



# What is TinkerCell?

## Contents

[Computer-Aided Design in Synthetic Biology](#)

[Abstract yet detailed diagrams](#)

[Current functionality](#)

[Long term goals](#)

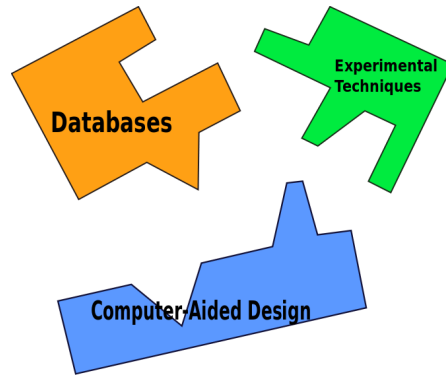
[TinkerCell is for the community](#)

## Computer-Aided Design in Synthetic Biology

TinkerCell is a Computer-Aided Design (CAD) tool for synthetic biology. Since synthetic biology is a rapidly evolving field, here are the issues which were taken into consideration when designing TinkerCell:

- modeling techniques in biology are not robust
- experimental techniques are constantly evolving, especially automation and directed evolution methods
- databases of biological components are still maturing

In summary, TinkerCell is designed with the anticipation that the future of synthetic biology will be an intricate interplay between a variety of experimental techniques, databases that store results from experiments, and mathematical models explaining different aspects of the experiments.



The ideal situation of the CAD program is this:

- step 1: user *draws* a biological system
- step 2: user performs some *analysis*
- step 3: go back to step 1 if analysis is not satisfactory

The *analysis* step can potentially include:

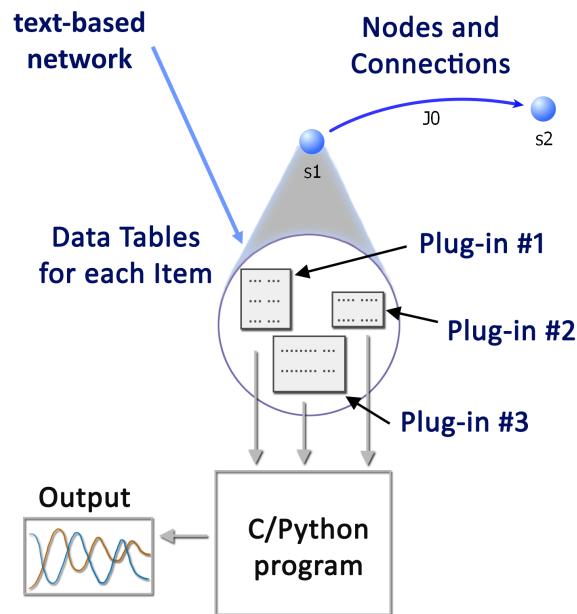
- mathematical analysis of non-linear systems
- stochastic simulations and other methods from systems biology
- prediction of evolutionary trajectories for directed evolution
- analysis and optimization of the sequence
- database look-up to find suitable components
- generate different mathematical representations for the same diagram
- ...more...

It is too much to have all of these functions in TinkerCell from the start. So TinkerCell uses a flexible plug-in framework, which will allow others to contribute new functions to TinkerCell. The plug-ins can be written in C, C++, Python, or Octave. Other languages, e.g. Java, might also get added to this list.

## Abstract yet detailed diagrams

While TinkerCell diagrams are very abstract, they also contain all the details needed for different types of analysis. For example, here is an abstract diagram.

Using the type of each component and their data tables, TinkerCell plug-ins perform different types of analysis.



## Current functionality

- deterministic simulation (using COPASI)
- stochastic simulation (using COPASI)
- steady state computation (using COPASI)
- jacobian computation (using COPASI)
- eigenvalue computation (using COPASI)
- exact stochastic simulation (using COPASI)
- tau-leap based stochastic simulation (using COPASI)
- hybrid simulation (using COPASI)
- stoichiometry matrix structural analysis (using libStructural)

## Long term goals

The long term goal of TinkerCell is to become one of the key applications involved in efficient engineering of biological (cellular) systems. The way this will work is by connecting TinkerCell to lab automation tools. In this situation, users would draw a system in TinkerCell. Some TinkerCell plugin would send the information to a remote robot. When the results from the experiments return, another TinkerCell plugin would automatically recommend the models to

use based on the experimental evidence. None of the other existing modeling tools can do this because they lack the details for describing the underlying biology.

## **TinkerCell is for the community**

The purpose of TinkerCell's plugin interface is to welcome contributions from the community. Additionally, the TinkerCell source code is open-source (BSD) and the application itself is free of charge. All this is intended to make TinkerCell open to the community.