Global Optimization for Scaffolding and Completing Genome Assemblies

Sebastien François [1] Rumen Andonov [2,3]
Dominique Lavenier [4]

IRISA/INRIA, Rennes, France

Hristo Djidjev [5]

Los Alamos National Laboratory, Los Alamos, NM 87545, USA

Abstract

We develop a method for solving genome scaffolding as a problem of finding a long simple path in a graph defined by the contigs that satisfies additional constraints encoding the insert-size information. Then we solve the resulting mixed integer linear program to optimality using the Gurobi solver. We test our algorithm on several chloroplast genomes and show that it outperforms other widely-used assembly solvers by the accuracy of the results.

⋆ Preliminary version of this paper has been presented at the Workshop on Constraint-Based Methods for Bioinformatics 2016.

[1] Email: `sefra35@gmail.com`

[2] Email: `randonov@irisa.fr`

[3] `Corresponding author`

[4] Email: `lavenier@irisa.fr`

[5] Email: `djidjev@lanl.gov`

1 Modeling the scaffolding problem

1.1 Graph Modeling

We model the problem of scaffolding as path finding in a directed graph $G = (V, E)$ that we call a contig graph, where both vertices V and edges E are weighted. The set of vertices V is generated based on the set C of the contigs according the following rules: the contig i is represented by at least two vertices v_i and v_i' (forward/inverse orientation respectively). If the contig i is repeated k_i times, it generates $2k_i$ vertices. Denote $N = \sum_{i \in C} k_i$, therefore $|V| = 2N$.

The edges are generated following given patterns—a set of known overlaps/distances between the contigs. Any edge is given in the graph G in its forward/inverse orientation. We denote by e_{ij} the edge joining vertices v_i and v_j and the inverse of edge e_{ij} is $e_{j'i'}$. For any i, the weight w_i on a vertex v_i corresponds to the length of the contig i, while the weight l_{ij} on the edge e_{ij} corresponds to the value of the overlap/distance between contigs i and j. The problem then is to find a path in the graph G such that the total length (the sum over the traversed vertices and edges) is maximized, while a set of additional constraints are also satisfied:

- For any i, either vertex v_i or v_i' is visited (participates in the path).
- The orientations of the nodes does not contradict the constraints imposed by mate-pairs. This is at least partially enforced by the construction of G.

To any edge $e \in E$ we associate a variable x_e. Its value is set to 1, if the corresponding edge participates in the assembled genome sequence (the associated path in our case), otherwise its value is set to 0. There are two kinds of edges: edges corresponding to overlaps between contigs, denote them by O (from overlaps), and edges associated with mate-pairs relationships, denote them by L (from links). We therefore have $E = L \cup O$. Let l_e be the length of the edge $e = (u, v)$. We have $l_e < 0$ and $|l_e| < \min\{w(u), w(v)\}$, $\forall e \in O$, and $l_e > 0$ $\forall e \in L$. Let w_v be the length of the contig corresponding to vertex v and denote $W = \sum_{v \in V} w_v$.

Let $A^+(v) \subset E$ (resp. $A^-(v) \subset E$) denote the subset of arcs in E leaving (resp. entering) node v.

1.2 Mixed Integer Linear Programming Formulation

We associate a binary variable for any edge of the graph, i.e.

$$(1) \qquad \forall e \in O : x_e \in \{0, 1\} \text{ and } \forall e \in L : g_e \in \{0, 1\}.$$

Furthermore, to any vertex $v \in V$ we associate three variables, i_v, s_v, and

t_v, which stand respectively for intermediate, source, and target for some path, and satisfy

$$0 \leq i_v \leq 1, \quad 0 \leq s_v \leq 1, \; 0 \leq t_v \leq 1. \tag{2}$$

All three variables are set to zero when the associated vertex v participates in none of the paths. Otherwise, v can be either a source/initial (noted by $s_v = 1, t_v = 0, i_v = 0$), or a target/final ($t_v = 1, s_v = 0, i_v = 0$), or an intermediate vertex, in which case the equalities $i_v = 1, t_v = 0$ and $s_v = 0$ hold. Moreover, each vertex (or its inverse) can be visited at most once, i.e.

$$\forall (v, v') : i_v + i_{v'} + s_v + s_{v'} + t_v + t_{v'} \leq 1. \tag{3}$$

The four possibles states for a vertex v (to belong to none of the paths, or otherwise, to be a source, a target, or an intermediate vertex in some path) are provided by the following two constraints

$$s_v + i_v = \sum_{e \in A^+(v)} x_e, \quad t_v + i_v = \sum_{e \in A^-(v)} x_e. \tag{4}$$

Finally, only one sequence (a single path) is searched for

$$\sum_{v \in V} s_v = 1 \text{ and } \sum_{v \in V} t_v = 1. \tag{5}$$

Theorem 1.1 *The real variables $i_v, s_v, t_v, \forall v \in V$ take binary values.*

Proof. Given in [**?**]. □

We introduce a continuous variable $f_e \in R^+$ to express the quantity of the flow circulating along the arc $e \in E$

$$\forall e \in E : 0 \leq f_e \leq W x_e. \tag{6}$$

For $e \in O$, the value of x_e is set to 1, if the arc e carries some flow and 0, otherwise. In other words, no flow can use the arc e when $x_e = 0$.

We use the flows f_e in the following constraints, $\forall v \in V$,

$$\sum_{e \in A^-(v)} f_e - \sum_{e \in A^+(v)} f_e \geq i_v w_v + (i_v + t_v)(\sum_{e \in A^-(v)} l_e x_e) - W s_v, \quad W s_v \leq \sum_{e \in A^+(v)} f_e. \tag{7}$$

The purpose of the last two constraints is manifold. When a vertex v is a source ($s_v = 1$), (7) generates and outputs from it an initial flow of sufficiently big value (W is enough in our case). When v is an intermediate vertex ($i_v = 1$), constraint (7) forces the flow to decrease by at least $l_{(u,v)} + w_v$ units when it moves from vertex u to its adjacent vertex v. The value of the flow thus is decreasing and this feature forbids cycles in the context of (4). When v is a final vertex, (7) is simply a valid inequality for the input flow.

We furthermore observe that because of (4), the constraint (7) can be written as follows

$$\forall v \in V : \sum_{e \in A^-(v)} f_e - \sum_{e \in A^+(v)} f_e \geq i_v w_v + \sum_{e \in A^-(v)} l_e x_e - W s_v. \tag{8}$$

The constraint (8) is linear and we keep it in our model instead of (7).

Furthermore, binary variables g_e are associated with links. For $(s,t) \in L$, the value of $g_{(s,t)}$ is set to 1 only if both vertices s and t belong to the selected path and the length of the considered path between them is in the given interval $[\underline{L}_{(s,t)}, \overline{L}_{(s,t)}]$. Constraints related to links are :

$$g_{(s,t)} \leq s_s + i_s + t_s \text{ and } g_{(s,t)} \leq s_t + i_t + t_t \tag{9}$$

$$\forall (s,t) \in L : \sum_{e \in A^+(s)} f_e - \sum_{e \in A^-(t)} f_e + \sum_{e \in A^-(t)} l_e x_e \geq \underline{L}_{(s,t)} g_{(s,t)} - M(1 - g_{(s,t)}) \tag{10}$$

$$\forall (s,t) \in L : \sum_{e \in A^+(s)} f_e - \sum_{e \in A^-(t)} f_e + \sum_{e \in A^-(t)} l_e x_e \leq \overline{L}_{(s,t)} g_{(s,t)} + M(1 - g_{(s,t)}) \tag{11}$$

where M is some big constant.

We search for a long path in the graph and such that as much as possible mate-paired distances are satisfied. The objective hence is :

$$\max \left(W \sum_{e \in L} g_e + \sum_{e \in O} f_e \right). \tag{12}$$

We developed and tested algorithms for scaffolding and gap filling phases based on a version of the longest path problem and MILP representation. Our algorithms significantly outperform three of the best known scaffolding algorithms with respect to the quality of the scaffolds. Regardless of that, we consider the current results as a work in progress. The biggest challenge is to extend the method to much bigger genomes. We plan to use some additional ideas and careful implementation to increase the scalability without sacrificing the accuracy of the results.