solve a scaffold ILP for a large mammalian genome, the number of variables and constraints is simply too large. To overcome this hurdle and solve the problem optimally and we adopted the non-serial dynamic programming (NSDP) paradigm. This optimization technique exploits the sparsity of the scaffolding graph, which should be a bounded-width graph [3], to compute the solution in stages, each stage using the results of a previous one to efficiently solve the problem.

3.1 NSDP

A key concept in NSDP is the notion of an interaction graph which models the relationship between variables and constraints. An interaction graph $I = (V, E)$ contains a vertex for each variable and an edge is added between vertices if they appear in the same constraint or component of the objective function. NSDP is a process which eliminates variables in such a way that adjacent variables can be merged together [11]. We noted that our scaffolding graph is equivalent to the interaction graph.

The first step in applying NSDP is identifying the independent and weakly independent components of interaction graph. The completely independent components have no influence on other components, however a weakly independent component may share one or two nodes with another component. We use efficient algorithms to find the bi and tri connected components of our interaction graph. Then an elimination order must be found so that each component can be solved independently in such a way that the solutions can be merged to find the global solution. This order takes the shape of a tree, for bi-connected components the tree is found using DFS from an arbitrary node. The decomposition order from tri-connected components is given by the SPQR-tree [4] data structure.

The solution to each component of the interaction graph is found using a bottom up traversal. In general during the traversal the ILP for each component is solved 2 or 4 times. Once for each possible orientation of the common nodes. The objective value of each case is encoded in the objective of the components parent. After solving all components, top down DFS starting from the same root is performed to apply the chosen solution for each component.

3.2 Hierarchical

It was observed that the number of paired edges that span contigs is a parameter, called p that has the biggest effect on accuracy and scalability of the scaffolding program. In our work we attempted to solve the scaffolding problem without setting this parameter explicitly but in complex genomes it was observed that occasionally the ILP would be too large even after the complete decomposition procedure. In order to address this, we developed a hierarchical strategy to solving the scaffolding problem. The problem is first solved with high confidence pairs by requiring $p > 1$, then p is gradually decreased. After each solution utilized pairs are removed from consideration, and any edge in the scaffolding graph that is not compatible with a previous stage scaffold is removed.

4 Results

The input to a scaffolding program is simply a set of contigs (veritcies) and some number of paired reads (directed edges). A scaffolded genome can be though as a linear directed graph and the act of scaffolding is the attempt to predict directed edges between adjacent contigs. We treat scaffolding as a binary classification test where methods attempt to predict true adjacencies in the test dataset. However the traditional measure of a scaffold is the N50, or weighted median statistic such that 50% of the entire assembly is contained in scaffolds equal to or larger than this value. Unfortunately this measure does not reflect the accuracy of the scaffolds and rewards aggressive merging. Therefore we utilize the true positive N50 (TP-N50) which breaks scaffolds when an adjacency was falsely predicted.

4.1 Experimental Setup

The test datasets were derived from a simulated denovo draft assembly of an individual human [6]. The set of Sanger style reads used to create this finished version of the draft genome were sub sampled to about 4x base coverage. A total of 11200000 reads were placed into the Celera 6.1 [8] assembler and assembled using the recommended parameters for large mammalian genomes. The assembler generated 422837 contigs with an N50 of 7704bp. These contigs were then mapped against the finished version of the genome using a gapped alignment tool, only non-overlapping contigs with 95% identity were utilized. From this alignment a reference scaffold was created to serve as the test case. Smaller test cases were generated by randomly choosing a percentage of scaffolds from each chromosome.

Accuracy The primary metric we use to demonstrate our algorithm is accurate is the TP-N50. We see that in all test cases our algorithm makes longer correct scaffolds than existing tools. In addition to this number we provide the two standard measures of accuracy in a binary classification test, sensitivity and positive predictive value. Using these measures we see that we are basically equivalent to OPERA.

Scalability The obvious measure of the scalability of an algorithm is runtime. We compared the runtime of all three tools on different size test cases. The objective of this test was to discover on which test case each method failed to produce a result given 120 hours. We chose 120 hours, or 5 days as our threshold because we believe it is sufficient amount of time for each tool. Longer times would represent an unreasonable amount of time to wait for an scaffold given that the assembly took approximately 10 days.

Acknowledgment

This work has been supported in part by awards IIS-0916948 and IIS-0916401 from NSF and Agriculture and Food Research Initiative Competitive Grant no. 201167016-30331 from the USDA National Institute of Food and Agriculture.

References

1. Adel Dayarian, Todd P. Michael, and Anirvan M. Sengupta. SOPRA: Scaffolding algorithm for paired reads via statistical optimization. *BMC Bioinformatics*, 11:345, 2010.
2. Song Gao, Niranjan Nagarajan, and Wing-Kin Sung. Opera: reconstructing optimal genomic scaffolds with high-throughput paired-end sequences. In *Proc. 15th Annual international conference on Research in computational molecular biology*, pages 437–451, Berlin, Heidelberg, 2011. Springer-Verlag.
3. Song Gao, Niranjan Nagarajan, and Wing-Kin Sung. Opera: reconstructing optimal genomic scaffolds with high-throughput paired-end sequences. In *Proc. 15th Annual international conference on Research in computational molecular biology*, pages 437–451, 2011.
4. J. E. Hopcroft and R. E. Tarjan. Dividing a graph into triconnected components. *SIAM Journal on Computing*, 2(3):135–158, 1973.
5. Daniel H. Huson, Knut Reinert, and Eugene W. Myers. The greedy path-merging algorithm for contig scaffolding. *J. ACM*, 49(5):603–615, 2002.
6. S. Levy *et al.* The diploid genome sequence of an individual human. *PLoS Biology*, 5(10):e254+, 2007.
7. J. Lindsay, J. Zhang, T. Farnham, Y. Wu, I. Mandoiu, R. O'Neill, H. Salooti, E. Bullwinkel, and A. Zelikovsky. Poster: Scaffolding draft genomes using paired sequencing data. In *Computational Advances in Bio and Medical Sciences (ICCABS), 2011 IEEE 1st International Conference on*, page 252, feb. 2011.
8. E. W. Myers, G. G. Sutton, A. L. Delcher, I. M. Dew, D. P. Fasulo, M. J. Flanigan, S. A. Kravitz, C. M. Mobarry, K. H. Reinert, K. A. Remington, E. L. Anson, R. A. Bolanos, H. H. Chou, C. M. Jordan, A. L. Halpern, S. Lonardi, E. M. Beasley, R. C. Brandon, L. Chen, P. J. Dunn, Z. Lai, Y. Liang, D. R. Nusskern, M. Zhan, Q. Zhang, X. Zheng, G. M. Rubin, M. D. Adams, and J. C. Venter. A whole-genome assembly of Drosophila. *Science (New York, N.Y.)*, 287(5461):2196–2204, March 2000.
9. Mihai Pop, Daniel S. Kosack, and Steven L. Salzberg. Hierarchical scaffolding with Bambus. *Genome research*, 14(1):149–159, 2004.
10. Leena Salmela, Veli Mäkinen, Niko Välimäki, Johannes Ylinen, and Esko Ukkonen. Fast scaffolding with small independent mixed integer programs. *Bioinformatics*, 27(23):3259–3265, December 2011.
11. Oleg Shcherbina. Nonserial dynamic programming and tree decomposition in discrete optimization. In *OR*, pages 155–160, 2006.