GECKO Documentation

Release 1.3.3

SysBioChalmers

Nov 02, 2018

Contents:


Gecko


Gecko

CHAPTER 1

About GECKO

The **GECKO** toolbox is a Matlab/Python package for enhancing a **G**enome-scale model to account for **E**nzyme **C**onstraints, using **K**inetics and **O**mics. It is the companion software to this publication, and it has two main parts:

- `geckomat`: Matlab+Python scripts to fetch online data and build/simulate enzyme-constrained models.
- `geckopy`: a Python package which can be used with cobrapy to obtain a ecYeastGEM model object, optionally adjusted for provided proteomics data.

Last update: 2018-11-02

This repository is administered by Benjamin J. Sanchez (@BenjaSanchez), Division of Systems and Synthetic Biology, Department of Biology and Biological Engineering, Chalmers University of Technology.

CHAPTER 2

geckomat: Building enzyme-constrained models

2.1 Required software - Python module

- Python 2.7
- setuptools for python 2.7
- SOAPpy:

```
easy_install-2.7 SOAPpy
```

2.2 Required software - Matlab module

- MATLAB (7.5 or higher) + Optimization Toolbox.
- The COBRA toolbox for MATLAB.
- The RAVEN toolbox for MATLAB.
- The libSBML MATLAB API (version 5.17.0 is recommended).

2.3 Usage

- **For creating an enzyme constrained model:**
  - Update the following data files in `/databases` with your organism infomation:
    - `databases/prot_abundance.txt`: Protein abundance Data from Pax-DB.
    - `databases/uniprot.tab`: Gene-proteins data from uniprot.
    - `databases/chemostatData.tsv`: Chemostat data for estimating GAM (optional, called by `fitGAM.m`).

    * `databases/manual_data.txt`: Kcat data from eventual manual curations (optional, called by `manualModifications.m`).

  – Adapt the following functions in `/geckomat` to your organism:

    * `geckomat/enhanceGEM.m`
    * `geckomat/change_model/manualModifications.m`
    * `geckomat/limit_proteins/sumProtein.m`
    * `geckomat/limit_proteins/scaleBioMass.m`
    * `geckomat/kcat_sensitivity_analysis/changeMedia_batch.m`
    * `geckomat/change_model/removeIncorrectPathways.m` (optional, called by `manualModifications.m`)
    * `geckomat/limit_proteins/sumBioMass.m` (optional, called by `sumProtein.m` & `scaleBiomass.m`)
    * `geckomat/limit_proteins/fitGAM.m` (optional, called by `scaleBiomass.m`)

  – Run `geckomat/get_enzyme_data/updateDatabases.m` to update `ProtDatabase.mat`.

  – Run `geckomat/enhanceGEM.m` with your metabolic model as input.

- **For performing simulations with an enzyme-constrained model:** Enzyme-constrained models can be used as any other metabolic model, with toolboxes such as COBRA or RAVEN. For more information on rxn/met naming convention, see the supporting information of Sanchez et al. (2017)

CHAPTER 3

geckopy: Integrating proteomic data to ecYeastGEM

If all you need is the ecYeastGEM model to use together with cobrapy you can use the `geckopy` Python package.

3.1 Required software

- Python 2.7, 3.4, 3.5 or 3.6
- cobrapy

3.2 Installation

```
pip install geckopy
```

3.3 Usage

```
from geckopy import GeckoModel
import pandas
some_measurements = pandas.Series({'P00549': 0.1, 'P31373': 0.1, 'P31382': 0.1})
model = GeckoModel('multi-pool')
model.limit_proteins(some_measurements)
model.optimize()
```

CHAPTER 4

Contributors

- Ivan Domenzain (@IVANDOMENZAIN), Chalmers University of Technology, Gothenburg Sweden
- Eduard Kerkhoven (@edkerk), Chalmers University of Technology, Gothenburg Sweden
- Benjamin J. Sanchez (@BenjaSanchez), Chalmers University of Technology, Gothenburg Sweden
- Moritz Emanuel Beber (@Midnighter), Danish Technical University, Lyngby Denmark
- Henning Redestig (@hredestig), Danish Technical University, Lyngby Denmark
- Cheng Zhang, Science for Life Laboratory, KTH - Royal Institute of Technology, Stockholm Sweden