Bioinformatics, 32(13), 2016, 2008–2016
doi: 10.1093/bioinformatics/btw107
Advance Access Publication Date: 26 February 2016
Original Paper

Systems biology

An algorithm for designing minimal microbial communities with desired metabolic capacities

Alexander Eng[1] and Elhanan Borenstein[1,2,3]*

[1]Department of Genome Sciences, [2]Department of Computer Science and Engineering, University of Washington, Seattle, WA, USA and [3]Santa Fe Institute, Santa Fe, NM, USA

*To whom correspondence should be addressed.

Associate Editor: Janet Kelso

Received on September 29, 2015; revised on January 29, 2016; accepted on February 19, 2016

Abstract

Motivation: Recent efforts to manipulate various microbial communities, such as fecal microbiota transplant and bioreactor systems' optimization, suggest a promising route for microbial community engineering with numerous medical, environmental and industrial applications. However, such applications are currently restricted in scale and often rely on mimicking or enhancing natural communities, calling for the development of tools for designing synthetic communities with specific, tailored, desired metabolic capacities.

Results: Here, we present a first step toward this goal, introducing a novel algorithm for identifying minimal sets of microbial species that collectively provide the enzymatic capacity required to synthesize a set of desired target product metabolites from a predefined set of available substrates. Our method integrates a graph theoretic representation of network flow with the set cover problem in an integer linear programming (ILP) framework to simultaneously identify possible metabolic paths from substrates to products while minimizing the number of species required to catalyze these metabolic reactions. We apply our algorithm to successfully identify minimal communities both in a set of simple toy problems and in more complex, realistic settings, and to investigate metabolic capacities in the gut microbiome. Our framework adds to the growing toolset for supporting informed microbial community engineering and for ultimately realizing the full potential of such engineering efforts.

Availability and implementation: The algorithm source code, compilation, usage instructions and examples are available under a non-commercial research use only license at https://github.com/borenstein-lab/CoMiDA.

Contact: elbo@uw.edu

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Complex microbial communities can be found everywhere on our planet, spanning marine communities inhabiting the deep ocean to symbiotic communities living on and within host organisms. These communities impact a broad set of processes ranging from environmental resource cycles to host organism health. For example, deep sea rock and vent communities play a fundamental role in oxidizing environmental methane (Marlow *et al.*, 2014), whereas the human gut microbiome crucially aids in drug metabolism, energy harvest and immune system response (Sekirov *et al.*, 2010). Microbial communities affect these processes through a variety of metabolic reactions catalyzed by enzymes encoded in the member species' genomes, and ultimately through the diverse compounds each community can degrade or produce.

These critical roles microbial communities play in shaping their environment, combined with the potential to manipulate these communities, suggest a promising route for numerous medical and environmental applications (Brenner *et al.*, 2008). Specifically, several such efforts to shift target communities toward preferred states have used samples from naturally occurring communities as an inoculation source. For instance, transplanting healthy donor microbiome samples into patient guts has recently been used to treat a variety of gut disorders (Aroniadis and Brandt, 2013). Such fecal microbiota transplants (FMTs) have been shown to perturb a patient's dysbiotic gut community, shifting it to a healthier state and ameliorating their condition (Hamilton *et al.*, 2013; Song *et al.*, 2013). These FMT-based therapies have had a >90% success rate at curing recurrent *Clostridium difficile* infections and have promising results for addressing other gut disorders including inflammatory bowel disease and metabolic syndrome (Rossen *et al.*, 2015). Similarly, wastewater treatment bioreactors are often seeded by microbial communities cultivated from naturally occurring wastewater microbes or from previously established bioreactors (Alleman and Prakasam, 1983). These seed communities colonize the new bioreactor and thereby provide the metabolic processes necessary to degrade biological matter in wastewater.

Following the success of such transplants, recent efforts have further aimed to use engineered, rather than naturally occurring, communities in an attempt to increase control over transplant outcomes. For example, a synthetic stool substitute was recently developed using a mixture of cultured bacterial isolates to mimic a healthy gut community (Petrof *et al.*, 2013). Such a synthetic community removes the need for sample donors, allows greater regulation over the bacteria present in the transplant community, and reduces the risk for inadvertent transfer of pathogens. This synthetic and markedly simpler community was shown to still be effective in treating *C. difficile* infections. Another effort applied a simple selection-based approach to optimize the species composition of a bioreactor seed community for increased biopolymer production from glycerol (Moralejo-Gárate *et al.*, 2011). The final community's biopolymer production rate was demonstrated to be noticeably increased compared to the original community.

Such community engineering approaches are clearly an important first step towards customizing microbial community composition, yet they still largely rely on imitating natural community structures or enhancing existing community capabilities and cannot, for example, produce communities with potentially desired abilities absent from the initial community. Indeed, even optimizing an existing community function involves developing a carefully controlled selection procedure tailored to the preferred function and may require a long time for the community to reach an optimal state. The applications of such engineering efforts are therefore inherently constrained and are often very system-specific and hard to generalize.

One approach to address these challenges is to rationally design and construct synthetic communities with desired and predefined metabolic capabilities. Such a design process would involve the careful selection of member species and their abundances, hopefully defining a community composition that would achieve the desired metabolic task in the target environment. The ability to design such communities would significantly broaden the applicability of community engineering, could alleviate the reliance on naturally occurring community functions, and would ultimately support the construction of communities tailored to perform specific tasks within the context of various environmental settings.

Designing microbial communities, however, is a daunting task. Microbial species are endowed with tremendously diverse and complex capacities, which may not be trivial or easy to discern. Moreover, the various species comprising each community do not function independently, and each community impacts its environment through the orchestrated activity of its members. Interaction between species can lead to emergent behaviors that cannot be attributed to the function of just a single species or to additive species functions (Pelz *et al.*, 1999; Pettit, 2009). One species can, for example, provide the necessary precursors that allow a second species to produce metabolites that it could not produce when growing in isolation (Chiu *et al.*, 2014). Similarly, costly metabolic tasks could be distributed among community members such that each member performs a specific part of a complex metabolic pathway (Moran, 2007). A successful design framework should therefore account for such interactions and their impact on the metabolism of the community as a whole.

As a first step to address this challenge, here we develop CoMiDA (Community Metabolism Design Algorithm), an algorithmic framework for designing simple communities with some predefined metabolic capacities. Specifically, we aim to identify a set of species that, as a community, have the metabolic potential to convert a set of metabolic substrates to a set of desired target metabolites. We further aim to discern the smallest set of species required to provide this desired metabolic potential, reducing downstream complexities and providing more streamlined communities. In other words, our goal is to identify a minimal set of species whose genomes collectively encode a set of enzymatic genes that can catalyze a collection of metabolic reactions forming metabolic paths to each desired product metabolite from the available substrates.

Communities designed with our framework will therefore have the required metabolic potential to achieve the specified metabolism. Obviously, there are additional factors and processes that should be ultimately considered in designing a stable and functional community that carries out a specific task. First and foremost, possessing the set of reactions leading from substrates to products does not necessarily imply that the community would actively and efficiently perform the desired metabolic function. Toxin production, signaling between microbes, the capacity to transport metabolites between cells, and the ability of the selected species to survive in the target environment could further affect the community behavior, stability and dynamics. Yet, having the metabolic potential to carry out the desired function is an important and essential *prerequisite* for any community that could achieve the specified task, and is therefore a natural first step in rational community design and a critical component of any design task (see also Section 4).

2 Methods

2.1 Problem statement and approach

The goal of our design task is, given a set of substrate metabolites and a set of target product metabolites, to find a minimal subset of the available species that can collectively synthesize this set of target products using the available substrates. Specifically, here we view each microbial species as a simple assemblage of metabolic reactions, corresponding to the set of enzymatic genes encoded in its genome. Each reaction is represented as a hyperedge, linking the reaction's substrates to the reaction's products. We further initially assume that metabolites can transfer freely between species, a common simplifying assumption in various community models (Gordon and Klaenhammer, 2011; Raymond and Segrè, 2006; Song *et al.*, 2014; Taffs *et al.*, 2009), though we relax this assumption later. The metabolic potential of each community can accordingly be viewed as the aggregate set of metabolic reactions of the member species. A solution to our design task is therefore a minimal set of species that collectively include some set of metabolic reactions

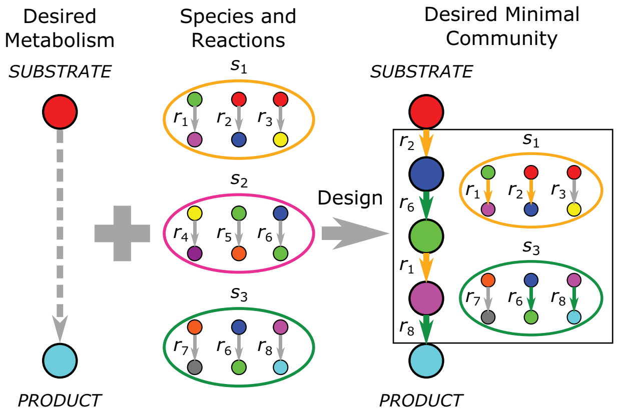

Fig. 1. Schematic representation of the design task. Circles represent metabolites, with arrows between metabolites representing metabolic reactions and ovals representing species. The presence of a reaction within a species indicates that this species can catalyze that reaction. Given desired products and available substrates (left) and a set of available species (middle), our algorithm aims to identify a minimal subset of species that can collectively synthesize the desired products from the available substrates (right)

sufficient to form valid paths to all target products from available substrates. This design task is depicted in Figure 1.

To solve the above design problem, we used an integer linear programming (ILP) formulation. ILP is a framework for defining a linear expression of variables to maximize/minimize, along with a set of linear equations and inequalities that constrain those variables. ILP is a well-established framework, with several efficient solvers and numerous applications (Wolsey and Nemhauser, 2014).

Below, we introduce an ILP formulation of our design task, inspired by ILP-based solutions to both the set cover problem and the network flow problem. To outline the different conceptual parts of our algorithm, we construct this ILP formulation in multiple steps. We first assume that all reactions are simple (connecting a single substrate to a single product) and that a set of reactions necessary to form paths from available substrates to all target products is specified. We show that, with these assumptions, identifying a minimal set of species that collectively encode this required set of reactions can be represented as a *set cover* problem and solved using an ILP formulation. Next, we relax our assumption of specified paths (or a specified set of reactions), introducing an array of *network flow*-inspired ILP constraints that defines possible paths from available substrates to target products using terms that can be linked to the set cover formulation. Finally, we consider the presence of multiple-substrate multiple-product reactions and adjust our network flow constraints to account for such hyperedges in the metabolic network.

2.2 Species, reactions and metabolites in a simple metabolic network representation

There are three main components to our community design problem: the set of available species, the metabolic reactions catalyzed by each species, and the metabolites these reactions consume and produce. Let $M = \{m_1, m_2, \ldots, m_n\}$ denote the set of possible metabolites where n is the number of metabolites. We additionally define $R = \{r_1, r_2, \ldots, r_p\}$ to be the set of reactions, where p is the number of reactions. Each reaction can then be defined as an ordered pair of metabolites, representing the reaction's substrate and product, respectively:

$$r_j = (m_{j\_\mathrm{substrate}},\ m_{j\_\mathrm{product}}).$$

For now assume that each reaction has one substrate metabolite and one product metabolite. This assumption will be relaxed later. Similarly, let $S = \{s_1, s_2, \ldots, s_q\}$ denote the set of species, where q is the number of species. Each species, s, in our formulation can be defined as the set of reactions it can catalyze:

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

We additionally define the set of available substrate metabolites and set of target product metabolites as:

$$\mathrm{SUBSTRATE} = \{m_{\mathrm{substrate}\_1},\ m_{\mathrm{substrate}\_2}, \ldots,\ m_{\mathrm{substrate}\_b}\},$$

$$\mathrm{PRODUCT} = \{m_{\mathrm{product}\_1},\ m_{\mathrm{product}\_2}, \ldots, m_{\mathrm{product}\_c}\},$$

where b is the number of substrates and c is the number of products. Notably, with these definitions, metabolites and reactions can also be viewed as a graph or a network, where nodes represent metabolites and edges represent reactions connecting substrates to products. Notice also that each species can be associated with some subgraph of this metabolic graph based on the set of reactions that species can catalyze.

2.3 Finding a minimal set of species with a pre-specified collection of metabolic capacities

To focus on the minimization aspect of the algorithm, first assume that there is a specified set of necessary metabolic reactions, $N \subseteq R$, that provides valid paths from SUBSTRATE to PRODUCT, such as the set of metabolic reactions in Figure 1 (right). Given this assumption, our aim is to identify a solution set of species that both can collectively catalyze this set of necessary reactions, and is minimal (in terms of the number of species). Since each species is viewed as some subset of the possible reactions, this task corresponds to identifying the minimal set of such subsets whose union contains the specified set of necessary reactions N. This representation of our task forms an instance of the well-defined set cover (SC) problem, which can be solved using an ILP formulation (see also Ye and Doak, 2009). Specifically, first we define a set of binary ILP species variables $I\_S = \{I\_s_1, I\_s_2, \ldots, I\_s_q\}$ such that each ILP species variable, I_s, corresponds to a species s:

$$I\_s_i \in \ \{0, 1\} : i \in \{1, 2, \ldots, q\},$$

with $I\_s_i = 1$ indicating that the ith species is included in the solution species set, and $I\_s_i = 0$ indicating that the ith species is not included. Given these ILP species variables, the objective function of minimizing the number of species included in the solution species set can be defined as:

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

To link the set of species to be included in the solution set to the sets of reactions each species can catalyze and the set of specified necessary reactions N, we define an additional set of binary ILP reaction variables $I\_R = \{I\_r_1, I\_r_2, \ldots, I\_r_p\}$ such that each ILP reaction variable, I_r, corresponds to a reaction r:

$$I\_r_j \in \{0, 1\} : j \in \{1, 2, \ldots, p\},$$

with $I\_r_j = 1$ indicating that the jth reaction is included in N, and $I\_r_j = 0$ indicating that the jth reaction is not included. Given these ILP reaction variables, the constraints ensuring that each necessary reaction can be catalyzed by at least one species can be defined as:

$$\sum_{\substack{\forall i \ s.t. \\ r_j \in s_i}} (I\_s_i) \geq I\_r_j : j \in \{1, 2, \ldots, p\}. \tag{2}$$

In other words, these constraints require that if a reaction is necessary ($I\_r_j = 1$), then there must be at least one species in the solution species set that catalyzes that reaction. The objective function (1) and the set of constraints (2) thus fully define an ILP formulation of the SC component of the algorithm, minimizing the number of species required to catalyze a known set of necessary metabolic reactions.

As a brief example of such a formulation, consider the set of available species in Figure 1 (middle) and the set of reactions in the displayed solution (right). By appropriately assigning ILP species variables, the objective function for this instance would be:

$$\min(I\_s_1 + I\_s_2 + I\_s_3).$$

Similarly, when we assign the ILP reaction variables and their values, we can formulate the set cover constraints associated with this instance (following the general form of constraint (2)):

$$\begin{aligned}
I\_r_2 = 1 \rightarrow I\_s_1 \qquad\qquad\quad &\geq 1\\
I\_r_6 = 1 \rightarrow \qquad\quad I\_s_2 + I\_s_3 &\geq 1\\
I\_r_1 = 1 \rightarrow I\_s_1 \qquad\qquad\quad &\geq 1\\
I\_r_8 = 1 \rightarrow \qquad\qquad\quad I\_s_3 &\geq 1.
\end{aligned}$$

Together, this specific objective function and these specific constraints define ILP problem associated with the task depicted in Figure 1 assuming the reactions in the presented path are necessary.

2.4 Considering all possible paths from available substrates to target products

When defining the SC component of the algorithm above, a set of necessary metabolic reactions connecting SUBSTRATE to PRODUCT was assumed to be predefined. Clearly, however, when considering the complex network of metabolic reactions that can be catalyzed by microbial species, there are likely numerous alternative paths connecting the available substrates to the desired target products. Since one cannot know *a priori* which paths require the fewest species to catalyze, a complete solution to our design problem must consider all possible paths when minimizing the number of species. To address this challenge and to remove the assumption of a specified set of necessary reactions, we use network flow (NF)-inspired constraints. Specifically, instead of predefining the values of the I_r variables to denote which reactions are necessary, we allow I_r values to vary freely and introduce a set of constraints that guarantee that the collection of reactions for which $I\_r = 1$ form valid paths from SUBSTRATE to PRODUCT. Intuitively, an NF problem considers a graph as a network of pipes where the task is to push the maximal flow through these pipes from a source node to a sink node. Here, we use this NF-based approach to define a valid path in the metabolic network as a set of reactions that allow flow to pass from SUBSTRATE to PRODUCT.

To define such a valid path, first we define a set of ILP flow variables, $F\_R = \{F\_r_1, F\_r_2, \ldots, F\_r_p\}$, where F_r_j denotes the amount of flow passing through the jth reaction. Since flow has to be non-negative and since real-valued flow variables are unnecessary and slow computation, we further limit the values for flow variables to non-negative integers:

$$F\_r_j \in \mathbb{N} : j \in \{1, 2, \ldots, p\}.$$

The first NF constraint requires that only metabolites in SUBSTRATE can be sources of flow, hence forcing any viable path to start from an available substrate metabolite (Fig. 2A):

$$\sum_{\substack{\forall j\ s.t.\\ m_j \in \text{SUBSTRATE}}} \left(- \sum_{\substack{\forall in\ s.t.\\ r_{in} = (m_i, m_j)}} F_{r_{in}} + \sum_{\substack{\forall out\ s.t.\\ r_{out} = (m_j, m_k)}} F_{r_{out}} \right) = |\text{PRODUCT}|, \tag{3}$$

where $i,j,k \in \{1,2,\ldots,n\}$, $i \neq j$, $j \neq k$, and $in, out \in \{1,2,\ldots,p\}$. In other words, the sum of flow leaving all available substrate metabolite nodes must be greater than the sum of flow entering these metabolites. Specifically, we require that the difference in flow be equal to the number of target product metabolites (|PRODUCT|), ensuring that each target product can receive one unit of flow if a viable path exists. Note here that even though we would not usually need flow to enter a substrate metabolite node, it may be necessary for problems involving forced substrate usage (see Supplementary Text 1).

The second NF constraint requires that metabolites in PRODUCT be flow sinks, forcing every viable path to end at a target product (Fig. 2B):

$$\sum_{\substack{\forall in\ s.t.\\ r_{in} = (m_i, m_j)}} F\_r_{in} - \sum_{\substack{\forall out\ s.t.\\ r_{out} = (m_j, m_k)}} F\_r_{out} = 1 \qquad : \forall j\ \ s.t.\ \ m_j \in \text{PRODUCT}. \tag{4}$$

This forces the flow into any target metabolite node to be greater than the flow leaving that node by one unit of flow. Thus, each product metabolite must be reached by some viable path. It should be noted that the network flow solution does not necessarily reflect *all* metabolic activity and, for example, intermediate reactions' bi-products could still be generated even when no flow is associated with these bi-products.

The third NF constraint asserts that all metabolites not in SUBSTRATE or PRODUCT have zero net flow (i.e. neither sources nor sinks of flow), allowing such metabolites to serve as intermediate nodes in any viable path (Fig. 2C):

$$\sum_{\substack{\forall in\ s.t.\\ r_{in} = (m_i, m_j)}} F\_r_{in} = \sum_{\substack{\forall out\ s.t.\\ r_{out} = (m_j, m_k)}} F\_r_{out} \qquad : \forall j\ \ s.t.\ \ m_j \in R, m_j \notin \text{SUBSTRATE}, m_j \notin \text{PRODUCT}. \tag{5}$$

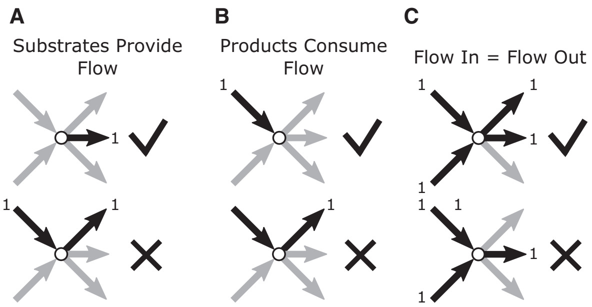

Fig. 2 The network flow constraints. (**A**) The net flow out of all available substrates must be equal to the number of target products. (**B**) The net flow into any target product must be equal to one. (**C**) The net flow for any intermediate metabolite must be zero. Together these constraints define any viable set of metabolic reactions that form paths from available substrates to target products

Given these NF constraints, any viable set of metabolic reactions for converting SUBSTRATE to PRODUCT will have non-zero associated ILP flow variables.

Finally, to appropriately set the I_r reaction variables to 1 if the reaction is used in the NF task and 0 otherwise, an additional set of conversion constraints is added:

$$F\_r_j \le |\text{PRODUCT}| \times I\_r_j : \forall j \in \{1, 2, \ldots, p\}, \qquad (6)$$

ensuring that if a reaction's flow variable is greater than 0, then the ILP reaction variable for that reaction must be 1.

Combining the objective function (1) and the constraints (2)–(6) for the SC and NF tasks therefore provides a complete ILP formulation for minimizing the number of species in the solution species set while ensuring the existence of viable paths from available substrates to all target products.

2.5 Allowing for metabolic reactions with multiple substrates and products

The ILP formulation above relies on the assumption that each metabolic reaction has a single substrate and a single product. This is a common simplification in metabolic network analysis and various protocols exist to reconstruct metabolic networks in which this assumption holds (Levy and Borenstein, 2013; Parter *et al.*, 2007). Yet, a more complete and accurate metabolic network formulation allows metabolic reactions to have multiple substrates (accounting, for example, for co-factors) and/or multiple products. To account for such metabolic reactions, we modify our metabolic network representation and instead of connecting substrate nodes to product nodes directly, we introduce a new type of node, representing reactions, and connect the (potentially multiple) substrates of each reaction to the (potentially multiple) products through the appropriate reaction node (Fig. 3A). In this representation, reactions, *r*, are therefore no longer representing edges in the network, but rather nodes that connect

$$\{m_{j\_\text{substrate}\_1}, m_{j\_\text{substrate}\_2}, \ldots, m_{j\_\text{substrate}\_d}\}$$

to

$$\{m_{j\_\text{product}\_1}, m_{j\_\text{product}\_2}, \ldots, m_{j\_\text{product}\_e}\},$$

where *d* is the number of substrates and *e* is the number of products for the *j*th reaction. Specifically, we define two new classes of edges: a set of reaction input edges, $I = \{i_1, i_2, \ldots, i_t\}$ where *t* is the number of substrate metabolites across all reactions, connecting a substrate metabolite *m* to a reaction *r*:

$$i_j = (m_{j\_\text{input}}, r_{j\_\text{reaction}}),$$

and a set of reaction output edges, $O = \{o_1, o_2, \ldots, o_u\}$ where *u* is the number of product metabolites across all reactions, similarly connecting a reaction *r* to its product metabolite *m*:

$$o_j = (r_{j\_\text{reaction}}, m_{j\_\text{output}}).$$

Together, these new nodes and edges thus define a bipartite graph where edges only exist between one metabolite node and one reaction node, but never between two metabolites or two reactions (Fig. 3A).

Now we redefine the set flow variables in this network as two sets, $F\_I = \{F\_i_1, F\_i_2, \ldots, F\_i_t\}$ and $F\_O = \{F\_o_1, F\_o_2, \ldots, F\_o_u\}$ where F_i and F_o represent flow along input and output edges respectively. Notably, most of the flow constraints defined above are still valid when applied to both metabolite and reaction nodes; however, constraint (6), which aimed to link the flow variables to the ILP reaction variables, I_r, needs to be updated to represent the link between a reaction's multiple substrates and products. Specifically, one set of constraints is introduced to ensure that a reaction can be active only if all its substrates are present (in other words, if all reaction input edges have flow):

$$F\_i_j \ge I\_r_k : \forall j, k \ \ s.t. \ \ i_j = (m_l, r_k). \qquad (7)$$

Then, another set of constraints is introduced to allow active reactions to generate products (by providing flow on the reaction output edges):

$$\sum_{\substack{\forall i \ s.t. \\ o_i = (r_j, m_k)}} (F\_o_i) \le (|I| + |O|) \times I\_r_j : \forall j \in \{1, 2, \ldots, p\}. \qquad (8)$$

The only difference between constraints (6) and (8) is that the maximum flow across a single edge is no longer bounded above by just the number of target products but instead the number of edges in the network. This difference is due to the need for all substrates of a reaction to provide flow instead of just a single substrate, which may require multiple units of flow to reach a single target product. Combined, constraints (7) and (8) guarantee that a reaction's products can be available only if all of the reaction's substrates are available, and that if the reaction is required for generating flow the ILP reaction variable for that reaction must be 1.

2.6 Compartmentalizing species and defining transport reactions

The algorithm so far has treated the community as a single superorganism, whereby metabolites can transfer freely between species. A more realistic scenario, however, assumes that metabolites are compartmentalized within each species and requires species to have the necessary transport reactions to allow environmental metabolite

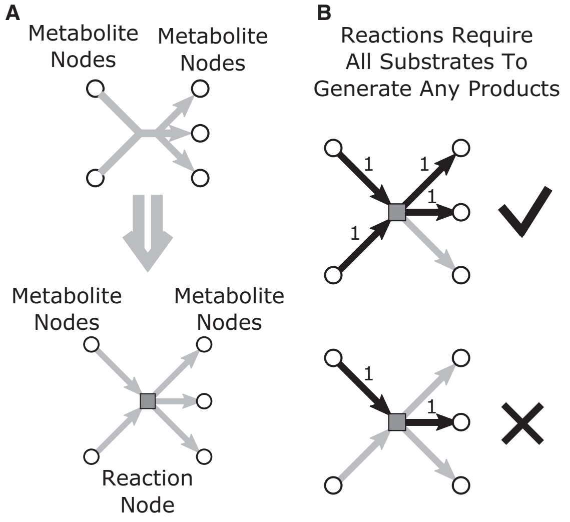

Fig. 3. The bipartite graph representation of multi-substrate, multi-product metabolic reactions and associated flow constraints. (**A**) Each metabolic reaction is replaced by a new node type representing the enzyme that catalyzes that reaction. New edges are added to indicate the input and output metabolites for each reaction. (**B**) The new flow constraints require that all reaction substrates can provide flow to use a reaction and that reaction output edges can have flow only if all input edges have flow

uptake and secretion. Such a compartmentalized problem can in fact be solved by the algorithm as currently defined simply by modifying the sets of metabolites and reactions. Specifically, rather than having a single variable to denote each metabolite (regardless of its compartment), a *set* of variables should be defined to denote the metabolite in each compartment in which it exists (be it a specific species or the shared environment). Given this extended set of metabolite variables, metabolic reactions are now viewed as operating within a species (and accordingly connect species-specific substrates to species-specific products). An additional set of transport reactions (which correspond to each species' uptake and secretion capacities) can then convert species-specific metabolites to environmental metabolites and vice-versa. In this compartmentalized setting, one species can only use metabolites produced by another species if both species have the appropriate transport reactions. For example, for a given metabolite *m* to transfer from species A to species B, species A must include a transport reaction to convert the A-specific version of *m* to an environmental version, and similarly species B must include a transport reaction to convert the environmental version of *m* to a B-specific version.

More formally, we now define $M = \{m_{0,1}, m_{0,2}, \ldots, m_{0,n}, m_{1,1}, \ldots, m_{q,n}\}$ as the set of metabolites such that $m_{i,k}$ where $i \in \{0,1,\ldots,q\}$ and $k \in \{1,2,\ldots,n\}$ denotes metabolite *k* in species *i*. We interpret the 0^{th} species as the shared environment. We then replace each reaction, r_j, present in species *I* in our previous formulation with a new reaction:

$$r_{i,j} = \left(\left\{m_{i,j\_\mathrm{substrate}\_1}, m_{i,j\_\mathrm{substrate}\_2}, \ldots, m_{i,j\_\mathrm{substrate}\_d}\right\},\right.$$

$$\left.\left\{m_{i,j\_\mathrm{product}\_1}, m_{i,j\_\mathrm{product}\_2}, \ldots, m_{i,j\_\mathrm{product}\_e}\right\}\right).$$

Each species may also include a set of transport reactions that convert environmental metabolites to species-specific metabolites (reflecting uptake reactions):

$$r_{i,\mathrm{transport}\_l} = \left(m_{0,k}, m_{i,k}\right),$$

or species-specific metabolites to environmental metabolites (representing secretion):

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

Together, these new metabolite and reaction definitions relax the assumption of freely transferred metabolites and allow our algorithm to solve problems in a compartmentalized species setting.

2.7 Forcing substrate usage and incorporating species costs

The above ILP-based formulation can be further extended to force the obtained solution to meet additional requirements or to consider additional factors. Specifically, we have developed and implemented algorithm extensions to handle two biologically relevant considerations, the first forcing the solution to utilize (or degrade) specified substrates, and the second to weigh species' predefined desirability when constructing a community. For a detailed description of the associated constraints and modifications, see Supplementary Text 1.

3 Results

3.1 Algorithm implementation and availability

We implemented the algorithm outlined above as a C++ program which takes as input a file describing the various parameters of the design task, including available substrates, target products and the set of available species with their associated metabolic reactions. The program then generates the associated ILP instance in the Mathematical Programming System (MPS) format (default) or the CPLEX format (depending on the requirements of the ILP solver used). The source code for the algorithm is available under a non-commercial research use only license at https://github.com/borenstein-lab/CoMiDA. To obtain solutions for our test cases and dataset analysis, we used the COIN-OR Branch and Cut (CBC) solver (Lougee-Heimer, 2003).

3.2 Unit test validation

We first aimed to verify our algorithm using a set of simple design problems. Specifically, we generated a suite of toy problems as unit test cases for our algorithm. These toy problems test whether our algorithm identifies an optimal solution under different scenarios that cover a variety of edge cases. These test cases focus on simple design tasks, with up to five species and up to seven associated metabolic reactions. For example, some cases examined scenarios in which the minimal species solution requires a longer metabolic path from substrate to product than a non-minimal species solution. Other cases examined scenarios in which a solution does not exist (e.g. because no path exists from substrate to product, regardless of which species are used). We have applied our algorithm to each of these test cases and confirmed that our algorithm correctly produces the ILP formulation and ultimately identifies an optimal solution for each design task (or the absence of one). These toy problems, along with their expected ILP formulations, can be found (with the source code) at https://github.com/borenstein-lab/CoMiDA, providing users with simple examples of the expected input/output format and allowing users to confirm that the algorithm is working properly.

3.3 Glycolysis pathway validation

The toy problems described above are limited in size and may not be comparable in scale to many real-world scenarios. To examine our algorithm's performance on datasets of a more practical size, we next focused on a well-characterized metabolic pathway, the Embden-Meyerhof glycolysis pathway (KEGG entry *M00001* (Kanehisa *et al.*, 2014; Ogata *et al.*, 1999)), defining *glucose* and *pyruvate* as the available substrate and target product respectively (Supplementary Fig. S1). For the set of available species, we selected all 284 species identified from the 2013 Human Microbiome Project (HMP) (Human Microbiome Project Consortium, 2012a) stool sample datasets that contained the entire set of metabolic reactions in the glycolysis pathway as predicted by PICRUSt (Langille *et al.*, 2013). Combined, this set of species corresponds to an aggregate metabolic network containing 1803 metabolites and 3120 metabolic reactions. Since each species in this set can catalyze the entire pathway from glucose to pyruvate, our algorithm identifies, as expected, a single species solution (one of the 284 possible choices). To test our algorithm's performance when minimal solutions required multiple species, we next therefore modified the metabolic network of each species, deleting various reactions and forcing a multi-species solution. Specifically, we first removed all alternate reaction paths between glucose and pyruvate by removing the first reaction in each alternate path that was not also part of the glycolysis pathway (Supplementary Fig. S1), filtering out 39 reactions and leaving a total of 3081 metabolic reactions in the aggregate network. We then removed selected reactions in the glycolysis pathway from subsets of the available species such that no single species contained all reactions in the path (e.g. by removing one reaction in the pathway from half of the available species and a different reaction from the other half). Through numerous such modifications, we forced minimal solutions for providing the glycolysis pathway to require multiple species, fully controlling the size of these minimal solutions. We confirmed that our algorithm was able to handle such cases and to provide a correct minimal solution in each such scenario.

3.4 Analysis of minimal communities of gut microbiome species

Naturally occurring microbial communities often comprise an extremely complex and diverse collection of species (Human Microbiome Project Consortium, 2012b; Lozupone and Knight, 2007). This diversity can be the product of numerous factors, including a variety of niches species can occupy (Escalante *et al.*, 2015; Rainey and Travisano, 1998), metabolic specialization of individual species within the community (Johnson *et al.*, 2012; Zhou *et al.*, 2002), intricate multi-species interactions (Doebeli and Ispolatov, 2010), or functional redundancy (Ley *et al.*, 2006; Nemergut *et al.*, 2013). Yet, when designing synthetic communities, markedly fewer species may be required (Petrof *et al.*, 2013). To explore this possibility and to characterize potential redundancy in naturally occurring communities, we used our algorithm to identify minimal communities required to perform various simple metabolic syntheses within the context of a diverse natural community. Specifically, given the promise of gut microbiome-based therapies, we focused on minimal communities that consist of gut dwelling species. To this end, we selected a set of 2051 species (represented as Operational Taxonomic Units; OTUs) detected via 16S sequencing of HMP stool samples. The set of metabolic reactions each OTU could catalyze was determined using PICRUSt (Langille *et al.*, 2013). Combined, the aggregate metabolic network of this set of species included 2225 unique metabolites. We then selected 10 000 random pairs of metabolites from this set, one as the available substrate and one as the target product, and used our algorithm to identify minimal communities that could provide a pathway from substrate to product. We specifically used our algorithm in three different settings: one with the metabolic network simplified such that each reaction has a single substrate and a single product (see above), one with the full bipartite graph representation of the metabolic network (allowing each reaction to have multiple substrates and/or multiple products), and one with the full bipartite graph representation but also including a set of common currency metabolites (based on (Greenblum *et al.*, 2012)) as available substrates.

As shown in Figure 4, for most random metabolite pairs, no set of species had the capacity to perform the desired synthesis (potentially owing to various gaps in the aggregate metabolic network and incomplete annotation of the various species). Yet, when a solution existed, it generally required only very few species (≤ 5 for all metabolite pairs tested). Notably, since the simplified graph representation requires only one substrate of a reaction to be available to generate any of that reaction's products (ignoring, for example, the need for additional co-factors), many more solutions existed when this simple graph representation was used compared with the complete bipartite representation. Making currency metabolites available evidently allowed additional reactions to be active and therefore recovered some of the metabolic capacity that could not be realized in the bipartite graph representation. Given the small minimal communities identified and the small number of unique species used in these communities across all pairs (379 OTUs), one might suspect that a small number of metabolic generalist species are responsible for providing the required metabolic capacity in many of these minimal communities. Indeed, the number of reactions a species can catalyze was found to be positively correlated with the frequency with which that species was used in the identified minimal solutions ($r = 0.496$, P-value $\leq 10^{-15}$; Pearson correlation test). Importantly, however, even the species that appears most frequently in identified minimal solutions occurs in only $<10\%$ of the solutions, with the next most frequent species occurring in $<5\%$. Together, these results suggest that even though a small number of species are required to perform any relatively simple target synthesis, a variety of species may be needed to catalyze a range of simple substrate/product conversions.

4 Discussion

Recent efforts to manipulate various naturally occurring communities and to impact their activities have shown tremendous promise. For example, efforts to modify the human gut microbiome have demonstrated that properly perturbing this community can treat or ameliorate certain conditions (Aroniadis and Brandt, 2013). Expanding this approach to effectively treat a wider variety of diseases, as well as alter the functions of environmentally and industrially-relevant microbial communities, requires methods for rationally designing communities with specific metabolic capacities. Above, we take a first step towards this goal, introducing and validating a novel algorithm which identifies minimal microbial communities that provide specified and desired metabolic capacities.

Clearly, various biological factors are currently not considered by our algorithm, including, for example, species-level interactions (Hansen *et al.*, 2007), the expected flux through each metabolic reaction, and whether member species, or even the community as a whole, can survive in the target environment (Ley *et al.*, 2006). Ignoring such factors may render communities designed by our algorithm markedly different than naturally occurring communities and synthetic communities constructed based on such designs may consequently fail to survive or to perform specific desired tasks. For example, the discrepancy between the small communities identified in our analysis above and the extreme diversity observed in many naturally occurring communities (Human Microbiome Project Consortium, 2012b; Lozupone and Knight, 2007) may be likely accounted for, at least partly, by such factors. Yet, our algorithm

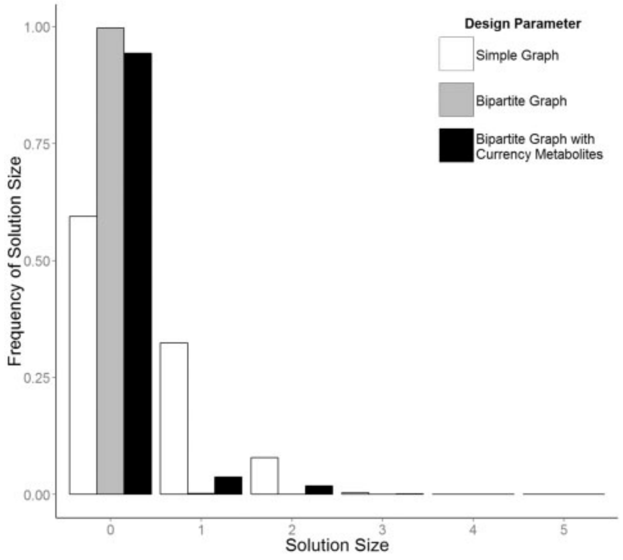

Fig. 4 Solution sizes identified for 10 000 random substrate/product metabolite pairs, using species from the Human Microbiome Project

provides a starting point for such design efforts and for future method development in this area. Specifically, selecting a community based initially on the presence of desired metabolic capacities provides a simple way to address an important prerequisite for community metabolism; any community designed to consume or produce given metabolites or to have some metabolic activity must obviously also have the metabolic capacity to carry out those functions. Our attempt to identify minimal communities may again not necessarily be aligned with biological assembly rules, but offers simple candidate communities for further design refinement. Moreover, by formulating our program as an ILP algorithm, we provide an easy way to introduce additional design considerations. As our understanding of the various constraints affecting community assembly improves, such considerations can be added to this framework by devising equations and inequalities that encode these constraints.

Of the various considerations that could be implemented to further refine any design approach, two stand out as logical next steps. First, the expected stability of designed communities could be improved by examining the likelihood that a combination of species will coexist in a community. Specifically, information on species co-occurrence in natural communities can be used to estimate the tendency of various species pairs to co-exist or the exclude one another from a shared environment (Faust *et al.*, 2012; Levy and Borenstein, 2013). Such information would allow an algorithm to prioritize communities that minimize the risk of losing member species due to antagonistic species interactions, ultimately stabilizing community structure. Second, considering the *predicted* activity of candidate communities, rather than just the presence of specific metabolic capacities, could increase the likelihood that designed communities would perform the desired task. Several frameworks for predicting the metabolic activity of microbial communities have recently been introduced (Chiu *et al.*, 2014; Harcomb *et al.*, 2014; Zhuang *et al.*, 2011; Zomorrodi *et al.*, 2014), potentially allowing future design algorithms to consider predicted rates of metabolite consumption and production and predicted changes in species abundances over time. Our algorithm could be used, for example, as an initial filtering step, providing a set of candidate minimal communities that have the capacity for some desired metabolism, followed by a metabolic model-based prediction of the metabolic activity of each candidate community to further refine the design process. Moreover, such metabolic modeling could allow the design process to account for important factors that our current algorithm may not be able consider. For instance, our algorithm does not explicitly prevent community members from degrading one or more of the specified target products. Such inadvertent target metabolite degradation may depend on the set of microbes present, other available substrates, and various environmental conditions, and could therefore be predicted and potentially avoided using metabolic modeling-based design.

The ability to computationally design microbial communities will be a useful tool for many purposes. For example, designed synthetic communities could be ultimately used in place of FMTs, removing the need for screening donor samples while also optimizing treatments to target specific conditions. Communities could also be created for industrial resource and pharmaceutical production, potentially obviating the need for extensive microbial genetic engineering and providing novel mechanisms for production control in the form of inter-species signaling (Brenner *et al.*, 2008). Clearly, such applications are not yet feasible and the development of a comprehensive, general-purpose design framework may still be out of our reach for years to come. We hope, however, that our framework will encourage future developments of such design methodologies and will lay the foundation for future efforts in microbial community design.

Acknowledgements

We are grateful to all members of the Borenstein lab for helpful discussions.

Funding

This work has been supported by New Innovator Award DP2 AT007802-01 to EB from the National Institutes of Health.

Conflict of Interest: none declared.

References

Alleman,J.E. and Prakasam,T.B.S. (1983) Reflections on seven decades of activated sludge history. *J. Water Pollut. Fed.*, **55**, 436–443.

Aroniadis,O.C. and Brandt,L.J. (2013) Fecal microbiota transplantation: past, present and future. *Curr. Opin. Gastroenterol.*, **29**, 79–84.

Brenner,K. *et al.* (2008) Engineering microbial consortia: a new frontier in synthetic biology. *Trends Biotechnol.*, **26**, 483–489.

Chiu,H.C. *et al.* (2014) Emergent biosynthetic capacity in simple microbial communities. *PLoS Comput. Biol.*, **10**, e1003695.

Doebeli,M. and Ispolatov,I. (2010) Complexity and diversity. *Science*, **328**, 494–497.

Edwards,J.S. and Palsson,B.O. (1998) How will bioinformatics influence metabolic engineering? *Biotechnol. Bioeng.*, **58**, 162–169.

Escalante,A.E. *et al.* (2015) Ecological perspectives on synthetic biology: insights from microbial population biology. *Front. Microbiol.*, **6**, 143.

Faust,K. *et al.* (2012) Microbial co-occurrence relationships in the human microbiome. *PLoS Comput. Biol.*, **8**, e1002606.

Gordon,J.I. and Klaenhammer,T.R. (2011) A rendezvous with our microbes. *Proc. Natl. Acad. Sci. U. S. A.*, **108**, 4513–4515.

Greenblum,S. *et al.* (2012) Metagenomic systems biology of the human gut microbiome reveals topological shifts associated with obesity and inflammatory bowel disease. *Proc. Natl. Acad. Sci. U. S. A.*, **109**, 594–599.

Hamilton,M.J. *et al.* (2013) High-throughput DNA sequence analysis reveals stable engraftment of gut microbiota following transplantation of previously frozen fecal bacteria. *Gut Microbes*, **4**, 125–135.

Hansen,S.K. *et al.* (2007) Evolution of species interactions in a biofilm community. *Nature*, **445**, 533–536.

Harcomb,W.R. *et al.* (2014) Metabolic resource allocation in individual microbes determines ecosystem interactions and spatial dynamics. *Cell Rep.*, **7**, 1104–1115.

Human Microbiome Project Consortium. (2012a) A framework for human microbiome research. *Nature*, **486**, 215–221.

Human Microbiome Project Consortium. (2012b). Structure, function and diversity of the healthy human microbiome. *Nature*, **486**, 207–214.

Johnson,D.R. *et al.* (2012) Metabolic specialization and the assembly of microbial communities. *ISME J.*, **6**, 1985–1991.

Kanehisa,M. *et al.* (2014) Data, information, knowledge and principle: back to metabolism in KEGG. *Nucleic Acids Res.*, **42**, D199–D205.

Langille,M.G.I. *et al.* (2013) Predictive functional profiling of microbial communities using 16S rRNA marker gene sequences. *Nat. Biotechnol.*, **31**, 814–821.

Levy,R. and Borenstein,E. (2013) Metabolic modeling of species interaction in the human microbiome elucidates community-level assembly rules. *Proc. Natl. Acad. Sci. U. S. A.*, **110**, 12804–12809.

Ley,R.E. *et al.* (2006) Ecological and evolutionary forces shaping microbial diversity in the human intestine. *Cell*, **124**, 837–848.

Lougee-Heimer,R. (2003) The Common Optimization INterface for Operations Research: promoting open-source software in the operations research community. *IBM J. Res. Dev.*, **47**, 57–66.

Lozupone,C.A. and Knight,R. (2007) Global patterns in bacterial diversity. *Proc. Natl. Acad. Sci. U. S. A.*, **104**, 11436–11440.

Marlow,J.J. *et al*. (2014) Carbonate-hosted methanotrophy represents an unrecognized methane sink in the deep sea. *Nat. Commun*., **5**, 5094.

Moralejo-Gárate,H. *et al*. (2011) Microbial community engineering for biopolymer production from glycerol. *Appl. Microbiol. Biotechnol*., **92**, 631–639.

Moran,N.A. (2007) Symbiosis as an adaptive process and source of phenotypic complexity. *Proc. Natl. Acad. Sci. U. S. A*., **104**, 8627–8633.

Nemergut,D.R. *et al*. (2013) Patterns and processes of microbial community assembly. *Microbiol. Mol. Biol. Rev. MMBR*, **77**, 342–356.

Ogata,H. *et al*. (1999) KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Res*., **27**, 29–34.

Parter,M. *et al*. (2007) Environmental variability and modularity of bacterial metabolic networks. *BMC Evol. Biol*., **7**, 169.

Pelz,O. *et al*. (1999) Towards elucidation of microbial community metabolic pathways: unravelling the network of carbon sharing in a pollutant-degrading bacterial consortium by immunocapture and isotopic ratio mass spectrometry. *Environ. Microbiol*., **1**, 167–174.

Petrof,E.O. *et al*. (2013) Stool substitute transplant therapy for the eradication of Clostridium difficile infection: "RePOOPulating" the gut. *Microbiome*, **1**, 3.

Pettit,R.K. (2009) Mixed fermentation for natural product drug discovery. *Appl. Microbiol. Biotechnol*., **83**, 19–25.

Rainey,P.B. and Travisano,M. (1998) Adaptive radiation in a heterogeneous environment. *Nature*, **394**, 69–72.

Raymond,J. and Segrè,D. (2006) The effect of oxygen on biochemical networks and the evolution of complex life. *Science*, **311**, 1764–1767.

Rossen,N.G. *et al*. (2015) Fecal microbiota transplantation as novel therapy in gastroenterology: a systematic review. *World J. Gastroenterol. WJG*, **21**, 5359–5371.

Sekirov,I. *et al*. (2010) Gut microbiota in health and disease. *Physiol. Rev*., **90**, 859–904.

Song,Y. *et al*. (2013) Microbiota dynamics in patients treated with fecal microbiota transplantation for recurrent Clostridium difficile infection. *PloS One*, **8**, e81330.

Song,H.S. *et al*. (2014) Mathematical modeling of microbial community dynamics: a methodological review. *Processes*, **2**, 711–752.

Taffs,R. *et al*. (2009) In silico approaches to study mass and energy flows in microbial consortia: a syntrophic case study. *BMC Syst. Biol*., **3**, 114.

Wolsey,L.A. and Nemhauser,G.L. (2014) *Integer and Combinatorial Optimization*. John Wiley & Sons, New York.

Ye,Y. and Doak,T.G. (2009) A parsimony approach to biological pathway reconstruction/inference for genomes and metagenomes. *PLoS Comput. Biol*., **5**, e1000465.

Zhou,J. *et al*. (2002) Spatial and Resource Factors Influencing High Microbial Diversity in Soil. *Appl. Environ. Microbiol*., **68**, 326–334.

Zhuang,K. *et al*. (2011) Genome-scale dynamic modeling of the competition between *Rhodoferax* and *Geobacter* in anoxic subsurface environments. *ISME J*., **5**, 305–316.

Zomorrodi,A. *et al*. (2014) d-OptCom: dynamic multi-level and multi-objective metabolic modeling of microbial communities. *ACS Synth. Biol*., **3**, 247–257.