Bug ID S-40: Time step of Export to CSV

Table of contents

1. Problem

There are two different ways of writing simBio simulation data to a CSV file:

1. Adding a csv maker to the model xml file
2. Selecting "Export to CSV" from the simBio File menu

When the CSV files produced by these two different methods are compared, the time ranges do not match; the data in the CSV file produced by the "Export to CSV" menu option starts later and finishes earlier than the data produced by the csv maker. The time ranges should be the same for both.

2. Solution

The time range of the "Export to CSV" option should be changed so that the initial and final times are the same as those of the csv maker.

3. Modified file list

- org.simBio.sim.analyzer.graph.AbstractGraph
- org.simBio.sim.analyzer.graph.BasicGraph
- org.simBio.sim.analyzer.graph.Graph
- org.simBio.sim.gui.CsvDialog

4. Code explanation

The early finish of the CSV data was fixed by changing a loop ending condition in CsvDialog.java and adding code to the end() method of Graph.java. The csv maker has code in the CsvMaker end() method to write the final data to the CSV file. "Export to CSV" did not have such code, and so failed to write the final data to its CSV file.

In order for "Export to CSV" to calculate and store the final data, a call to analyze() has now been added to the end() method of Graph.

```
analyze(elapsedTime);
```

Also, this code in the csvMaker method of CsvDialog

```
for ( long t = idxMin; t < idxMax; t++ ) {
```

was changed to this

```
for ( long t = idxMin; t <= idxMax; t++ ) {
```

so that the program will continue writing data to the CSV file until it reaches the end time, idxMax.

The late start of the data in the CSV file was fixed by changing code in the prepare() method of AbstractGraph from this

```
maxTime = 0;
```

to this

```
maxTime = -1;
```

The above change stops the following code in the analyze() method of AbstractGraph from returning without doing any calculations when time t is 0.

```
if (t <= maxTime) {
        log.warn("time=" + t + " : ignored. (current max time=" + maxTime + ")");
        return;
}
```

In the original code, when time t was equal to 0, maxTime was also equal to 0, and so the analyze() method returned without doing any calculations for time 0. Therefore the data for time 0 was missing from the CSV file. Now however, when the time t is equal to 0, maxTime is equal to -1, so the condition in the above code is not met, and the program continues to execute the rest of the code in the analyze() method. The unnecessary code below was also deleted from the prepare() method of BasicGraph.

```
maxTime = values.getTimeMax()
```