



Automated metabolic modelling

Building, analysing and simulating
genome-scale metabolic models in
Python

Carolin Brune & Gwendolyn O. Döbel
22.09.2025

Schafft Wissen. Seit 1502.



MARTIN-LUTHER-UNIVERSITÄT
HALLE-WITTENBERG

Table of Content

Time	Content	Subparts
9:00 - 10:00h	I Introduction, Background & Basic functions	0 Introduction 1 Working with GEMs 1. Available Software 2. Basics → Notebook 01
10:00h - 11:00h	II Advanced Functions	3. Advanced Usages → Notebook 02
11:00h - 11:30h		Coffee break
11:30h - 12:30h	III Workflows	2 Piecing it together: Workflows → Notebook 03



0 Introduction

An Introduction to Systems Biology

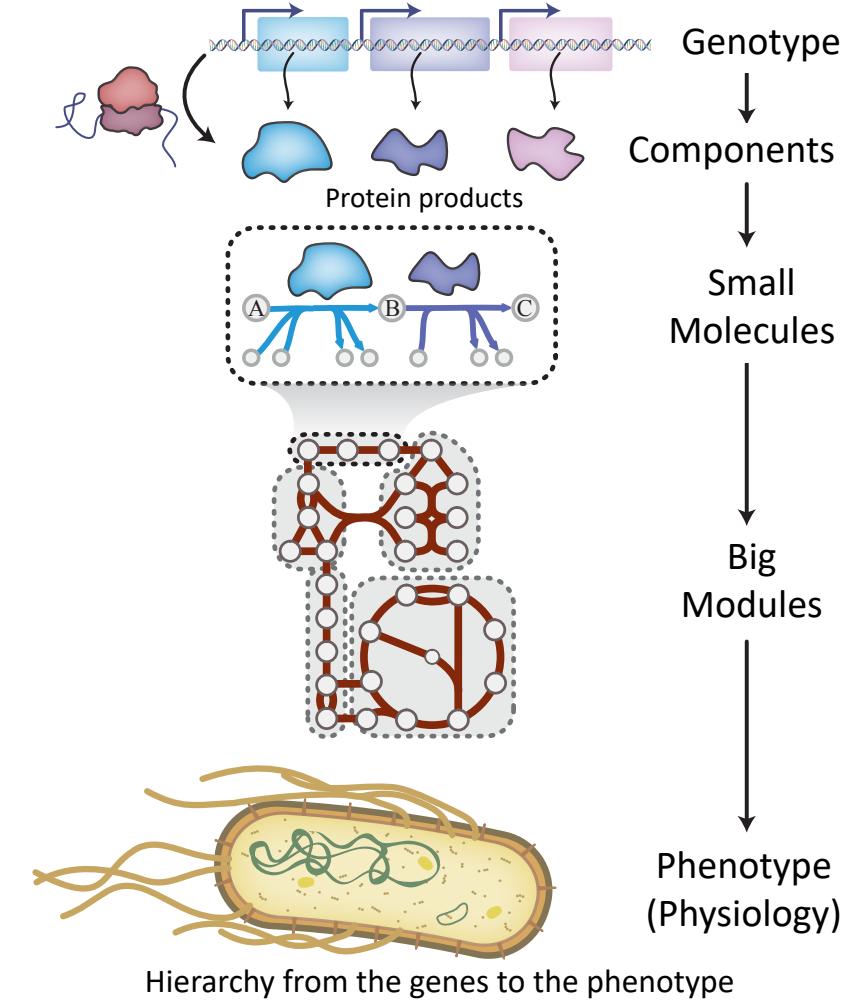
Prediction of the phenotype on the basis of the genotype

An attempt at a definition (there are many):

- Computer-based mathematical modelling of complex biological systems
- An interdisciplinary scientific discipline focusing on complex interactions in biological systems
- Integrative, Utilisation of reductionist insights

Aim:

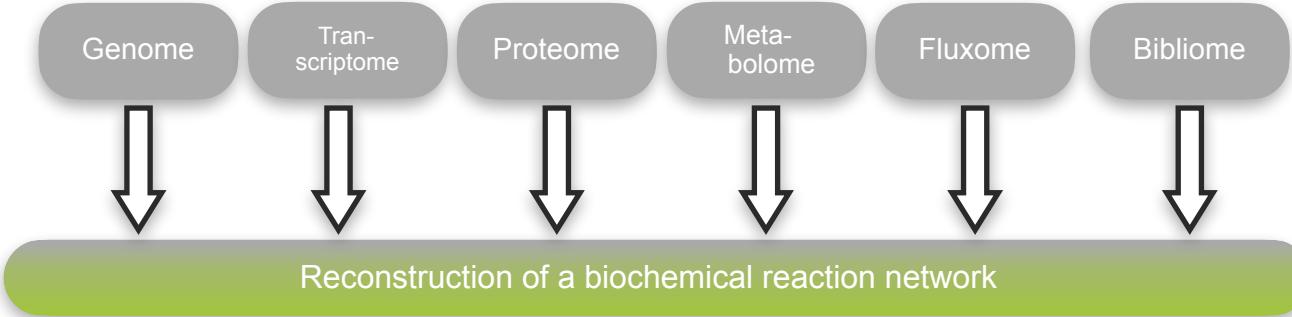
Understanding the characteristics and functionalities of the system as a whole beyond the detailed knowledge about its components



An Introduction to Systems Biology

A brief overview

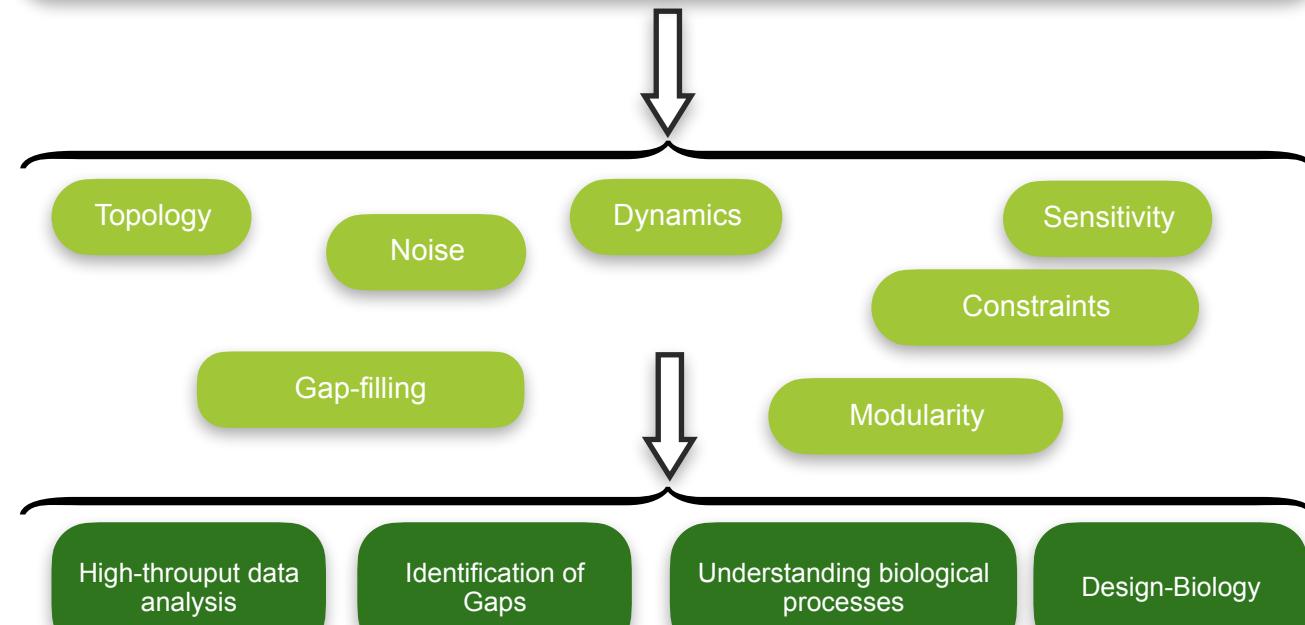
1. Databases:
Diversity of OMICS-Data



2. Knowledge base:
Set of reactions encoded
on the genome

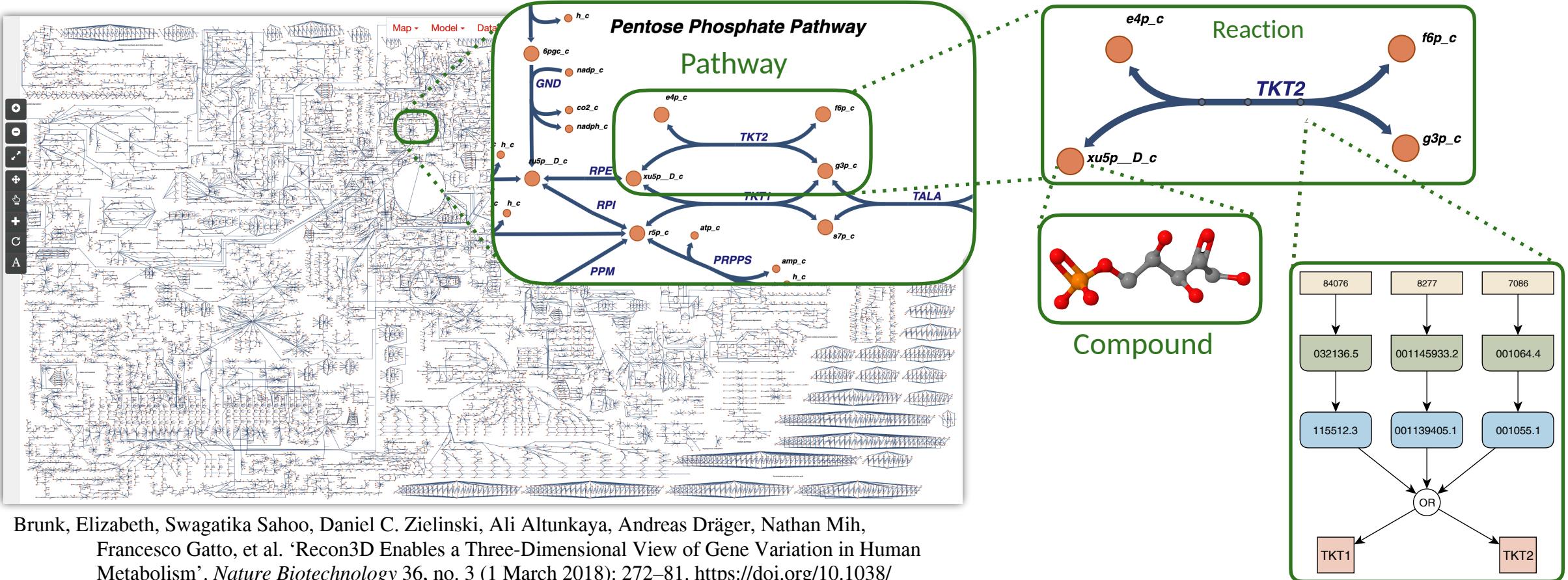
3. Methods of *in silico*
modelling

4. Validation, analysis
and application



```
<?xml version="1.0" encoding="UTF-8"?>
<sbml xmlns="http://www.sbml.org/sbml/level3/version1/core" xmlns:fbc="http://biomodels.net/biology-qualifiers/" xmlns:vCard="http://www.w3.org/2001/vcard-rdf/3.0#" xmlns:vCard4="http://biomodels.net/biology-qualifiers/" xmlns:bqmodel="http://biomodels.net/mode...>
<model metaid="meta_Kp_std" id="Kp_std" name="Genome-scale metabolic model">
  <notes>
    <html xmlns="http://www.w3.org/1999/xhtml">
      <p>Description: This model was built with CarveMe version 1.5.1</p>
    </html>
  </notes>
  <listOfUnitDefinitions>
    <unitDefinition id="mmol_per_gDW_per_hr">
      <listOfUnits>
        <unit kind="mole" exponent="1" scale="-3" multiplier="1"/>
        <unit kind="gram" exponent="-1" scale="0" multiplier="1"/>
        <unit kind="second" exponent="-1" scale="0" multiplier="3600"/>
      </listOfUnits>
    </unitDefinition>
  </listOfUnitDefinitions>
  <listOfCompartments>
    <compartment id="c" name="cytosol" constant="true"/>
    <compartment id="p" name="periplasm" constant="true"/>
    <compartment id="e" name="extracellular space" constant="true"/>
  </listOfCompartments>
  <listOfSpecies>
    <species metaid="meta_M_10fthf_c" sboTerm="SB0:0000247" id="M_10fthf_c" hasOnlySubstanceUnits="false" boundaryCondition="false" constant="false">
      <notes>
        <html xmlns="http://www.w3.org/1999/xhtml">
          <p>FORMULA: C20H21N7O7</p>
          <p>CHARGE: -2</p>
          <p>SBOTerm: SB0:0000247</p>
          <p>creation: via template</p>
        </html>
      </notes>
      <annotation>
        <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#>
```

What are genome-scale metabolic models (GEMs)?

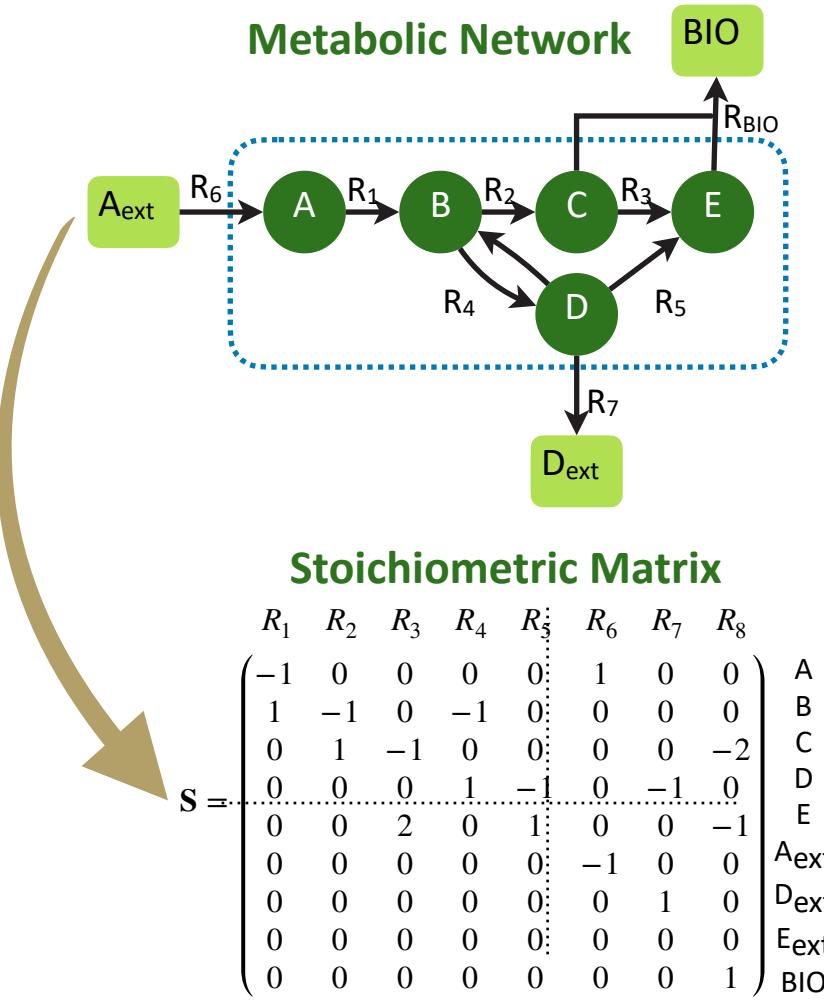


Brunk, Elizabeth, Swagatika Sahoo, Daniel C. Zielinski, Ali Altunkaya, Andreas Dräger, Nathan Mih,
Francesco Gatto, et al. 'Recon3D Enables a Three-Dimensional View of Gene Variation in Human
Metabolism'. *Nature Biotechnology* 36, no. 3 (1 March 2018): 272–81. <https://doi.org/10.1038/nbt.4072>.

Recon3D

The quantitative backbone

Adapted from: Orth, J. D., Thiele, I., & Palsson, B. Ø. (2010). What is flux balance analysis?. *Nature biotechnology*, 28(3), 245-248.



- Internal calculations based on Dynamic Mass Balance Equation

$$\frac{d\vec{x}}{dt} = \mathbf{S} \cdot \vec{v}$$

- Common analysis: Flux Balance Analysis (FBA)

$$\text{Maximise } Z = \vec{c}^T \cdot \vec{v}$$

Subject to:

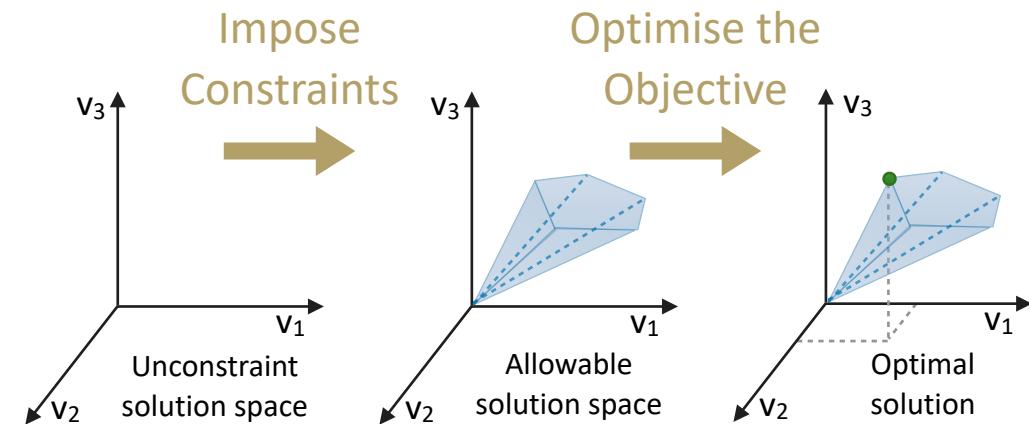
$$\mathbf{S} \cdot \vec{v} = \vec{0}$$

$$\forall r \in I : v_r \geq 0$$

$$v_{min} \leq v_r \leq v_{max} \quad \forall r \in \{1, \dots, n\}$$

- Assumption for calculations:
Steady State

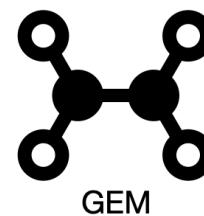
$$\mathbf{S} \cdot \vec{v} = \vec{0}$$



What can we do with GEMs?

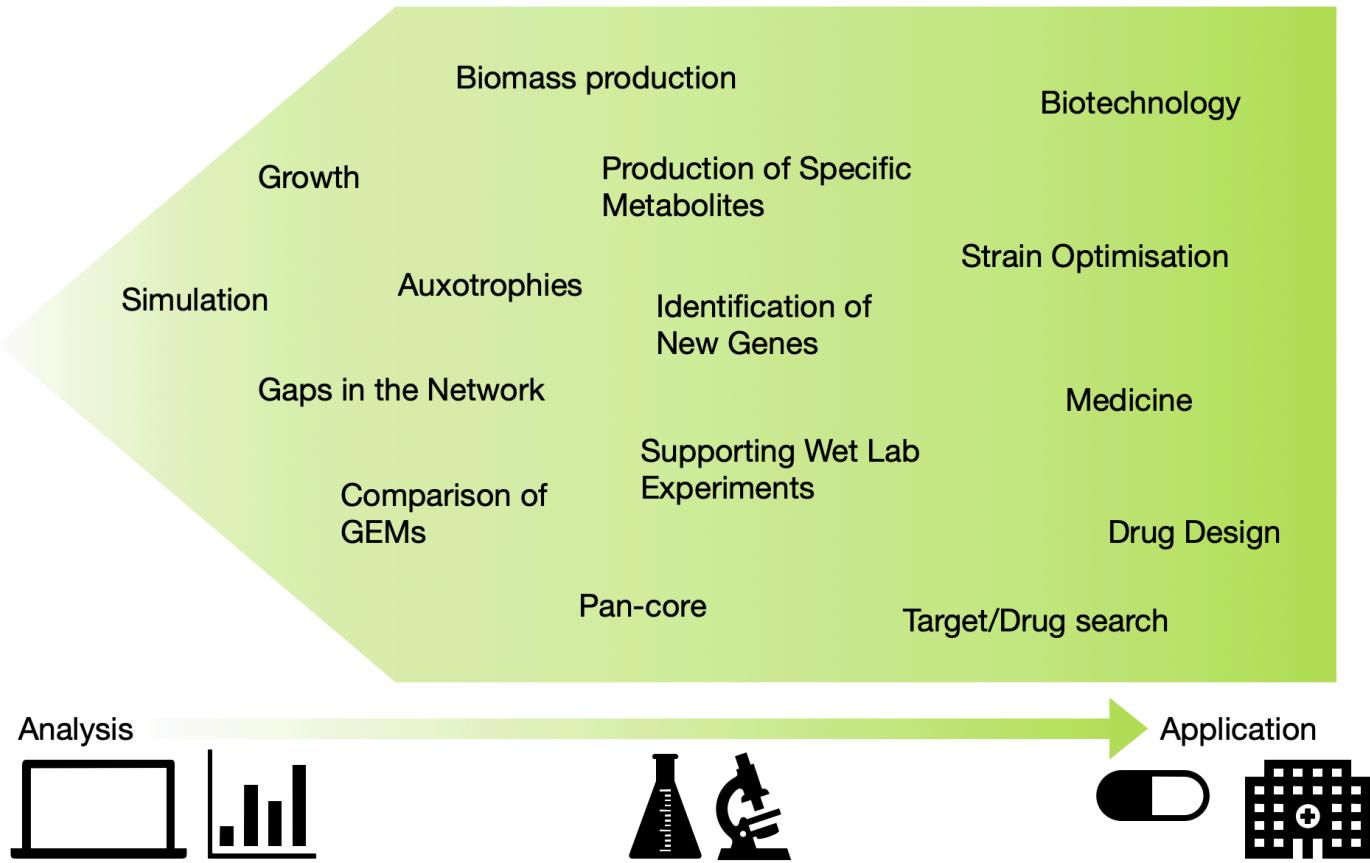
GEMs can be used for numerous analyses ...

- Growth simulation
- Minimal growth requirements
- Optimisation towards different objectives
- Flux balance analysis
- Flux variability analysis
- Effects of knockouts
- Auxotrophies
- Source Tests



... in different scenarios

- Microbial Engineering
- Drug Discovery
- Host-Pathogen Interactions
- Microbial Communities





Use case: *Methanothermobacter* and renewable energy

Problem:

Fluctuating availability of renewable energy and limited storage capacity of batteries

Bavarian Start-Up



Idea:

Energy storage in chemical form as biogas (methane, CH₄) generated from CO₂

But:

The chemical formation of CH₄ via the Sabatier process is expensive and energy-intensive (high temperature and pressure, metal catalysts)



Solution:

Use of optimised methanogenic archaea. Candidate: *Methanothermobacter*: 65 °C, Standard pressure, low costs, low-salt media, 2-3 h doubling time

iScience 26:10, 2023, doi: 10.1016/j.isci.2023.108016

Energy-to-Gas-Technology



Use case: Phenformin and Atpenin A5 as broad-spectrum antiviral drugs

1. *In silico* generation of a genome-scale metabolic model using the viral components and tissue-specific human model

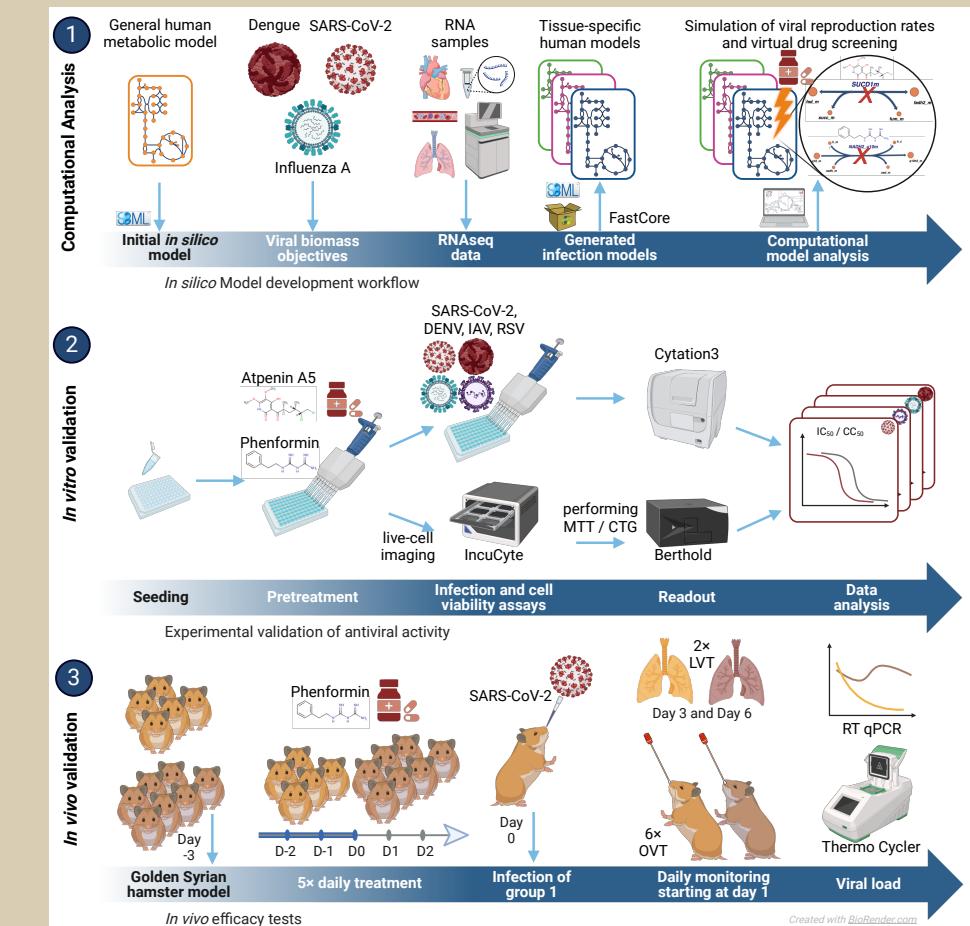
→ Virtual drug screening

2. *In vitro* validation using viral infection assays and multiple human cancer cell lines

→ Broad *in vitro* antiviral activity against all tested viruses for Atpenin A5

3. *In vivo* using hamster models of infection

→ *In vivo* antiviral effect on SARS-CoV-2 for Phenformin



Renz et al., *Communications Biology* volume 8, Article number: 791 (2025)

1 Working with GEMs

Working with GEMs

Available Software

Libraries

- COBRA Toolbox (MatLab)
- COBRApy (Python)
- libSBML (Python API)
- **refineGEMs (Python)**

Reconstruction tools (examples)

- Out-of-the-box reconstruction
 - Architect
 - CarveMe
 - **SPECIMEN-CMPB**
- Template-based reconstruction
 - Bactabolize
 - **SPECIMEN-HQTB**

Curation & analysis tools for specific steps

- SBO term annotation (SBOannotator, Python)
- Mass Charge Curation (MCC, Python)
- Semantic validity & annotation enhancement (ModelPolisher, Java + Python API)
- Biomass objective function improvement (BOFdat, Python)
- Model quality assessment suite (MEMOTE, Python)

Problem:

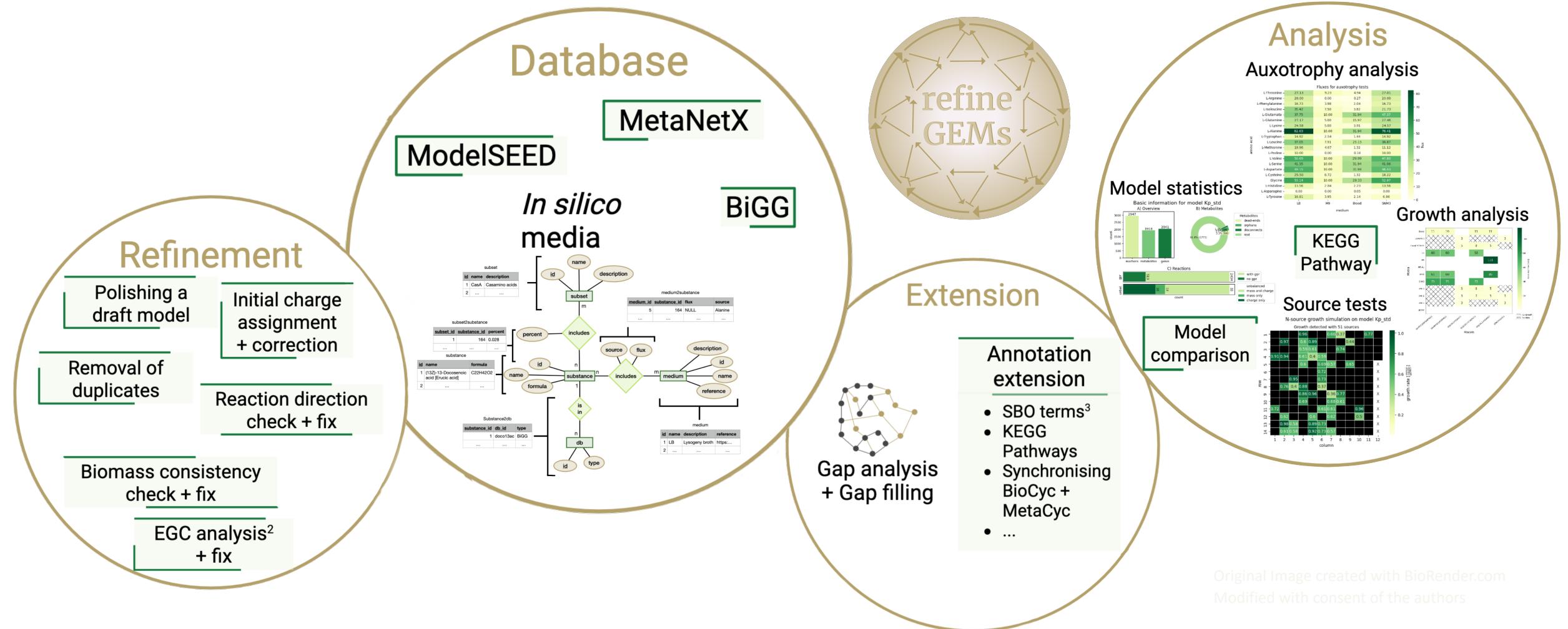
Multiple library and curation packages, multiple databases and different file formats complicate the handling of models.

- ➡ refineGEMs provides generalised functions, reducing the burden of the user, while retaining versatility and the ability to connect to other tools
- ➡ We will present a number of these functionalities in the upcoming slides

Working with GEMs



Toolbox refineGEMs: Overview



Working with GEMs - Basics

Loading, Saving



Loading/writing models

- Four widely used formats: XML/SBML, mat, JSON, YAML
- Function for each in COBRApy; libSBML can only handle XML/SBML
- refineGEMs is able to handle all combinations:

Code

```
from refinegems.utility.io import load_model, write_model_to_file

model = load_model(path, package='cobra')

write_model_to_file(model,
                    filepath='some_name.xml')
```

Working with GEMs - Basics

Building



Building reactions and metabolites from databases

- Usually, model entities are built based on package-specific setup (much manual input needed)
- refineGEMs provides functions for building these (mostly automatic) from database IDs/strings/entries
 - Based on COBRApy
 - MetaNetX, KEGG and BiGG available
 - Either the reaction string or a database ID can be used for reconstruction (or both)

Code Example for building a reaction entity from KEGG

```
from refinegems.utility.entities import build_reaction_kegg

build_reaction_kegg(model, id = "R00134")
# or
build_reaction_kegg(model, reac_str = "C00058 + C00006 <=> C00011 + C00005 + C00080")
```

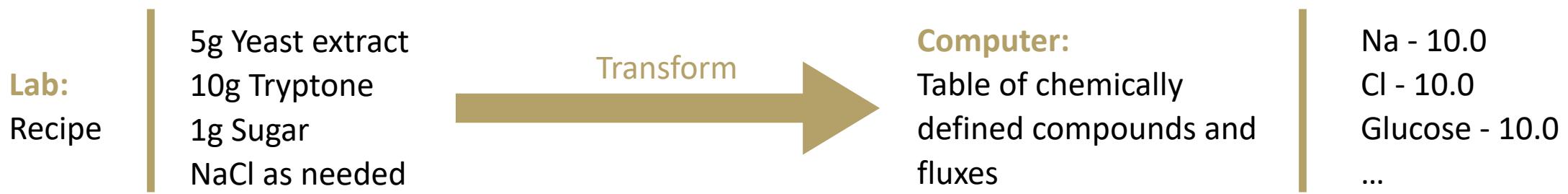
Working with GEMs - Basics



Growth analysis: *In silico* media

- “Nothing happens in isolation”
- Organisms take up nutrients from their environment
- ▶ The medium needs to be modelled as well

From *in vivo/in vitro* to *in silico*:



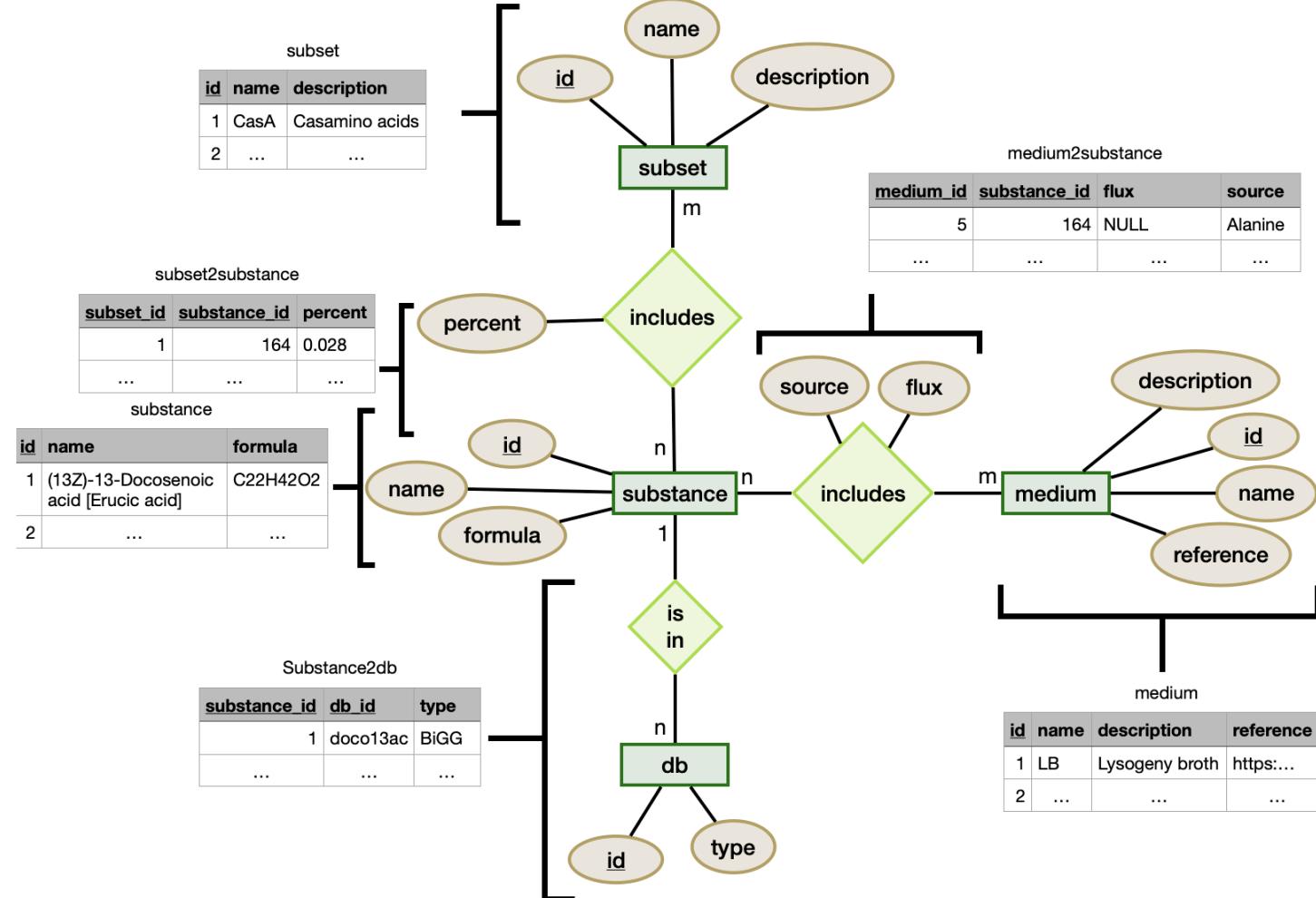
Implementation in refineGEMs:

- *In silico* media database
- Class/functionality for creating, manipulating and using the media for better and more reproducible model simulations (growth simulations)

Working with GEMs - Basics



Growth analysis: *In silico* media - database setup



Working with GEMs - Basics



Growth analysis: *In silico* media - an example

The loaded media objects can be further manipulated:

- Addition of media, subsets and substances
- Removing substances
- Changing fluxes
- ➡ Methods of the medium class

Code Creating media with refineGEMs

```
from refinegems.classes.medium import *

# Empty medium
m_emp = Medium(name)

# Load medium directly from the database
m = load_medium_from_db(name='LB')

# Load media configuration file
media_list, suppl_list = load_media(config)
```

Small example for a
media configuration file (YAML)

```
Code params:
    aerobic: True
    supplement: None
media:
    LB:
        aerobic: False
    LB_o2:
        base: LB
        default_flux: 5.0
        o2_percent: 2.0
    SNM3:
    M9:
        add_substance:
            Glycine: Null
            Glycerol: 0.87
            Guanosine: '20%'
```

Working with GEMs - Basics



Growth analysis

Growth simulation:

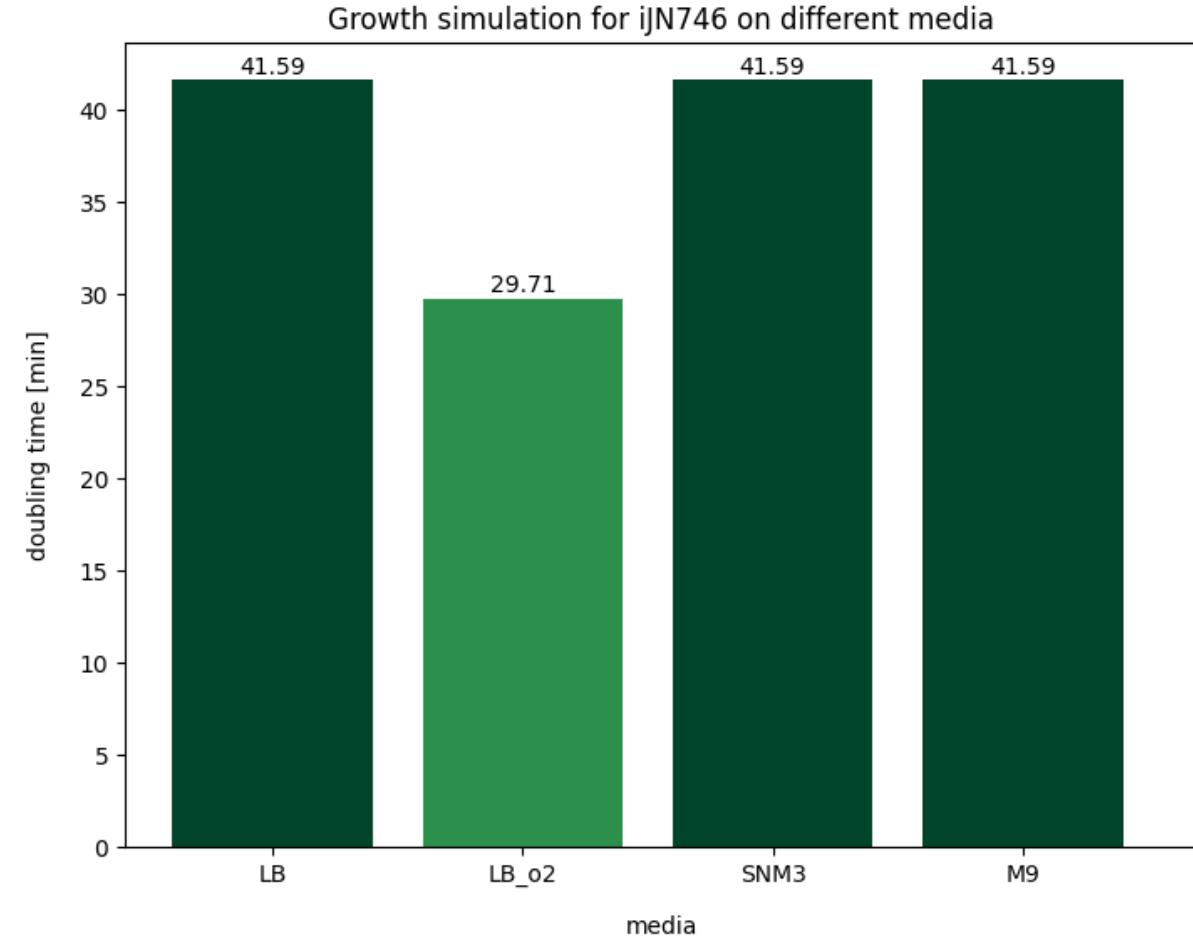
Is the model able to grow on a number of *in silico* media?

Code

```
from refinegems.analysis.growth import growth_analysis

report = growth_analysis(models, media,
                         retrieve="report")

report.visualise()
```





Notebook I

Working with GEMs - Advanced Usages



Amino Acid Auxotrophies

Amino acids auxotrophy test:

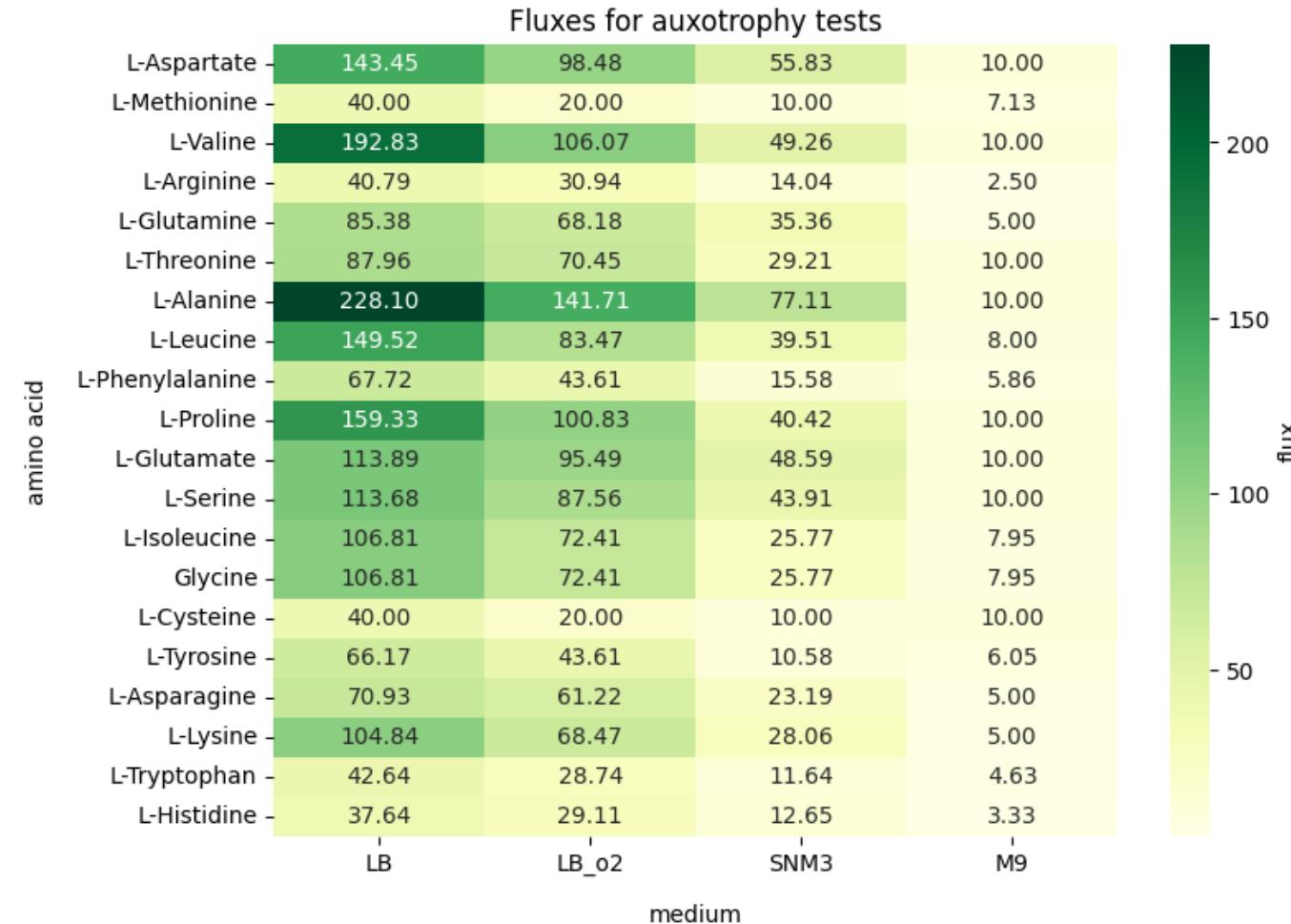
Given a specific medium, which (proteinogenic) amino acids can be produced, when it's not in the medium?

Code

```
from refinegems.analysis.growth import
test_auxotrophies

report = test_auxotrophies(model, media_list,
                           supplement_list)

fig = report.visualise()
```



Working with GEMs - Advanced Usages



Source Tests

Source test:

Which sources for a specific chemical element (here: nitrogen, N) can be utilised by the organism, if its the only one in the medium?

Code

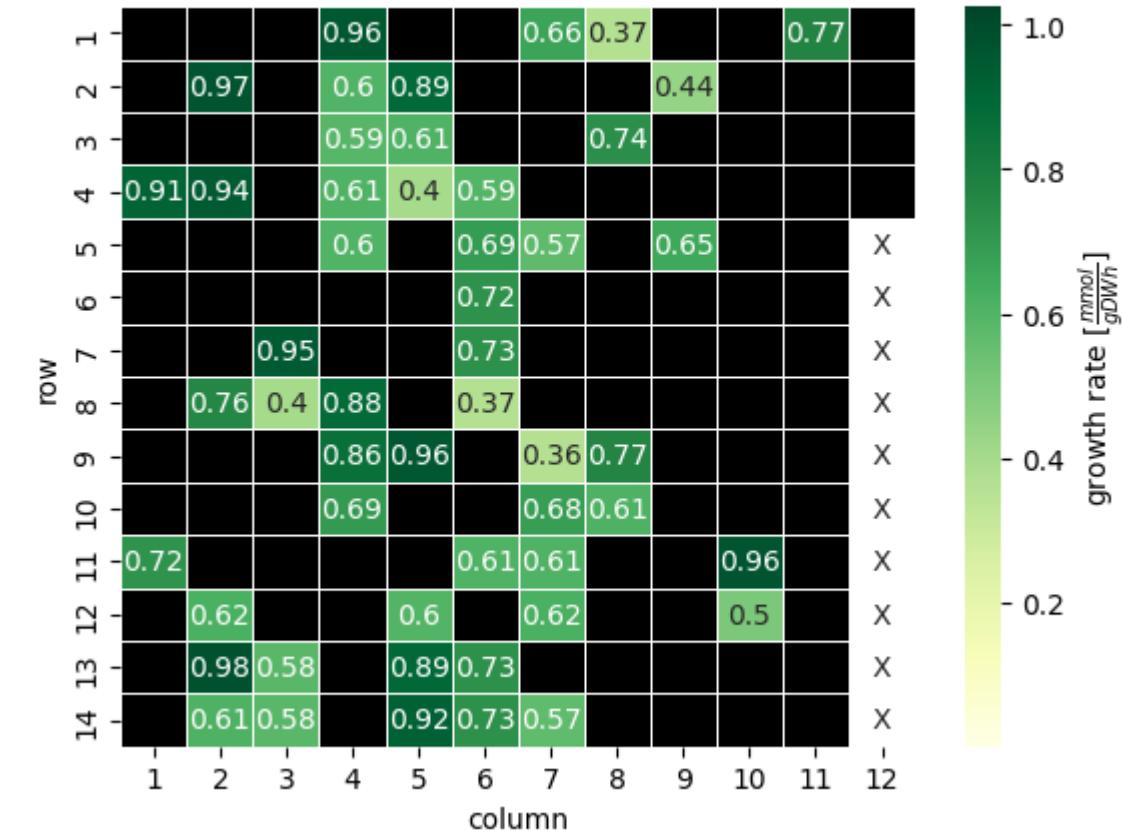
```
from refinegems.analysis.growth import
test_growth_with_source

report = test_growth_with_source(model, element = "N",
                                 Medium)

fig = report.visualise()
```

N-source growth simulation on model Kp_std

Growth detected with 51 sources



Working with GEMs - Advanced Usages



Handling duplicates in the model

Problem: Duplicates

- Can lead to broken pathways, dead ends, orphans, etc.
- Ideally should be identified and resolved, as to not influence growth and metabolite production / consumption
- Question: How to identify duplicates? ➔ Multiple options
 - ID ➔ depends on the namespace
 - Name ➔ trivial, high amount of variability
 - Annotation ➔ best way, as that is what they are for

Code

```
from refinegems.utility.io import load_model
from refinegems.curation.curate import resolve_duplicates

model = load_model(path, package='cobra')

model = resolve_duplicates(model, check_reac = True, check_meta = "default",
                           replace_dupl_meta = True, remove_unused_meta = True,
                           remove_dupl_reac = True)
```

Working with GEMs - Advanced Usages

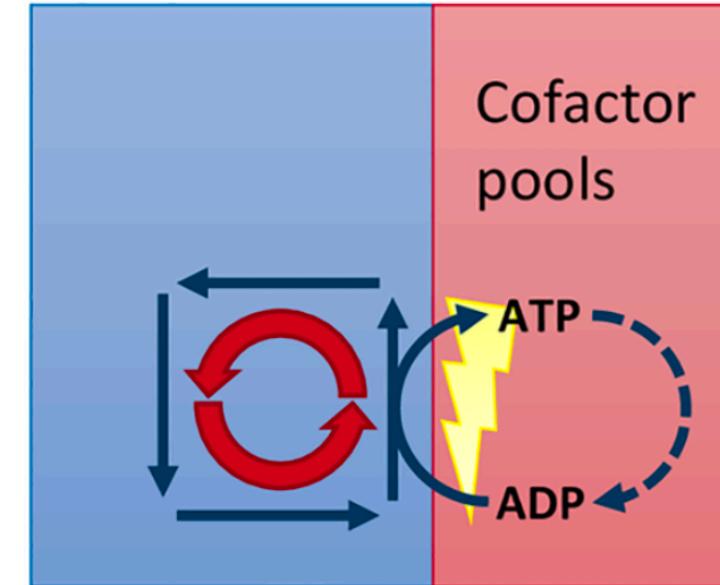


Handling EGCs

Problem:

Energy-generating cycles (EGCs) are cycles in the model, where energy metabolites are produced, even when no metabolites are taken up by the model

- ➡ Thermodynamically infeasible
- ➡ Can lead to wrong simulations
- ➡ Need to be removed!



Toy example EGCs (Fritzemeier et. Al, 2017)

Working with GEMs - Advanced Usages



Handling EGCS

Finding EGCS

Easy task:

Algorithm:

- For each type of energy metabolite,
 - Create a sink reaction,
 - Set as the objective function
 - Close all exchanges with the environment
 - Simulate growth

➡ If there is still a flow through the objective, there is an EGC for that energy metabolite

Code

```
from refinegems.classes.egcs import EGCSolver

egcsolve = EGCSolver()
results = egcsolve.find_egcs(model,
                             with_reacs=True,
                             compartment=[“c”, “e”])
```

Working with GEMs - Advanced Usages



Gap-Filling

Problem: Gaps

- Idea implemented in refineGEMs:

Gap-filling via gene content

- Depends on gene content in GFF file or databases (e.g. KEGG, BioCyc)
- Algorithm:
 - Input:** model, KEGG Organism ID (optional), GFF file (optional), tables from BioCyc (optional), FASTA
 - Output:** gap-filled model, statistics, tables for manual curation
 - Idea:** identify missing genes, map to EC number / reaction and try to construct model entities accordingly

Code Filling gaps via KEGG

```
from refinegems.utility.io import load_model, write_model_to_file
from refinegems.classes.gapfill import KEGGGapFiller

# Load model with COBRApy & libSBML
m = load_model(modelpath, 'libsbml')
cm = load_model(modelpath, 'cobra')
# Initialise GapFiller
gfk = KEGGGapFiller(kegg_organism_id)
# Find missing genes
gfk.find_missing_genes(m)
# Find missing reactions
gfk.find_missing_reactions(cm)
# Fill gaps
m = gfk.fill_model(m)
# Get statistics & files for manual curation
gfk.report(output_dir)
```

Working with GEMs - Advanced Usages



And so much more ...

refineGEMs provides further functionalities from different areas of genome-scale metabolic modelling:

Curation

- Annotations (including MIRIAM compliance)
- Biomass
- Charges
- KEGG pathways
- Compartments

Analyses & Investigation

- Pan-core
- Pathways
- Minimal medium
- Multiple statistics reports
- Model comparison

Connections & Databases

- MassChargeCuration
- SBOannotator
- BOFdat
- Functionalities for accessing databases, e.g. MetaNetX, BiGG or ChEBI

... to name just a few



Notebook II

Part III Workflows



SPECIMEN

Piecing it together: Workflows

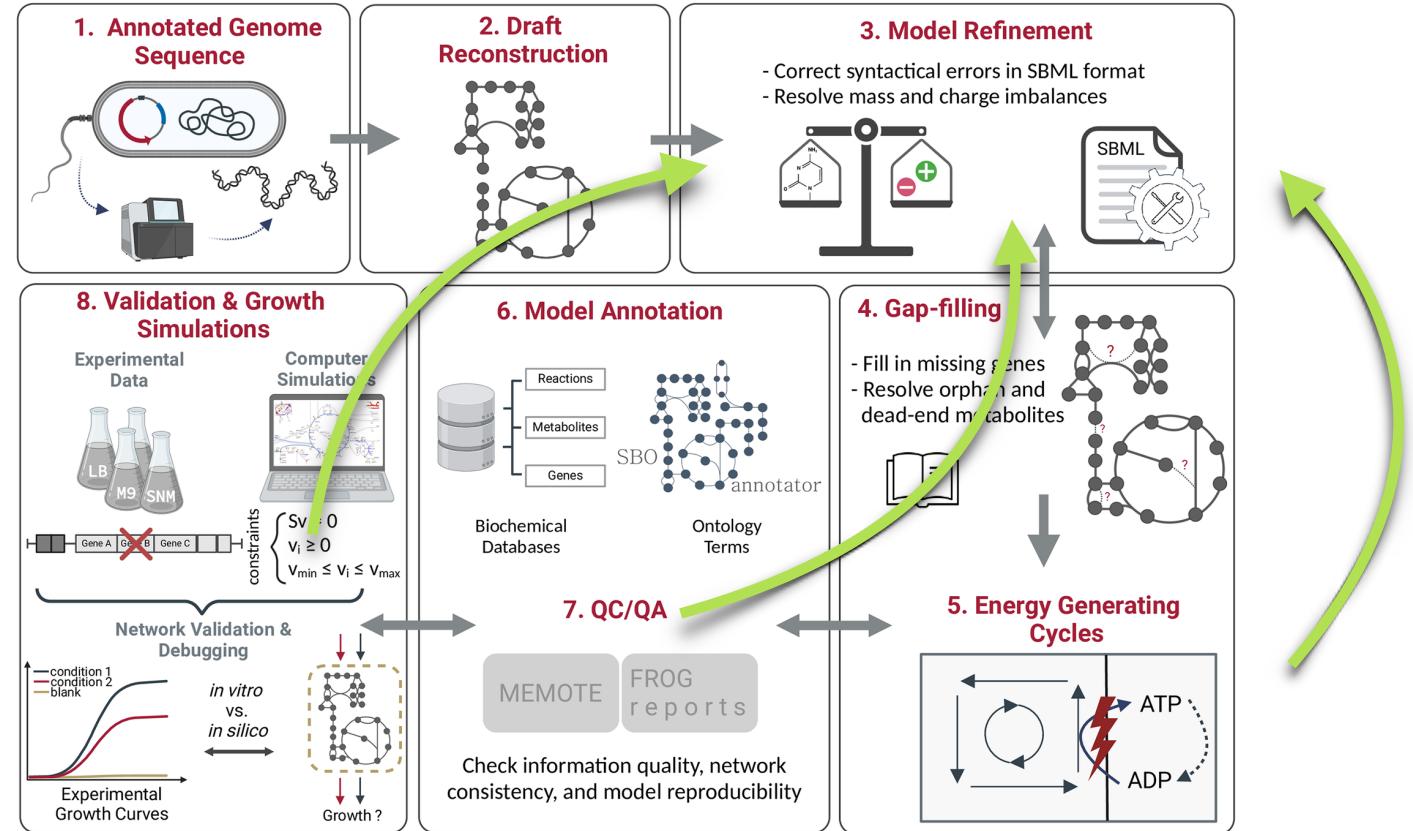
Piecing it together: Workflows

A classic example of a common workflow

Exemplary workflow of a model reconstruction progress for *Acinetobacter baumannii*:

If something does not fit ...

... one goes back to refinement



From Leonidou et al. "Exploring the metabolic profile of *A. Baumannii* for antimicrobial development using genome-scale modelling." PLoS pathogens 20.9 (2024): e1012528

Piecing it together: Workflows

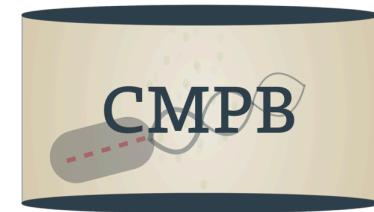
Workflow collection SPECIMEN: Overview

SPECIMEN

- GEM reconstruction is a process of running steps subsequently and/or iteratively
- Idea: Reduce the amount of manual work by chaining multiple functionalities and tools into workflows
- Solution:

SPECIMEN (Strain-specific metabolic modelling)

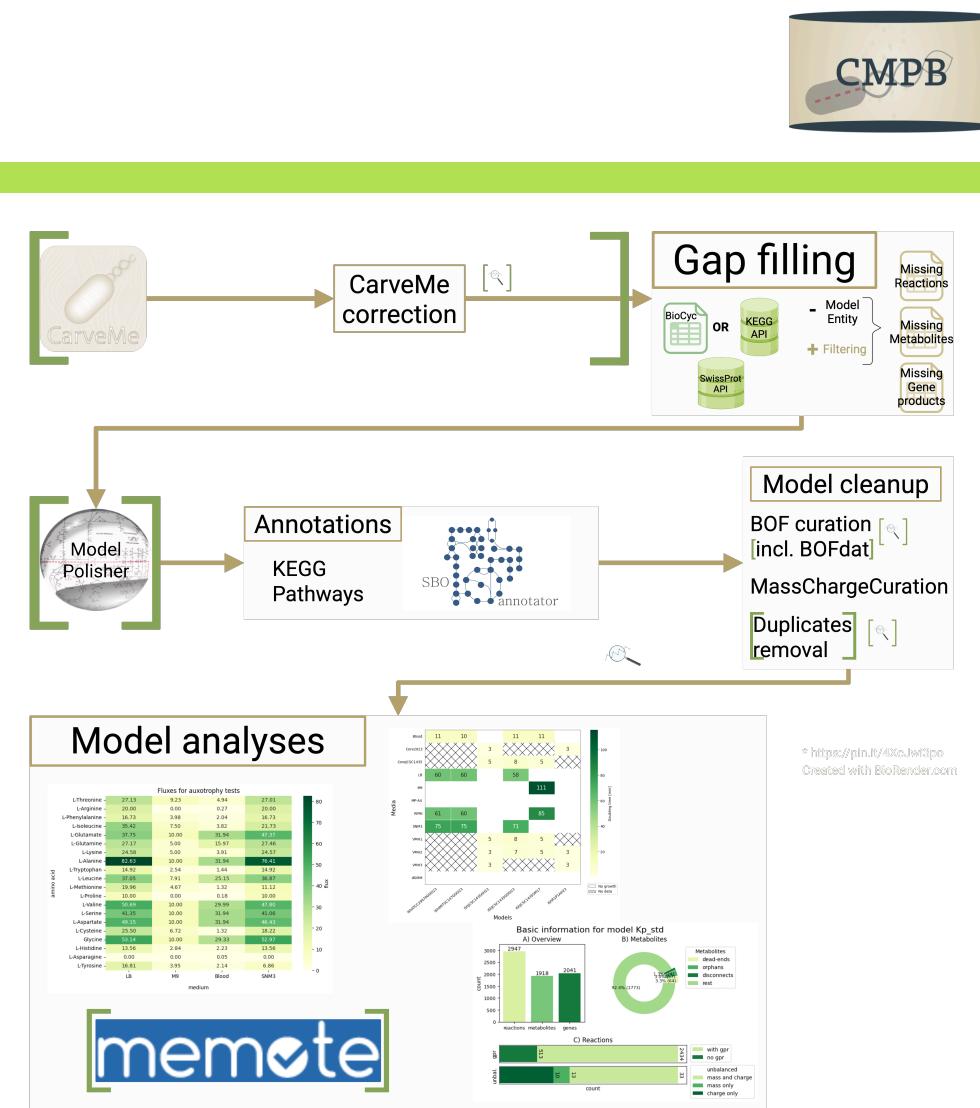
- Python package containing a collection of workflows for GEM reconstruction
- (Currently) Available workflows:



Note: Some manual curation afterwards oftentimes still necessary, as not everything can be solved in an automated manner

CMPB: CarveMe & ModelPolisher based

- Based on the “traditional” workflow using CarveMe for the draft model reconstruction
 - Idea: Find matches against a universal model
 - While often leading to reasonable results, the universal model has several shortcomings (Old, Artefacts, Unspecific)
- Workflow tries to combat the shortcomings, adding further refinements and analyses.
- Works with an annotated genome or model as input
 - Works with numerous kinds of data and organisms
 - If the organism was not included when the universal model was built, it might not be specific or miss critical pathways.



Overview of the CMPB workflow

CMPB: CarveMe & ModelPolisher based



Input

- Model or protein FastA
- Many optional files are possible

Output

- Final model
- Analyses
- Logs, information about fails during steps, etc.; files for manual curation

Access points

- Python: specimen.cmpb.workflow.run
- Command line: specimen cmpb --help
- Notes:
 - Parameters can be set either directly via a configuration file (YAML, left side)

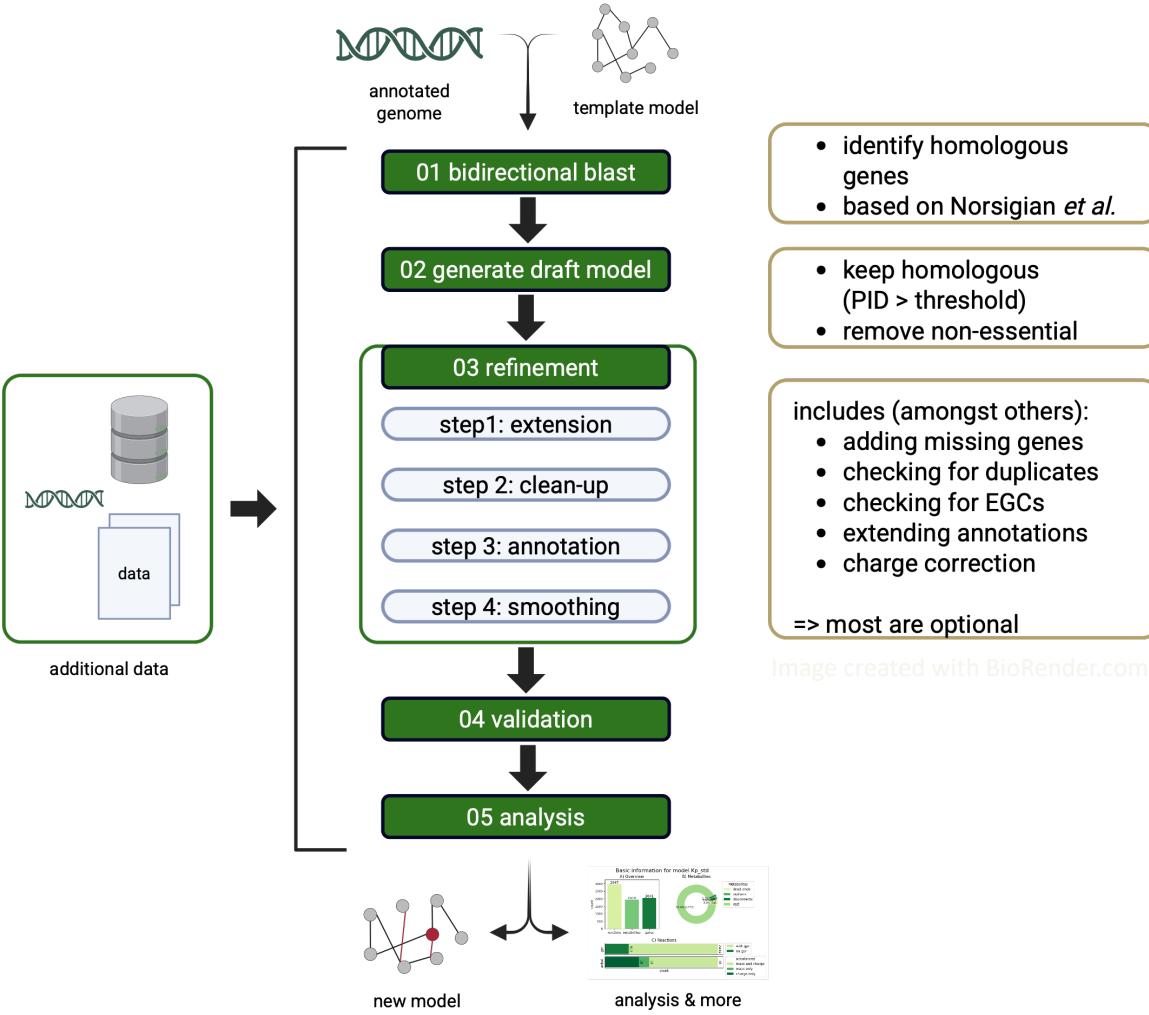
YAML

```
input:
    modelpath: NULL          # Optional, path to a model
                                # If not given, runs CarveMe
    mediapath: __USER__

general: # just a small selection
    modelname: NULL
    organism: USER           # Abbreviation for the organism
    save_all_models: True    # Save a model per step
    memote_always_on: False   # Run MEMOTE after every step
    stats_always_on: False
    kegg_organism_id: USER
    protein_fasta: USER

tech-resources:
    email: USER              # User Mail to use for Entrez (accessing
                                # NCBI).
    threads: 2                # Number of threads available for tools
                                # like DIAMOND
...
```

HQTB: High-quality template based



- Based on the draft model construction approach by Norwegian et al.
- **Idea:**
Use an already curated high-quality model as a template for a model of a similar species to keep as much information from previous work as possible
- **Algorithm:**
 - Perform bidirectional blast
 - Keep homologous genes, throw out everything else (save for complexes and essential ones)
 - Further extend, refine and improve the model
 - Useful, if e.g. lab strains do not have an entry or multiple strains of one species should be modelled

HQTB: High-quality template based



Input

- Subject: GFF + annotated genome
- Template: model + annotated genome

Output

- Final model
- Analyses
- Logs, information about fails during steps, etc.; files for manual curation

Access points

- Python: specimen.hqtb
- Command line: specimen hqtb --help
- Notes:
 - Most steps/parts can be run independently of the full workflow
 - Parameters can be set either directly or via a configuration file (YAML, right side)

YAML

```
# information about the genome to be used to generate
# the new model
subject:
    annotated_genome: <cds.faa>
    gff: <genomic.fna>

# information about the template model/genome
template:
    annotated_genome: __USER__
    model: __USER__
    Namespace: BiGG

# information about the output
# -----
general:
    dir: __USER__
    modelname: iLPhqtb
    ...
    memote: False
    ...
```



Notebook III

Summary

Topics covered in this tutorial:

- Introduction to genome-scale metabolic models (GEMs)
- refineGEMs - a toolbox for working with GEMs
- Introduction to the reconstruction and curation process of GEMs
- SPECIMEN - a workflow collection for (semi-)automated, high-quality GEM construction

What you learned:

- ◆ How to use refineGEMs and SPECIMEN
- How to do basic and advanced analyses on GEMs
- How to perform different model curation steps
- How to run a workflow from SPECIMEN



Find the tools on:



Or reach out to us:

Martin Luther University Halle-Wittenberg,
“Datenanalytik und Bioinformatik”
Carolin Brune
Gwendolyn O. Döbel



Fritzemeier, C. J., D. Hartleb, B. Szappanos, B. Papp und M. J. Lercher (2017). „Erroneous energy-generating cycles in published genome scale metabolic networks: Identification and removal“. In: PLoS computational biology 13.4, e1005494.

Norsigian, C.J., Fang, X., Seif, Y. et al. A workflow for generating multi-strain genome-scale metabolic models of prokaryotes. *Nat Protoc* 15, 1–14 (2020). <https://doi.org/10.1038/s41596-019-0254-3>