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,
  - Nucleotide sequence,
  - 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, $contigs[v] = 1$

if v belongs to the solution and 0 otherwise. Similarly, for an extremity v_y, $ext[(v, y)] = 1$ if v belongs to the solution and 0 otherwise. For an edge e, $links[e] = 1$ 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 $cap[e]$ of the edge as the minimum of the read depths rd of u and v. Thus, $cap[e] = min(rd_u, rd_v)$. For an edge (S, u_x) (resp. (v_y, T)), $cap[e] = rd_u$ (resp. $cap[e] = rd_v$). The flow through an edge e is denoted by $flows[e]$. 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, $F = \sum_{e\ out\ of\ S} flows[e]$.

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 c, we estimate the probability $GC_{c,b}$ that the contig belongs to a sequence that belongs to each block b. In other words, we determine how likely is it that the contig c is part of a larger sequence with GC content in block b. If we use n blocks to split the range of GC content, then for each contig c, $\sum_{b=1}^{n} GC_{c,b} = 1$. Note that the values $GC_{c,b}$ will be obtained via simulation $\forall c, b$.

We use a set of binary variables $plas\_GC[b]$ to denote if the GC content of the plasmid falls in block b. We wish to to output a solution with relatively uniform GC content. Hence, if $plas\_GC[k] = 1$ for some block k, we penalize the total probability that the GC content of a chosen contig c 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. Note that, depending on the identification tool, we may not always have the chromosomal probability as $1 - p_v$. In that case, we simply change the term to $ln\frac{pl_v}{chr_v}$ where pl_v and chr_v denote the probabilities of the contig origin being plasmid and chromosome respectively.

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

$$F - \sum_{c,b}((1 - GC_{c,b}) * contigs[c] * plas\_GC[b]) + \sum_{c}(contig[c] * ln\frac{pl_c}{chr_c})$$

Here, $GC_{c,b}$ is the given probability the contig c belongs to a sequence with GC content in bin b.

Contig attributes: We store the following attributes in a dictionary *contigs_dict*. For each contig c,

- Length - Length of the contig (int)
- Read_depth - Overall read depth of the contig (float)
- GC_cont - GC content of the contig (float)
- Seed - Indication if the contig is a seed (binary)
- Sequence - Sequence of a contig (string)
- PrPl - Probability that a contig is of plasmidic origin (float)
- PrChr Probability that a contig is of chromosomal origin (float)
- log_ratio - Log ratio of the above probabilities, $ln\frac{pl_c}{chr_c}$ (float)

Constraints:

1. A link e is in the plasmid only if both its endpoints are in the plasmid. Say $e = (u_y, v_z)$ where $y, z \in \{h, t\}$. Then:

$$links[e] \leq ext[u_y]$$
$$links[e] \leq ext[v_z]$$

   Additionally, we also add these constraints for all links (S, u_y) and (u_y, T).

2. An extremity is in the plasmid only if at least one link is incident on it.

$$ext[u_y] == min(1, \sum_{e\ inc.\ on\ u_y} links[e])$$

   Note: Constraint needs modification

3. A contig c is in the plasmid if at least one of its endpoints is in the plasmid.

$$contigs[c] \geq ext[c_h]$$
$$contigs[c] \geq ext[c_t]$$

   Both extremities must feature equal number of times in the solution.

$$ext[c_h] == ext[c_t]$$

4. A contig is considered to be a "counted" seed if it is eligible to be a seed contig (from input) and is considered to be part of the solution (by solving the MILP).

$$counted\_seed[c] == contigs\_dict[c]['Seed'] * contigs[c]$$

Every solution must contain a seed contig.

$$\sum_{c} counted\_seed[c] >= 1$$

5. F should equal the flow out of S and into T. Exactly one edge exits S and exactly one enters T.

$$\sum_{e=(S,u_y)} flows[e] == F$$
$$\sum_{e=(u_y,T)} flows[e] == F$$
$$\sum_{e=(S,u_y)} links[e] == 1$$
$$\sum_{e=(u_y,T)} links[e] == 1$$

6. Flow conservation constraints:
Flow into u_h (resp. u_t) should be equal to flow out of u_t (resp. u_h).

$$\sum_{e=v_y,u_h} flows[e] == \sum_{e=u_t,w_z} flows[e]$$
$$\sum_{e=v_y,u_t} flows[e] == \sum_{e=u_h,w_z} flows[e]$$

Number of links into u_h (resp. u_t) should be equal to those of u_t (resp. u_h).

$$\sum_{e=v_y,u_h} links[e] == \sum_{e=u_t,w_z} links[e]$$
$$\sum_{e=v_y,u_t} links[e] == \sum_{e=u_h,w_z} links[e]$$

7. Capacity constraints:
The maximum flow into a vertex u should be at most the capacity (read depth) of the vertex itself. Here, we represents the read depth of a contig u by rd_u for ease of representation. Thus, $rd_u = contigs\_dict[u]['Read\_depth']$.

$$\sum_{e=v_y,u_h} flows[e] + \sum_{e=v_y,u_t} flows[e] \leq rd_u$$

Similarly, the maximum flow through an edge has to be at most the capacity ($cap[e]$) of the edge (i.e. the minimum read depth of the contigs involved in the edge). For an edge $e = (u_y, v_z)$, $cap[e] = min(rd_u, rd_v)$.

$$flows[e] \leq cap[e]$$

8. The overall flow F through link e is "counted" only if e is part of the solution.

$$counted\_F[e] <= F * links[e]$$

However, as F and $links[e]$ are both variables, we add the following constraints:

$$counted\_F[e] <= UBD * links[e]$$
$$counted\_F[e] <= F$$
$$counted\_F[e] >= F - (1 - links[e]) * UBD$$
$$counted\_F[e] >= 0$$

Here, UBD is an upper bound on the flow. We can use $UBD = max_u(rd_u)$. Thus, if $links[e] = 1$ then $counted\_F[e] = F$, else it is equal to 0.

9. The overall plasmid read depth should not exceed that of an individual contig. Thus, the overall flow cannot exceed the flow through any active link e.

$$counted\_F[e] <= flows[e]$$

This constraint forces all flows to be 0. Thereby satisfying all flow constraints. Hence, it needs to be modified.

10. Constraint to handle product of variables $contigs[c]$ and $plas\_GC[b]$ in the GC-content term in the objective function:
If a putative plasmid is assigned a GC content bin b, the penalties for the GC content for a particular contig c will be considered only if the contig is part of the putative plasmid. Furthermore, this penalty will be exactly equal to the sum of probabilities that the GC content of the contig does *not* fall under bin b.

$$contig\_GC[c][b] \leq contig[c]$$
$$contig\_GC[c][b] \leq plas\_GC[b]$$
$$contig\_GC[c][b] \geq contigs[c] + plas\_GC[b] - 1$$

The GC content of the solution can be in exactly one bin.

$$\sum_b plas\_GC[b] = 1$$

11. We wish to extract a path from the graph. Considering the two dummy vertices S and T, we need one more link than the number of contigs.

$$\sum_e links[e] == \sum_c contigs[c] + 1$$

Constraint or associated variables need modification to account for repeated contigs.