List of assumptions to deal with.

- There is a single set of reactions that we can use to convert substrates to products.
- Each reaction has one substrate and one product.
- All species can uptake and secrete all metabolites.

1 Integer Linear Programming

- We need a framework that allows us to combine our solutions to the two components of the problem (species set minimization and ensuring a sufficient reaction set).
- Linear programming provides a convenient framework for this.
- Consider the following function of variables x, y, and z:

$$f(x,y,z) = 2x - y + 0.5z.$$

- **Now what if we want to know what the maximum value of f is?**
- Clearly, the maximum value is ∞ since x, y, and z can be equal to anything.
- **What happens if we introduce constraints on the values of x, y, and z?**
- For example, let's specify the following constraints:

$$\begin{aligned} x + y &\leq 10 \\ z - x &= 2 \\ y &\geq 4. \end{aligned}$$

- **What is the maximum value of f now?**
- This set of constraints makes the problem fairly easy, and we find that f is maximized given the following conditions:

$$\begin{aligned} x &= 6 \\ y &= 4 \\ z &= 8 \\ f &= 12. \end{aligned}$$

- We will specifically be using a version of linear programming called Integer Linear Programming (ILP), which always contains the additional constraint that all variables must be integer-valued.

- As it turns out, ILP is NP-hard (and the decision version is NP-complete).
- This will become evident since we are going to reduce the set cover problem to an ILP problem and we know set cover is NP-hard (and the decision version is NP-complete).
- Side note: decision versions of optimization problems instead ask "Is there a solution with value at most/at least X".

2 Set Cover

- The first component of the problem asks us to minimize the set of species that together catalyze a particular set of metabolic reactions.
- Let us define the set of species:

$$S = \{s_1, s_2, ..., s_n\}.$$

- We will also define the set of metabolic reactions that can be catalyzed as:

$$R = \{r_1, r_2, ..., r_p\}.$$

- Given that we care about which species can catalyze which metabolic reactions, each species in this problem will be defined as the set of metabolic reactions that it can catalyze:

$$s_i = \{r_{i\_1}, r_{i\_2}, ..., r_{i\_a}\}.$$

- Thus, given a set of reactions, we want to minimize the set of species that collectively contain those reactions.
- Side note: this is equivalent to the Set Cover (SC) problem, which is defined as finding the minimum number of given subsets whose union contains all elements in a specified set.
- Side note continued: the decision version of set cover is to ask whether there is a collection of subsets of size X or less whose union contains all elements in a specified set.
- **So in our community design problem, what is the specified set of elements?** The reactions.
- **What are the available subsets we have to choose from?** The species.
- **What is the final collection of subsets?** The community.
- **Now given all of this, what do we need to encode the problem in ILP terms?** We need to convert these terms to integer-valued variables.

- Let's start with the variables we'll be using in the function we want to minimize.
- **What should the species-associated variables be?** They should be binary indicator variables that are 1 if the species is used in the community and 0 otherwise:

$$I\_s_i \leq 1,$$

$$I\_s_i \geq 0.$$

- **So what is the function we are minimizing then?** The sum of the species indicator variables:

$$\min \sum_{i=1}^{n} I\_s_i.$$

- Now let's define the set of reactions we want to cover. **How should we do this with ILP variables?** Again, use binary indicator variables, 1 for if we want the reaction and 0 if we don't care:

$$I\_r_j \leq 1,$$

$$I\_r_j \geq 0.$$

- **Now how do we specify using constraints that the minimized set of species should contain all the reactions we want? What about for a single reaction?** For each reaction, have the sum of all species variables that contain that reaction be greater than or equal to the reaction's indicator variable:

$$\sum_{\forall i \text{ s.t.} r_j \in s_i} (I\_s_i) \geq I\_r_j,$$

- For example, consider the set of species and reactions shown in the example. **What would the constraint look like for reaction r_6?**

$$I\_s_2 + I\_s_3 \geq I\_r_j.$$

- Together, these constraints and the minimized function solve our first problem, minimizing the set of species that provide a specified set of metabolic reactions.

3 Network Flow

- The set cover solution works provided we know what set of metabolic reactions we want the community to catalyze, but now let's try to remove this assumption.

- However there are many possible paths from a particular substrate metabolite to a particular product metabolite.

- Thus, our goal is to provide ILP constraints such that the set of reaction indicator variables will equal 1 when they provide a path from the substrate metabolite to the product metabolite.

- Note that since we define this as a set of constraints, then we are not requiring a specific path through the metabolic network, just some viable path.

- **What do we need to relate reactions to each other and the sets of available substrate and desired product metabolites?** We need to define the sets of metabolites.

$$M = \{m_1, m_2, ..., m_q\},$$

$$SUBSTRATES = \{m_{substrate_1}, m_{substrate_2}, ..., m_{substrate_b}\},$$

$$PRODUCTS = \{m_{product_1}, m_{product_2}, ..., m_{product_c}\}.$$

- **Given these metabolite variables, how do we define reactions (assuming each reaction has one substrate and one product)?** We define each reaction as the ordered pair of substrate and product metabolite, which defines a graph with metabolites as vertices and reactions as edges.

$$r_j = (m_{j\_substrate}, m_{j\_product}).$$

- At this point, we are going to introduce the idea of flow in the graph (this allows us to avoid problems with cycles in the graph).

- We imagine that the graph is somewhat akin to a series of directional pipes. Our available substrate metabolites are where we can put water into the graph, and our product metabolites are where we need water to come out. In this analogy, a sufficient set of reactions we want will allow water to reach each product metabolite.

- In particular, we are going to assign a flow variable F_r_j to each reaction that indicates how much water is passing through that pipe/edge.

- **How do we allow our substrate metabolites to be the starting points of any paths to the product metabolites (be the source of any flow)? In particular, what do we need to say about the relationship between incoming flow and outgoing flow?** The amount of flow leaving a substrate metabolite's vertex can be greater than the amount of flow entering the substrate metabolite's vertex.

$$\sum_{\forall j \text{ s.t.} m_j \in SUBSTRATES} (- \sum_{\forall in \text{ s.t.} r_{in}=(m_i,m_j)} F\_r_{in} + \sum_{\forall out \text{ s.t.} r_{out}=(m_j,m_k)} F\_r_{out}) = |PRODUCTS|.$$

- **Now how do we ensure that all product metabolites are the end of some path starting from substrate metabolites (are receiving flow)? Again, what should be true of incoming versus outgoing flow?** The amount of flow entering a product metabolite's vertex must be greater than the amount of flow leaving the product metabolite's vertex.

$$\sum_{\forall in \text{ s.t.} r_{in}=(m_i,m_j)} F\_r_{in} - \sum_{\forall out \text{ s.t.} r_{out}=(m_j,m_k)} F\_r_{out} = 1,$$

$$\forall j \text{ s.t.} m_j \in PRODUCTS.$$

- **And finally, how do we ensure that no non-substrate (intermediate) metabolites are the starts of any paths (no intermediate metabolite produces flow)?** The flow into an intermediate metabolite must be equal to the flow out of an intermediate metabolite.

$$\sum_{\forall in \text{ s.t.} r_{in}=(m_i,m_j)} F\_r_{in} = \sum_{\forall out \text{ s.t.} r_{out}=(m_j,m_k)} F\_r_{out},$$

$$\forall j \text{ s.t.} m_j \notin SUBSTRATES, m_j \notin PRODUCTS.$$

- Together, these constraints require that flow comes from substrate metabolites, passes through intermediate metabolites without increasing, and ends up at all product metabolites, thus encompassing all paths that we want to consider.

- **These are the flow variables, but how do we convert these to determining whether the reaction indicator variables are 1's or 0's? How do we state that a non-zero flow ensures a non-zero reaction indicator variable?** Make the flow variable be less than the reaction variable multiplied by a large constant.

$$F\_r_j \leq \text{MAXINT} \times I\_r_j.$$

- Those constraints satisfy the requirement that we can use any possible path from substrates to products. **Now we have removed the assumption that we have a pre-specified set of reactions to provide.**

4 Multiple-substrate multiple-product reactions

- Remember that we have been assuming that each reaction has a single substrate and a single product. However, in reality reactions often have multiple substrates and/or cofactors and generate more than one product.

- The new definition of a reaction is then:

$$r_j = (\{m_{j\_substrate_1}, m_{j\_substrate_2}, ..., m_{j\_substrate_d}\},$$

$$\{m_{j\_product_1}, m_{j\_product_2}, ..., m_{j\_product_e}\}).$$

- **How might we handle the scenario with multiple-substrate multiple-product reactions? What about the scenario where a reaction has multiple products? Is there an easy way to deal with that?** Since we don't care about unnecessary byproducts in this problem, we can just split up each reaction into a separate reaction for each of its products.
- **What about when a reaction has multiple substrates? How can we encode this in our graph? Could we replace the reaction edges with something else that can have multiple inputs in our framework?** We can turn each reaction into a vertex in the network, with edges only from metabolites to reactions or reactions to metabolites.
- Now there are two types of edges in the network, input edges and output edges:

$$I = i_1, i_2, ..., i_t,$$
$$O = \{o_1, o_2, ..., o_u\},$$
$$i_j = (m_{j\_input}, r_{j\_reaction}),$$
$$o_j = (r_{j\_reaction}, m_{j\_output}).$$

- We also redefine flow to be over input and output edges.
- **How do our flow rules need to change? Do the substrate metabolite vertex flow rules change? How about for product metabolite vertices? What about intermediate metabolite vertices?** We actually only need to introduce one new flow rule for reaction vertices. Specifically, each input edge must provide flow equal to the flow along the output edge:

$$F\_i_j = F\_o_j.$$

- **Why don't we take the sum of the input edges?** Because that would allow cycles to incorrectly contribute flow.
- We also slightly redefine how we force reactions with flow to have the correct indicator variable.
- **What must be true of each input edge for a reaction to be used?** Each input edge to that reaction must have flow:

$$F\_i_j \geq I\_r_k : \forall j, k \text{ s.t.} i_j = (m_l, r_k).$$

- **And how do we ensure that if a reaction is used, the reaction indicator variable is 1?** We use the same trick as before, the outgoing flow must be less than the indicator variable times some large number:

$$F\_o_j \leq \text{MAXINT} \times I\_r_k : \forall j, k \text{s.t.} o_j = (r_k, m_l).$$

- These modifications are sufficient to encode multiple-substrate multiple-product reactions in the ILP formulation. **Now we have removed the constraint that reactions have a single substrate and a single product.**

5 Transport between species

- Up to now, we have assumed that metabolites can be freely transferred between different microbes. However, it is often the case that a microbe will lack certain transporters, making it impossible for it to take up certain metabolites. Secretion is also an issue, though a common assumption made is that there is a high enough turnover of individual cells that metabolite release on death is equivalent to all microbes being able to secrete all metabolites.

- **How might we compartmentalize metabolites in this algorithm? That is, distinguish which microbe has access to which metabolites?** The simple solution to remove this assumption is to first replace the set of general metabolites from before with a set of species-specific metabolites (the first index in the subscript is the species, the second is the type of metabolite):

$$M = \{m_{1,1}, m_{1,2}, ..., m_{1,q}, m_{2,1}, ..., m_{n,q}\}.$$

- **How do we represent environmental metabolites?** We can treat the environment like another species (species 0):

$$M = \{m_{0,1}, m_{0,2}, ..., m_{0,q}, m_{1,1}, ..., m_{n,q}\}.$$

- **Now what do transport reactions correspond to?** Reactions that convert environmental metabolites to species-specific metabolites and vice-versa:

$$r_{i,transport\_l} = (m_{0,k}, m_{i,k}),$$
$$r_{i,transport\_l} = (m_{i,k}, m_{0,k}).$$

- Thus, given specified transport reactions for each species, we can compartmentalize the reactions to each species. **Now we have removed the assumption that metabolites can transfer freely between species.**

6 Forcing substrate usage and species costs

- As a final remark, there are a couple of other use cases that we can address with simple modifications. First, it is possible that you might want to ensure that reactions exist to ensure that certain substrate metabolites are used in generating the product metabolites. For instance, if you want to degrade certain pollutants or metabolize certain toxins.

- **What simple constraint would you add to make sure a particular substrate is used in some path to a product metabolite?** Just force the outgoing flow for that particular substrate to be positive.

- Another possible use case is that certain species may be more or less desirable to use. For instance, maybe certain species are more difficult to obtain and/or culture in the lab for easy community construction.
- **How can we modify how we treat particular species when determining a minimal community?** We can add coefficients to the function we're minimizing to weight each species differently.