Plasmid MILP flow formulation

Aniket Mane

April 2022

PlasBin is an MILP-based hybrid approach for plasmid binning. Here, we look at the problem of obtaining plasmid bins from the assembly graph as a flow problem and design an MILP formulation accordingly.

Input:

- Contigs,
- Edges,
- For each contig, the following information is known
  - Probability of the contig origin being plasmidic (explained later),
  - GC content probability (explained later),
  - Read depth of contig,
  - Length of contig.

Output:

- Ordered sets of contigs defining a walk in the assembly graph

Idea:
An ideal plasmid bin will consist of a set of contigs that have similar read depth, similar GC content and high probability of originating from a plasmid. In this formulation, the plasmid bin will be defined by the flow that optimizes an objective function based on the above-mentioned criteria.

Consider an assembly graph G. Every contig u in G has two extremeties, a head and a tail, denoted by u_h and u_t for a contig u. The edges of the assembly graph are pairs of contig extremeties. For instance, an edge between u_h and v_h is represented as (u_h, v_h). We introduce a source vertex S and a sink vertex T. We add edges from S and T to all the extremities. We associate a binary variable each with every vertex, extremity and edge in the assembly graph to denote if they are part of the solution. Thus, for a vertex v, $x_v = 1$ if v belongs to the solution and 0 otherwise. Similarly, for an extremity v_y, $x_{v_y} = 1$ if v

belongs to the solution and 0 otherwise. For an edge e, $x_e = 1$ if e belongs to the solution and 0 otherwise.

Flow: For each edge $e = (u_x, v_y)$, where $x, y \in \{h, t\}$ we define the capacity c_e of the edge as the minimum of the read depths rd of u and v. Thus, $c_e = min(rd_u, rd_v)$. For an edge S, u_x or u_x, T, $c_e = rd_u$. The flow through an edge e is denoted by f_e. The total value F of the flow will be the total flow out of S. For now, we limit ourselves to the case where exactly one edge out of S (and into T) has a positive flow across it. All other edges out of S have zero flow. We wish to maximize the flow from S to T. Thus, $F = \sum_{e\ out\ of\ S} f_e$.

GC content: We discretize the range of GC content of contig or overall plasmid bin using blocks. These blocks can either be defined using a plasmid database or through other means (yet to be discussed). For each contig v, we estimate the probability $GC_{v,i}$ that the contig belongs to a sequence that belongs to each block i. In other words, we determine how likely is it that the contig v is part of a larger sequence with GC content in block i. If we use n blocks to split the range of GC content, then for each contig v, $\sum_{i=1}^{n} GC_{v,i} = 1$. Note that the values $GC_{v,i}$ will be obtained via simulation $\forall v, i$.

We use a set of binary variables GC_i to denote if the GC content of the plasmid falls in block i. We wish to to output a solution with relatively uniform GC content. Hence, if $GC_k = 1$ for some block k, we penalize the total probability that the GC content of a chosen contig v not being in block k.

Probability of origin: For each contig, we obtain the probability that it belongs to a plasmid. To do so, we use plasmid identification tools such as mlplasmids that provide this information. For each contig v, we have the probability p_v that it belongs to a plasmid. In order to discourage the MILP from choosing contigs that are probably chromosomal in origin, we add the term $ln\frac{p_v}{1-p_v}$ for each vertex v chosen in the solution.

Objective function: Thus, the objective function for the formulation is as follows:
Maximize:

$$F - \sum_{v,i}((1 - GC_{v,i}) * x_v * GC_i) + \sum_{v}(x_v * ln\frac{p_v}{1-p_v})$$

Constraints:

1. Product of binary variables:
   x_v and GC_i are both binary variables. We define $xGC_{v,i}$ as their product. Thus, $xGC_{v,i} = 1$ only when both x_v and GC_i are 1. To ensure this, we add the following constraints,

$$xGC_{v,i} \leq x_v$$

$$xGC_{v,i} \leq GC_i$$
$$xGC_{v,i} \geq x_v + GC_i - 1$$

2. Flow conservation constraints:
   For a contig v, the incoming and outgoing flow should be equal, considering edges that are part of the solution. Thus, for all contigs, we add constraints as follows,
$$\sum_{e\ into\ v} f_e * x_e = \sum_{e\ out\ of\ v} f_e * x_e$$

3. Capacity constraints:
   The maximum flow through an edge has to be at most the capacity of the edge (i.e. the minimum read depth of the contigs involved in the edge).
$$f_e \leq c_e$$
   Similarly, the maximum flow into a vertex should be at most the capacity (read depth) of the vertex itself.
$$\sum_{e\ into\ v} f_e * x_e \leq rd_v$$

4. Value of the flow F:
   The value of the flow F should be equal to the total flow out of the source vertex.
$$F = \sum_{e\ out\ of\ S} f_e * x_e$$
   If we are looking for exactly one walk from S to T,
$$\sum_{e\ out\ of\ S} x_e = 1$$

5. Uniformity of read depth:
   The overall plasmid read depth should not exceed that of an individual contig. Thus, we want the flow value F to be at most the minimum flow through an edge in the chosen walk. Thus, if $x_e = 1$ for an edge e, we want $F \leq f_e$.
$$F * x_e \leq f_e$$

6. Product between a continuous and a binary variable:
   We see that several constraints mentioned above require a product between a continuous variable (either f_e or F) and a binary variable x_e. To obtain equivalent linear constraints, we use an upper bound on the continuous variables, U and variable $xCVAR$ for the product $x_e * CVAR$. We then add the constraints,
$$xCVAR \leq U * x_e$$

$$xCVAR \leq CVAR$$

$$xCVAR \geq CVAR - (1 - x_e) * U$$

$$xCVAR \geq 0$$

7. Constraints to define the contigs and edges of the walk:
   An edge is a part of the solution only if both its endpoints are part of the solution. For an edge $e = (u_x, v_y)$

$$x_e \leq x_{u_x}$$

$$x_e \leq x_{v_y}$$

   A contig is part of the solution if at least one of its extremities is part of the solution.

$$x_u \geq x_{u_h}$$

$$x_u \geq x_{u_t}$$

8. GC content of plasmid bin:
   GC content of the solution can be in exactly one block.

$$\sum_i GC_i = 1$$