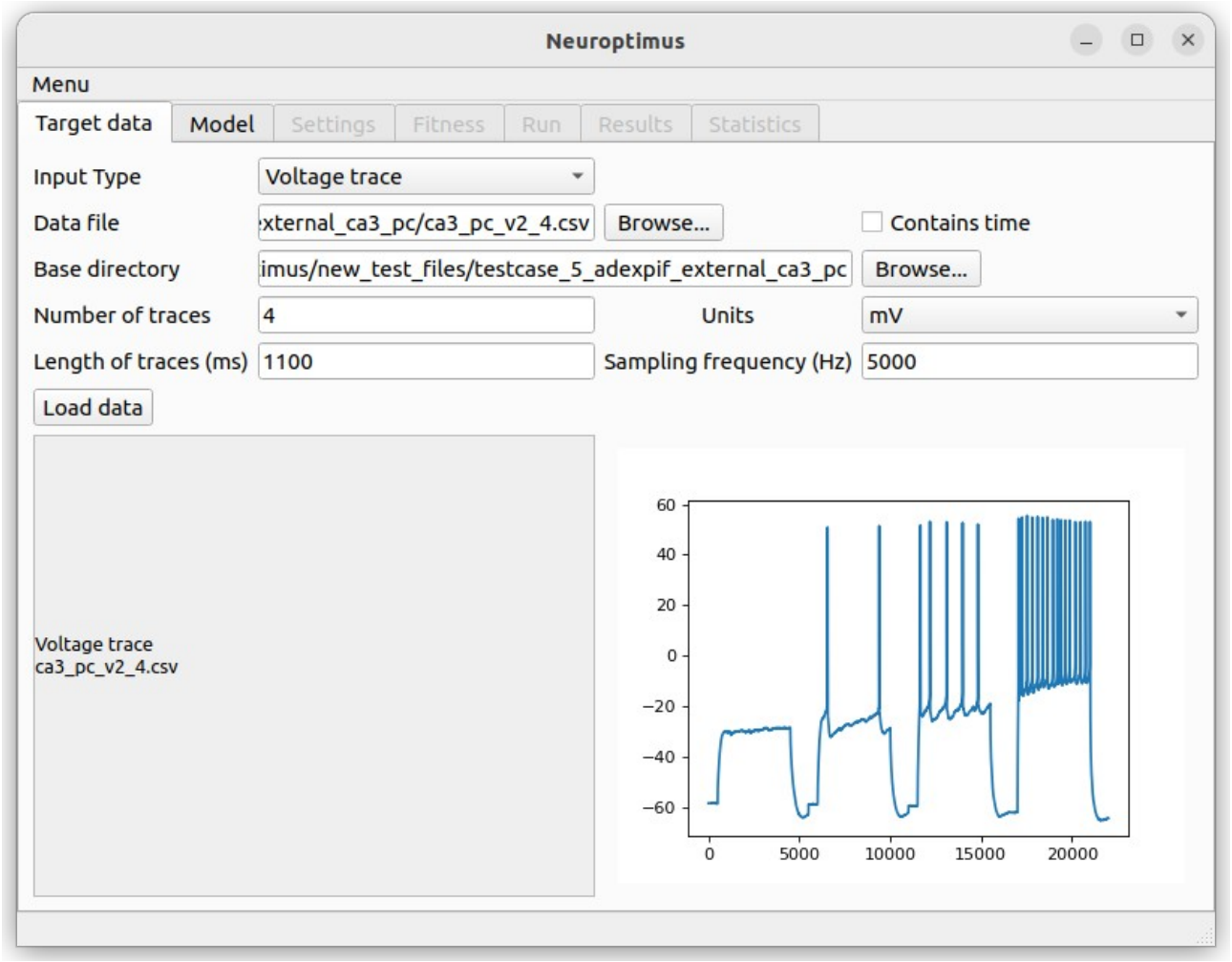
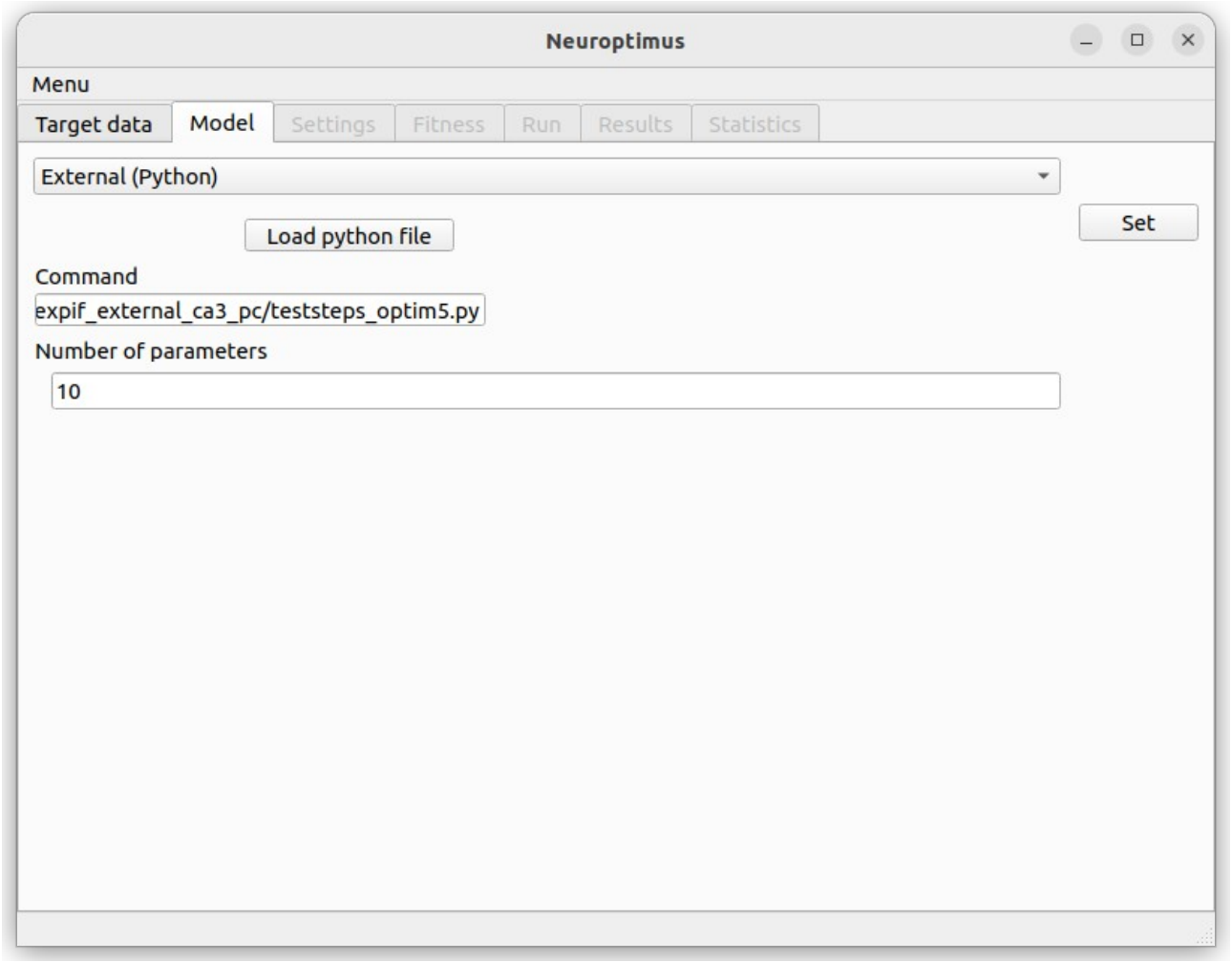
Summary of example:
The input data file (ca3_pc_v2_4.csv) contains 4 voltage traces, corresponding to the responses of a CA3 pyramidal neuron to current steps of different amplitudes. The traces are 1100 ms long, the step starts at 100ms and lasts for 900 ms.
The amplitudes are the following:
0.30 nA
0.35 nA
0.40 nA
0.45 nA
The sampling frequency was 5 kHz.
The model to be optimized is an adaptive exponential integrate-and-fire (AdExpIF) neuron. The model is implemented using the Python interface to the NEST simulator (see teststeps_optim5.py), and is handled as a black box by Neuroptimus. Specifically, the following 10 parameters are the subject of optimization, and are passed to the Python script (along with an extra parameter corresponding to the index of the stimulus) through the file "params.param":
g_L
tau_m
E_L
t_ref
V_reset
V_th
a
b
delta
tau_w
Some additional parameters (C_m and V_peak) of the model are calculated from these parameters in the Python script.
The model script saves the results of the simulation in the files trace.dat (containing the voltage trace) and spike.dat (containg explicit spike times from the same run), which are then used by Neuroptimus to compute the corresponding cost (fitness) value.

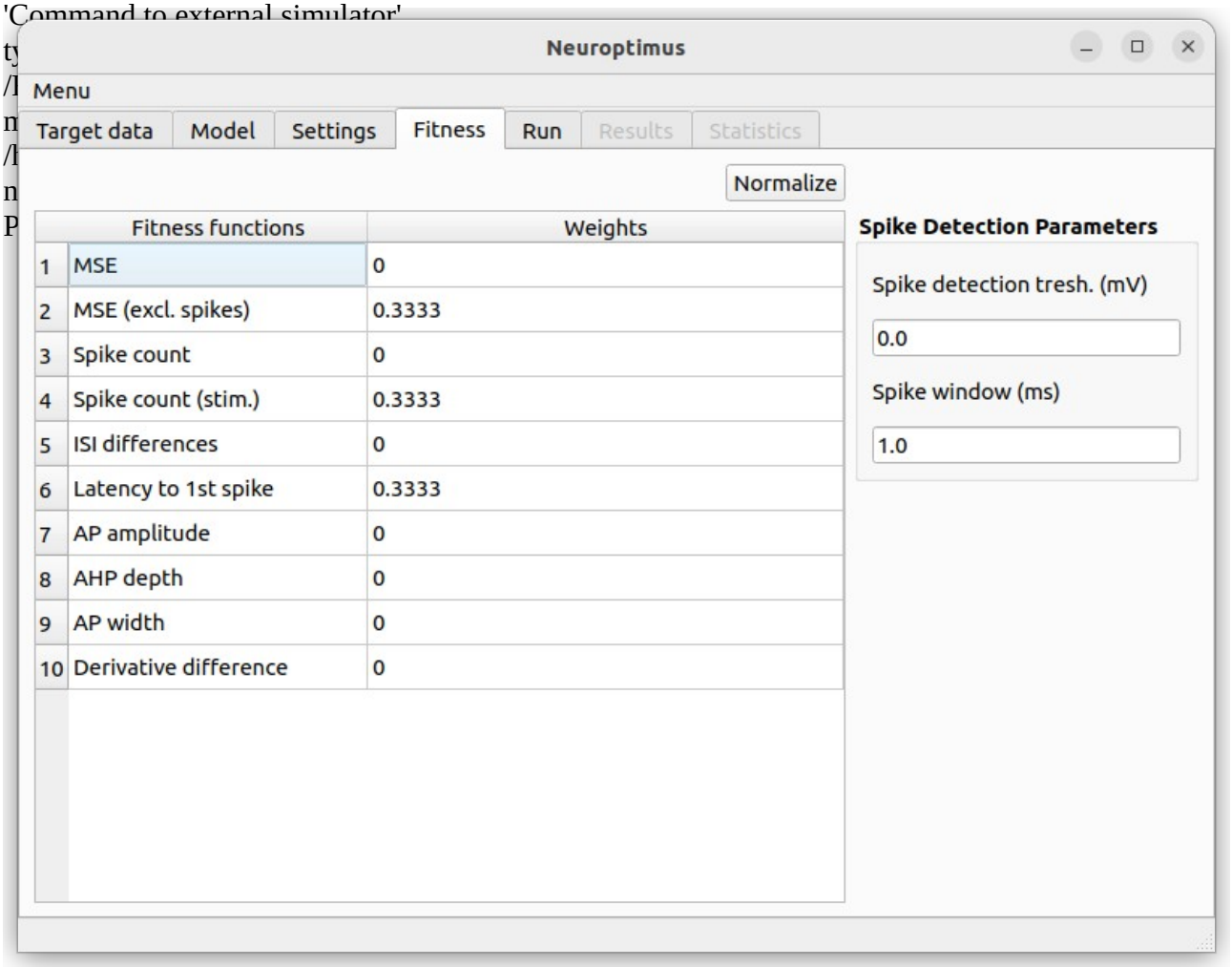
Step-by-step instructions to run the example from the Neuroptimus GUI:
Run „python3 neuroptimus.py -g” to start the GUI

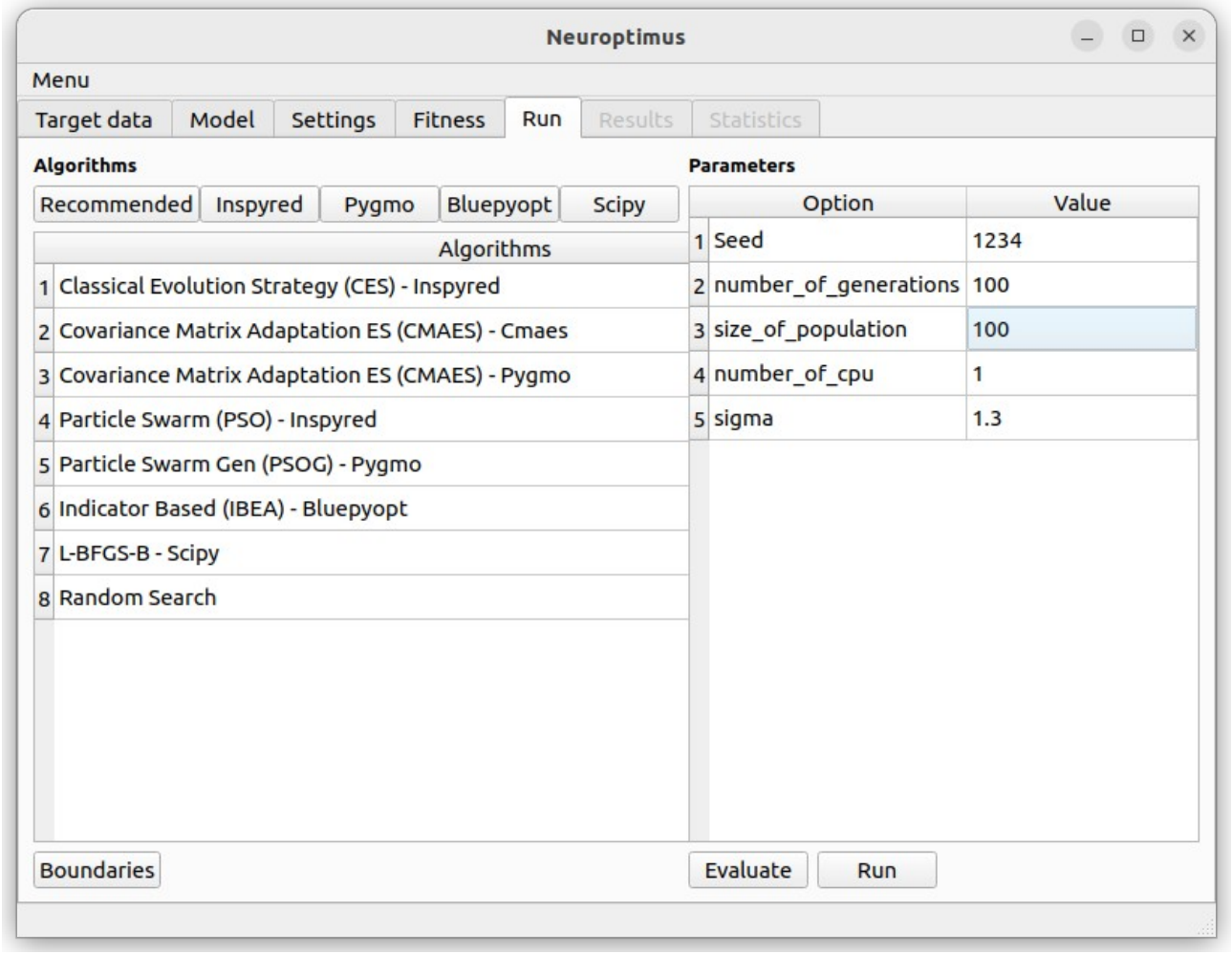


At ’Data File’ load the target data, at ’Base Directory’ choose the directory where you want to save the results. Fill out all the cells and press ’Load data’. Go on by pressing the Model Tab.



For model type choose 'external (python)' or ‘external’. For external load a python file or o
'Command to external simulator'
t
/
n
/
n
P





Choose fitness function(s), and define their weights. Go on by pressing the Run Tab.

Select an algorithm, and set its parameters.
Press the ’Boundaries’ button to define the boundaries of the parameters to be optimized: Press ’Set’.
Boundaries can also be loaded from a file.