Problem Statement  
Currently, community flux balance analysis models are optimized through non-linear systems. They elegantly describe community models mathematically, however non-linear optimization methods are often difficult and sometimes even impossible to solve. We have developed a new pipeline which uses only linear programming to perform community flux balance analysis.

Methods  
First, we create metabolic models, in JSON file format, for each individual species that are present in the community. Each metabolic model is tailored to contain extra external metabolites and reactions that are contributed from other species in the community. Thus, the extracellular space of each metabolic model is unique to each individual species despite the fact that all members belong to the same community. After the creation of individual metabolic models, we iterate through each model and use COBRApy to optimize for each model’s biomass objective function and subsequently store the solutions in text files.

Using the solutions, we calculate and store the averages and standard deviations for all shared reactions by the community in new text file. With this, we change the upper and lower bounds, of each reaction of all the species’ models, to the average flux value plus two standard deviations and the average flux value minus two standard deviations respectively. With new constraints on shared reactions, we perform flux balance analysis again iteratively for each model with COBRApy, again optimizing for each respective biomass objective function. We then store each flux value returned by the objective function of each species in another new text file.

We take the flux values and calculate z-scores compared to each other. Fractional biomass coefficients will be calculated for each species by taking their respective z-score and diving over the sum of z-scores for all species and will be stored in another text file. The sum of all fractional biomass coefficient should equal to one.

Lastly, a community metabolic model will be created where species are treated as just additional compartments. However, the constraints in the model and/or variables in the objective function for this community model will be weighed by their respective fractional biomass coefficients depending on which species the constraint or variable belongs to. Constraints and/or variables for reactions that are shared between species will be weighed as the sum of the fractional biomass coefficients for the species involved. The final step is to then use COBRApy to optimize for the community biomass objective function, which is defined as the weighed summation of biomass objective functions of all species.

The resultant vector of fluxes should be representative of real-world experimental data.

Overall comments:

* Needs more “rapid advances”
* Needs more references. Way more. Like there are none right now.
* The core obstruction to successful cFBA is inclusion of product formation/inhibition rates in the extracellular space. You can see in Henson’s review, the dashed lines, which do not consider these effects, do not correlate as well with experimental data. However, the dFBA framework, which, as I understand, computes differential equations at each iteration to determine uptake values (thus being able to model product buildup, inhibition, etc.) requires kinetics for these values; i.e. experimental data or some random-ass values. As I see it, the core foundation of this framework acFBA (Anthony’s cFBA) is that taking the average of flux values for shared reactions that contribute to products secreted into or absorbed from the mutual extracellular space (and iterating over this process multiple times) would reflect the inhibition to a specie’s growth/functioning/flux by a specific metabolite. The alternative is to use cFBA as described by Norway people (PLOS One) which assumes a steady-state for external metabolites (preventing the modelling of buildup of metabolites to lead to inhibitions since by the steady-state assumption what is made must be equally consumed) and provides an LP problem which can be solved efficiently for the entire community.
* In regards to “but it only does 5”. Well, a lot of work of cFBA recently has only focused on 2 species. Mostly for co-cultures producing bio-fuels and the like. And by assuming steady-state for secreted metabolites, they turn the problem from a non-linear one into a linear one (I think – I don’t have a developed understanding of the properties of linear vs. non-linear problems – For now I am associating “non-linear” with differential equations used in dFBA). We will see how the acFBA scales. Although, I don’t think “doing more than 5” should be a huge goal of ours because a) I doubt we can create a method that correlates well with experimental data that we don’t have anyways, and b) something I’ll mention in my next bullet
* In Henson’s review (2014, #recent #yolo), it was mentioned that:

“Incorporation of the necessary tools for species genome annotation and metabolic reconstruction, *substrate uptake identification* [(this is I believe the main factor that prevents us from using the more powerful dcFBA in our situation, due to lack of experimental data)], community dynamic modelling, and dynamic simulation and optimisation within a single software platform would be very beneficial”

and our competition:

“the recent release of the Model SEED within the DOE Systems Biology Knowledgebase (KBase) is a positive step in this direction, and plans to expand KBase to microbial communities are under development”

Thus, an *easy* solution to avoiding the “is your cFBA good” attack is that we can simply allow the use of multiple frameworks, and also provide a way to compare flux values generated in them. This will lead to standardising FBA methodologies. And it gets us off the hook for developing a revolutionary cFBA algorithm for communities of +5 species where hardly any experimental data is available and maybe there are complete metabolic models made for some of them… I can code the backend for this in a day or two. The main problem is that we can’t have auto-upload scripts to the server, there needs to be a review process. “Frameworks” can be hosted as repos on GitHub and have a standard way of taking data in and sending it out. This may seem unnecessary but it leads to a modern, standardinizable web framework for FBA and gets us off the hook for revolutionizing cFBA.

Things I have heard about, need to read more about, can include in our “FBA frameworks and analsysis web package – FBAFAWP, read as “FBA fap”

* OptCom
* DMMMMMMMMMM… (DMMM)
* cFBA as described by Norway people
* dcFBA as described in Henson’s review (needs uptake params!!!)
* acFBA
* “geometric FBA” as described in “Flux balance analysis: a geometric perspective” Smallbone and Simeonidis
* multitude of existing single species FBA frameworks, tools (MATLAB, Python, COBRA, etc.)

Refs.

“Hensons review” -> Dynamic flux balance analysis for synthetic microbial communities by Henson and Hanly

“Norway people” -> actually Netherlands.. -> Community flux balance analysis for microbial consortia at balanced growth”

non-linear => can’t be solved as LP