Poster: Accurate Scaffolding of Large Genomes using Integer Programming and Non-Serial Dynamic Programming

J. Lindsay, I. Măndoiu
Computer Science & Engineering Dept.
University of Connecticut, Storrs, CT
{*james.lindsay,ion*}@*engr.uconn.edu*

H. Salooti, A. Zelikovsky
Computer Science Dept.
Georgia State University, Atlanta, GA
{*hsalooti1,alexz*}@*cs.gsu.edu*

The precipitous drop in sequencing costs has generated much enthusiasm for very large scale genome sequencing initiatives. However, assembling high-quality genome sequences from the short reads currently generated by high-throughput sequencing (HTS) technologies represents a formidable computational challenge that has yet to be met. In this work we address the scaffolding step of genome assembly pipelines, whereby sets of assembled contigs are oriented, ordered, and combined into larger structures called *scaffolds*. Scaffolding is a critical step of practical genome assembly, as the larger structures significantly increase the usefulness of assembled genomes to biologists.

For conventional Sanger assemblies, long-range linkage information used in scaffolding is obtained by sequencing both ends of clones of up to hundreds of kilobases. HTS paired reads are typically generated with much shorter inserts. Furthermore, the linkage information is noiser due to HTS library preparation artifacts and erroneous mapping of short reads. These difficulties, along with the large number of HTS read pairs and contigs that must be handled, render scaffolding methods developed for Sanger pairs ineffective on HTS data. While recent algorithmic advances have led to improved scaffolding accuracy from HTS paired reads, scaling these methods to datasets consisting of up to millions of contigs and hundreds of millions of read pairs, as expected for a vertebrate genome, remains a significant challenge. To achieve accurate scaffolds efficiently, we decompose the scaffolding problem into smaller subproblems that can be solved exactly via Integer Linear Programing (ILP), and then optimally combine solutions to the subproblems using Non-Serial Dynamic Programming (NSDP).

The input to our scaffolding algorithm consists of contig sequences along with read pairs mapped onto contigs. We retain only pairs of uniquely mapping reads, and filter out pairs for which the inferred insert size lower bound is too large. The unfiltered paired read data can be conveniently represented as a multigraph $G = (V, E)$ with each vertex i representing a contig and each edge $e = (i, j)$ representing a paired read mapped onto contigs i and j, respectively. The relative orientation of the mapped reads in a pair can be consistent with the current orientation of the contigs, or consistent with switching the orientation of one or both of the contigs. We assign to each pair a weight based on the "uniqueness" of mapping regions, with pairs mapping to repetitive contig regions receiving lower weight.

To represent final contig orientations we use variables $S : V \rightarrow \{0, 1\}$, where a contig i has $S_i = 0$ if it retains its current orientation and $S_i = 1$ if its orientation is flipped. To express ordering constraints, for each pair of connected contigs (i, j) we introduce a variable S_{ij} which is 0 $S_i = S_j$ and 1, otherwise, as well as four state variables, $A_{ij}, B_{ij}, C_{ij}, D_{ij}$ representing the four mutually exclusive order and orientation configurations of two adjacent contigs (all four variables are set to 0 when contigs i and j are not adjacent). Each such state has an associated weight, A^w_{ij}, B^w_{ij}, C^w_{ij}, respectively D^w_{ij}, obtained by summing the weights of the read pairs consistent to the corresponding order and orientation state. The objective of the ILP is to maximizes the weight of concordant edges between adjacent contigs, $\sum_{(i,j)\in E} A^w_{ij} \cdot A_{ij} + B^w_{ij} \cdot B_{ij} + C^w_{ij} \cdot C_{ij} + D^w_{ij} \cdot D_{ij}$, subject to the constraint that each contig be immediately preceded and followed by at most one other contig (resulting cycles, if any, are resolved by removing the lowest weight edges in a postprocessing step).

Solving the above ILP directly for very large scaffolding multigraphs G is impractical. However, we take advantage of the sparsity of G to independently scaffold its tri-connected components, generated in linear time using the SPQR-tree datastructure, and optimally combine them using the NSDP paradigm. For cases when tri-connected components are still too large to handle directly we adopt a hierarchical scaffolding scheme that solves multiple ILPs for increasingly denser subgraphs of G obtained by filtering low confidence edges.

Acknowledgments

This work is 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.

978-1-4673-1321-6/12/$31.00 2012 IEEE