ULNA: Development Platform for Artificial Pancreas Algorithms

Tutorial on how to start implementing your own control algorithms in Java

Dr. Mohamed R. Smaoui

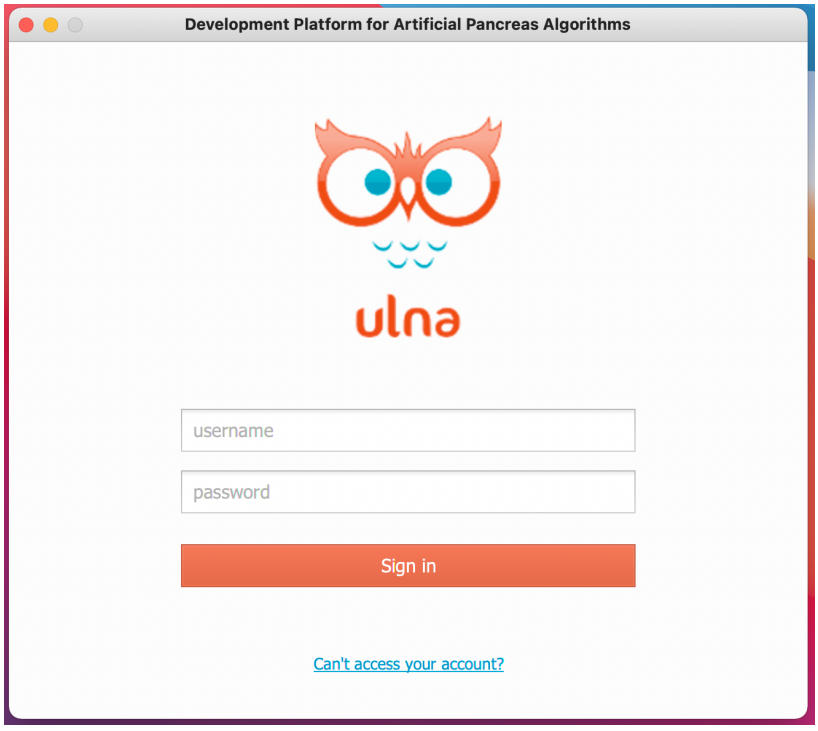
Brief:

Diabetes is a growing epidemic and will affect 1 in every 10 people in the coming years. In this tutorial we will use Ulna, the Development Platform for Artificial Pancreas Algorithms, to develop an algorithm for Type I Diabetes insulin management.

In the sections below, you will walk through steps to launch your first Type I Diabetes control algorithm!

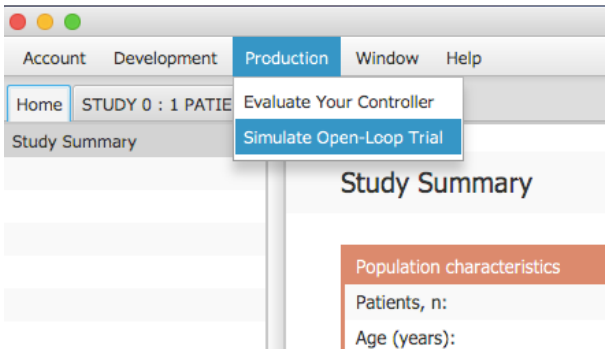
Section 1: Launch Ulna

1. Search for the application Ulna on your computer. Double click and you should be greeted with the following screen

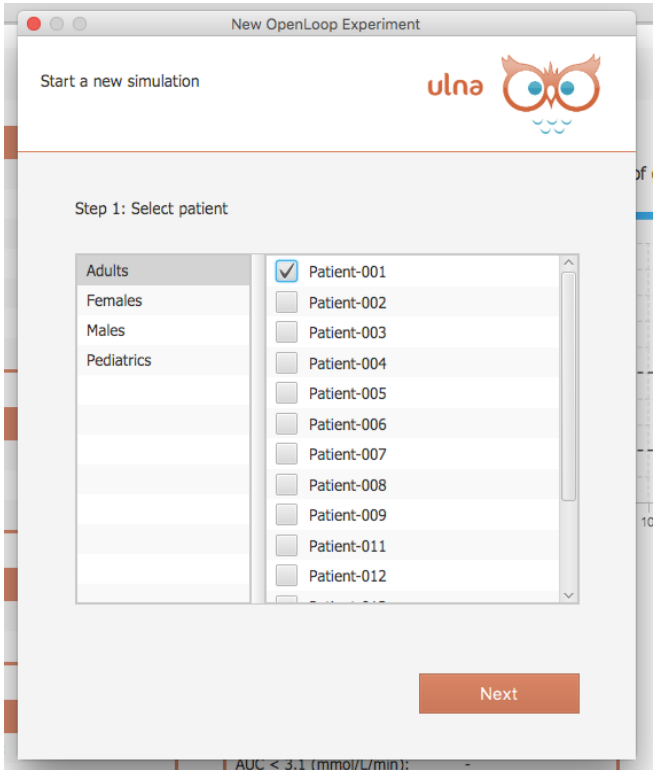


2. Log in with the username and password that have been provided to you in your license file.

3. Once you are logged in, click on “Production” followed by “Simulate Open-Loop Trial” to run a simulation on an “Open-loop” algorithm.



4. From the list of patients, select patient 1 and proceed with “Next” 3 times until the simulation starts.



5. The simulation will record the results of a 24 hour intervention on our virtual patient. Observe the performance of the “Open-Loop algorithm” and take a screenshot of the “Study Summary” page.

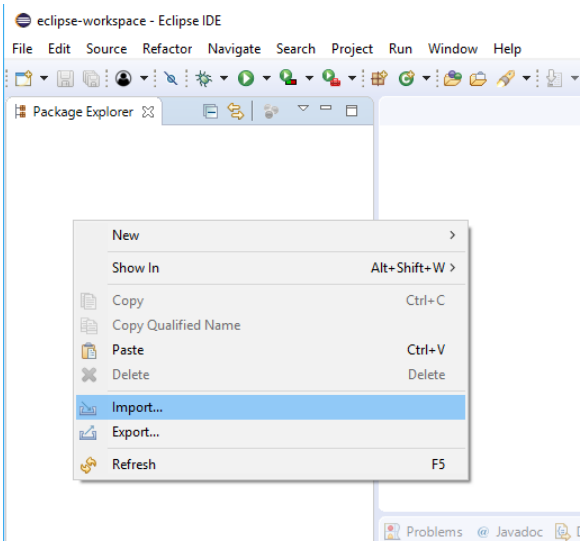
Section 2: Connect your algorithm (JAVA, eclipse)

The Ulna platform allows us to integrate our own algorithms and test their performance against the virtual patient population. To plug an algorithm of your own, follow these steps:

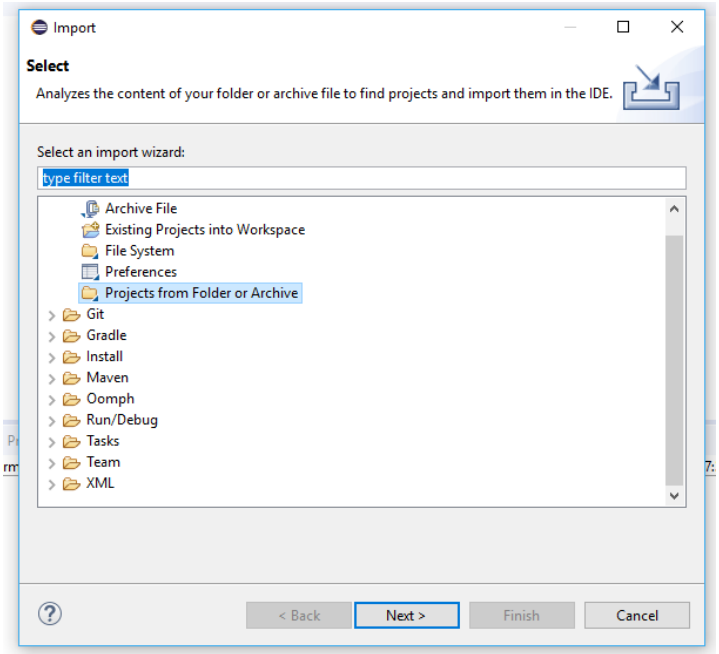
1. Download the AlgorithmPlugin-JAVA-11.zip file from the GitHub repository.

2. Start an IDE for your java project such as Eclipse.

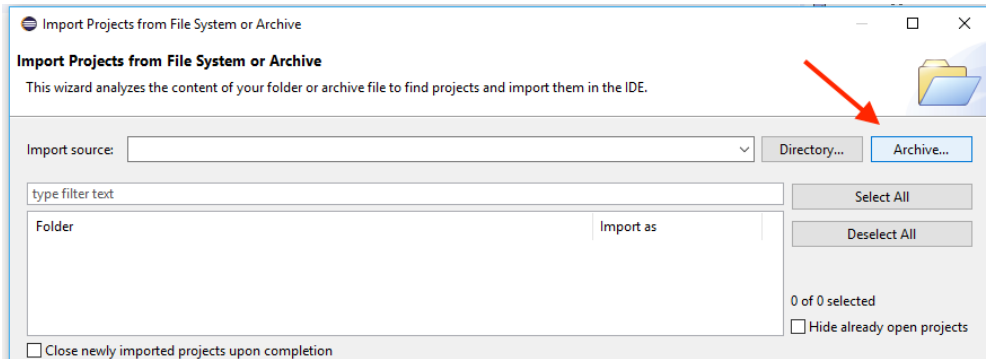
3. Right click inside the “Package explorer section” and select “Import”



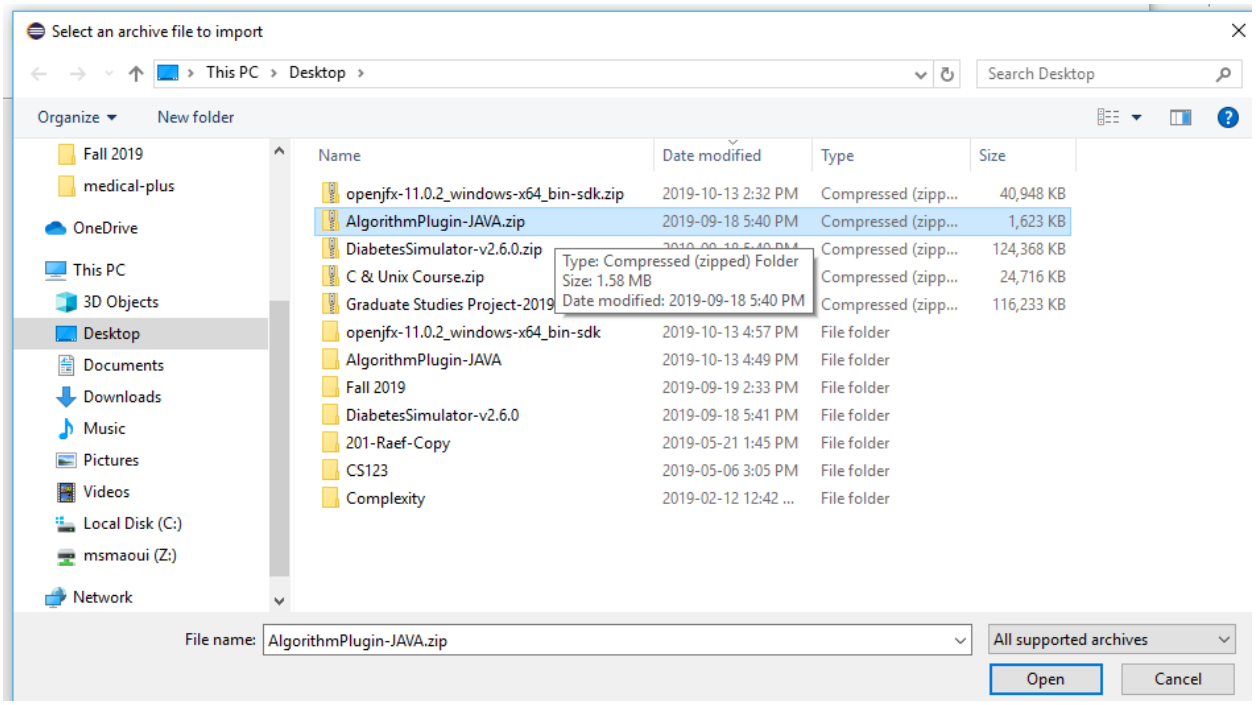
4. Under “General”, select “Projects from Folder or Archive”, and click “Next”



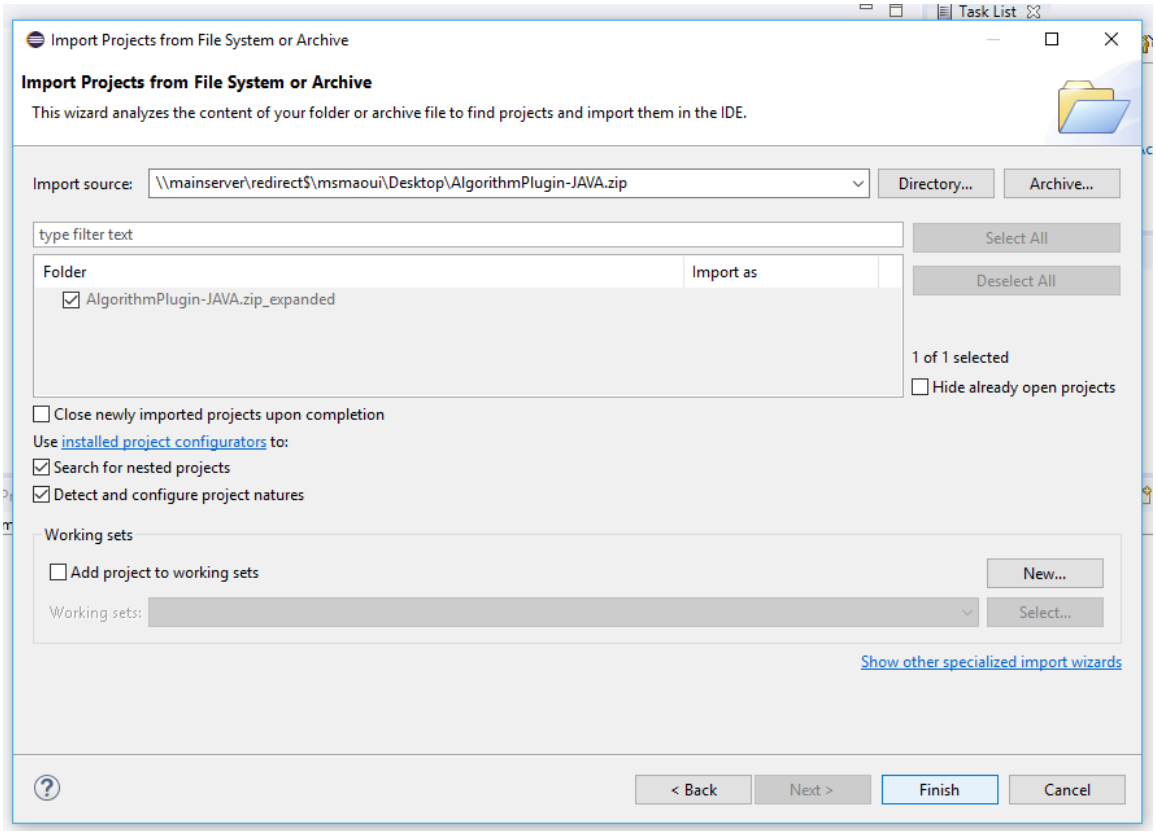
5. Select “Archive” from the next panel



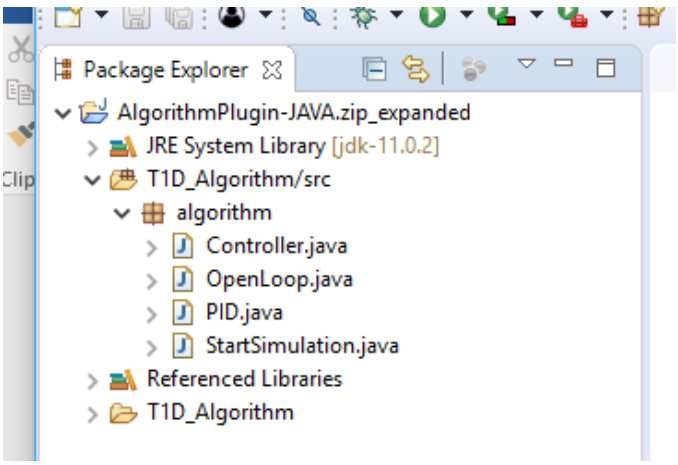
6. Select the “AlgorithmPlugin-JAVA-11.zip” file that you downloaded earlier from the GitHub repository, and click open.



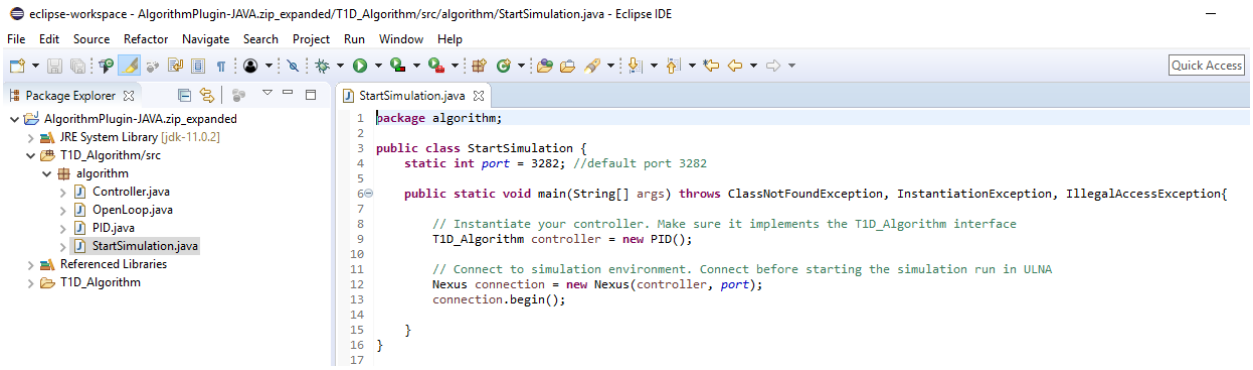
7. Select the “AlgorithmPlugin-JAVA-11.zip” file that you downloaded earlier from the GitHub repository. You should see something like the following. Click Finish.



8. You should be able to see a project open for you with 4 files: Conroller.java, OpenLoop.java, PID.java, and StartSimulation.java

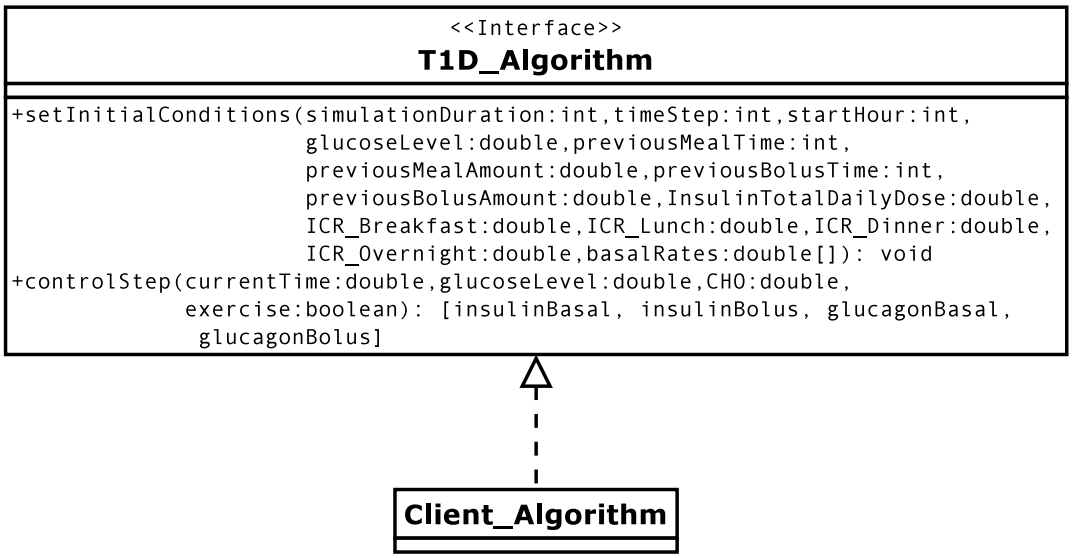


9. Double click on the StartSimulation.java file. This file is responsible for connecting your algorithm with the Development Platform (Ulna). Notice that this file instantiates a PID object puts it connection with Ulna.



You can change the PID() object to any other object that implements the T1D_Algorithm interface.

Your code will have to implement this interface to run.

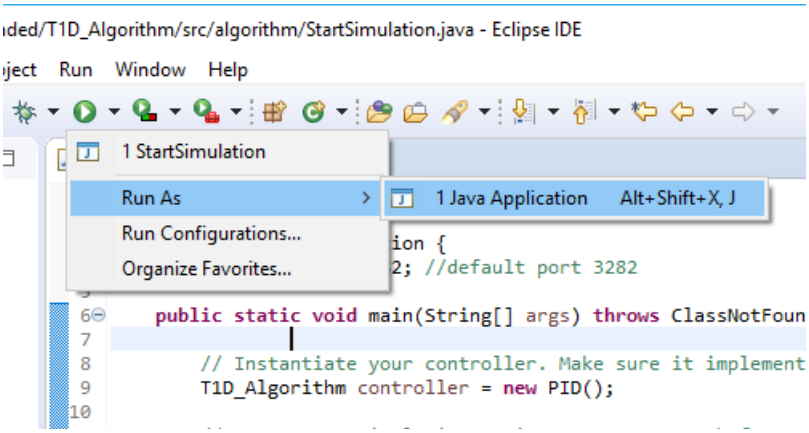


10. Open the PID file and explore the 2 functions that are implemented. The `controlStep()` function gets called every 10 simulation minutes to compute the Insulin measurement that should be given to the patient given their current glucose level and any meals.

    The algorithm is expected to use the current glucose reading and the meal information to make an educated measurement of what the next insulin basal amount should be. This result is then given back to Ulna and Ulna passes it to the virtual patient. Ulna will get the response back from the patient (glucose measurement), and in turn, provide this response back to the algorithm.

This loop of exchange continues until the simulation time is over.

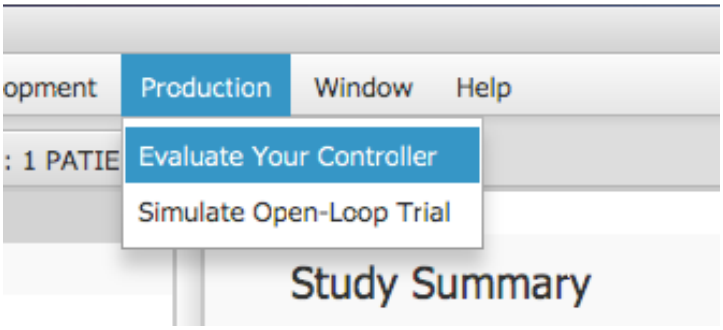
11. Run this file by clicking on the “Run Simulation” button, as shown:



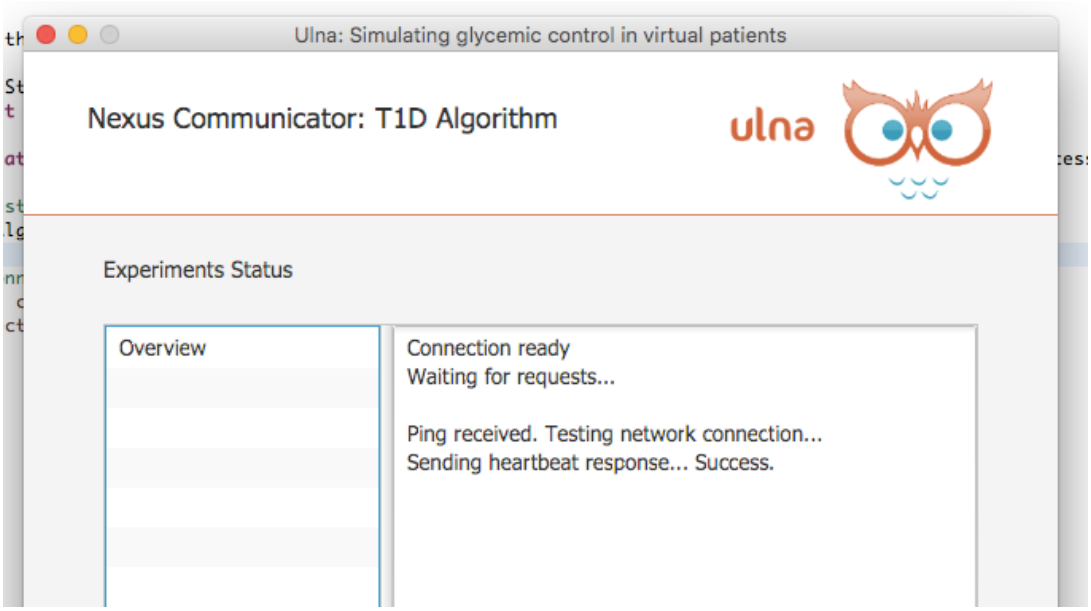
12. The algorithm PID is now on standby waiting for instructions from the Ulna program. You should see a dialog box (Nexus Communicator) that will log for you the communication between your algorithm and the platform.
13.

Section 3: Run a virtual trial

Go back to the Ulna program and click on “Production” -> “Evaluate Your Controller”.

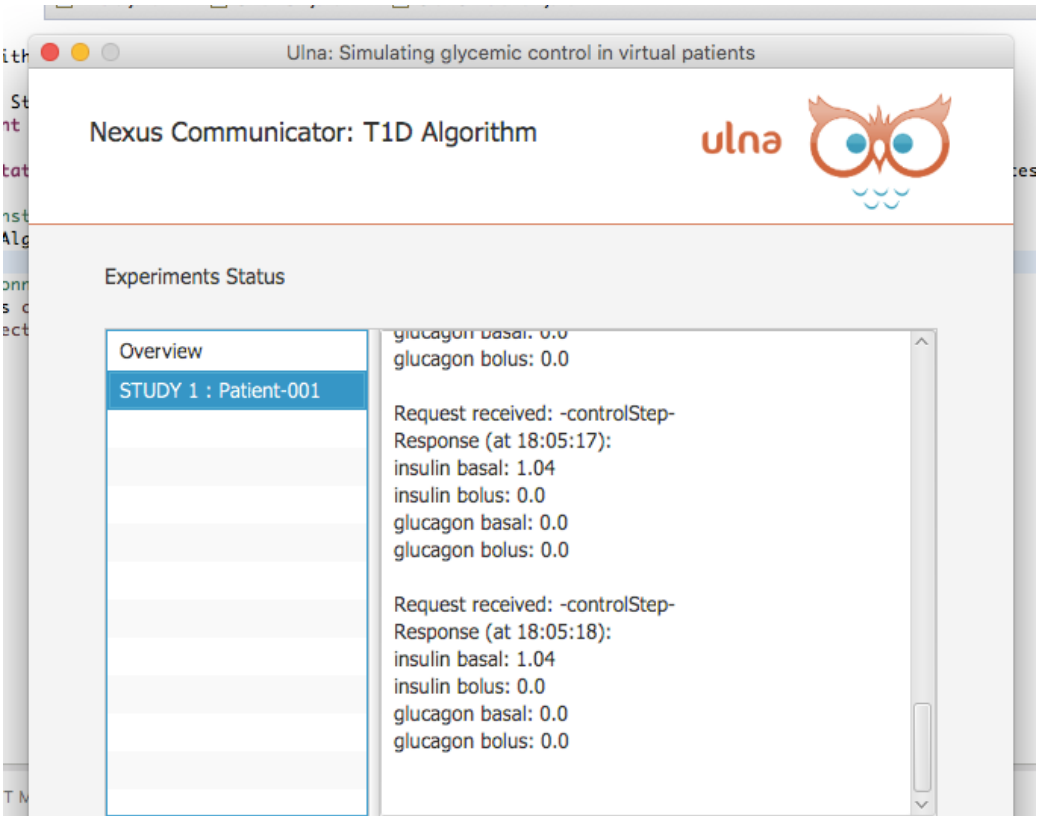


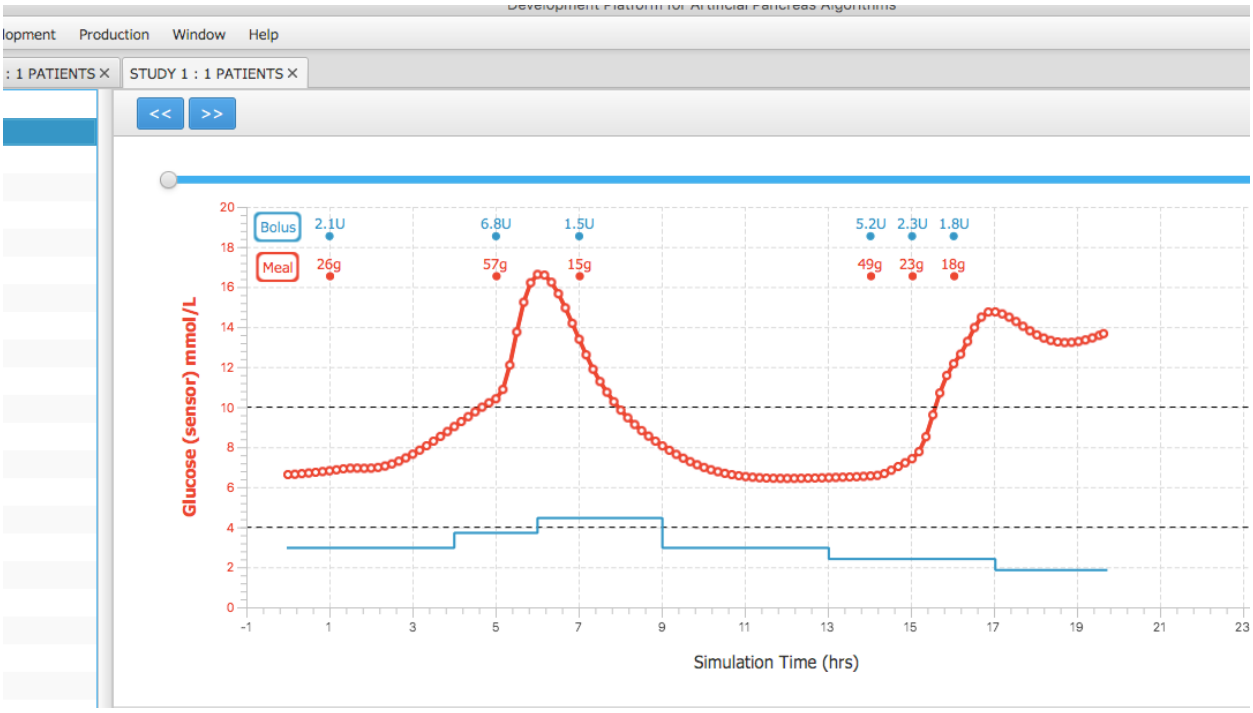
Ulna will now connect to the algorithm running on eclipse and you will see an updated message on the Nexus Communicator



Proceed with the virtual trial run. Select Patient-001 and click Next a few times to start the simulation.

Once the simulation starts, you can view the output of the algorithm on the Nexus Communicator and view the progress of the patient from Ulna


Ulna: Simulating glycemic control in virtual patients
Nexus Communicator: T1D Algorithm
ulna
Experiments Status
Overview
STUDY 1 : Patient-001
glucagon bolus: 0.0
Request received: -controlStep-
Response (at 18:05:17):
insulin basal: 1.04
insulin bolus: 0.0
glucagon basal: 0.0
glucagon bolus: 0.0
Request received: -controlStep-
Response (at 18:05:18):
insulin basal: 1.04
insulin bolus: 0.0
glucagon basal: 0.0
glucagon bolus: 0.0


Production
Window
Help
: 1 PATIENTS
STUDY 1 : 1 PATIENTS
<<
>>
Bolus
2.1U
6.8U
1.5U
5.2U
2.3U
1.8U
Meal
26g
57g
15g
49g
23g
18g
Glucose (sensor) mmol/L
Simulation Time (hrs)

Section 4: Report results

Report the following results of your simulation run. You should find these values on the "Study Summary Page":

Percentage Time in Target 4-8 mmol/L:
Percentage Time in Target 4-10 mmol/L:

Hypoglycemia Time < 4.0 mmol/L:
Incidence of Hypoglycemia events:

Hyperglycemia Time > 8 mmol/L:
Incidence of Hyperglycemia events: