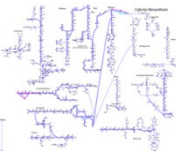


# Introduction to Systems Biology: Constraint-based Metabolic Reconstructions & Analysis

*Course Introduction Lab*

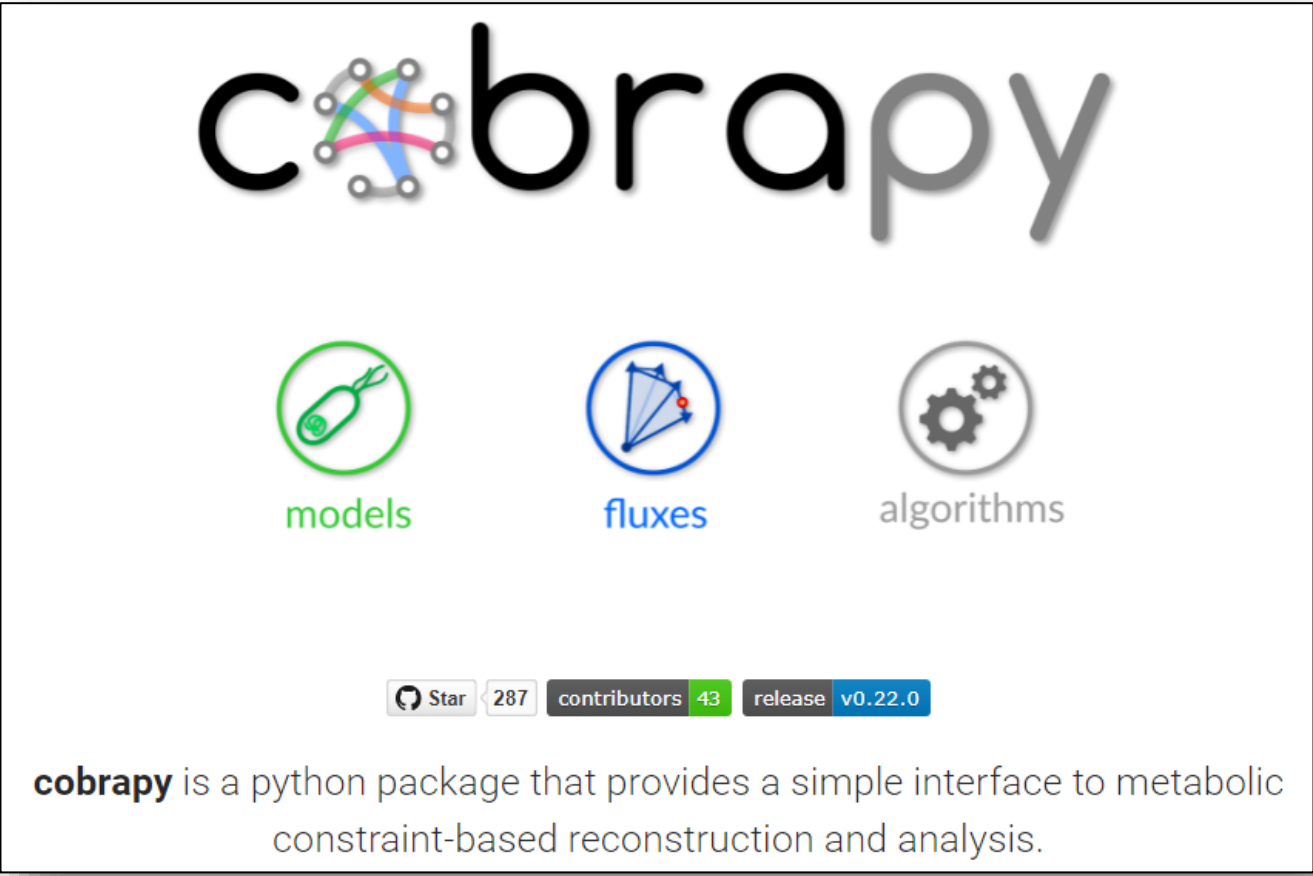


## Course Introduction Lab

- Content Overview
- Course Website
- Course Learning Process
- Course Grading & Expectations

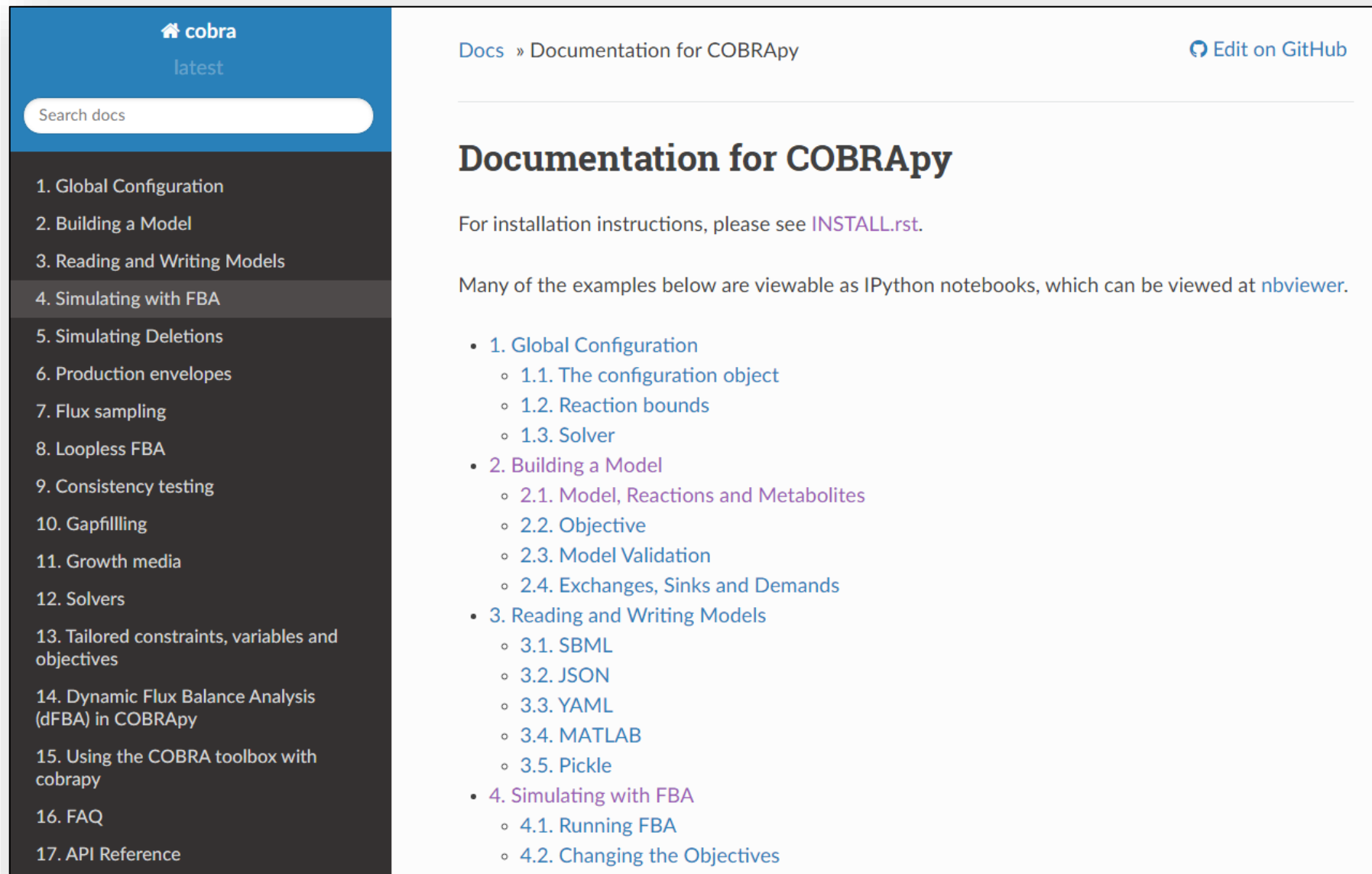
# COBRApy Toolbox Website

<https://opencobra.github.io/cobrapy/>



The screenshot shows the COBRApy Toolbox Website. At the top is the COBRApy logo, which consists of the word "cobrapy" in a sans-serif font, with the "c" and "b" in black and "o", "r", "a", and "p" in a light gray. The "o" is replaced by a network diagram with nodes and colored edges. Below the logo are three circular icons: a green circle with a green outline and a green icon of a bacterium (labeled "models" in green), a blue circle with a blue outline and a blue icon of a network graph (labeled "fluxes" in blue), and a gray circle with a gray outline and a gray icon of two interlocking gears (labeled "algorithms" in gray). Below these icons is a GitHub repository statistics bar showing "Star 287", "contributors 43", and "release v0.22.0". At the bottom, a paragraph states: "cobrapy is a python package that provides a simple interface to metabolic constraint-based reconstruction and analysis."

# COBRApy Documentation



The screenshot displays the COBRApy documentation website. On the left, a sidebar with a dark blue header contains the 'cobra' logo and a 'latest' version indicator. Below this is a search bar and a list of 17 topics, with '4. Simulating with FBA' selected. The main content area has a light blue header with 'Docs » Documentation for COBRApy' and an 'Edit on GitHub' link. The title 'Documentation for COBRApy' is prominently displayed. The text 'For installation instructions, please see [INSTALL.rst](#).' and 'Many of the examples below are viewable as IPython notebooks, which can be viewed at [nbviewer](#).' are present. A bulleted list of topics follows, with sub-topics indented.

**Documentation for COBRApy**


For installation instructions, please see [INSTALL.rst](#).

Many of the examples below are viewable as IPython notebooks, which can be viewed at [nbviewer](#).

- 1. Global Configuration
  - 1.1. The configuration object
  - 1.2. Reaction bounds
  - 1.3. Solver
- 2. Building a Model
  - 2.1. Model, Reactions and Metabolites
  - 2.2. Objective
  - 2.3. Model Validation
  - 2.4. Exchanges, Sinks and Demands
- 3. Reading and Writing Models
  - 3.1. SBML
  - 3.2. JSON
  - 3.3. YAML
  - 3.4. MATLAB
  - 3.5. Pickle
- 4. Simulating with FBA
  - 4.1. Running FBA
  - 4.2. Changing the Objectives

<https://cobrapy.readthedocs.io/en/latest/index.html>

# CAMEO



## Welcome to cameo!

[Watch](#) 23

[chat](#)
[on gitter](#)
[pypi](#)
[v0.13.6](#)
[license](#)
[APACHE2](#)
[build](#)
[error](#)
[coverage](#)
[67%](#)
[DOI](#)
[10.5281/zenodo.2575046](#)

[Shipping faster with ZenHub](#)
[launch](#)
[binder](#)

[build](#)
[error](#)

[codecov](#)
[64%](#)

### Navigation

- [Installation](#)
- [Tutorials](#)
- [Contributing](#)
- [Development](#)
- [API Docs](#)

### Quick search

## What is cameo?

**Cameo** is a high-level python library developed to aid the strain design process in metabolic engineering projects. The library provides a modular framework of simulation and strain design methods that targets developers that want to develop new design algorithms and custom analysis workflows. Furthermore, it exposes a high-level API to users that just want to compute promising strain designs.

Curious? Head over to [try.cameo.bio](https://try.cameo.bio) and give it a try.

Please cite <https://doi.org/10.1021/acssynbio.7b00423> if you've used cameo in a scientific publication.

## High-level API (for users)


Compute strain engineering strategies for a desired product in a number of host organisms using the high-level interface (runtime is on the order of hours).

```
from cameo.api import design
design(product='L-Serine')
```


Cardoso, João GR, et al. "Cameo: a Python library for computer aided metabolic engineering and optimization of cell factories." *ACS synthetic biology* 7.4 (2018): 1163-1166.

## CAMEO Tutorials


<https://cameo.bio/tutorials.html>



### Tutorials

 Watch
 23

build
error

 codecov
 64%

#### Navigation

- [Installation](#)
- [Tutorials](#)
  - [Import models](#)
  - [Simulate models](#)
  - [Analyzing models](#)
  - [Predict gene knockout strategies](#)
  - [Predict expression modulation targets](#)
  - [Predict heterologous pathways](#)
  - [Easy strain design using a high-level interface](#)
- [Contributing](#)

The following tutorials are based on [Jupyter](#) notebooks that are also available as live code at [try.cameo.bio](https://try.cameo.bio). Furthermore, [course materials](#) are available for a 2-day course in cell factory engineering.

- [Import models](#)
  - [Import models from files](#)
  - [Import models from the internet](#)
- [Simulate models](#)
  - [Primer: Constraint-Based Modeling](#)
  - [Flux Balance Analysis](#)
  - [Parsimonious Flux Balance Analysis](#)
  - [Setp 2: Simulate knockouts phenotypes](#)
- [Analyzing models](#)
  - [Flux Variability Analysis](#)
  - [Phenotypic Phase Plane](#)
  - [Flux Balance Impact Degree](#)
- [Predict gene knockout strategies](#)
  - [OptGene](#)
  - [OptKnock](#)
  - [References](#)

# Escher Visualization



<http://escher.github.io/>

# ESCHER

*Build, share, and embed visualizations of biological pathways.*

Filter by organism

All

Map

Core metabolism (e\_coli\_core)

Model (Optional)

e\_coli\_core

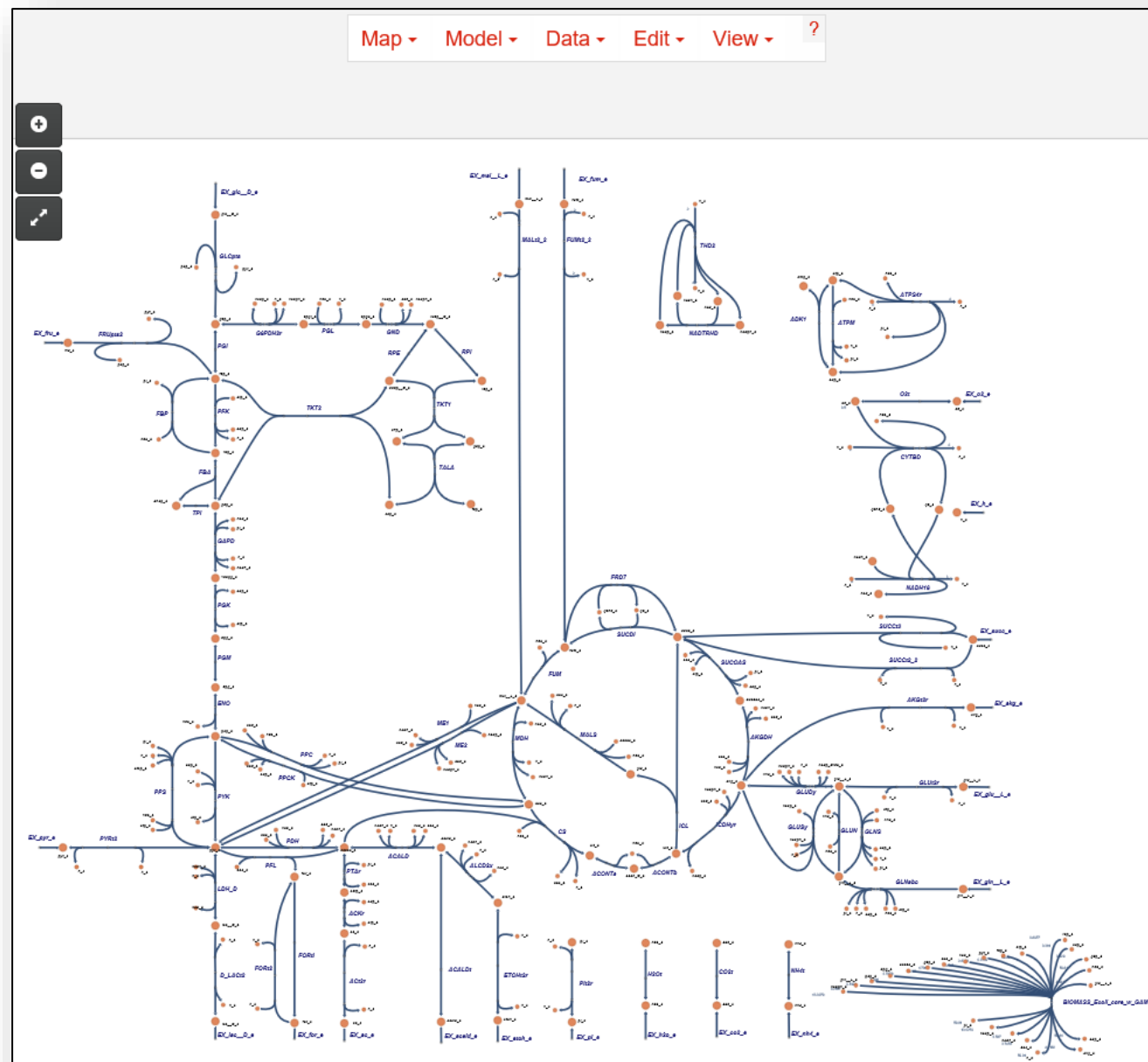
Tool

Builder

Options

- ☐ Scroll to zoom (instead of scroll to pan)
- ☐ Never ask before reloading
- ☐ Responsive pan and zoom

Load map





## Escher Documentation

<https://escher.readthedocs.io/en/latest/>

Escher

latest

Search docs

- Getting started with Escher visualizations
- Tips and Tricks
- Escher, COBRA, and COBRApy
- Escher Python tutorial
- Validate Escher maps
- Developing with Escher
- Developer Tutorial: Custom tooltips
- EscherConverter
- JavaScript API
- Python API
- License

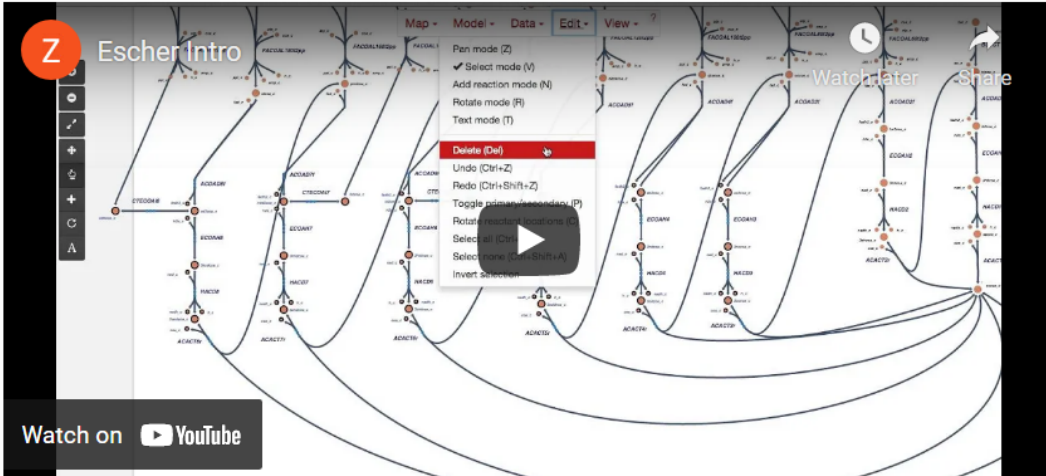
Docs » Welcome to the documentation for Escher

Edit on GitHub

### Welcome to the documentation for Escher

Escher is a web-based tool for building, viewing, and sharing visualizations of metabolic pathways. These 'pathway maps' are a great way to contextualize metabolic datasets. The easiest way to use Escher is to browse and build maps on the [Escher website](#). New users may be interested in the [Getting started with Escher visualizations](#) guide. Escher also has a [Python package](#) and, for developers, a [NPM package](#).

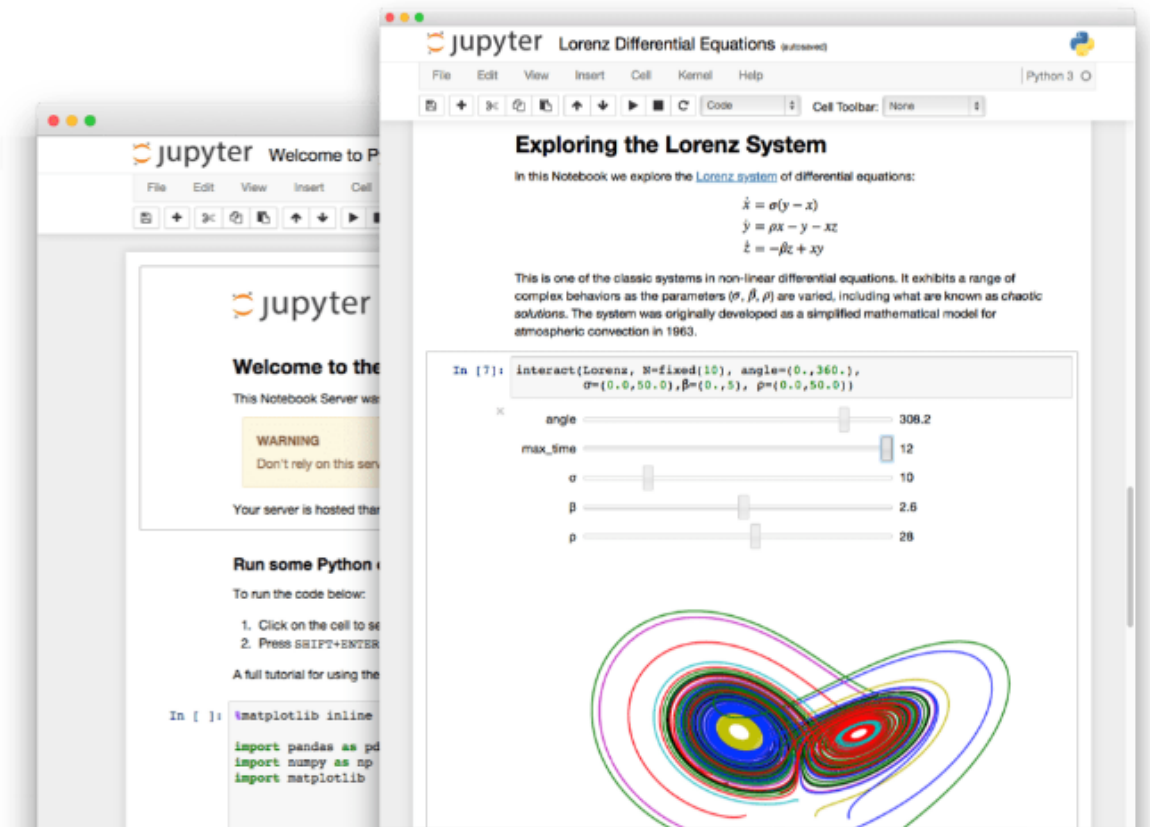
### Escher in 3 minutes





## Jupyter Notebooks

<https://jupyter.org/>

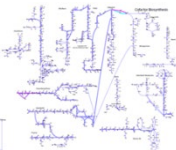


### Jupyter Notebook: The Classic Notebook Interface

The Jupyter Notebook is a web application for creating and sharing documents that contain code, visualizations, and text. It can be used for data science, statistical modeling, machine learning, and much more.

Try it in your browser


Install the Notebook



# The Jupyter Notebook Documentation

<https://jupyter-notebook.readthedocs.io/en/stable/>

The screenshot shows the Jupyter Notebook documentation website. On the left is a dark sidebar with a blue header containing a home icon, the text 'Jupyter Notebook', and 'stable' below it. A search bar labeled 'Search docs' is also in the header. The sidebar lists 'USER DOCUMENTATION' (The Jupyter Notebook, User interface components, Notebook Examples, What to do when things go wrong, Changelog, Comms) and 'CONFIGURATION' (Configuration Overview, Config file and command line options). The main content area has a breadcrumb '» The Jupyter Notebook' and a link to 'Edit on GitHub'. It features a large heading 'The Jupyter Notebook' followed by a list of links: 'Installation' and 'Starting the Notebook'. Below this is a heading 'User Documentation' followed by a list of links: 'The Jupyter Notebook', 'User interface components', 'Notebook Examples', 'What to do when things go wrong', 'Changelog', and 'Comms'.

 Jupyter Notebook  
stable

Search docs

USER DOCUMENTATION

The Jupyter Notebook

User interface components

Notebook Examples

What to do when things go wrong


Changelog


Comms

CONFIGURATION

Configuration Overview

Config file and command line options

 » The Jupyter Notebook

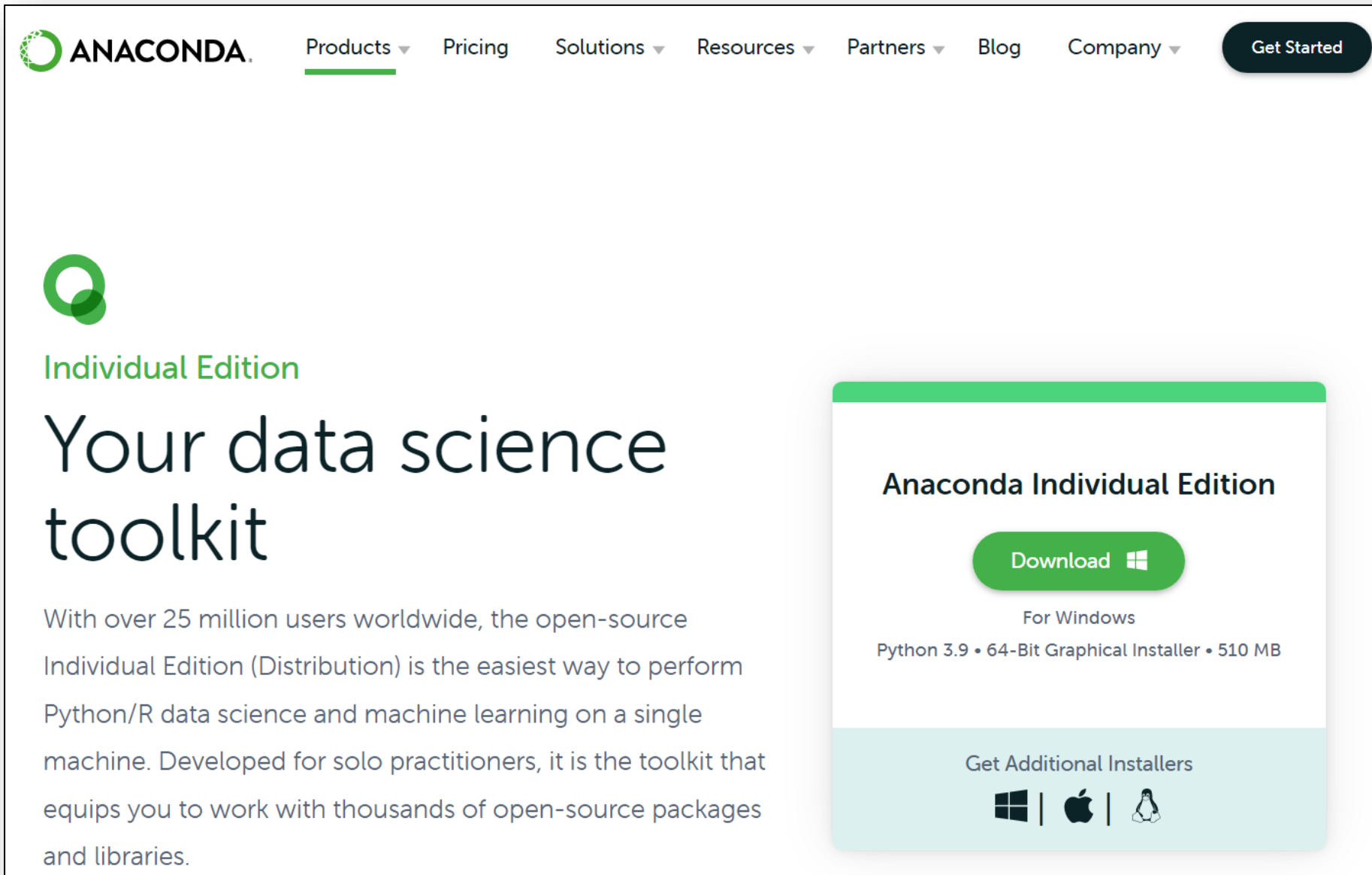
 Edit on GitHub

The Jupyter Notebook

- [Installation](#)
- [Starting the Notebook](#)


User Documentation

- [The Jupyter Notebook](#)
- [User interface components](#)
- [Notebook Examples](#)
- [What to do when things go wrong](#)
- [Changelog](#)
- [Comms](#)



The screenshot shows the Anaconda Individual Edition website. At the top is a navigation bar with the Anaconda logo, links for Products, Pricing, Solutions, Resources, Partners, Blog, and Company, and a 'Get Started' button. The main content area features the Anaconda logo, the text 'Individual Edition', and the headline 'Your data science toolkit'. Below this is a paragraph describing the toolkit. To the right is a download card for the Windows version, including a 'Download' button and system requirements. At the bottom of the card is a section for 'Get Additional Installers' with icons for Windows, macOS, and Linux.


**ANACONDA** Products Pricing Solutions Resources Partners Blog Company [Get Started](#)

 **Individual Edition**

## Your data science toolkit

With over 25 million users worldwide, the open-source Individual Edition (Distribution) is the easiest way to perform Python/R data science and machine learning on a single machine. Developed for solo practitioners, it is the toolkit that equips you to work with thousands of open-source packages and libraries.




### Anaconda Individual Edition

[Download](#) 

For Windows

Python 3.9 • 64-Bit Graphical Installer • 510 MB

Get Additional Installers

 |  | 

<https://www.anaconda.com/products/individual>

The screenshot shows the Anaconda Navigator desktop application. The interface includes a sidebar on the left with navigation options: Home, Environments, Learning, and Community. The main area displays a grid of applications available on the 'Combined' channel. Each application card includes an icon, name, version, description, and a button to launch or install it. A 'Sign in' button is located in the top right corner of the application window.

**Applications on Combined Channels**

Application	Version	Description	Action
Datalore		Online Data Analysis Tool with smart coding assistance by JetBrains. Edit and run your Python notebooks in the cloud and share them with your team.	Launch
IBM Watson Studio Cloud		IBM Watson Studio Cloud provides you the tools to analyze and visualize data, to cleanse and shape data, to create and train machine learning models. Prepare data and build models, using open source data science tools or visual modeling.	Launch
Jupyter Notebook	6.4.6	Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.	Launch
PyCharm Community	2021.2	An IDE by JetBrains for pure Python development. Supports code completion, listing, and debugging.	Launch
VS Code	1.46.1	Streamlined code editor with support for development operations like debugging, task running and version control.	Launch
CMD.exe Prompt	0.1.1	Run a cmd.exe terminal with your current environment from Navigator activated	Install

**ANACONDA NUCLEUS**  
Back up your environments in Nucleus for free  
[Join Now](#)  
Easily back up, port, and restore any environment  
[Documentation](#)  
[Anaconda Blog](#)

Twitter YouTube GitHub

# COBRAPY Toolbox Installation

## Installation

You can install cobrapy on linux, Mac OSX and Windows with pip.

On Linux and Mac a simple:

```
$ pip install cobra
```

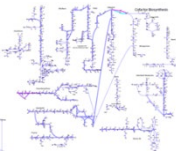
will be enough. On Windows substitute pip with pip.exe.

Or install cobrapy with conda from conda-forge:

```
$ conda install -c conda-forge cobra
```

For further help refer to the **detailed installation instructions**.

<https://opencobra.github.io/cobrapy/>



## Python Packages to Install

- COBRApy - Python COBRA Toolbox
  - `pip install cobra`
- Escher - Visualization Package
  - `pip install escher`
- Matplotlib - Python Graphics Package
  - `pip install matplotlib`
- Pytest - Testing Package
  - `pip install pytest`
- Scipy - Fundamental algorithms for scientific computing in Python
  - `pip install scipy`
- COBRApy BIGG Client - COBRApy Access to the BIGG database
  - `pip install cobrapy_bigg_client`
- Cameo - Metabolic Engineering Package
  - `pip install cameo`
- Plotly - Graphics Package
  - `pip install plotly`