Exploration of Metabolic Networks

Nora Beier Thomas Gatter Natasha Jorge Bruno Schmidt Peter F. Stadler

February 5, 2023

Dates: 06.02.-10.02.2023 and 20.02.-24.02.2023

Final Presentations: 24.02.2023

Work in Groups: 2-3 students per group

Introduction

Obesity is associated with numerous medical conditions. As such it carries high socio-economic costs while treatment options are limited to date. Both obese and lean phenotypes have been linked to the intestinal microbiome and its associated metabolic capacities. Accordingly, manipulation of this community promises potential new treatment options for obesity. The transfer of microbioms and thereby metabolic phenotypes via fecal microbiome transplantation (Fig. 1) is one such approach that has been successfully applied in medical studies. Fecal transplants have also been used as treatment for other (chronic) conditions of the digestive system. In order to increase effectiveness and avoid detrimental effects for the patient, gut communities need to be better understood, with special regard towards metabolic interaction within the microbiota. Mathematical modelling for microbial communities is a core goal of an ongoing research project in our work group.

The Simplified Human Intestinal Microbiota (SIHUMIx, see Tbl. 1) is an intestinal microbiota model community of reduced complexity well suited towards mathematical metabolic modeling. It exhibits highly reproducible growth in laboratory cultivation, thereby allowing consistent measurements to both verify models and increase model accuracy.

Commonly, metabolic pathways are portrayed as a (multi-)set of chemical compounds C and a set of reactions R, each defined as a transformation of a multiset of educts $Y \subseteq C$ to a multiset of products $Y' \subseteq C$. We write $Y \to Y'$. Chemical reaction networks (CRNs) link all reactions in a common datastructure that enables to us to track the conversion of specific compounds.

In order to reconstruct the chemical reaction network of a single organism, or of a consortium of organisms, first the set of reactions has to be established by some approach. While mass spectrometry can be used to track metabolites and thereby to infer reactions, this approach is usually both cost and time prohibitive. Instead, readily available and cheaper genomic and transcriptomic data serves as a basis for reconstruction. While certainly not perfect, robust and complex pipelines for genomic annotation based on a bouquet of methods have been well established (Fig. 2). Explicitly, the set of proteins and non-coding transcript are predicted. Based on homology to structures in already studied organisms, a layer of functional annotation can be added. Reaction databases, equally grounded in past research, link this functional annotation to known reactions. The resulting set of reactions is often incomplete, owing to a myriad of error sources, including un- or undersequenced genomic regions or transcripts, sequencing errors, genes missed by the annotation pipeline, missed functional homologies, or reactions unique to the organism. Errors typically manifest as gaps in the CRN. Some degree of manual curation is commonly necessary to close such gaps, although both databases and annotation tools have steadily improved. Automatic reconstruction strategies, referred to as gap filling, may also be employed. Gap filling can be optimized towards various constraints and cost functions, ideally related to data derived from experiments (Fig. 3). A straight forward optimization can be implemented if a medium is known in which an organism can be successfully cultivated. The chemical composition of the growth medium is defined and the gaps of the CRN are filled such that the organism can grow in simulation. Annotation, conversion to a set of reactions and gap filling based on two different media each have already been performed for all SIHUMIx species and serve as the base input for this project.

To reduce the complexity, we limit analysis to the amino acid synthesis pathway for this project, i.e. the reactions necessary to transform glucose to any amino acid under addition of water and energy (Fig. 4). In particular, we want to model protein synthesis.

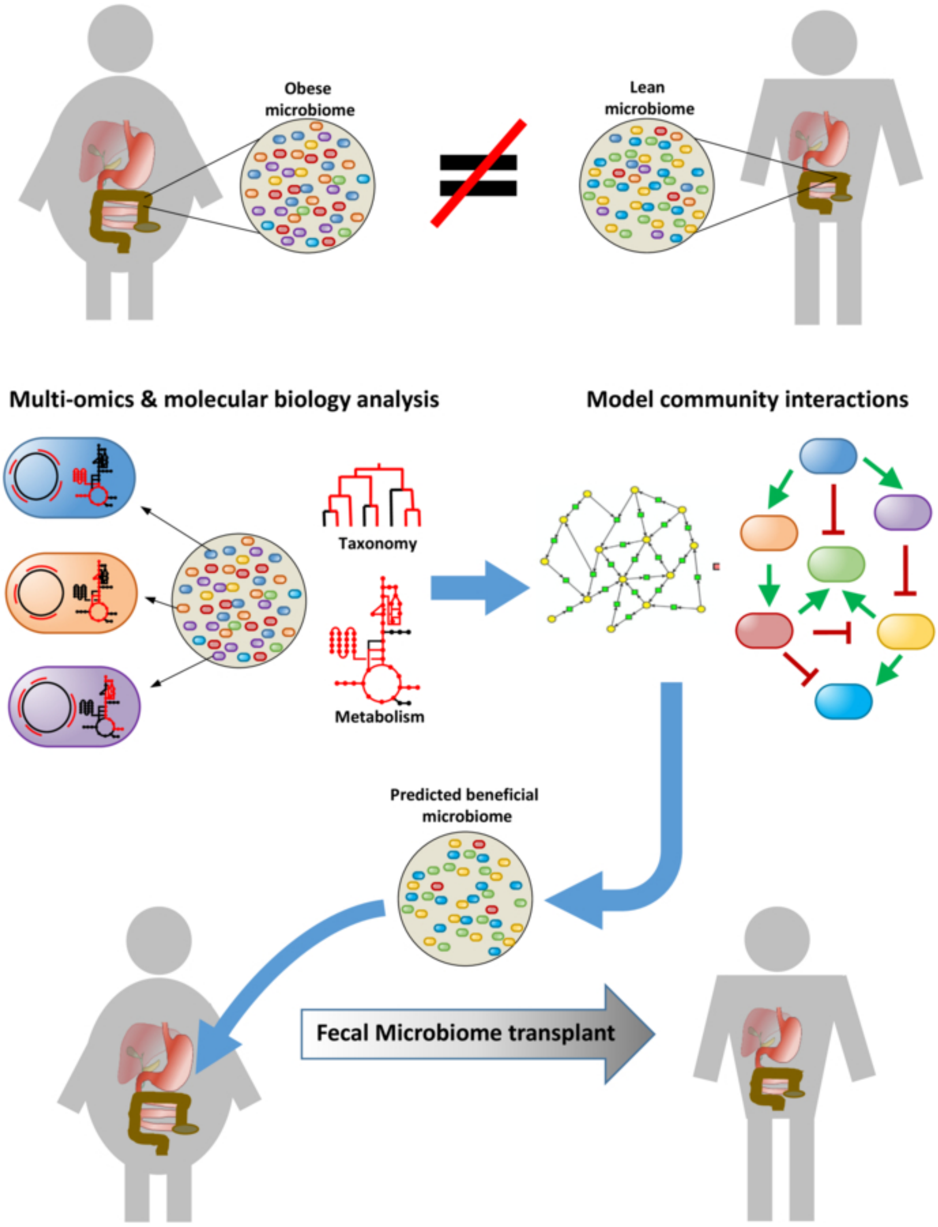

Obese microbiome
Lean microbiome
Multi-omics & molecular biology analysis
Model community interactions
Taxonomy
Metabolism
Predicted beneficial microbiome
Fecal Microbiome transplant

Figure 1: The differences in lean and obese microbioms inspires the metabolic modelling of model communities to guide the design of therapeutic microbioms.

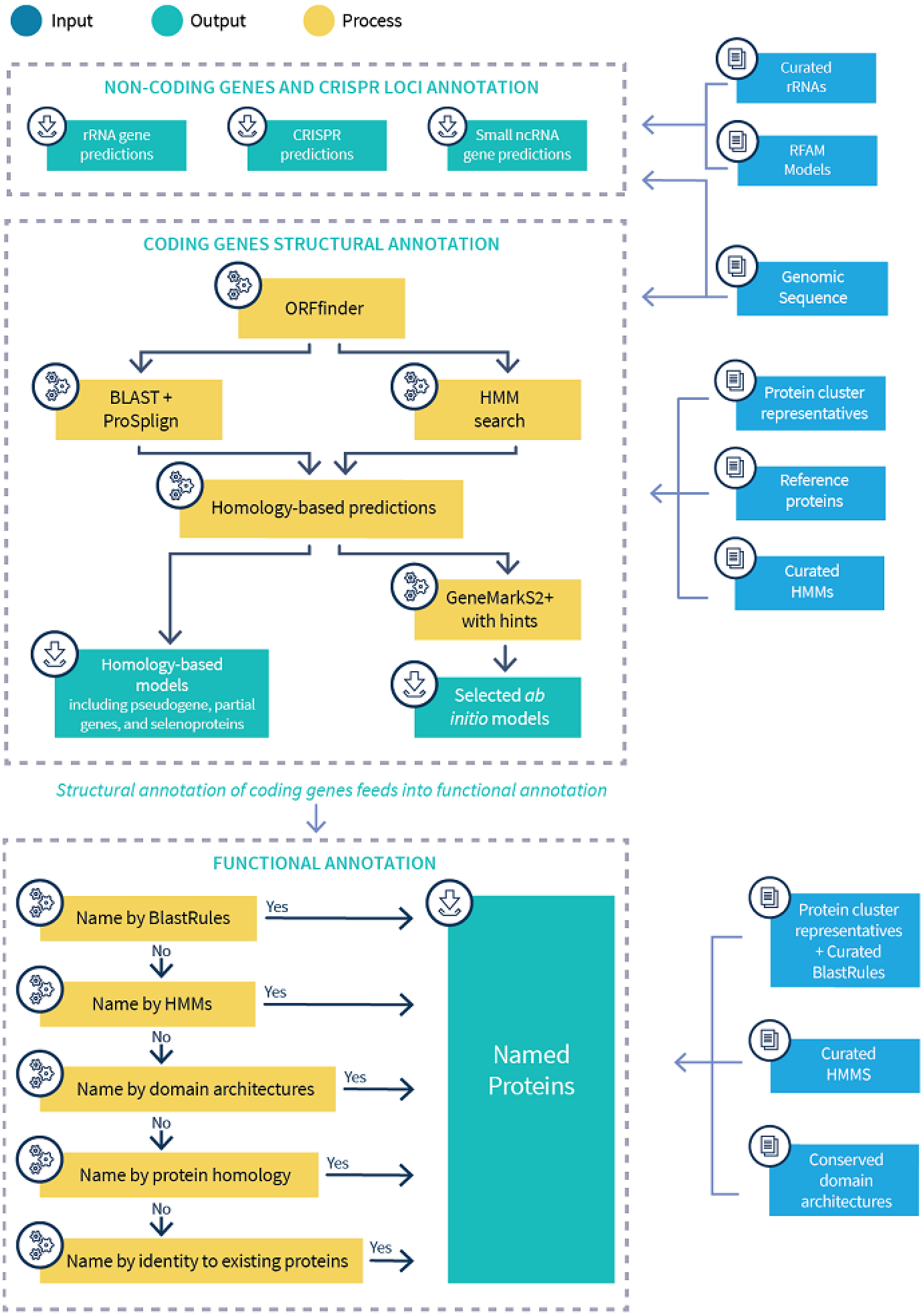

Input
Output
Process
NON-CODING GENES AND CRISPR LOCI ANNOTATION
rRNA gene predictions
CRISPR predictions
Small ncRNA gene predictions
Curated rRNAs
RFAM Models
Genomic Sequence
CODING GENES STRUCTURAL ANNOTATION
ORFfinder
BLAST + ProSplign
HMM search
Homology-based predictions
GeneMarkS2+ with hints
Homology-based models including pseudogene, partial genes, and selenoproteins
Selected ab initio models
Protein cluster representatives
Reference proteins
Curated HMMs
Structural annotation of coding genes feeds into functional annotation
FUNCTIONAL ANNOTATION
Name by BlastRules
Yes
No
Name by HMMs
Yes
No
Name by domain architectures
Yes
No
Name by protein homology
Yes
No
Name by identity to existing proteins
Yes
Named Proteins
Protein cluster representatives + Curated BlastRules
Curated HMMS
Conserved domain architectures

Figure 2: Overview of he PGAP Pipeline for genomic annotation of prokaryotes as designed and used by the NCBI genome consortium.

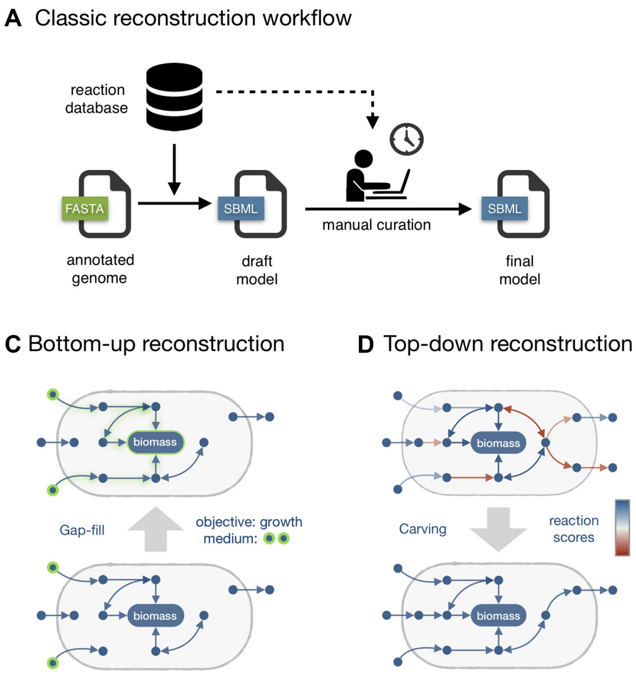


Figure 3: Genomic annotation is converted to a draft set of reactions. The remaining gaps in the system need to be filled by manual curation or through automated systems.

species	**phylum**	**function in the human gut**
Anaerostipes caccae (=Eubacterium entericum)	Firmicutes	short chain fatty acids producer
Bacteroides thetaiotaomicron	Bacteroidetes	mucosal barrier reinforcement immune system modulation nutrients metabolism
Bifidobacterium longum	Actinobacteria	produces acetate catabolism of oligosaccharides
Blautia producta	Firmicutes	glucose fermentation among the most abundant members
Clostridium butyricum	Firmicutes	short chain fatty acids producer dehydroxylation of bile acids
Clostridium ramosum	Firmicutes	conversion of bilirubin to urobilinogen dehydroxylation of bile acids
Escherichia coli	Proteobacteria	metabolism of high spectrum of glycoconjugates
Lactobacillus plantarum	Firmicutes	immunomodulation enhancement of the epithelial barrier functions

Table 1: List of all SIHUMIx species with basic properties.

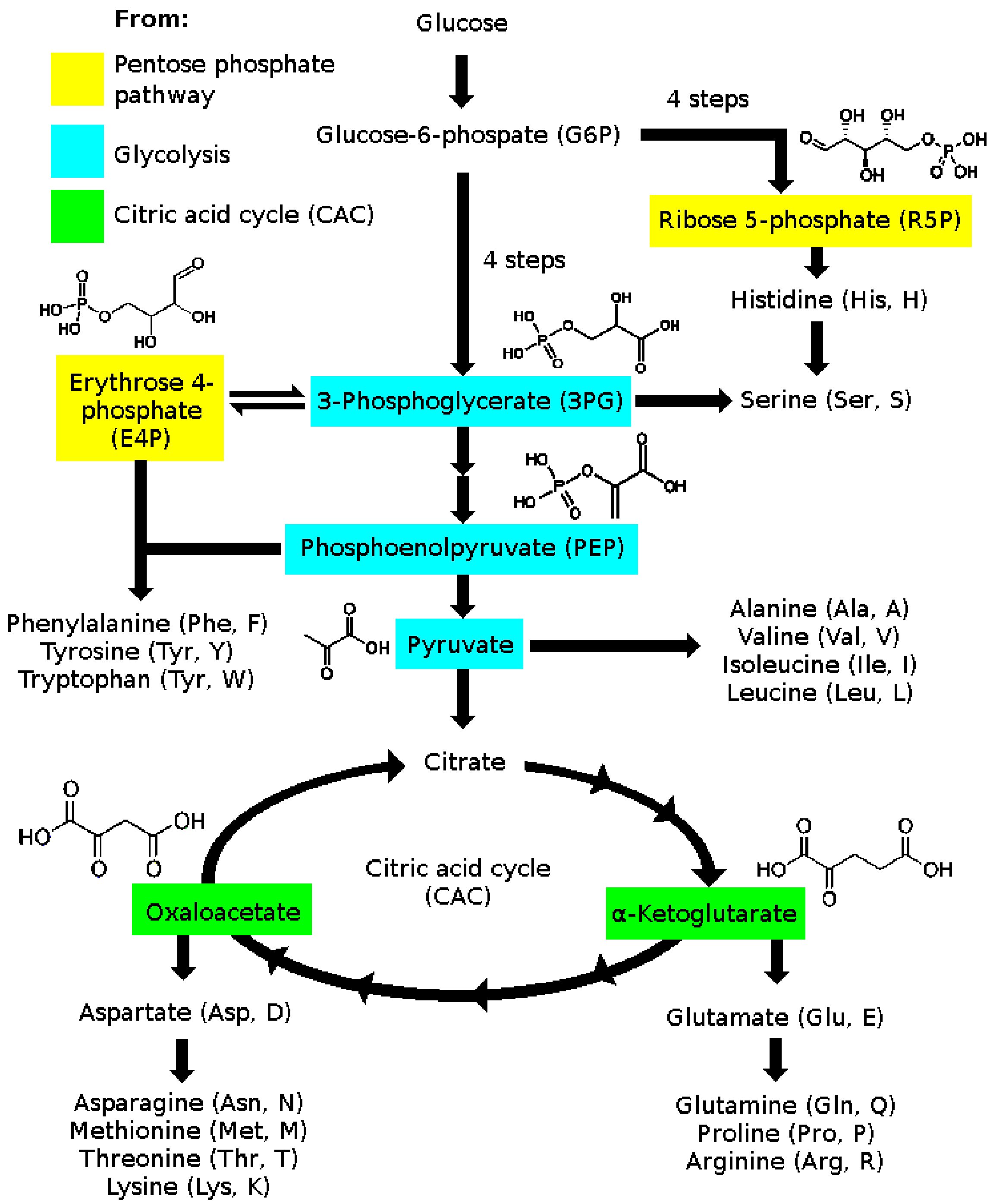


Figure 4: An overview of the amino acid biosynthesis pathway.

Chemical Reaction Networks

CRNs are commonly represented as directed hypergraphs. A directed hypergraph consists of a set of vertices $V \equiv C$, here equivalent to the set of chemical compounds, and a set of hyperedges H representing reactions. Each edge therefore does not just connect two nodes, but sets of nodes. Each edge $e \in H$ is defined as a tuple of (multi-)sets $e = (e^-, e^+)$ where $e^-, e^+ \subseteq V$. In CRNs we add a hyperedge (Y, Y') for each reaction. Notions such as paths can be naturally extended towards hypergraphs, e.g. a path is a sequence of hyperarcs connecting two nodes. In a CRN a path indicates the transformation of one compound into another via potential intermediary compounds. As each edge connects sets of nodes we may require that in order to continue a path with a hyperedge all incoming vertices need to have been previously visited. Otherwise themselves not visited incoming nodes of each hyperedge of the path reveal additional compounds necessary for the transformation. Likewise the set of themselves not visited outgoing nodes of each hyperedge define the set of additionally created compounds.

A CRN hypergraph may be represented as a common directed graph $G = (V, E)$ as follows: We add additional nodes for each reaction to the network $V = \{C \cup R\}$. Instead of hyperarcs we then add edges between all nodes of the the incoming set to the new reaction node and from the reaction to the outgoing node set $E = \{cr | \forall r \in R, \forall c \in e^- \text{ of } r\} \cup \{rc | \forall r \in R, \forall c \in e^+ \text{ of } r\}$. The following examples are based on this representation.

Chemically plausible CRNs must adhere to the laws of energy and mass conservation. More practically speaking, no cycles can exist where a product is transformed into itself without the loss of energy. No reactions can create or annihilate mass. As reaction databases sometimes abbreviate or even completely leave out chemically "uninteresting" parts of compounds, it may not always be directly apparent that the second condition holds. Nevertheless at steady state all mass that goes into the network must also "come out", i.e. is excreted or converted into biomass in a real organism.

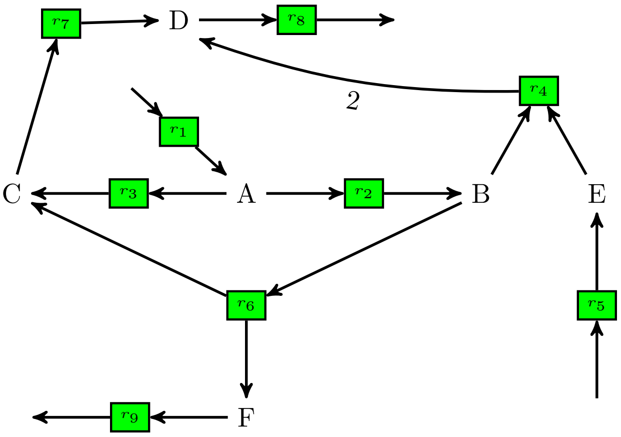
Flux Balances Analysis

Flux balance analysis is a mathematical approach for analyzing the flow of metabolites through a metabolic network. Every reaction is assigned a flux value, i.e. the rate of turnover of compounds through the reaction, based on application dependent constraints. For protein synthesis, constraints are set to mimic the exact production ratio of amino acid needed for specific (groups of) proteins. Linear programming is used to solve optimization problems where all the constraints, as well as the objective function, are linear equalities or inequalities. Methods for solving such linear systems have been extensively studied since even before the second world war. Although the problem is NP-hard for integer solutions, efficient solution for the general case in $\mathbb{R}$ exist. A number of ready made tools exists that we can use for this project without deeper concern on algorithmic details. Linear programming consist of 3 main components: 1) a vector of undetermined variables x, 2) a set of inequalities and equalities based on these variables, and 3) an optimization function. Commonly constraints are given in matrix and vector form for compact notation. However this notation is not intuitive for typical programming interfaces. Notably, the optimization does not need to include all variables. Depending on the set of inequalities, multiple or even infinite solutions may exist. In this case programs will report a single random solution by default. The set of reactions (or a thereon defined CRN) can be transformed to a linear program as follows:

- Extend the reaction set to include input and output reactions:
  - for every compound node c with in-degree zero in the CRN (i.e. it does not appear in any reaction as a product) add a reaction $\{\} \Rightarrow \{c\}$
  - for every compound node c with out-degre zero in the CRN (i.e. it does not appear in any reaction as an educt) add a reaction $\{c\} \Rightarrow \{\}$
- Create a variable r_i for each reaction in R. Those are the to be determined flux values for each reaction.
- As flux cannot be negative, we assume that $r_i \geq 0$.
- For each compound $c \in C$:
  - Let I be the set of all reactions containing c as educt ($c \in Y$)
  - and let O be the set of all reactions containing c as product ($c \in Y'$).

  - For the sake of notation note reactions as tuples (i, m) where i is the index of the associated flux variable and m is the multiplicity of c in the educt or product set.
  - Add $\sum_{i,m \in I} m \cdot r_i = \sum_{i,m \in O} m \cdot r_i$ as a constraint.
- Add further constraints and the optimization function based on the given task.

An Example: Objective function: $z = 0.75 \cdot r_8 + 0.5 \cdot r_9$, Constraints $r_1 = 10$, $r_3 >= 1$ and $r_5 = 6$ The FBA model consists of an **objective function** z, which shall be maximized, three **flux constraints** for r_1, r_3 and r_5 and six **flux balance equations** for each of the compound A – F.



```
     -> A     ; r1 = 10
     -> E     ; r5 = 6
   A -> C     ; r3 >= 1
   A -> B     ; r2
 E + B -> 2 * D ; r4
   B -> C + F ; r6
   C -> D     ; r7
   D ->       ; r8
   F ->       ; r9
```

$$
\begin{aligned}
&\max z = 0.75 \cdot r_8 + 0.5 \cdot r_9 \\
&\text{subject to:} \\
&\qquad r_1 = 10 \\
&\qquad r_3 \geq 1 \\
&\qquad r_5 = 6 \\[6pt]
&\qquad r_1 = r_2 + r_3 \\
&\qquad r_2 = r_4 + r_6 \\
&\qquad r_3 + r_6 = r_7 \\
&\qquad r_7 + 2 \cdot r_4 = r_8 \\
&\qquad r_5 = r_4 \\
&\qquad r_6 = r_9
\end{aligned}
$$

Atom Transition Networks

In a CRN all vertices themselves are molecules, which are not structure-less identifiers, but are (molecular) graphs themselves. Reactions thus are transformations of graphs that move bonds according to rules that, in a metabolic network, are determined by the enzymes that catalyse the reaction. In this regard, we are not only interested in the more coarse grained CRNs but also the transition of individual atoms in the reaction network. Modeling compounds and reactions on atomic details is essential for the design and interpretation of isotope labeling experiments, and enables a direct verification of compound fluxes using mass spectrometry.

As atom transitions are difficult to measure in practice, we have to resort to computational approximations. For each reaction a bijective mapping for each (non-hydrogen) atom of the educt to each atom in the product is computed, referred to as an atom-mapping. This problem is NP-hard. However, high quality heuristics exist. For this project we will rely on the tool **RXNMapper**, which uses machine learning to predict atom mappings.

In an atom transition network (ATN, see Fig. 5 for an example), each compound of the reaction set is added as a molecular graph, where (non-hydrogen) atoms act as nodes and edges correspond to the bonds between them. Further edges are added indicating the atom transitions in 2 categories as follows: i) the atom-mapping for every reaction is computed and edges are added for each pair of associated atoms. ii) Symmetries in metabolites may cause plausible alternative atom mappings, as the orientation of the molecule for the reaction

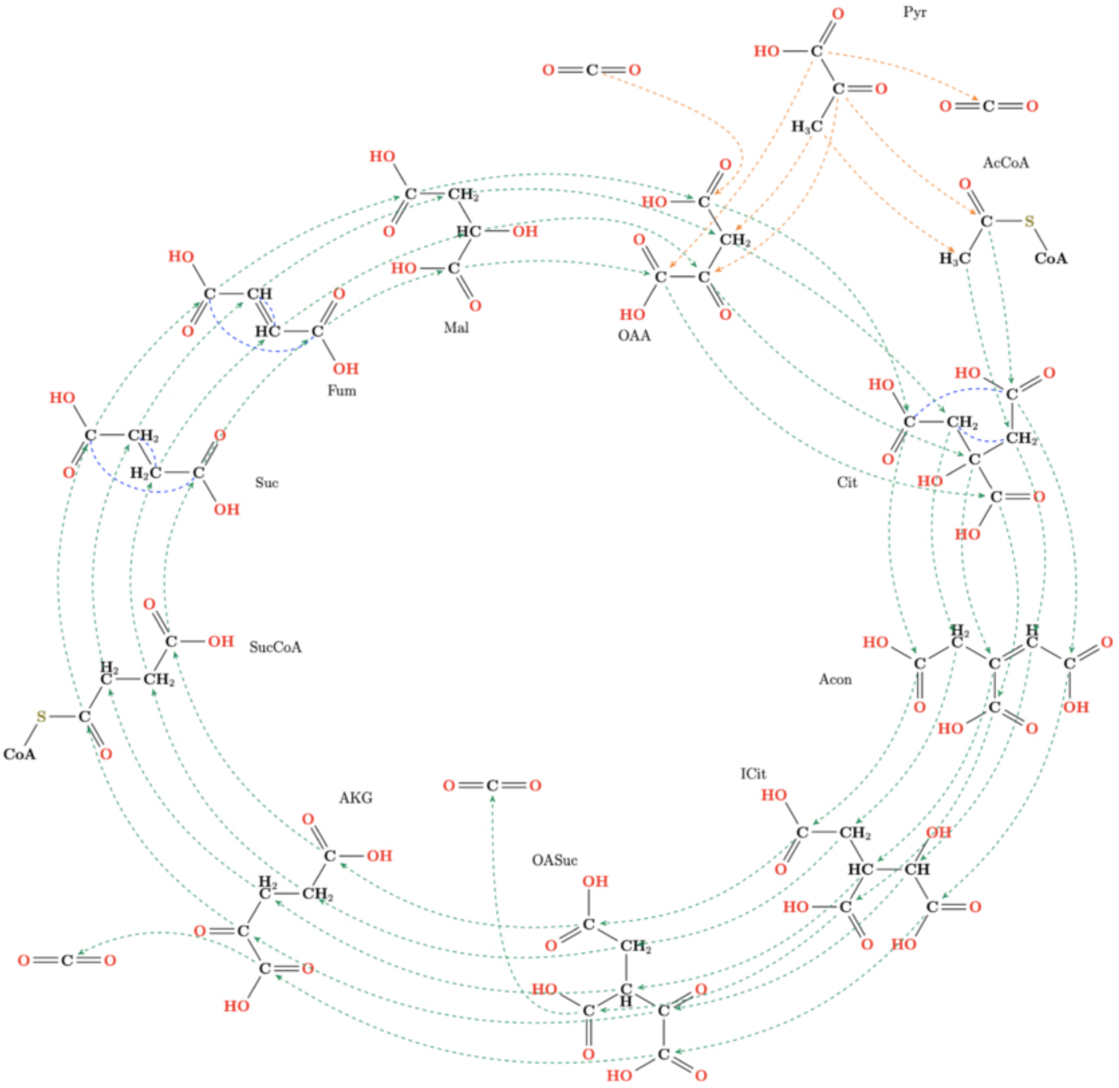


Figure 5: Example of an atom transition network for the citric acid cycle. Blue dotted lines indicate automorphisms of symmetrical molecules. Green and orange dotted lines define atom transitions as the occur in the transformation.

is random (Fig. 6). An automorphism test is used to detect such symmetries and transition edges are added between alternative atoms. The alternative transitions are thereby encoded as paths within the atoms of a single compound. This process has already been implemented in our workgroup.

SMILES

The simplified molecular-input line-entry system (SMILES) is a textbased specification system for describing the structure of chemical species. It is well suited for digital storage and processing, and can be converted into two-dimensional drawings or three-dimensional models of the molecule. There are several online tools available for visualizing SMILES, e.g. `http://www.cheminfo.org/Chemistry/Cheminformatics/Smiles/index.html`. Typically, hydrogen atoms are not explicitly added in the SMILES strings, but must be inferred through free valences.

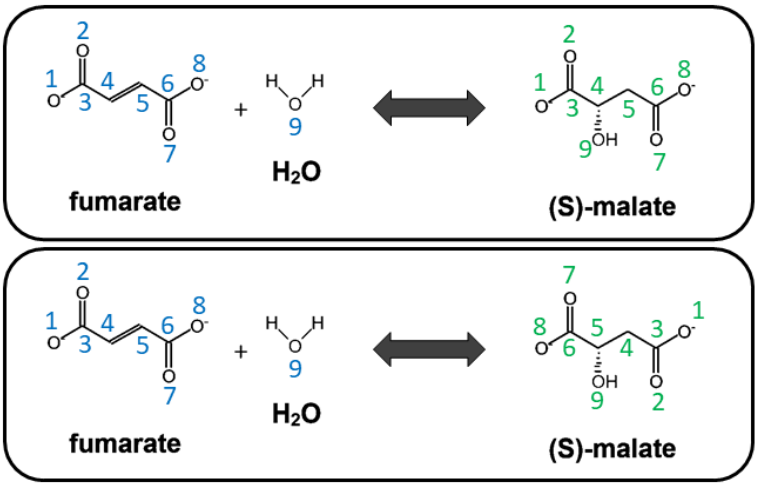


Figure 6: Possible atom-maps in the reaction of (S)-malate to fumarate and water. Due to the symmetrical property of fumarates, multiple valid atomic maps exist that occur by chance.

Tasks

A set of xml files in SMBL format containing the predicted reaction sets of all SIHUMIx species for 2 different cultivation media each serve as the main Input for this project. For your and our convenience we have created a small toolkit able to i) convert the reaction set to a more actionable format, ii) create atom-mappings, and iii) create an ATN. You will find the toolkits pre-installed on the lab computers or may download it from git on you personal PC: `https://github.com/TGatter/IsotopeMappingToolkit` As the computational capacities of the lab computers are farily limited, we have pre-computed all steps. Please download both XML files and pre-computed files here: `http://silo.bioinf.uni-leipzig.de/bioinf_lehre/sihumix_praktikum.tar.gz`.

Additionally you will need a Linear Programming Solver, e.g. PuLP (`https://www.gurobi.com/`) or Gurobi (`https://www.gurobi.com/`). You will find PuLP preinstalled on all lab computers.

The provided toolkit may be used as follows:

Script 01 converts the provided xml files into a custom format, including SMILES representations fo all reactions. The file is organized in blocks of 4 lines:

```
Bigg ID: [BIG] MetaNetXId: [MN] Reversible: [R=True/False]
ECs: [EC]
[r1: reaction with compound names]
[r2: reaction with SMILES]
```

where: [BIG] is the Id linking the reaction to the Bigg database (`http://bigg.ucsd.edu/`). [MN] is the Id in the MetaNetX database (`https://www.metanetx.org/`). [R] denotes wether the reactions can only be performed in the given direction (*False*) or can also be reversed (*True*), i.e. educts and products can be swapped. [EC] is the list of associated Enzyme Numbers relating to the KEGG PATHWAY Database (`https://www.genome.jp/kegg/`). Reaction line 1 [r1] indicates the reaction using chemical compound names as defined by MetaNetX. Educts and products are seperated by the character '='. Metabolites within educts or products are seperated by '+'. Reaction line 2 [r2] presents the same reaction in identical compound order, but using SMILES strings. Educts and products are seperated by the characters '¿¿'. Metabolites within educts or products are seperated by '.'. As a reminder, educts and products are multisets and metabolites may appear multiple times.

Run as:

```
./01_bigg_to_smiles_reactions.py [SMBL Xml] [SMILES]
```

Script 02 converts the output of script 01 to a include explicit hydrogens and atom maps. The output file is organized in blocks of 3 lines per reaction:

```
[r1: reaction with compound names]
```

```
[r2: reaction with SMILES inluding explicit hydrogens]
[r3: reaction with SMILES inluding atom maps]
```

Reaction line 1 [r1] is identical to the output of script 01. Reaction line 2 [r2] adds explicit hydrogens to the SMILES representation output of script 01. Reaction line 3 [r3] adds atom maps created by RXNMapper to the SMILES representation output of script 01. The metabolite order is identical for all reaction lines and also consistent to the output of script 01.

Run as:

```
./02_atommap_smiles_reactions.py [SMILES] [Mapped SMILES]
```

Script 03 converts the output of script 02 into an atom transition network. The network is exported in GML format, which can be parsed by most graph libraries such as NetworkX for python. The following properties for nodes are included:

- 'charge': The charge of this atom.
- 'hcount': The number of implicit hydrogens attached to this atom.
- 'aromatic': Whether this atom is part of an aromatic system.
- 'element': The element of this atom.
- 'compound_id': The internal id of the metabolite the atom belongs to in the network.

Edges derived from atom mapping are annotated by the list of reactions defining this edge. The field 'reaction_id' contains a string entry as a start element followed by one or more entriesfor each id of involved reactions.

Next to the graph, a csv file is produced with the list of compounds, including 3 fields.

```
[Compound ID] [Molecule Name] [SMILES String]
```

Reaction IDs in edges correspond to the index of the reaction in the input file containing the mapped reactions.

Run as:

```
./03_generate_ATN.py [Mapped SMILES] [ATN in GML format]
```

WP1: Extract Amino Acid Biosynthesis Pathways

First we seek to extract all reactions associated with amino acid biosynthesis for all reconstructed datasets.

Proceed as follows:

1. Use the script `01_bigg_to_smiles_reactions.py` to transform the XML file to an easier to read and parse format.
2. Parse reactions and create a hypergraph representation of them. Remember to encode the multiplicities of the multisets.
   - You may use the representation as a basic directed graph described above.
   - We recommend using the python package NetworkX.
3. Modify breadth-first traversal to enumerate the reachable subgraph for a metabolite, i.e. all metabolites that can be synthesized from it and the reactions for this transformation.
4. Compute the subgraph of reaction products derived from glucose S.
5. Compute the subgraphs of reaction educts leading to each amino acid $A = A_1, A_2, \dots A_n$. Use breadth-first search in reverse direction for this purpose
6. Extract all reactions in $S \cap \bigcup_{A_i \in A} A_i$ as the full Amino Acid Biosynthesis Pathway

Breadth-first traversal needs to be adapted for hypergraphs. We seek only products with glucose as the sole carbon source. Therefore, we block the addition of any compounds aside of the root compound for traversal. Hence, for traversal we can only follow edges if all nodes in the input set have been previously visited. As the only exception, we need to consider the constraints of energy conversion, as energy is brought into the system to catalyse reactions. Aside of the root compound, we therefore need to consider water and energy providing compounds to be available without limit. If the the graph is traversed in reverse order, their conversion products need to be considered available accordingly.

1. Create an empty queue Q
2. Mark the nodes for the root compound, energy providers, and other essential compounds as visited
3. Insert all hyperedges adjacent to only those nodes to Q
4. While Q is not empty:
   (a) Retrieve and deque the next hyperedge (Y, Y')
   (b) Mark the edge and all nodes Y' as visited
   (c) Enqueue all hyperedges that are connected to any node in Y' and for which all nodes of the in-set are marked as visited to Q
5. Extract the traversed subgraph as the set of all nodes and hyperedges marked as visited

After extraction of all amino acid biosynthesis pathways, we are interested in their basic statistics:

- How many/can all amino acids be synthesized?
- For the same organism, are there differences in the reconstruction based on the cultivation media?
- For the same medium, are there notable differences in the reconstructed pathways between species?
- Are there/How many alternative reaction paths exist to synthesize each amino acid?
  - Extract the subgraph induced by reverse traversal relating to each amino acid. Enumerate reaction paths.

  a) Approximate by the results by enumerating simples paths, i.e. only require that one incoming node hase been visited in order to continue a path.

  b) If we require that all incoming nodes have been visited as before, we enumerate disjoint sub-graphs rather than paths. Modify the subgraph traversal algorithm as follows:
     * Treat every added hyperedge as an alternative subgraph. Recursivly continue for each.
     * Regularly trim the results for identical subgraphs that were just traversed in a different order.

Compare results based on those statistics.

Hints:
Following this list of amino acid names based on the MetaNetX naming schema:

- L-arginine
- L-valine
- L-methionine
- L-glutamate
- L-glutamine
- L-tyrosine
- L-tryptophan
- L-proline

- L-cysteine
- L-histidine
- L-asparagine
- L-aspartate
- L-phenylalanine
- L-threonine
- L-lysine
- L-serine
- L-isoleucine
- glycine
- L-Leucine

List of energy provding compounds:

- AMP
- ATP
- ADP
- GDP
- NAD(+)
- NADH
- NADP(+)
- NADPH
- FAD
- FADH2
- UTP
- CTP
- heme b
- CoA
- FMN

List of further central compounds:

- H2O
- NH4(+)
- phosphate
- CO2

WP2: Metabolic Flux Analysis

Model the metabolic flux for the generated networks of WP1. You will be provided a list of proteins to synthesize. For each (set of) protein(s) find the amino acid composition. Let $C_{amino} = c_1^a, \ldots, c_n^a$ be the set of aminos acids in the network. Let be f_i be the fraction of the aminoacid with index i in the composition. Ensure that fractions sum up to 1 $\sum f_i = 1$.

We want to model the metabolic flux that is optimal for the creation of this composition using glycose as the only carbon source, i.e. how reactions are utilized in order to create each amino acid at the right fractions.

Here we assume that metabolites that are consumed have negative flux. Metabolites that are produced have positive flux. We restrict the flux to $0 \leq r_j < \infty$ for each directed reactions, and to $-\infty \leq r_j < \infty$ for reversible reactions. Accordingly we restrict every input reaction to also $-\infty \leq r_j < \infty$ and output reactions to $0 \leq r_j < \infty$.

As glycose is the limiting compound for this network we need to give its input reaction a tighter bound such as $-10 \leq r_{glyc} < \infty$

Please note that for praktical reasons we cannot use true inifinity in our programs. We instead use a fixed value chosen to be significantly larger than other limits in the network. Here you may choose $-\infty \rightarrow -1000$ and $\infty \rightarrow 1000$.

Although we want the flux to show the creation of aminoacids in the exact fractions described above, we cannot directly use them as an objective function. Instead, create a new compound b "biomass" and its output reaction. Add a new reaction $\sum\limits_{i:c_i^a \in C_{amino}} f_i c_i^a \rightarrow b$. Please note that this modifies all derived constraints in the Linear Program.

We then maximize the output of b.

WP3: Characterize Atomic Transition Network (ATN)

Use `02_atommap_smiles_reactions.py` and `03_generate_ATN.py` to generate the atom transition networks for each species and medium. We are interested in basic statistics of these networks:

- Number of components
- Components sizes
- Density
- Path/Cycle lengths
- ...

WP4: Stability and Characteristics on Manipulation

Find reactions that are not essential, i.e. all amino acids can still be produced without them. You may refer to alternative reaction path detected in WP1. Remove the reactions and repeat WP2 and WP3.

Remove random (sets of) reactions. Remove the reactions and repeat WP2 and WP3. Compare results.