

Biocellion Command Line Tools

1 biocellion.py

`biocellion.py` runs biological models described with XML using the biocellion agent based simulation system. The XML model description file may be specified directly via the `--xml-file` option, or it may be created from a template XML file using the `--template-file` option and repeated use of the `--parm` option.

The tool operates on the model in the following stages:
ex

1. Template model XML description file. This file is an XML model description, with some constants replaced by variables. The tool will translate the variables into values specified on the command line.
2. Model XML description file. This is a complete XML model description.
3. Model C++ implementation files. This is a library of C++ files and auto generated C++ files created from the model XML description file.
4. Model compiled library file. This is the compiled version of the model C++ files.
5. Model run output data files. This is the result of running the model using biocellion and the compiled library file.

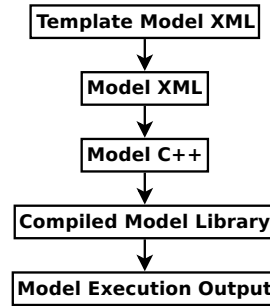


Figure 1: Model processing stages.

1.1 Command Line Options

Command line options control which of the stages to process and create.

1.1.1 Stage Processing Commands

- `-A` | `--all-commands` process all starting with template xml model file, ending with running model

- `-T | --process-xml-template` process template xml model file into xml model file
- `-X | --process-xml-model` process xml model file into cpp model files
- `-C | --compile-cpp-model` compile cpp model files into model library file
- `-R | --run-model` run the model

1.1.2 XML Options

- `-t | --template-file file-name` name of input model template XML file
- `-p | --parm parm=value` name and value of parameter for the model; may be specified more than once
- `-x | --xml-file file-name` name of output model XML file
- `-m | --model-name model-name` descriptor for model
- `-d | --model-dir model-dir` location of template and model files

Use `-t` and `-x` OR `-m` and `-d`. Don't mix them.

1.1.3 Directory and Miscellaneous Options

- `-c | --cpp-dir cpp-dir` location of cpp model library files. You should not need to override the default.
- `-b | --build-dir build-dir` directory to create with c++ code
- `-o | --command-timeout seconds` seconds to let the subprocess run the simulation
- `-h | -? | --help` show a help message and quit

1.2 Examples

- The following command creates a model instance from the `cell-sorting-template.xml`, using adhesion values specified on the command line. It stores the temporary model XML file in `foo.xml`, creates the C++ model in the directory `bar`, builds and runs the model. The model is only allowed 180 seconds to execute.

```
bin/biocellion.py --all-commands -t examples/cell-sorting-template.xml
-x foo.xml -p RED_RED_ADHESION=0.2 -p BLUE_BLUE_ADHESION=0.3 -p
RED_BLUE_ADHESION=0.0 --build-dir bar --command-timeout 180
```