How to reproduce this work:

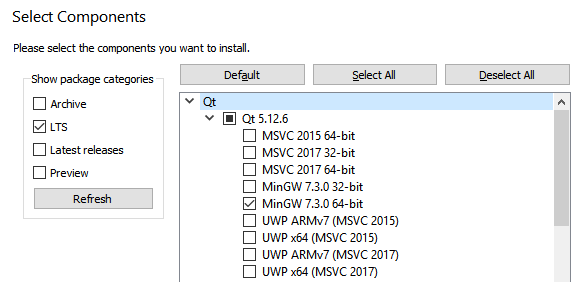
1. **Install the Qt framework and a C++ compiler**

The code is entirely programmed in C++, and further versionns will be available at gitlab.com/Moonfit/Balthyse. Running Moonfit requires a C++ compiler and the Qt library to be installed. No other library is required (some are included as C++ files directly inside the folder)[1]. It is possible to use boost solvers, in which case the boost library should be installed.

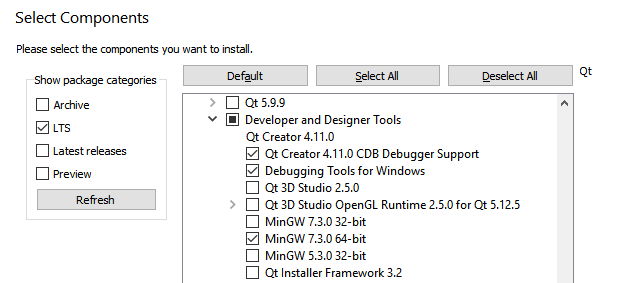
***In Linux:***g++ is recommanded (sudo apt-get install g++). The QT framework can either be

installed as complete version (sudo apt-get install qt-sdk and sudo apt-get install qtcreator), or as core libraries one by one (qtbase5-dev, libqt5svg5, libqt5printsupport5, qtcreator). If boost is wanted, (sudo apt-get install libboost-dev).

***In Windows:***A full version of Qt can be downloaded from https://www.qt.io/download, by choosing the Open Source version. <https://www.qt.io/download-open-source>. The C++ compiler can be included during installation:



And

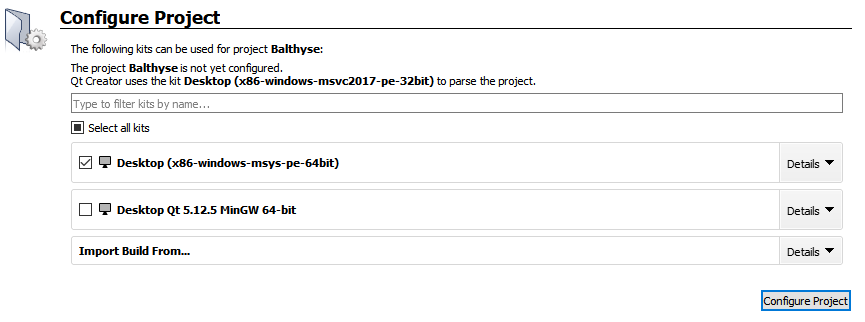


If you wish to use boost, download the latest version http://www.boost.org/users/download/) and unzip it anywhere.

***In MAC:***The clang C++ compiler can be installed with brew by running (brew install -with-clang llvm) on command line, and the QT platform can be downloaded from <https://www.qt.io/download>, by choosing the Open Source version.

1. **Unzip the attached zip with Open Balthyse.pro using QtCreator**

Choose a compiler (whatever one),



Note: remove Balthyse.pro.user when you move the main folder or change computer.

1. **Write inside mainThymus.cpp the folder of the DATA,**

Step 1: Give the location to the data files, and a folder to export results

static string folder = "C:/Users/…/Balthyse/DATA/";

static string folderBaseResults = "C:/Users/…/Balthyse/Results/";

1. **Decide the model structure and if equal export between populations**

Inside modelStructABC.h,

//1: choose model structure, A, B or C

#define ModelStructureB

//2: for hypotheses with coupled parameters, meaning factorF has same impact on multiple outputs, choose:

//- for increased export of all SP cells equally, including SP8 [the parameter for Tconv applies to all then]

//#define equalExportAll

//- if not, for equal increased export of Tregs and progenitors, [the parameter for mature DPTregs applies to all then]

#define equalExportTregs

// and/or, for equal increased export of Tconv and SP8 [the parameter for Tconv applies to SP8], but Tregs have their own differen export,

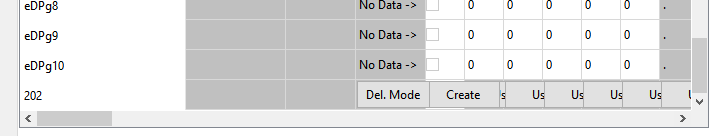
//#define equalExportSP4SP8

(by default, this is structure B with equal export of all Treg populations (and precursors).

1. **Start the program (green button)**

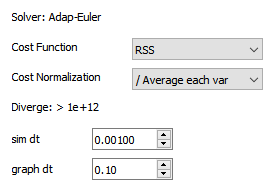
****

1. **Choose the combination of parameters to optimize (click Use)**

****

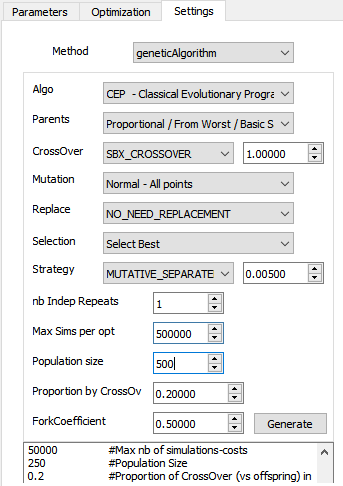
1. **Select the cost function:**

(in settings panel => Choose RSS and /average each var)

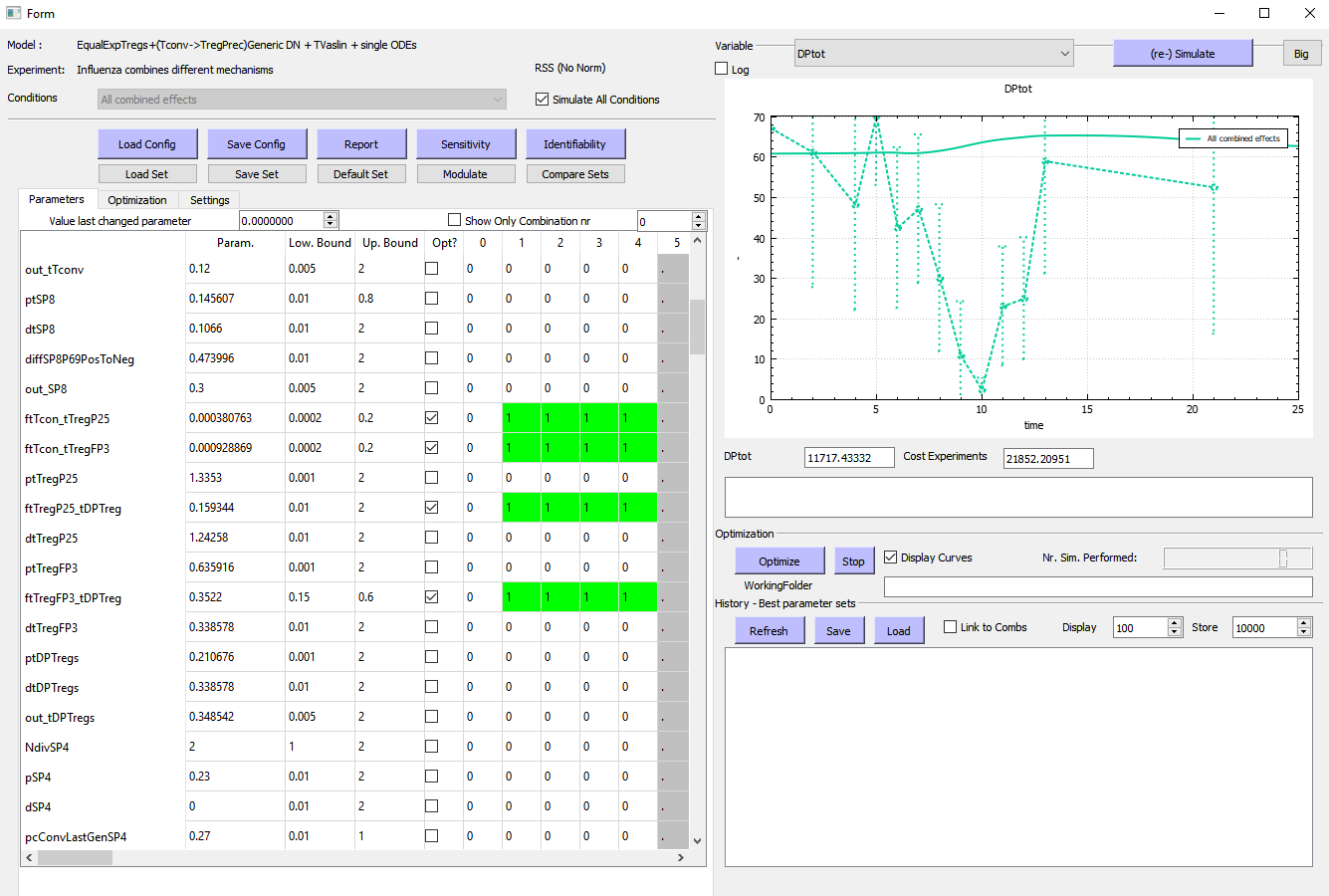


1. **Decide how many steps during optimization**

*Suggest 500 000 iterations, divided into generations of 500 individuals*

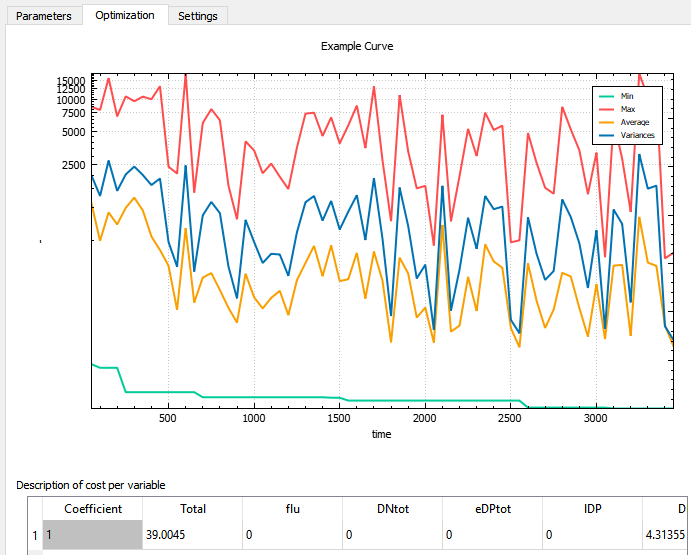


1. **then run Optimize**

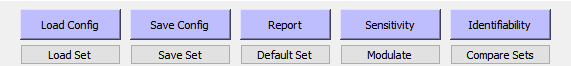
****

(Uncheck ‘display curves’ to run faster, and push stop when you like) –

1. **You can follow the cost over time during optimization (panel optimization, green curve)**

****

1. **Export report**

****