AequilibraE for QGIS

Pedro Camargo

Oct 30, 2025

CONTENTS

CHAPTER

ONE

GETTING STARTED

In this section we describe how you can install AequilibraE's QGIS plugin.

Note

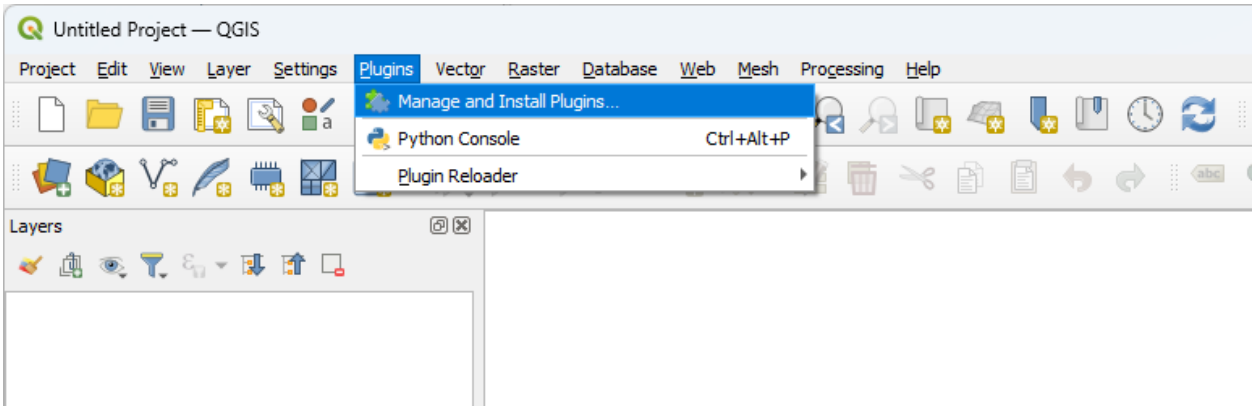
The recommendations on this page are as current as of April 2024.

1.1 Installation

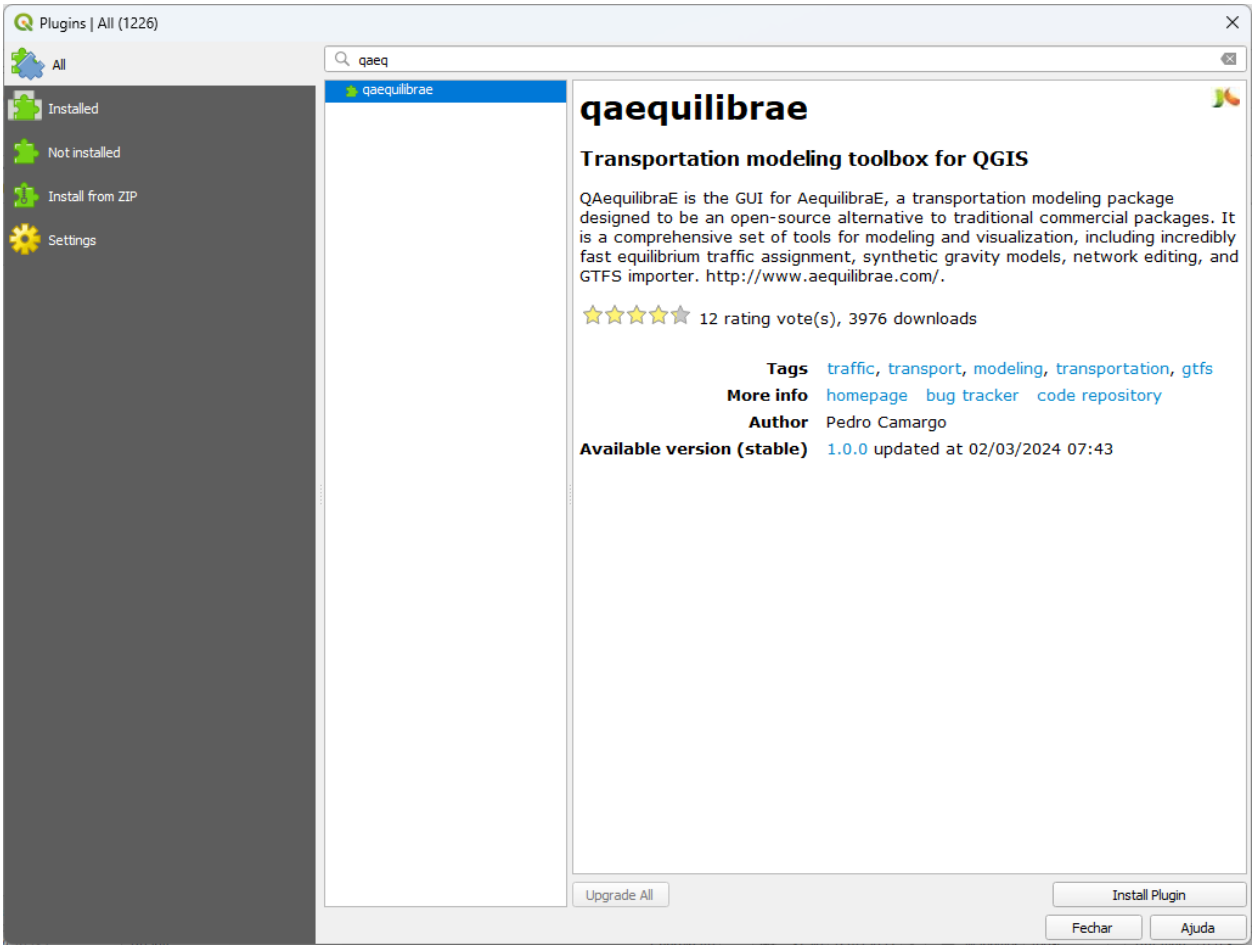
AequilibraE is available from the QGIS plugin repository, and we recommend you download it using the instructions below.

1.1.1 Step-by-step installation

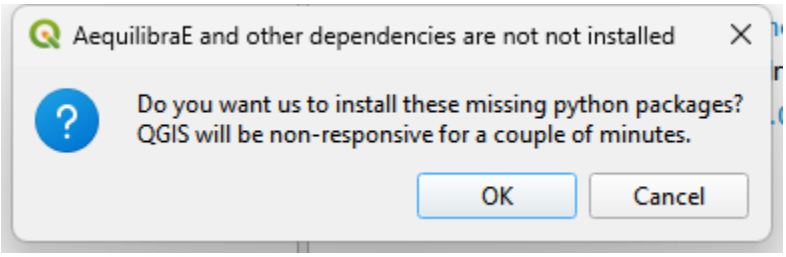
The steps for installing AequilibraE are the same as for any QGIS plugin. Go to the Plugins panel and click on **Manage and Install Plugins**.



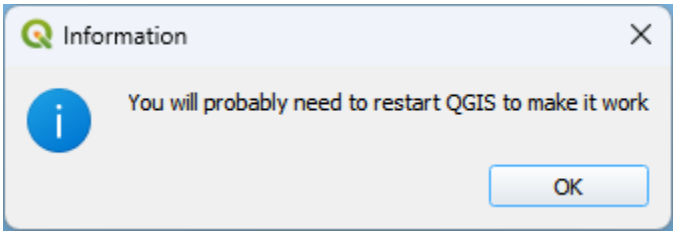
In the tab *All*, search for QAequilibraE.



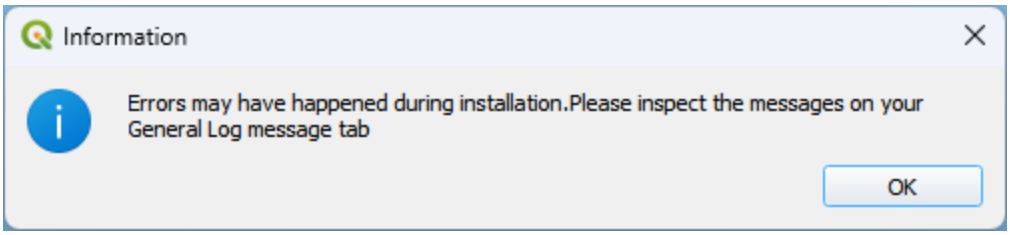
After selecting the plugin installation, you will be faced with the question of whether you want to download its dependencies, which are required for using most of the features. This is necessary because AequilibraE's algorithms rely on compiled extensions, but it is against the QGIS's community guidelines to upload binaries to the repository.



If you select to download the packages, QGIS will freeze for a few seconds before showing the image below.



Otherwise, a message warning about installation problems will be shown, and your plugin will be non-functional.

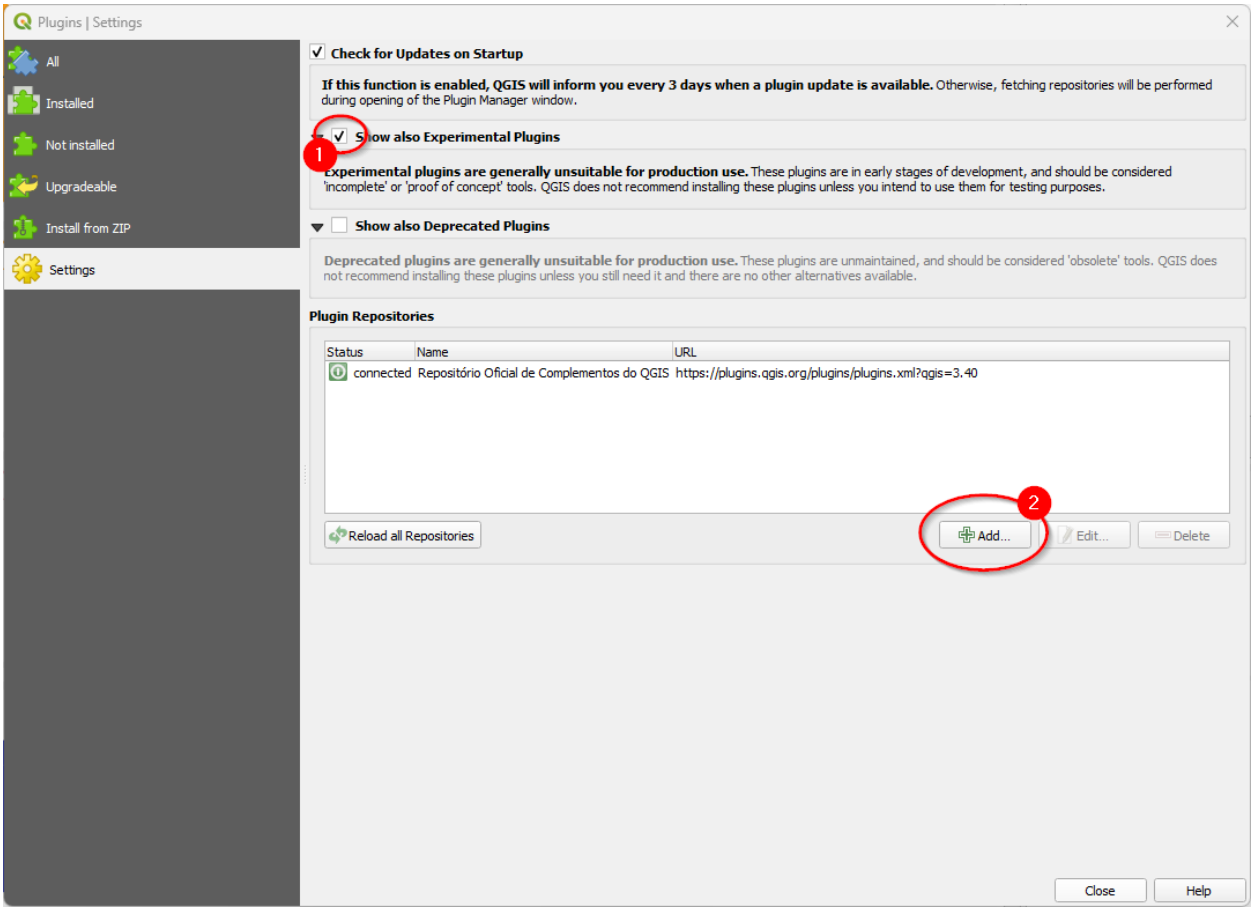


1.2 Plugin Repository

With the plugin repository, it is now possible to use the latest version in develop without waiting for the next release!

To configure it, let's go once again to the Plugins menu, as shown *here*.

Select the **Settings** tab and check the box for **Show also experimental plugins** (step 1). The versions of QAequilibraE made available at the plugin store are not the versions for release and are labelled as experimental.

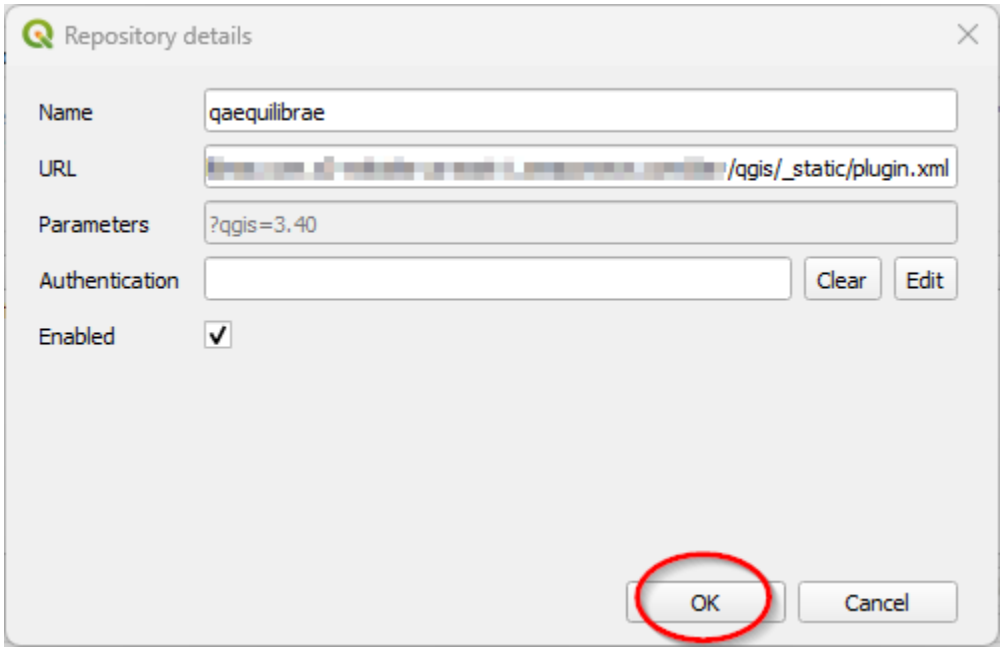


Then, click on the **Add** button (step 2). A new window will open. Fill the name and URL fields with the following data:

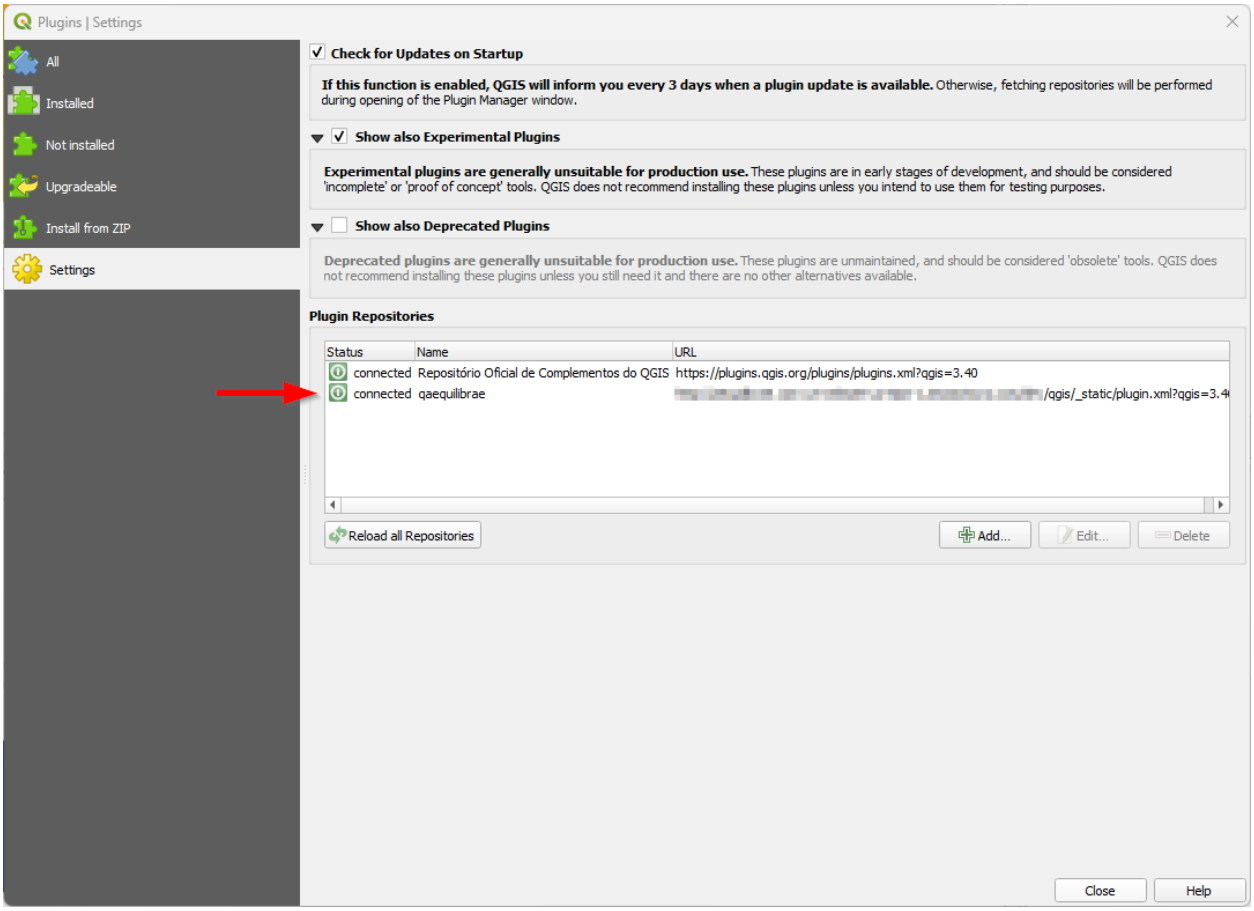
Listing 1: Plugin Repository

```
name: qaequilibrae
URL: https://aequilibrae.com/develop/qgis/_static/plugin.xml
```

Then, just click on the **OK** button.

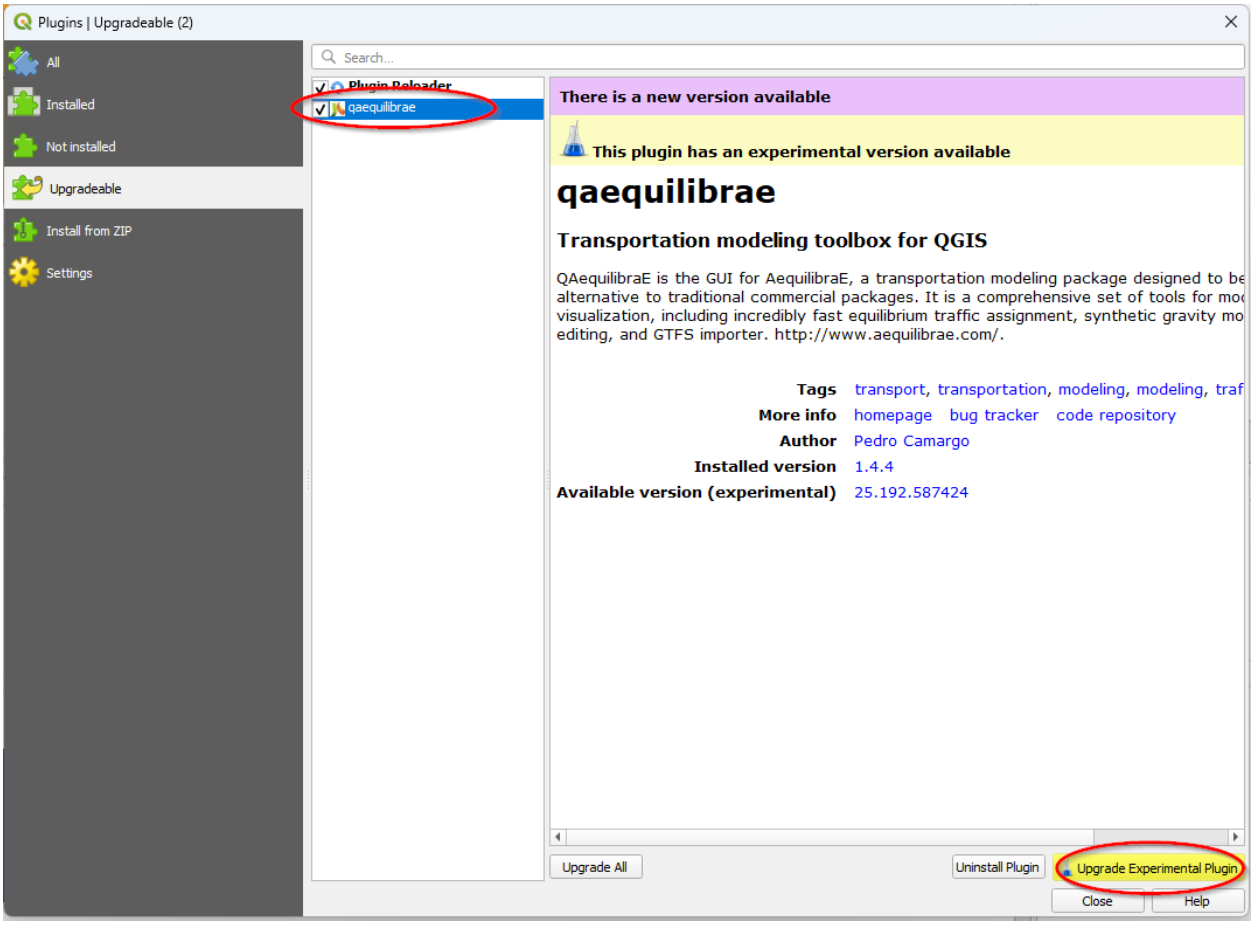


The QGIS is going to validate the provided URL. It should be really quick. You'll notice that qaequilibrae is now appearing at your plugin repositories.



Finally, select the **Upgradeable** tab. You'll notice that QAequilibraE has a newer version to be installed. Click on

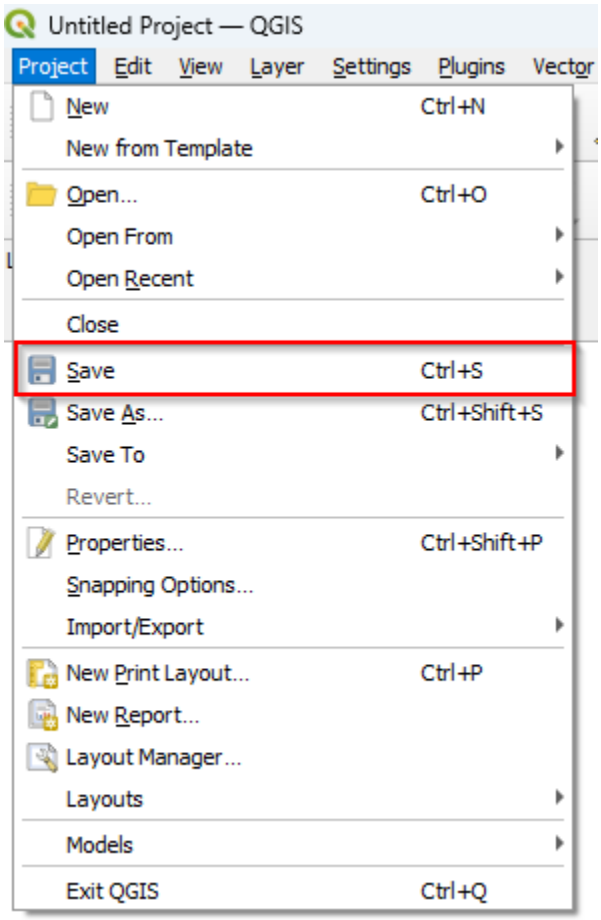
qaequilibrae, and hit the **Upgrade Experimental Plugin** button. The installation process should be the same as above, as you need to allow the installation of external libraries.



If you have any problems with this step, please check the official QGIS documentation.

1.3 Saving as QGIS Project

Since version 1.0.1, our users can save their on-going projects directly through the QGIS saving menu! This feature allows you to save both your AequilibraE project and temporary layers. The temporary layers are stored in **qgis_layer.sqlite**, a database automatically created to store these layers. All you have to do is go to the Project panel and select **Save** or **Save as**, indicate where you want to store your project file, and press save!



In the interest of data integrity, if you have open AequilibraE layers into your QGIS Project and close the AequilibraE project, these layers are removed from your open QGIS project.

When reopening the QGIS project containing an AequilibraE model, you will notice that the project stored is automatically reopened by QAequilibraE.

CHAPTER

TWO

MENUS IN DETAIL

In this section you can find a deep dive into modeling with AequilibraE, as a start guide to a complete view into AequilibraE's data structure. Each and every one of the following sub-sections are related to one AequilibraE's main functionalities.

Most screenshots in this section are related to Sioux Falls, a standard example in transportation network algorithm studies.

In the following sections, we present an illustrated example of a realistic modeling workflow for the beginner modeler out there. You can easily create a Sioux Falls model using our *create example* tool. The remaining of the data used in the examples can be downloaded here.

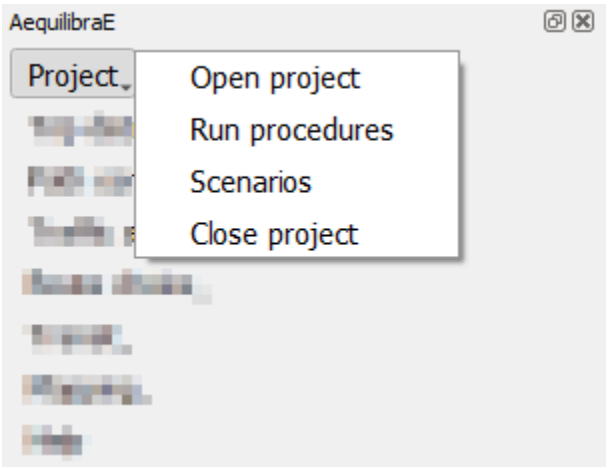
As to not upset those who think that Sioux Falls is not a realistic example (you would be right to think so), the example data is also available for the Chicago regional model, which has nearly 40,000 links and almost 1,800 zones.

Got a bigger instance we could use as an example? Send it over!

2.1 AequilibraE Project

This page is dedicated to a practical implementation of the AequilibraE project. In case you are interested in better understanding its structure, please visit its documentation webpage.

Under the project menu, there are some options to choose from and the following sections explore some of these actions.



2.1.1 Open & Close project

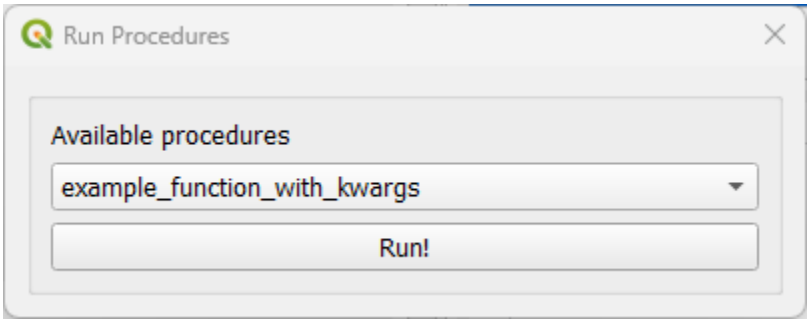
These options are pretty straightforward and are used either to open or close a project. You just have to click **Project > Open project** to open a project, and **Project > Close project** to close it.

Keep in mind that to open another project or to create a new one, you **must** close the currently open project, otherwise AequilibraE is going to return an error.

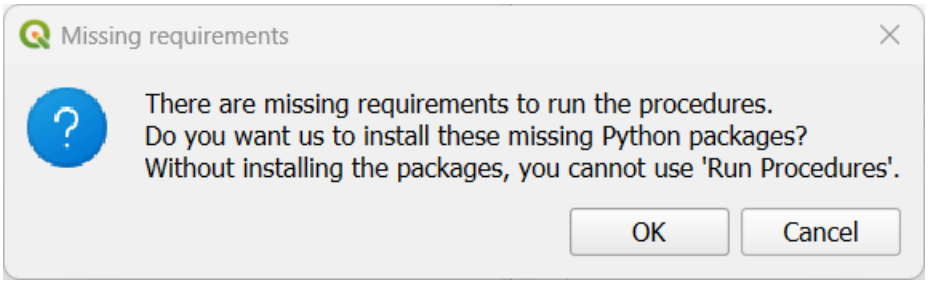
2.1.2 Run procedures

The run procedures allows you to define model entry points and their default arguments, and run models to the model itself. Usage at QAequilibraE is pretty straightforward: select one of the available functions, click on the *Run!* button, and wait for the log file to open with the output results of the model.

To better understand the application of the run module, we encourage you to read about it at the AequilibraE documentation.



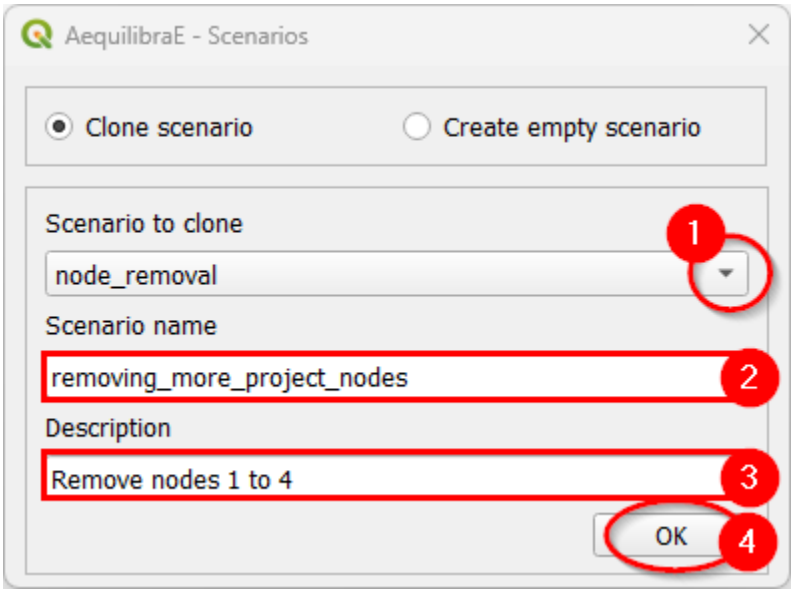
It is possible to use one of the default functions or create your own, including function calls to external libraries. To do so, don't forget to include a `requirements.txt` file with the dependencies in your project's run folder. The next time you open 'Run procedures', a message box asking about dependencies installation will open. If you choose to install the dependencies, the process resembles the one when installing QAequilibraE. Wait until it's complete, and restart the 'Run procedures' to validate the installation.



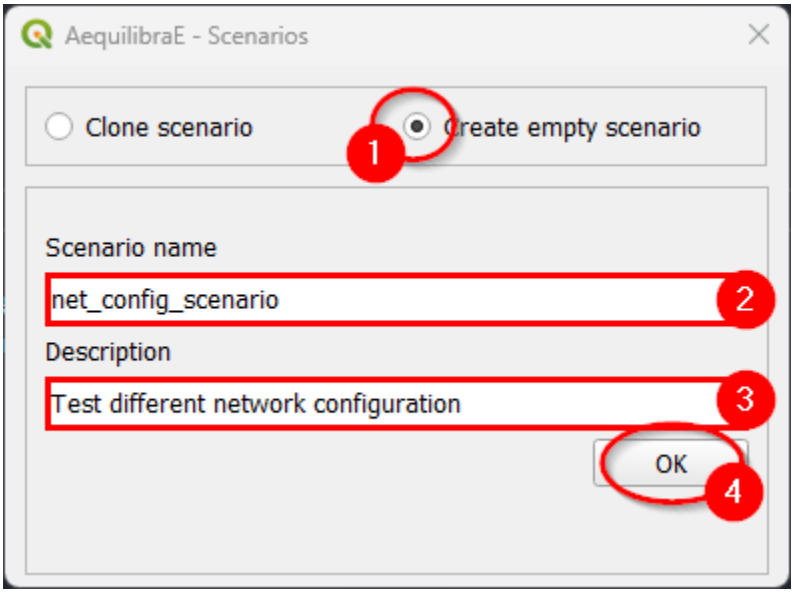
2.1.3 Scenarios

QAequilibraE now presents a scenario system, in which you can manage multiple scenario variants within a single project.

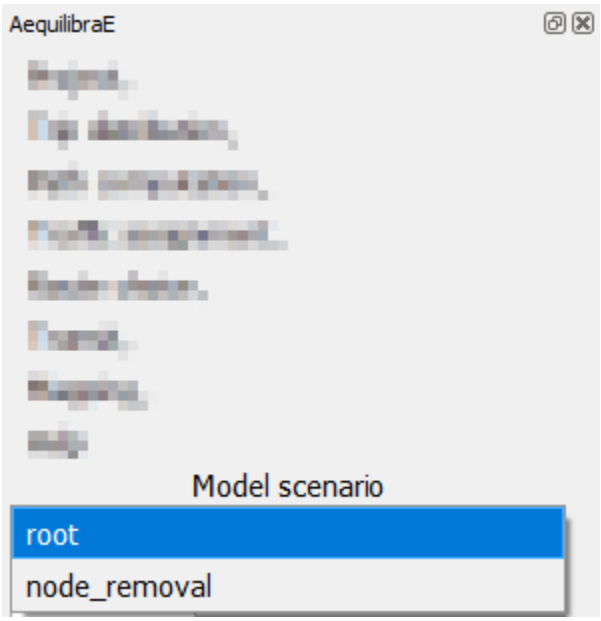
When a project is created, its default scenario is 'root'. QAequilibraE allows you to clone a scenario or create an empty scenario. To clone a scenario, you first choose the base scenario to clone (1) and the name of the scenario (2). An useful scenario description can also be added at the 'Description' box (3). By default, the scenario to clone is the currently active scenario, but you can choose anyone. To clone the scenario, just click on the 'OK' button at the bottom of the screen (4).



To create an empty scenario, choose the 'Empty scenario' option (1), and set the scenario name (2) and description (3). To create an empty scenario, just click on the 'OK' button at the bottom of the screen (4).

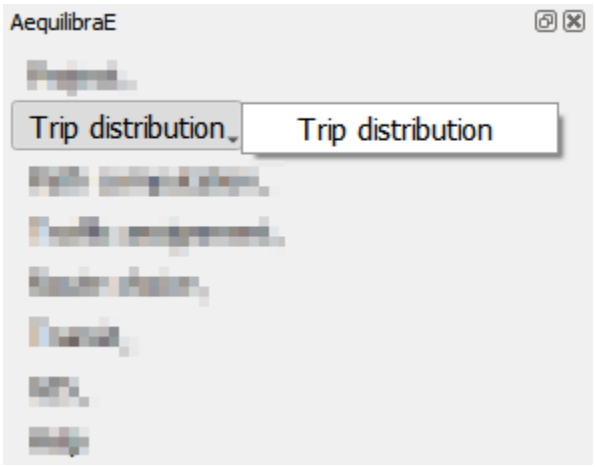


A list containing all project scenarios is presented at the bottom of the widget screen, and it can be used to change the currently open scenario. When changing the scenario, all geometric layers available at the "Geo layers" tab also change.



2.2 Trip Distribution

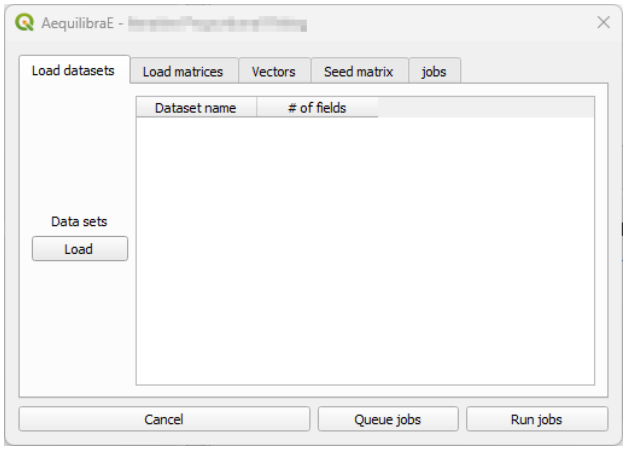
On the trip distribution menu, the user can perform Iterative Proportional Fitting (IPF) with their available matrices and vectors, as well as calibrate and apply a Synthetic Gravity Model.



Unlike the other menus in QAequilibraE, all three procedures in trip distribution share some configuring stepa. We'll go over each tab and, in the end, we'll run a basic workflow using Sioux Falls example.

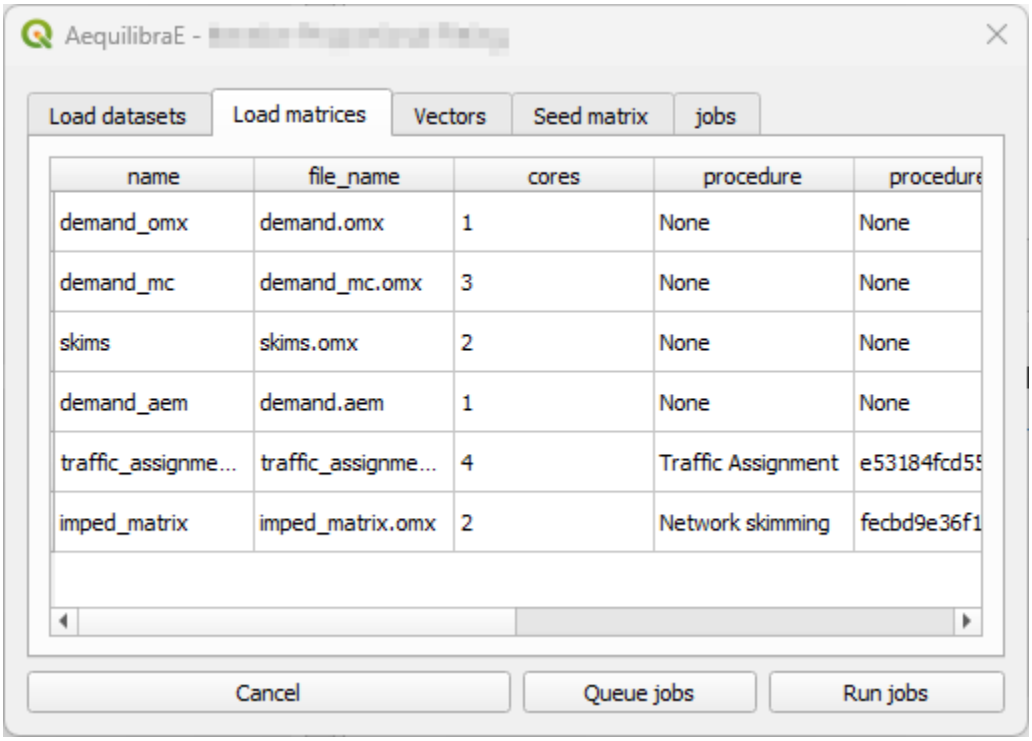
"*Load datasets*" is the first tab and contains a loading button and a dataset table at the right side. Currently, QAequilibraE allows import dataset data from a `*.csv` or `*.parquet` file or loading data from an open layer. This tab is configured for IPF and Apply Gravity.

The second tab is "*Load matrices*", which is configured in all processes. It consists in a table view of all matrices available in the project.

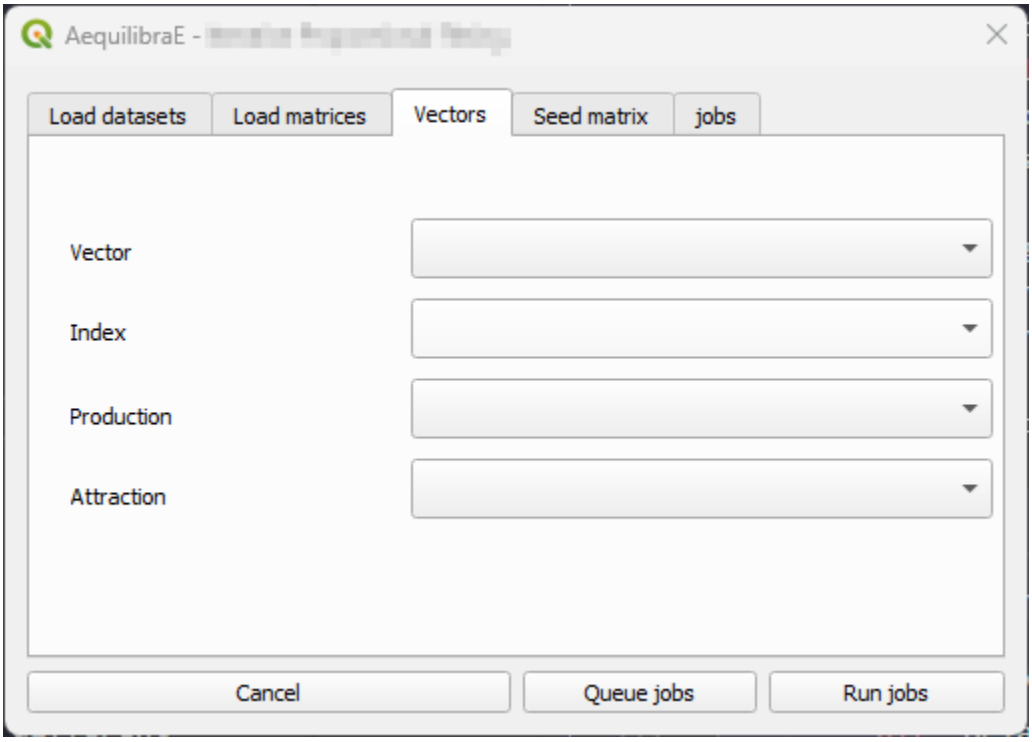


(a) Load datasets

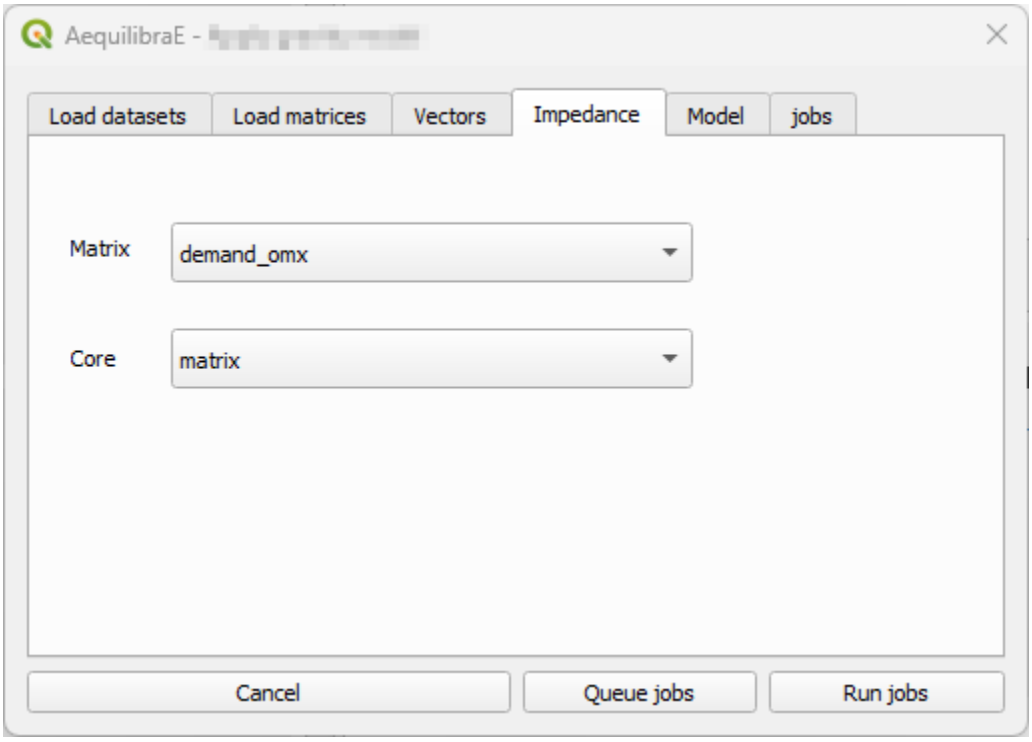
(b) Dataset file format



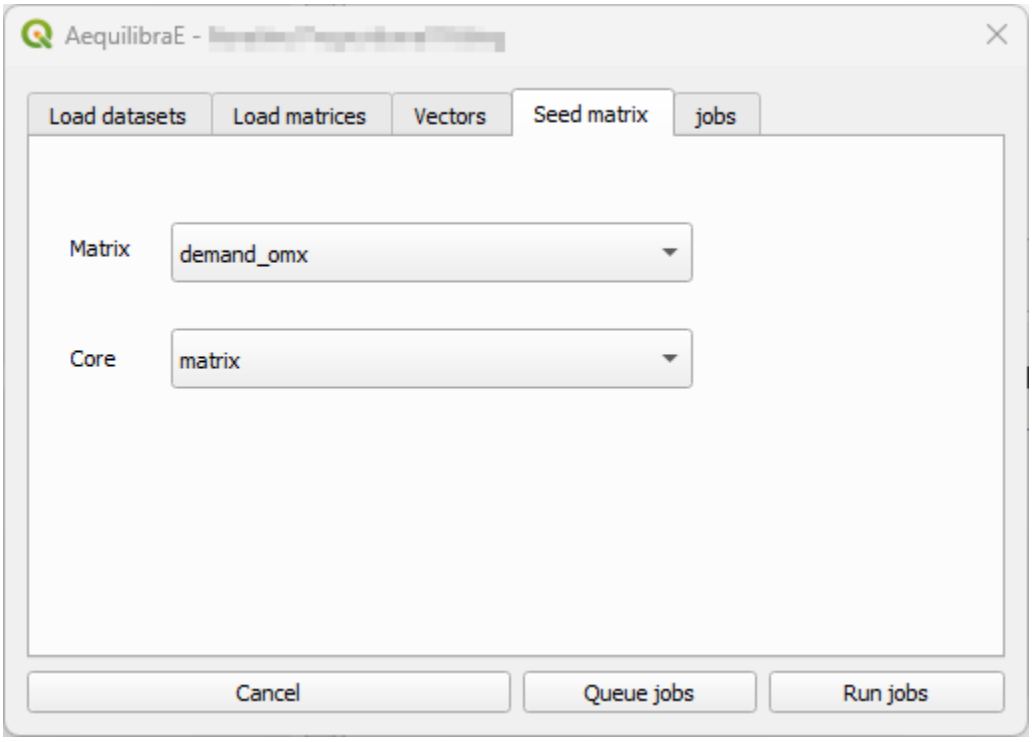
In the tab "*Vector*", we indicate the vector fields for computation. If no dataset was loaded in the "*Load datasets*" tab, no fields are displayed here. This tab is configured for IPF and Apply Gravity procedures.



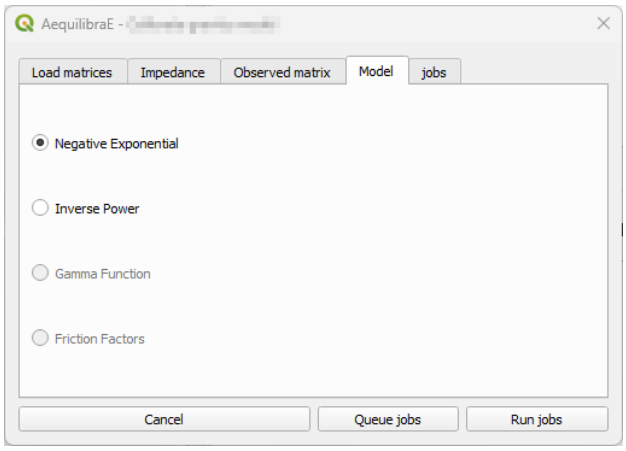
In the tab "*Impedance*" we select the matrix and matrix core that will be used for computation. We configure this tab at the Apply Gravity procedure.



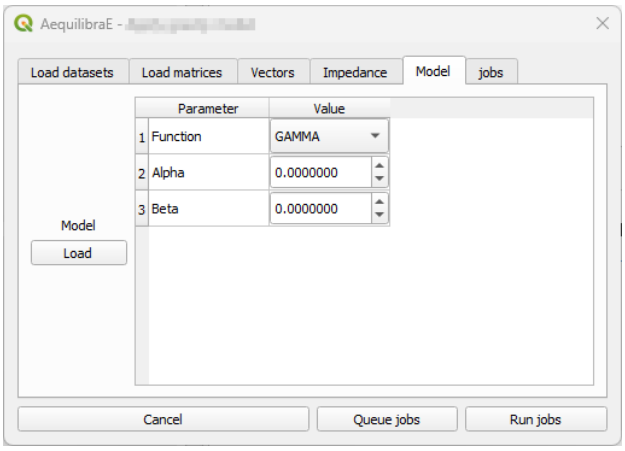
The tab "*Seed matrix*" (for IPF procedure) is analogous to the "*Observed matrix*" tab for the Calibrate Gravity procedure, and allows the user to indicate the impedance/observed matrix.



The tab "*Model*" exists for Calibrate and Apply Gravity procedures, however each procedure presents a different window layout. For the Calibrate Gravity, we choose the model's deterrence function, while for the Apply Gravity, we can load the calibrated model parameters for use.

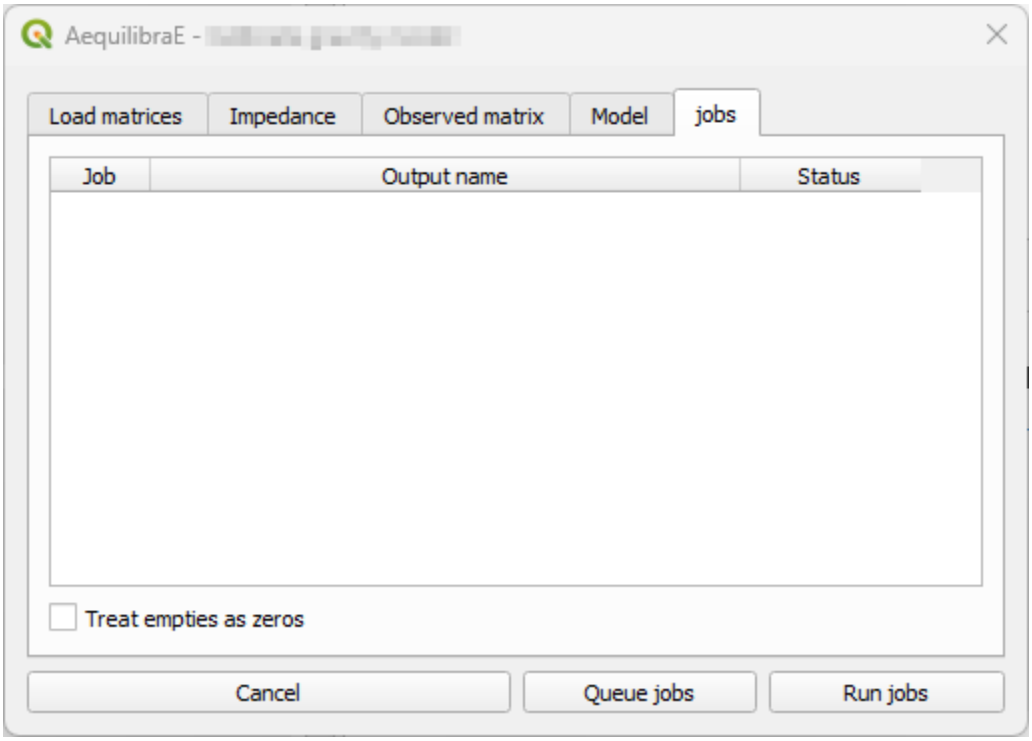


(a) Model tab - Calibrate Gravity



(b) Model tab - Apply Gravity

Finally, the tab "*Jobs*", we can queue and/or check the jobs that are already queued and are going to be executed, and run them!



2.2.1 Basic workflow

We present a full forecasting workflow using the Sioux Falls example. We start creating the skim matrices, running the assignment for the base-year, and then distributing these trips into the network. Later, we estimate a set of future demand vectors which are going to be the input of a future year assignnment.

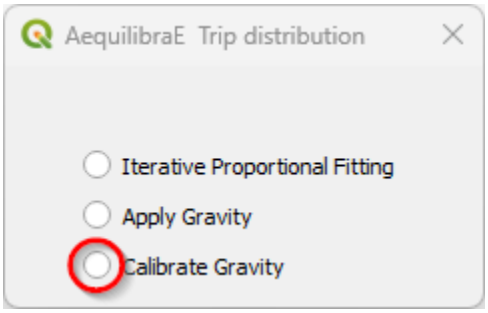
This workflow is based on the AequilibraE Python Forecast example.

Before running the trip distribution procedures, we encourage you to run the *traffic assignment procedure* for the base-year.

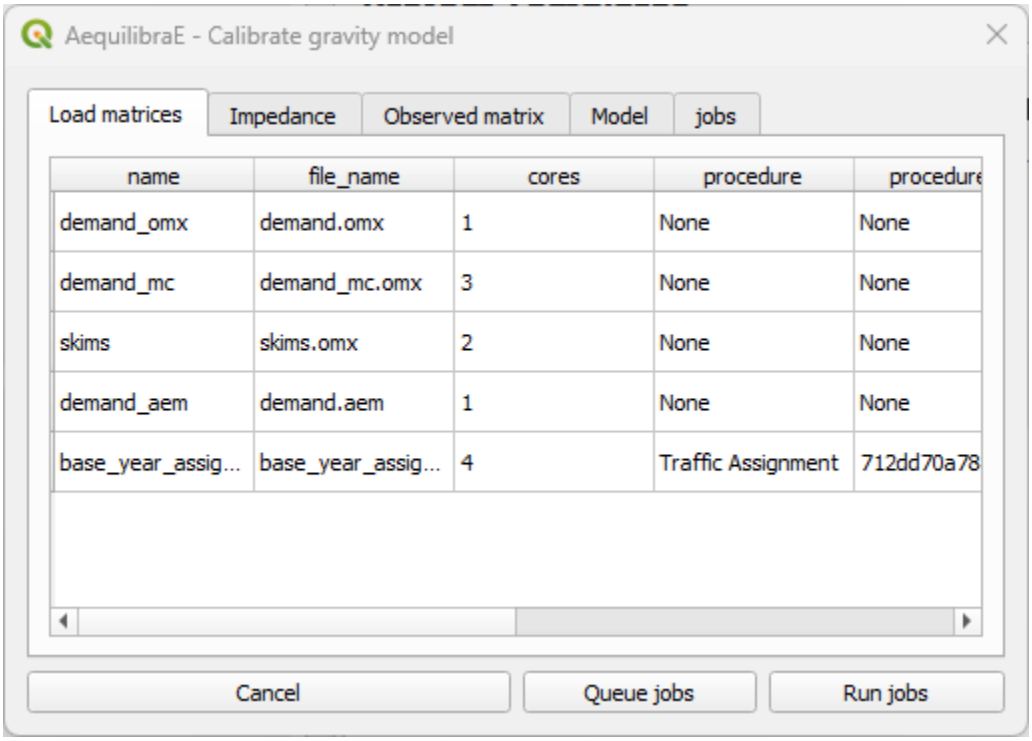
Calibrate Gravity Model

Now that we have the demand model and a fully converged skim, we can calibrate a synthetic gravity model.

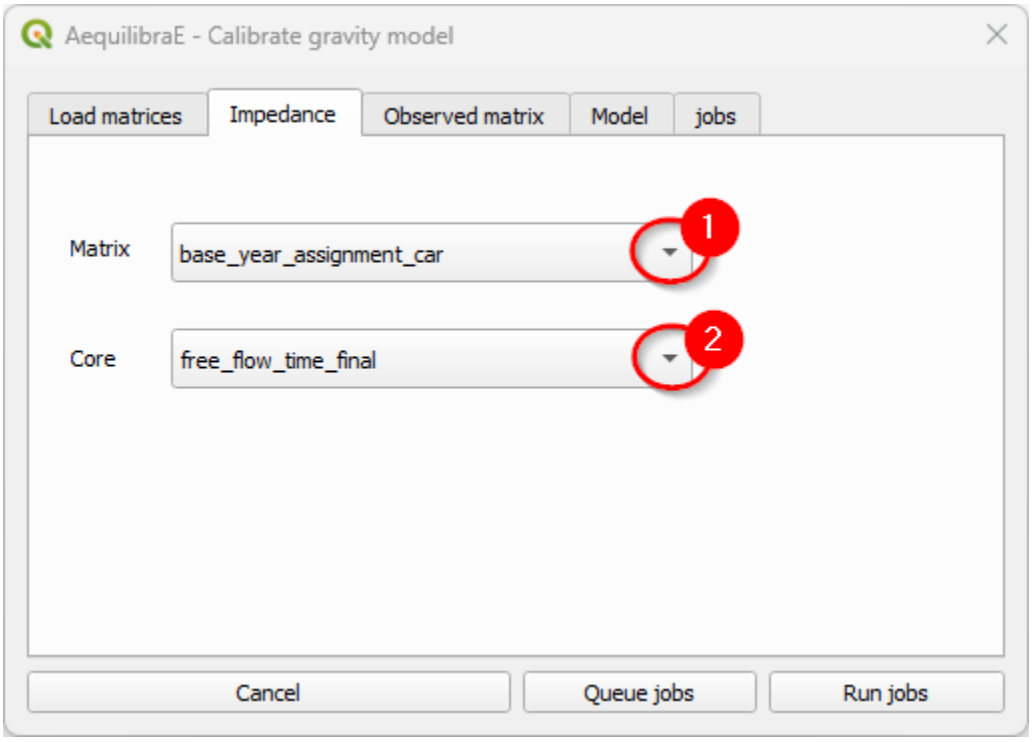
We click on Trip distribution in the AequilibraE menu and select the Calibrate Gravity model option.



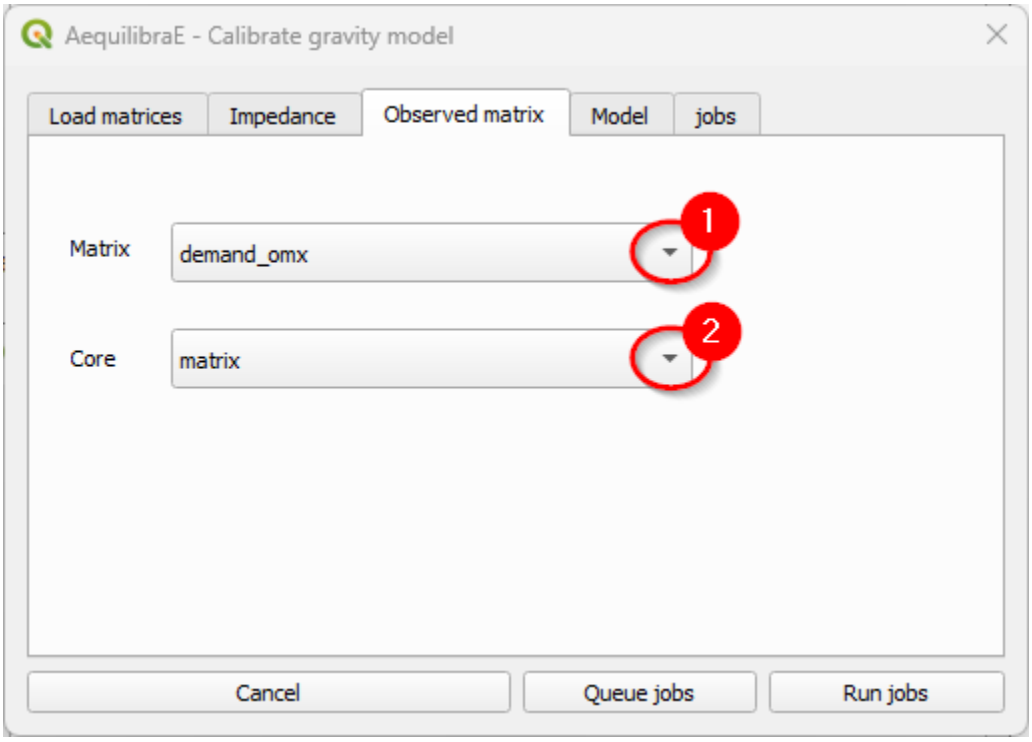
The first thing to do is to check if all matrices we need (skim and demand) are in the project folder.



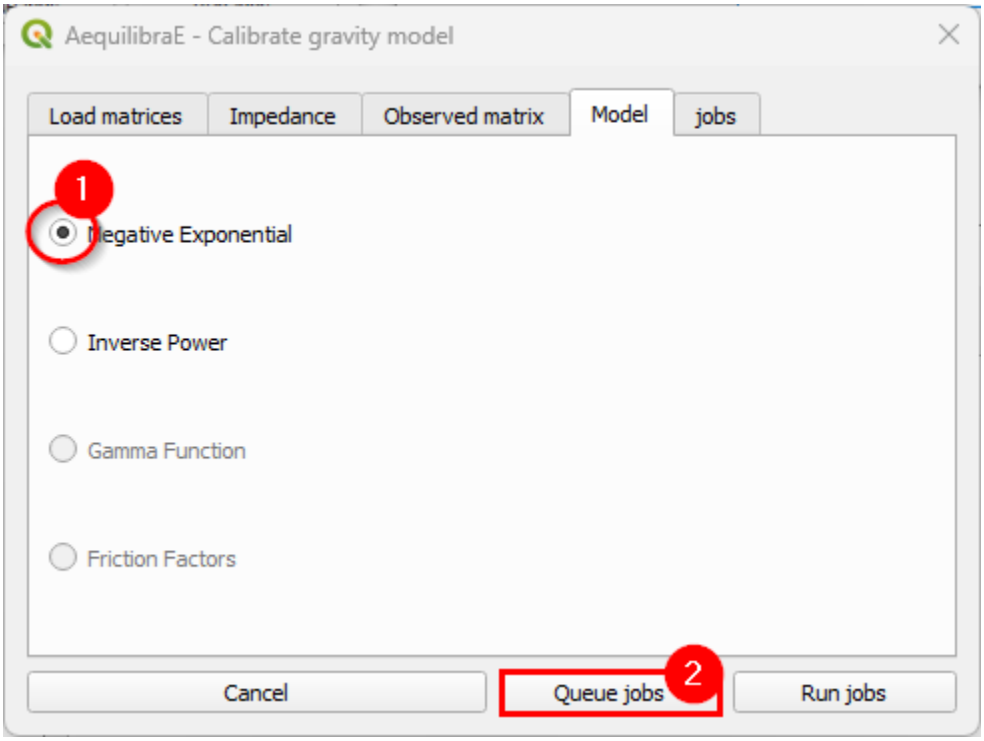
Select which matrix/matrix core is to be used as the impedance matrix.



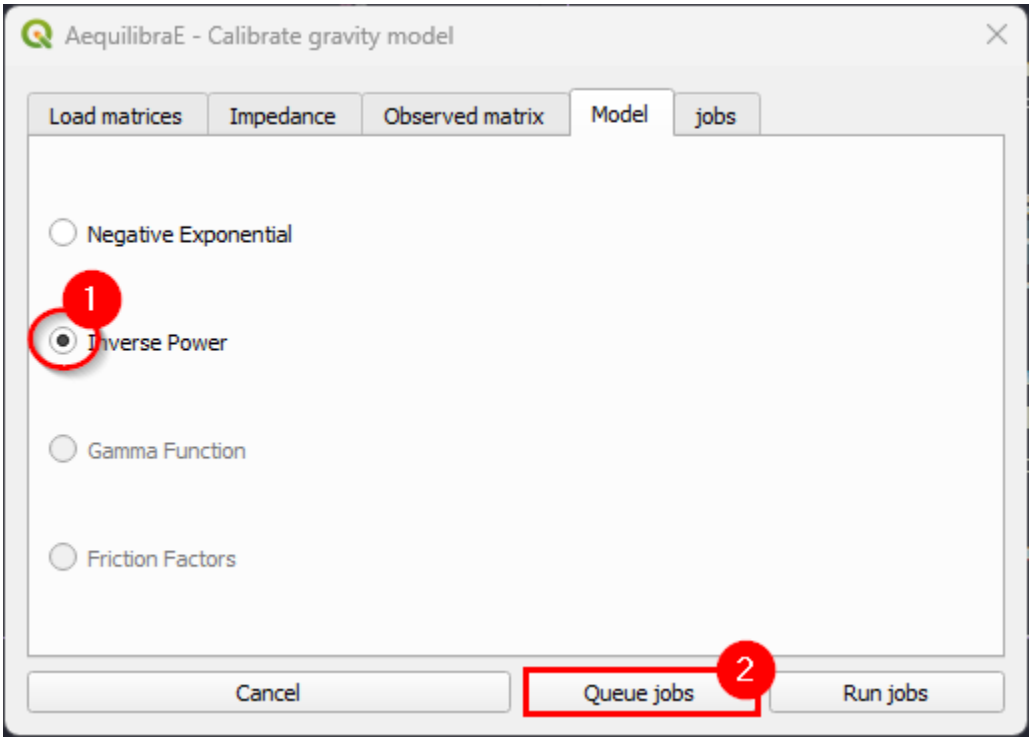
And which one corresponds to the observed (demand) matrix.



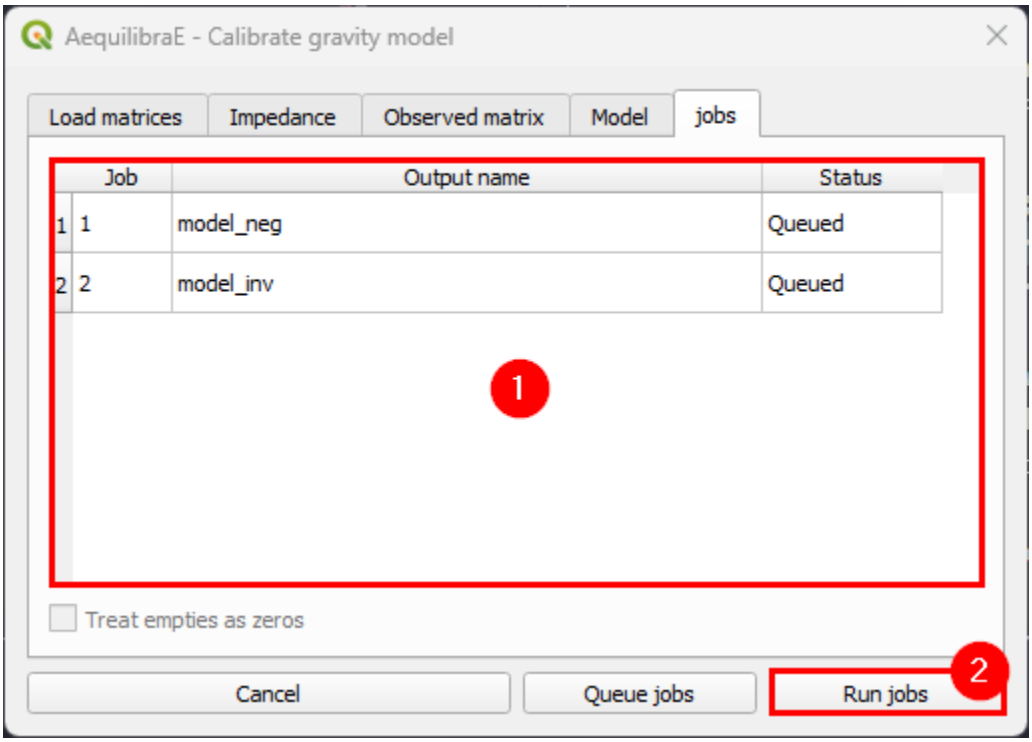
We then select which deterrence function we want to use (1) and choose a location to store the model by clicking on *Queue jobs* (2). A new window will open and you can choose your preferred place. Remember to pick up a place where you can easily find the model files: we'll use them to apply the gravity model.



Let's queue another job.

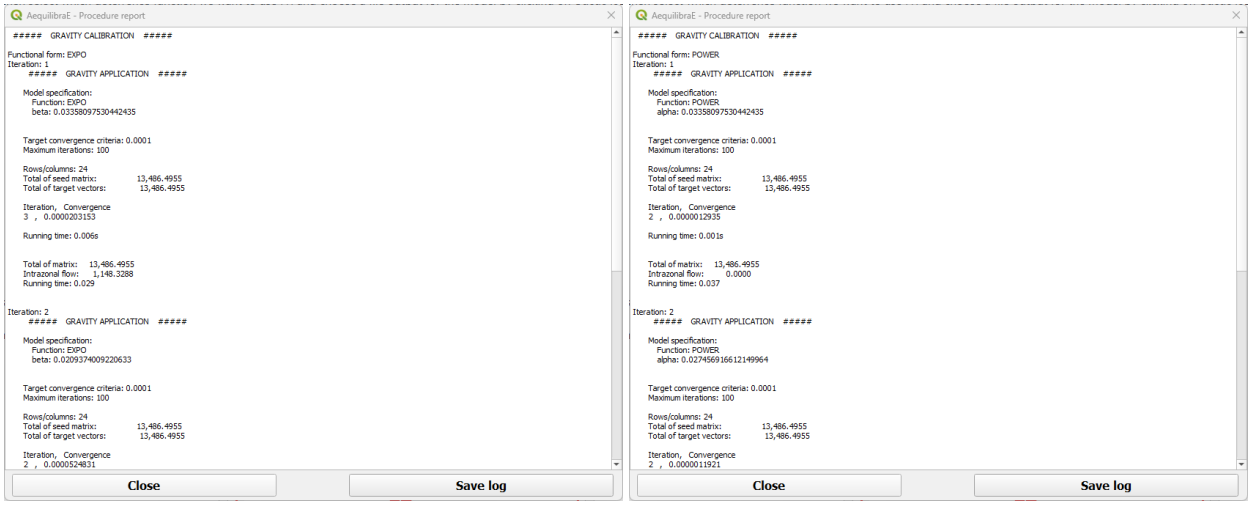


In the jobs tab, we can check all jobs we queued (1) and then run the procedures (2). You need to click once in the button to execute all of them.



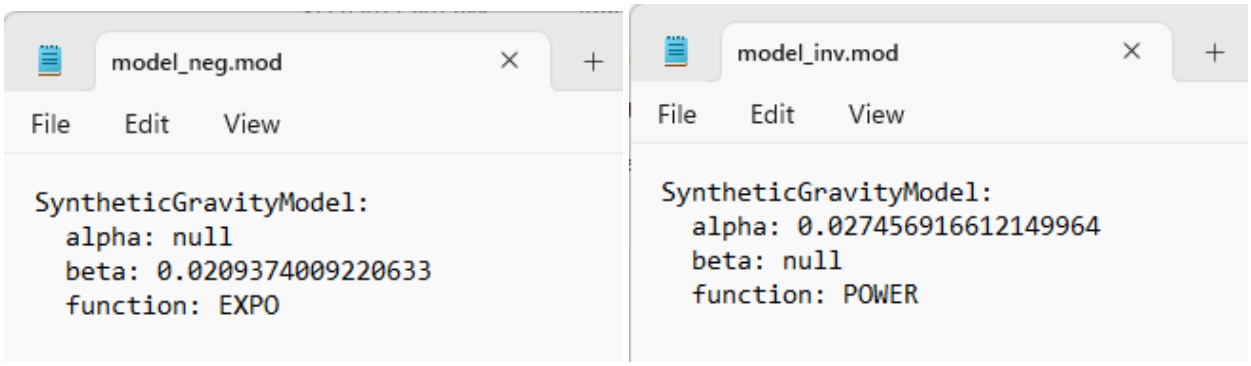
When the procedures are done, a window with each of the procedures report opens. You can inspect the outputs and save them.

The resulting file is of type `*.mod`, but that is just a YAML (text file).



(a) Negative exponential procedure output

(b) Inverse power procedure output

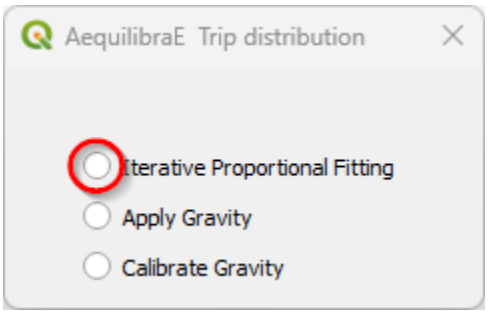


(a) Negative exponential model

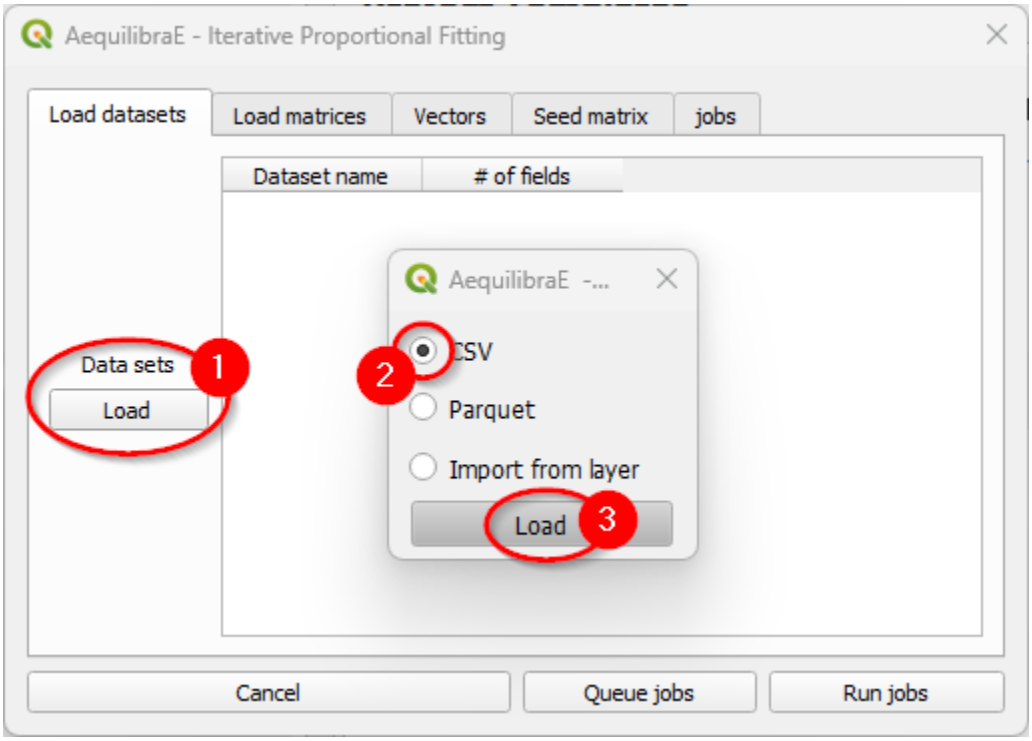
(b) Inverse power model

Iterative Proportional Fitting (IPF)

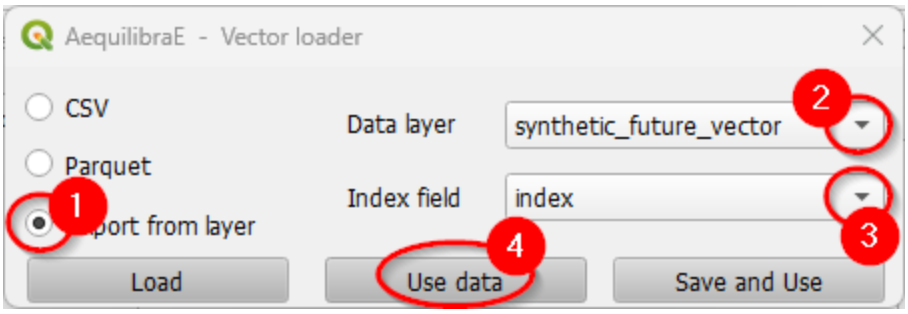
It is possible to balance the production/attraction vectors using Iterative Proportional Fitting (IPF). Let's click on the Trip Distribution menu and select Iterative Proportional Fitting.



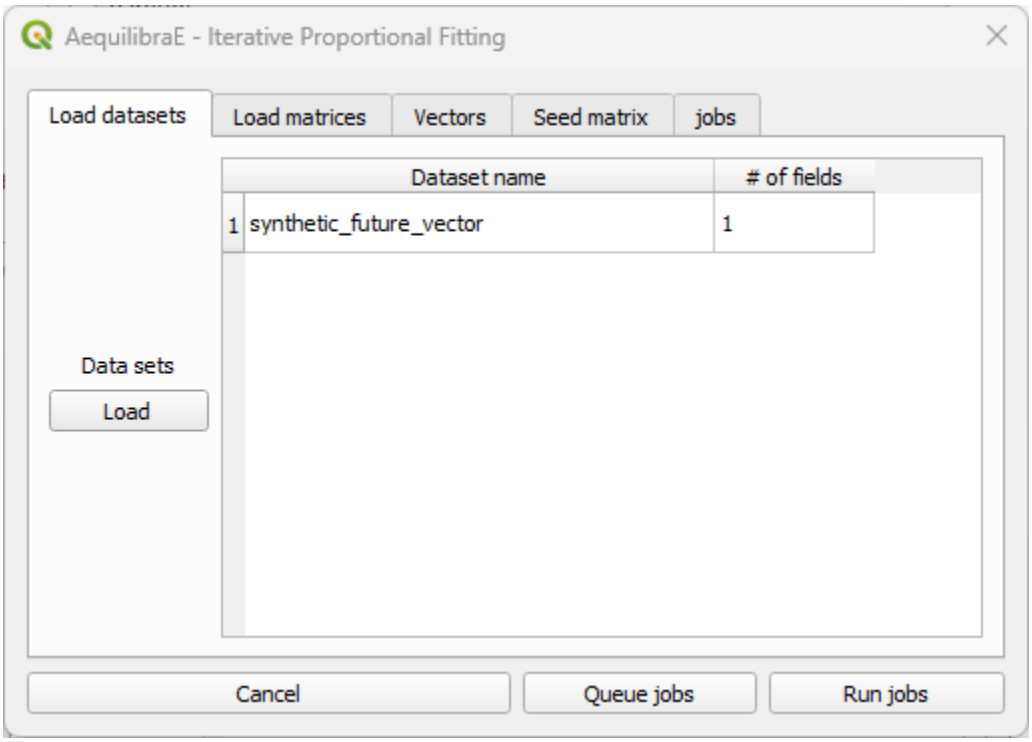
There are three different ways to load a vector's data: loading a `*.csv` or `*.parquet` file or loading data from an open layer. Click on the *Load* button under "Data sets" (1). A new window opens. Loading the vector from a file is quite the same: select your preferred file format in the menu (2), and click *Load* (3), pointing to the location of the vector file in your machine.



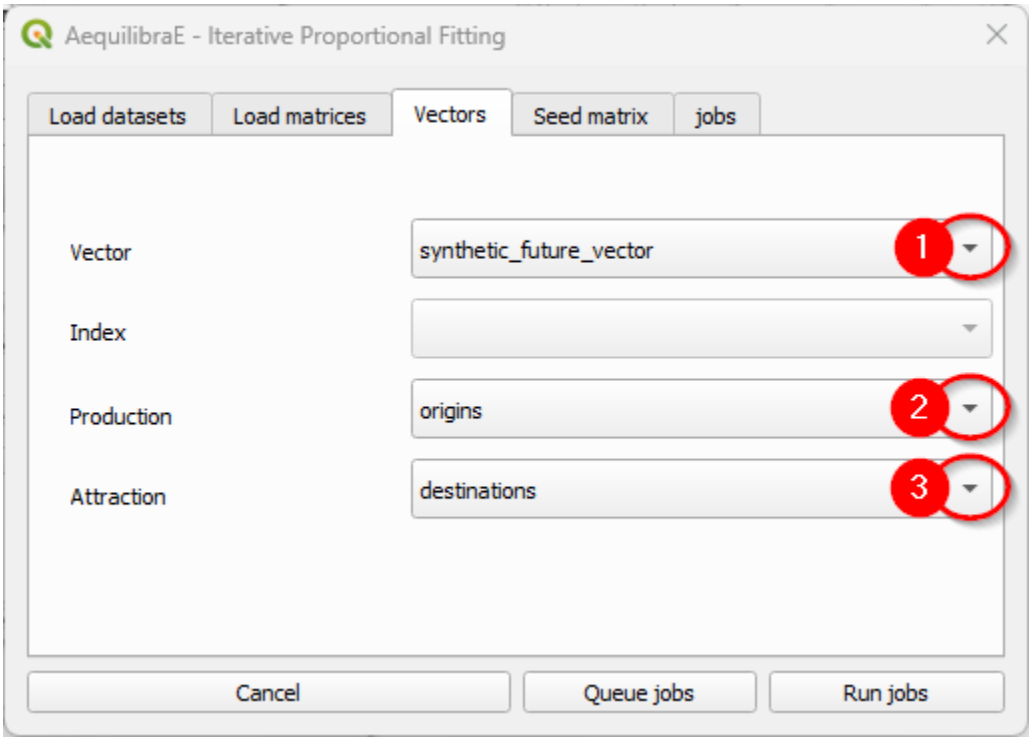
Case you are loading from an open layer, just click *Import from layer*, point the available data layer (1), and the name of its index column (2). Let's choose to only *Use data* (3).



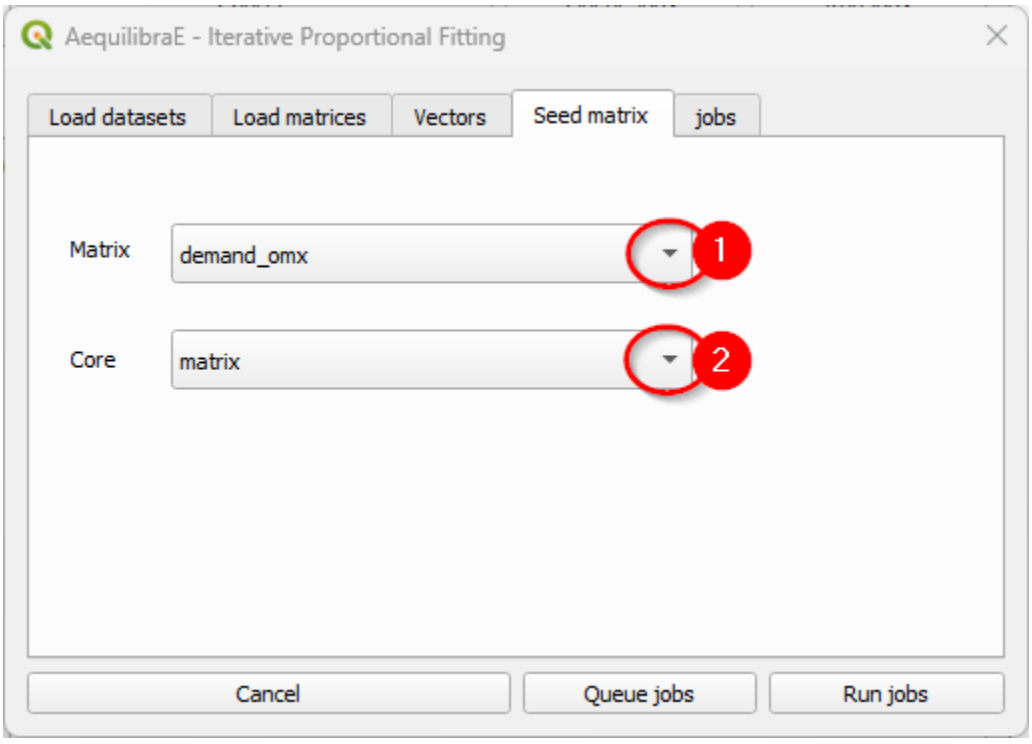
When the vector is loaded, it will appear in the *Load datasets* table.



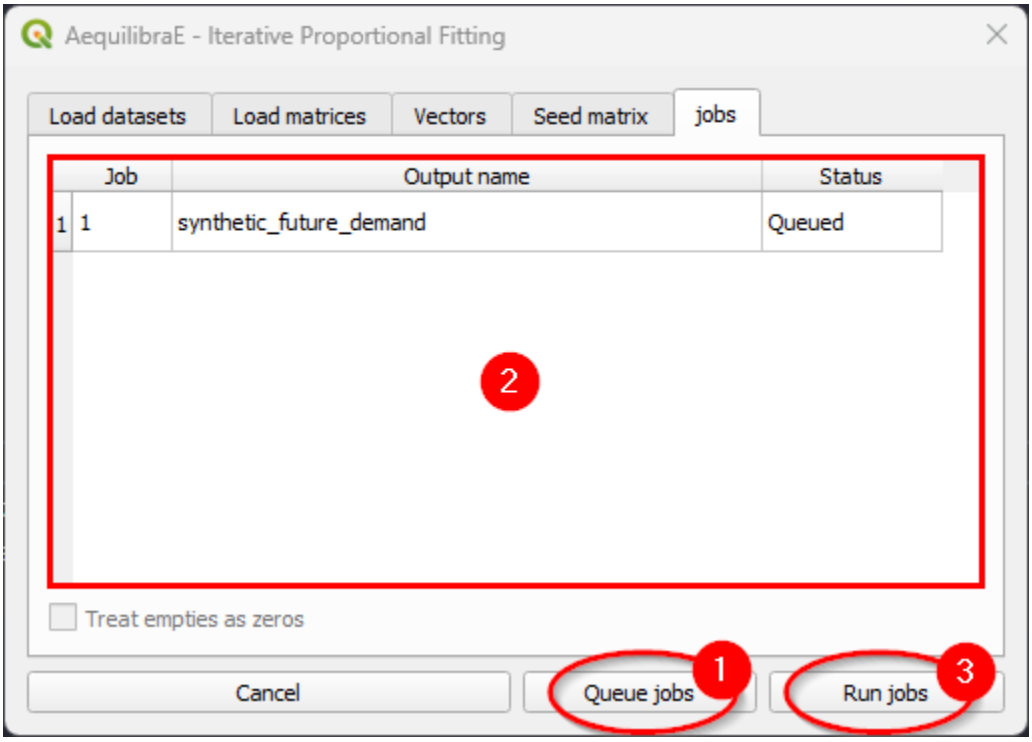
You can now select the production/attraction (origin/destination) vectors. If your data comes from a QGIS layer, you'll notice that the *Index* list is deactivated because the data index was configured when loading the data.



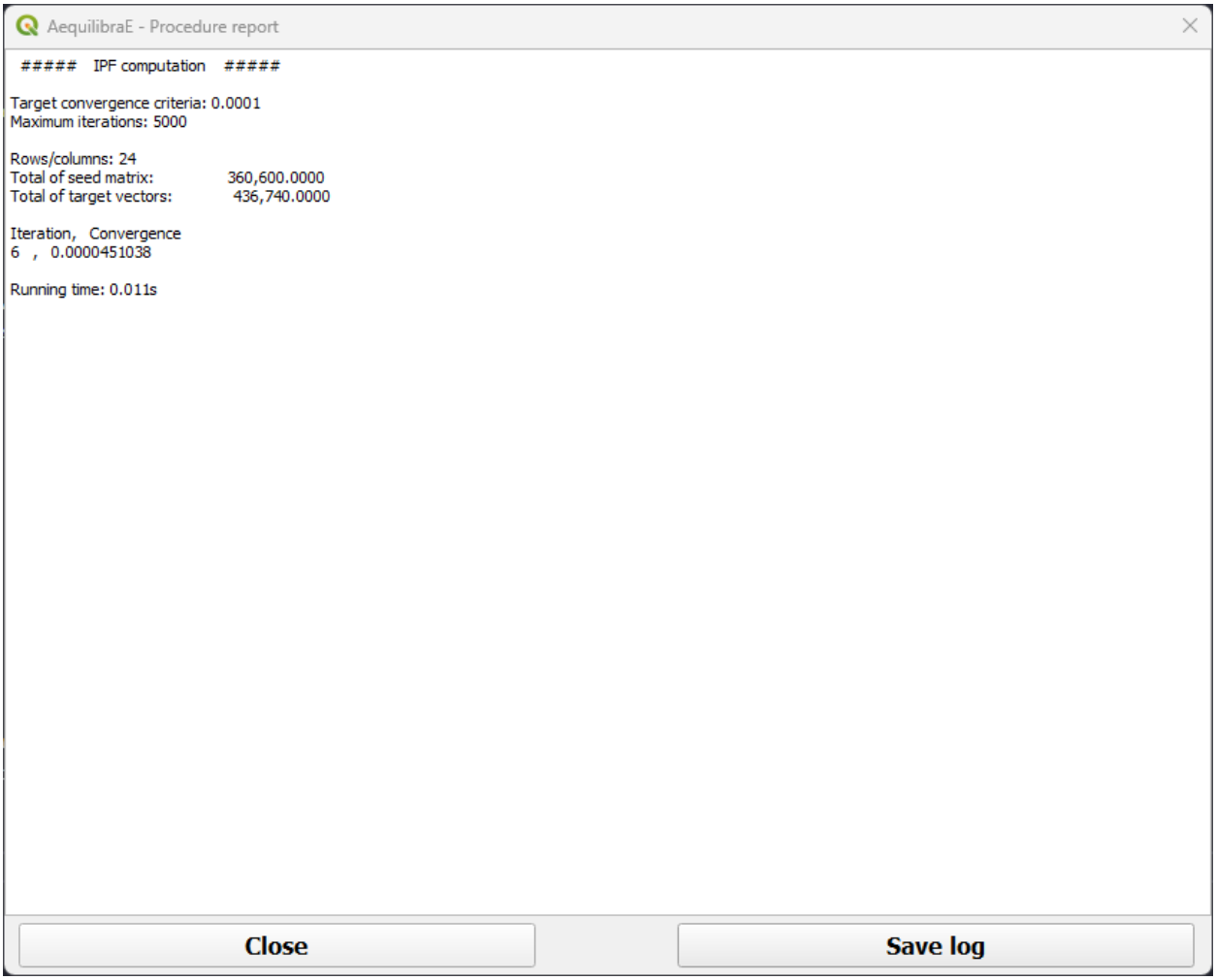
And select the seed (demand) matrix to be used.



To run the procedure, queue the job (and select the where the output file will be saved). You'll notice that a job with the output file name will appear in the jobs table with a status *queued* (2). Finally, press *Run jobs* (3).



After the job is completed, a new window showing its procedure report will open. We can close it after checking the procedure report.



Important

Production and Attraction vectors **must be** balanced before running IPF.

If you want to use the same data as we did, you can save the following code block as a CSV file in your machine!

Listing 0: Synthetic future vector

```
index,origins,destinations
1,5220.000000,29197.959184
2,20648.000000,41952.857143
3,2204.000000,23665.714286
4,7656.000000,12293.877551
5,10208.000000,2766.122449
6,57420.000000,31349.387755
7,25636.000000,6146.938776
8,2784.000000,5071.224490
9,10440.000000,4302.857143
10,2668.000000,8298.367347
11,18908.000000,18748.163265
```

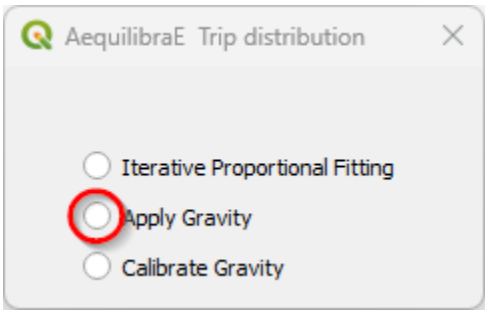
(continues on next page)

(continued from previous page)

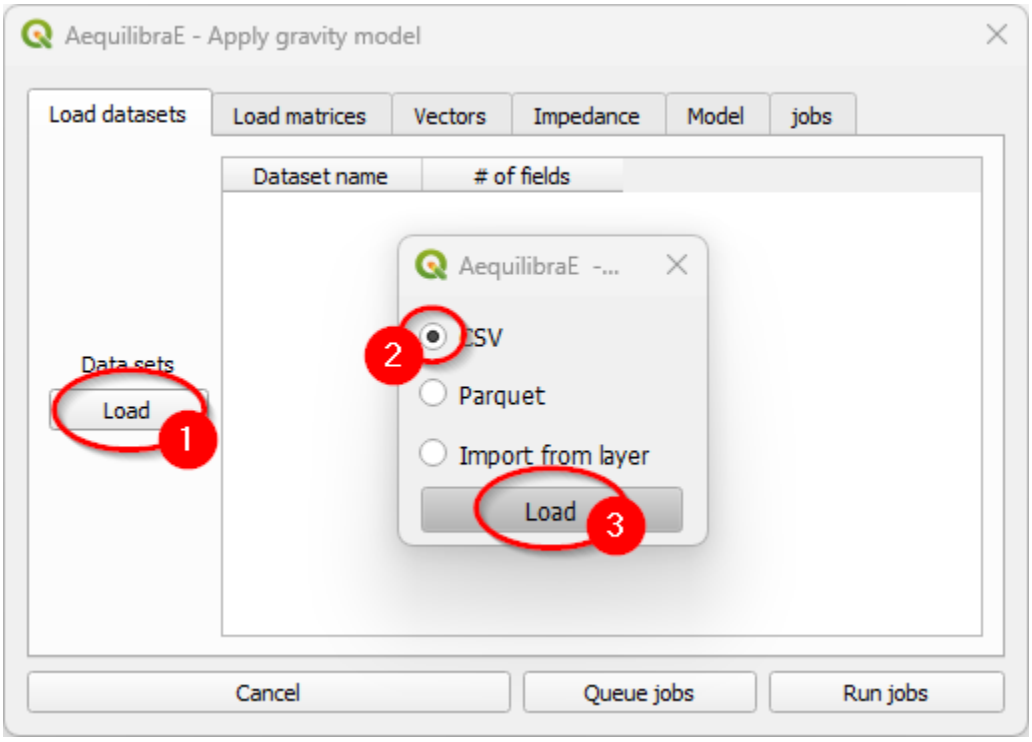
```
12,67164.000000,58242.244898
13,10440.000000,59471.632653
14,29348.000000,12447.551020
15,26912.000000,15060.000000
16,14848.000000,4302.857143
17,14848.000000,10449.795918
18,21344.000000,15213.673469
19,8236.000000,13523.265306
20,13456.000000,31195.714286
21,19140.000000,13369.591837
22,464.000000,307.346939
23,11716.000000,9681.428571
24,35032.000000,9681.428571
```

Apply Gravity Model

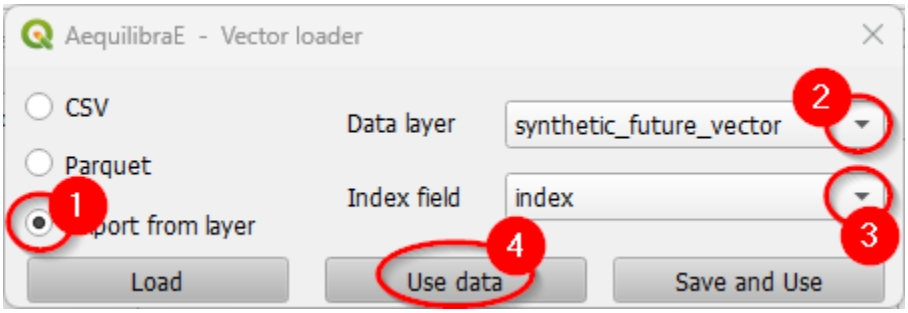
If one has future matrix vectors (there are some provided with the example dataset), they can either apply the Iterative Proportional Fitting (IPF) procedure available, or apply a gravity model just calibrated. Here we present the latter.



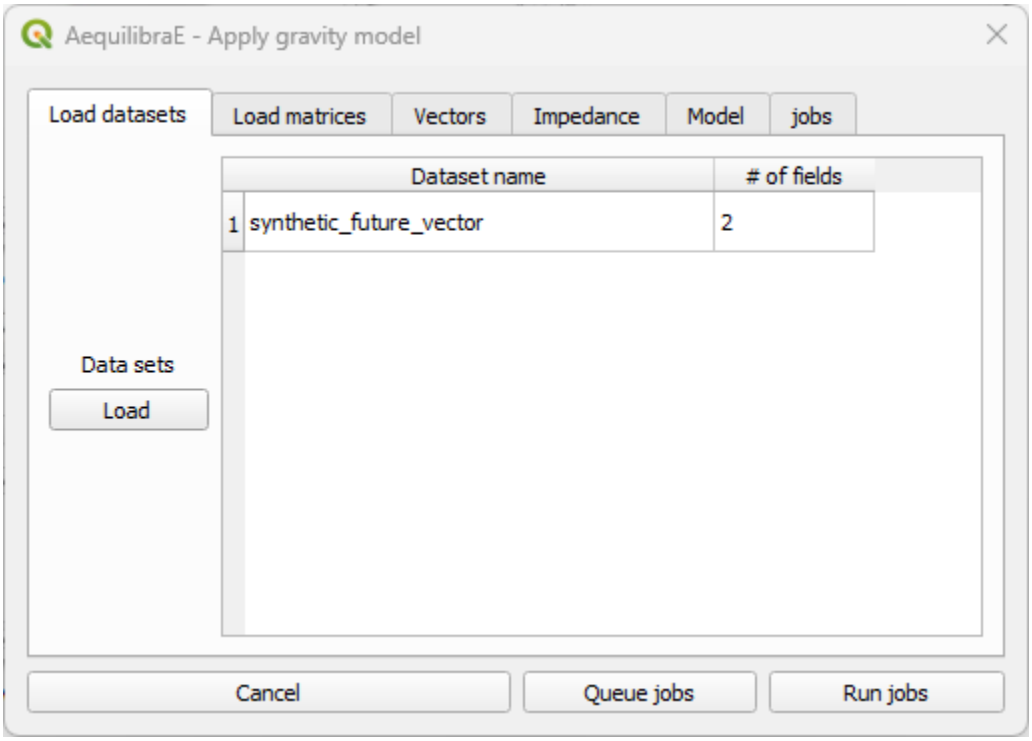
With the menu open, let's load the dataset(s) with the production/origin and attraction/destination vectors. We can add data into the model by loading a file or using an open layer, just like the IPF procedure. Let's click to load the dataset (1). A new window opens. Loading the vector from a file is quite the same: select your preferred file format in the menu (2), and click *Load* (3), pointing to the location of the vector file in your machine.



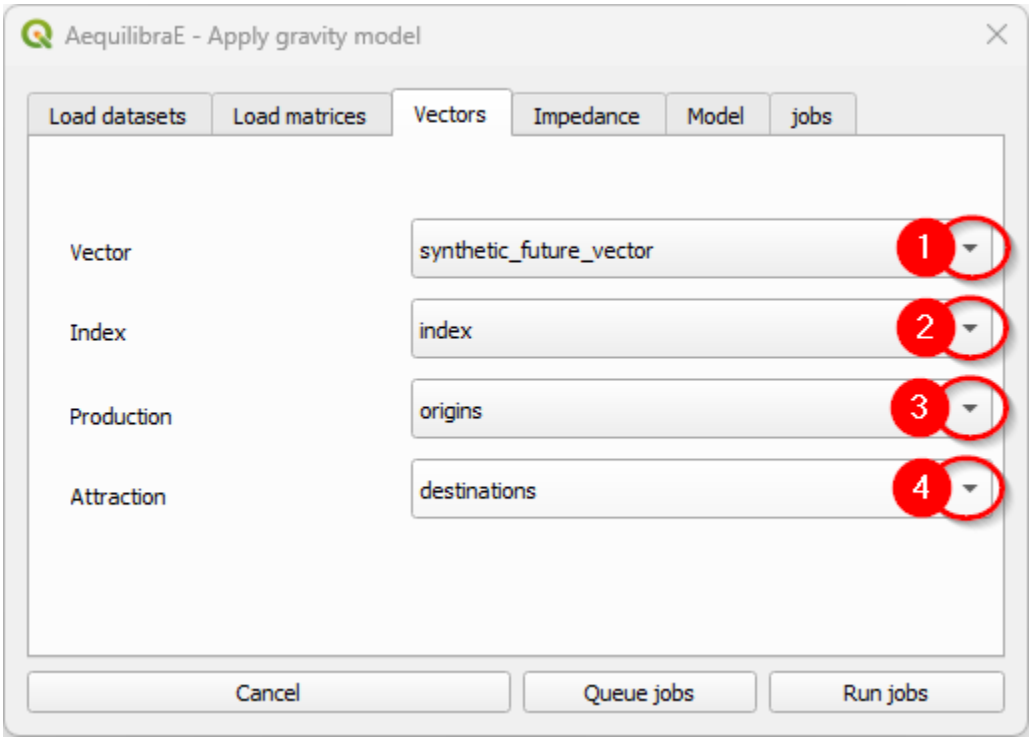
Case you are loading from an open layer, just click *Import from layer*, point the available data layer (1), and the name of its index column (2). Let's choose to only *Use data* (3).



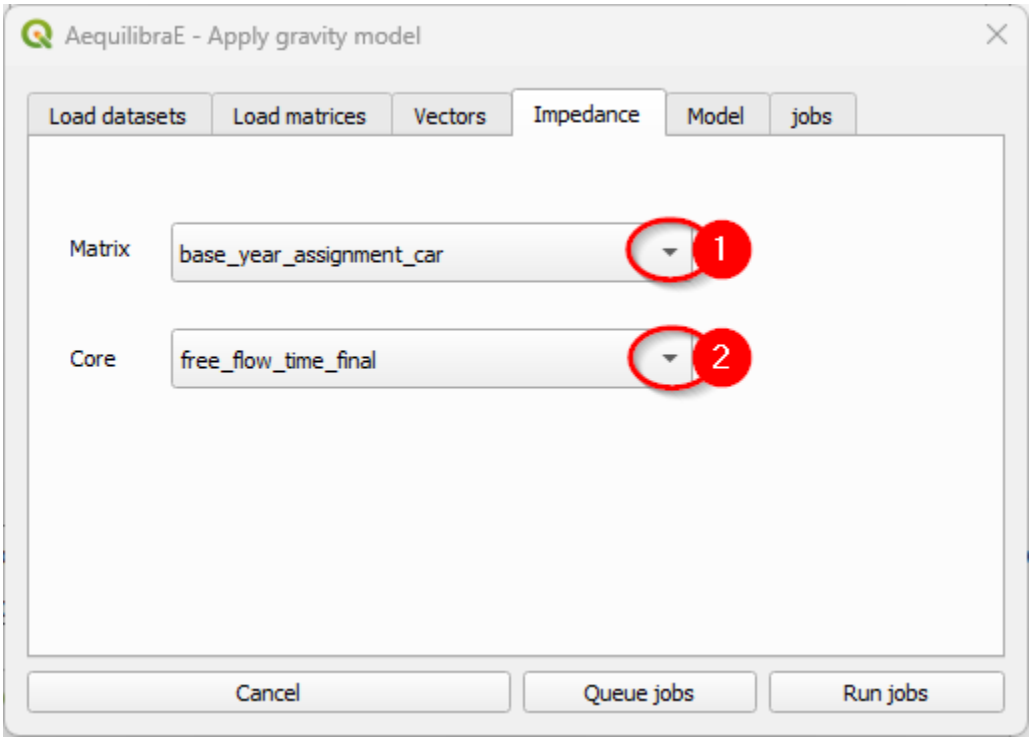
When the vector is loaded, it will appear in the *Load datasets* table.



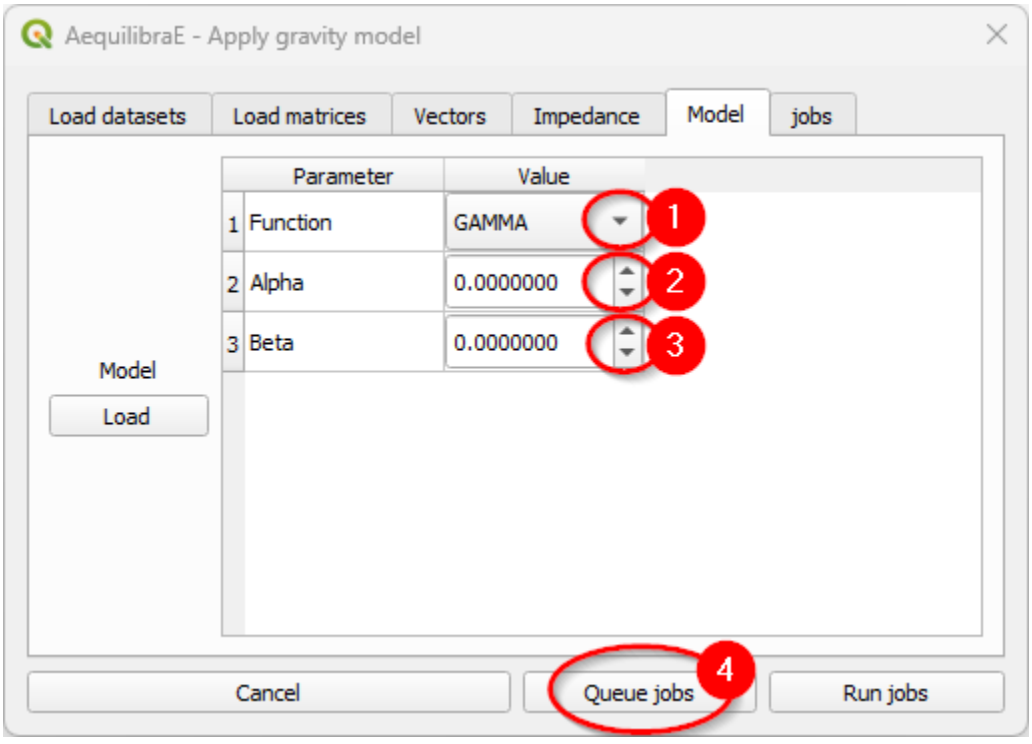
Select the production/attraction (origin/destination) vectors.



And the impedance matrix to be used. We can select one matrix core to use in computation.

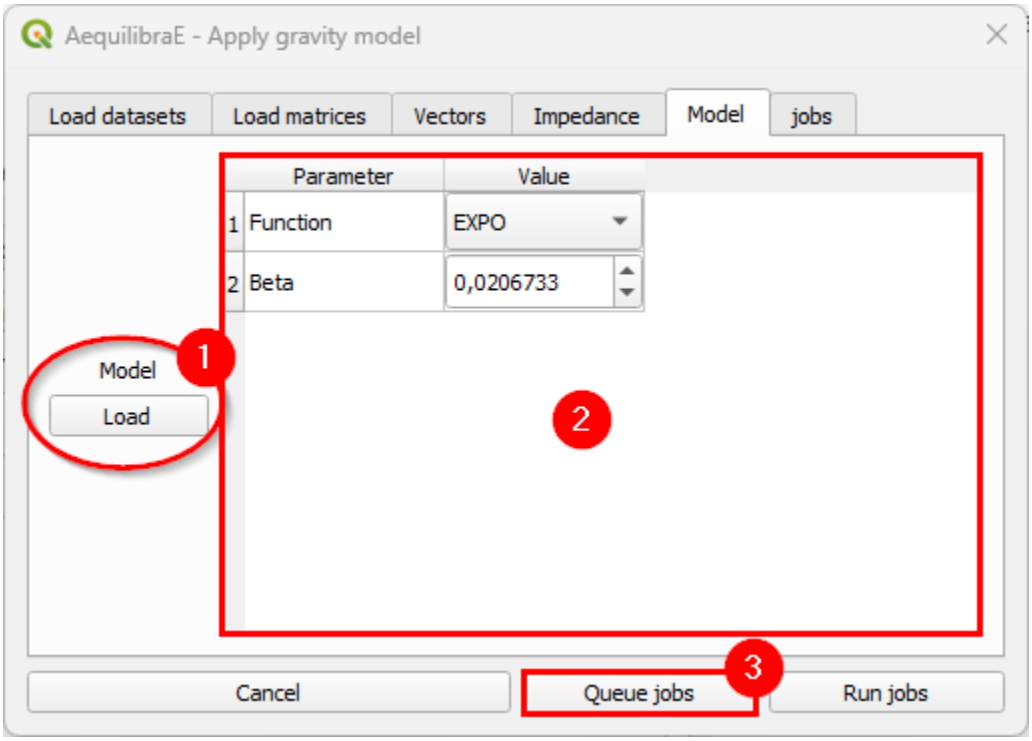


The last input is the gravity model itself, which can be done by loading a model that has been previously calibrated, or by selecting the deterrence function from the drop-down menu and typing the corresponding parameter values. To select a deterrence function, select one function among the available ones (1) and configure the values for the fields *alpha* and *beta* (steps 2 ane 3). For each function you select, queue it into the jobs (4) table.

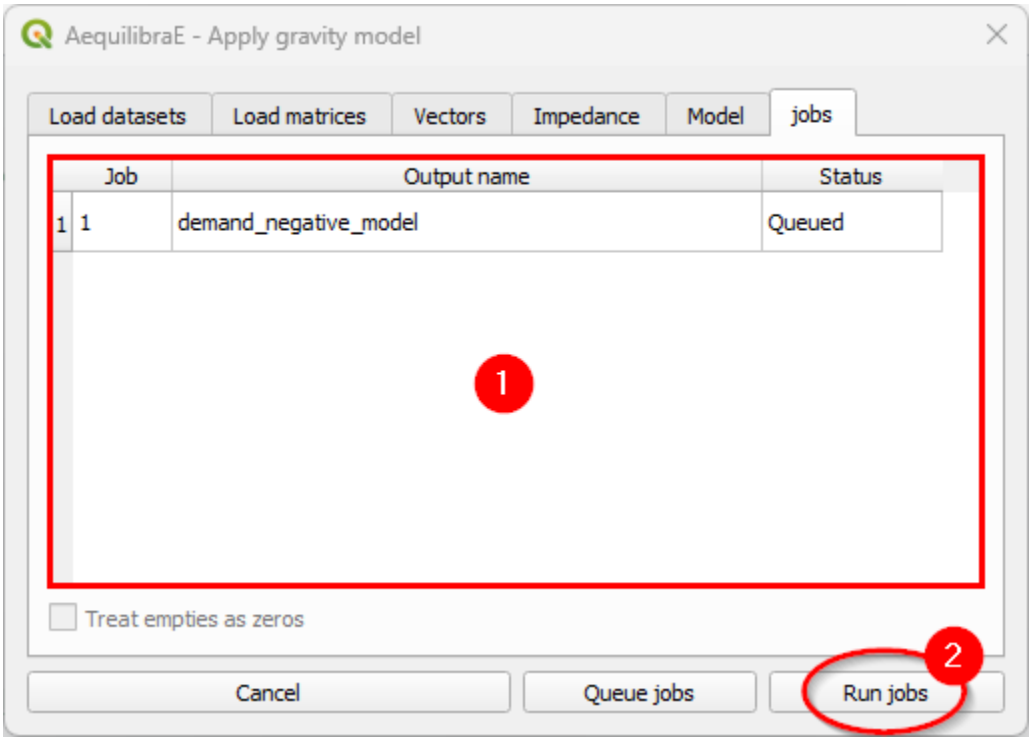


As we already have calibrated models, we'll load its configurations. When clicking *Load* (1) a new window opens. Point to the path where your `*.mod` file is stored, and once its loaded, you'll notice that the parameters in the table view

now correspond to the model data (2). Queue the jobs by hitting the *Queue jobs* button (3).

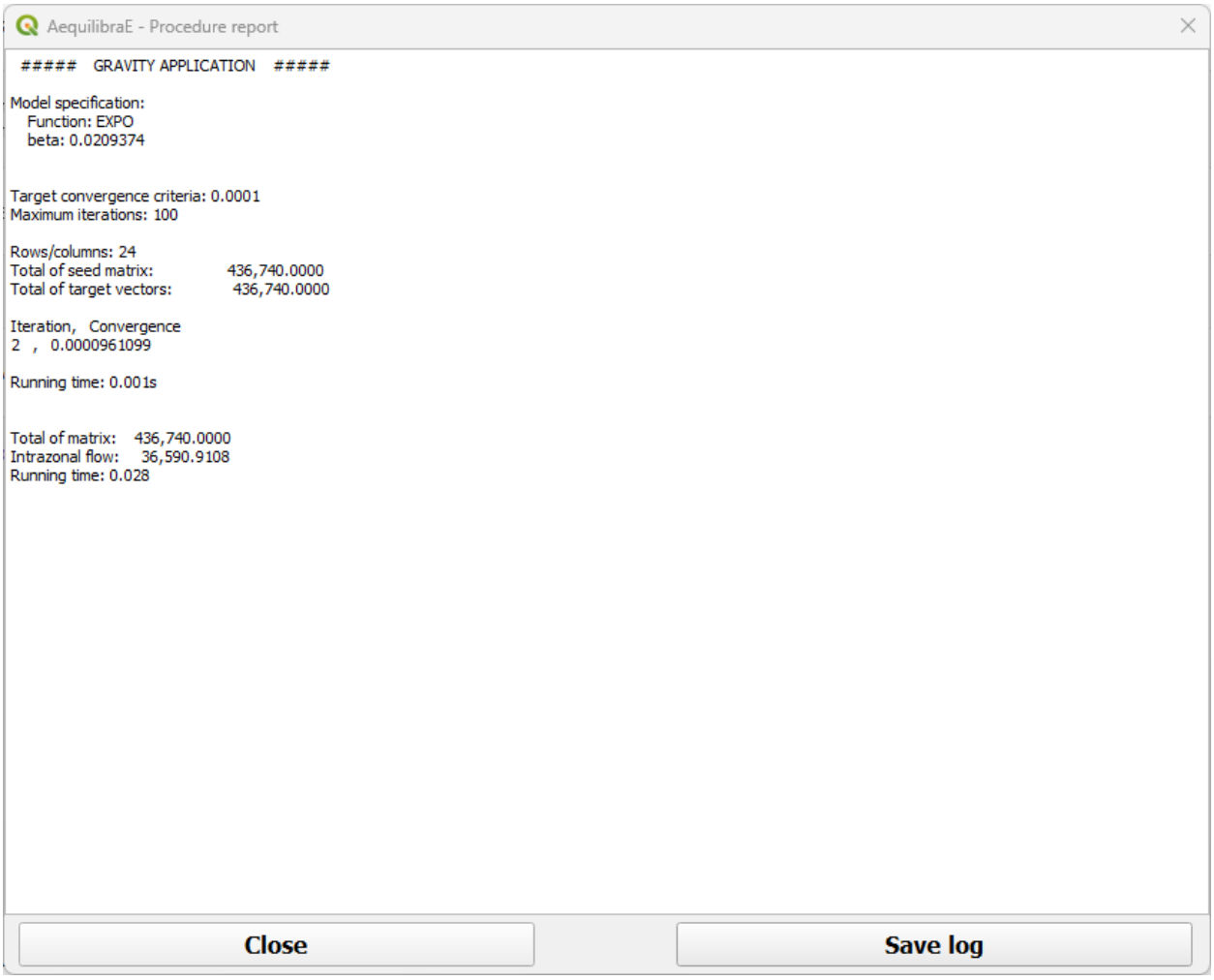


It is possible to check the jobs qeued before running the model in the tab *Jobs* (1). If all jobs look ok, just click on the *Run jobs* button (2).



We have to repeat this process of configuring, loading the calibrated models and running the jobs twice: one for each model we have!

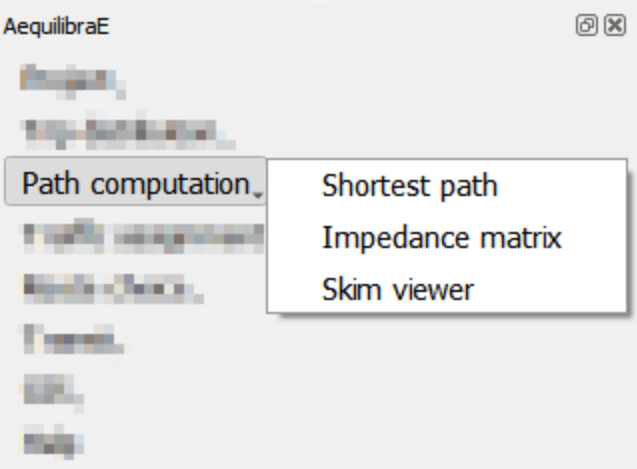
Once the process is finished, a new window with the procedure report output opens. You can check its output and close it.



The result of this future demand matrix can also be assigned, which is what we will generate the outputs being used in the *scenario comparison*. To do so, run a *traffic assignnment workflow* using the 'demand_negative_model' as input! Try it on!

2.3 Path Computation

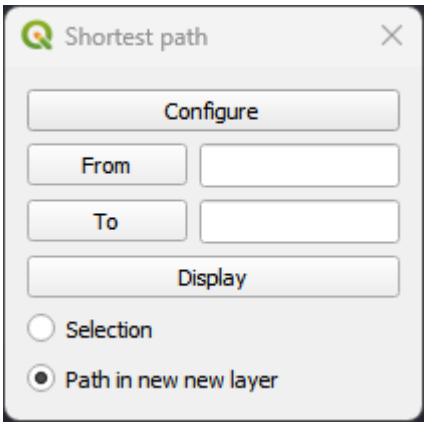
From version 0.6, AequilibraE plugin does not require the user to create the graph to perform path computation as in previous versions. In this version, as you set up your own configurations, the software already computes the graph for you.



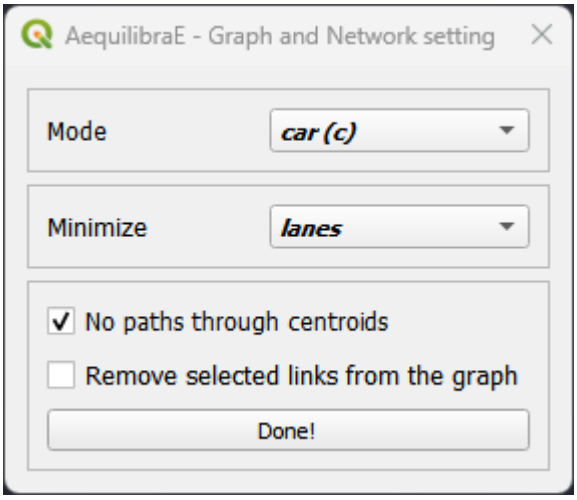
2.3.1 Shortest Path

The first thing we can do with a project is to compute a few arbitrary paths to see if the network is connected and if paths make sense.

The shortest path menu consists in two complimentary windows: one for selecting the to/from nodes and how we want to visualize the output and the other for configuring the network graph.



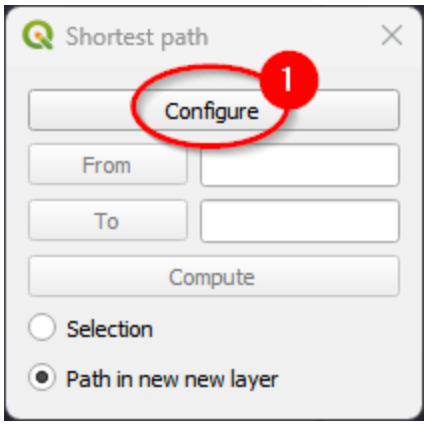
(a) Shortest path settings



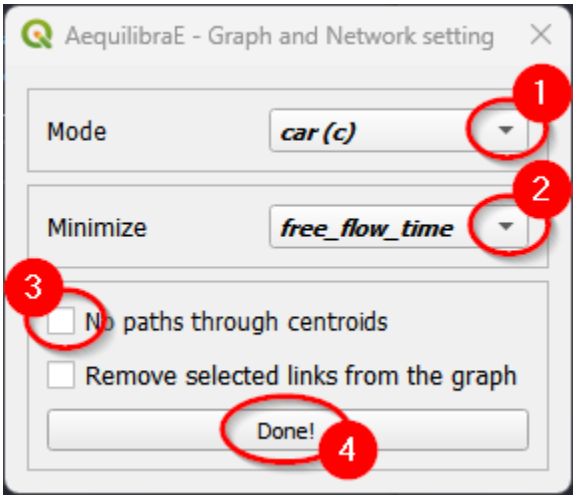
(b) Network graph configuration

Basic workflow

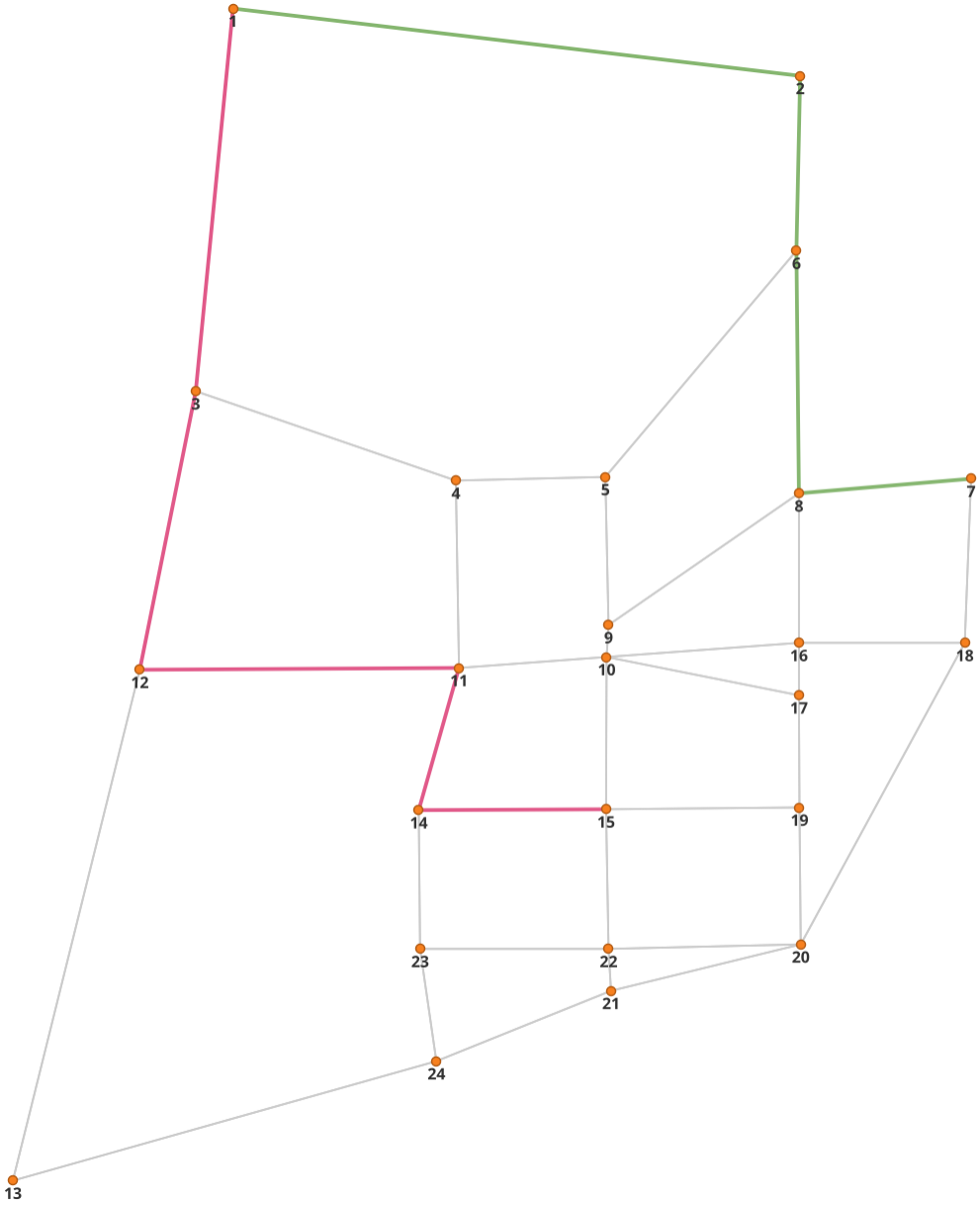
We'll use Sioux Falls example in this workflow. Open the shortest path window and notice that the only action available is to configure the network. We'll click on the "*Configure*" button (1).



For the case of Sioux Falls, we need to configure the graph to accept paths going through centroids (all nodes are centroids), but that is generally not the case. For zones with a single connector per zone it is slightly faster to also deselect this option, but use this carefully.

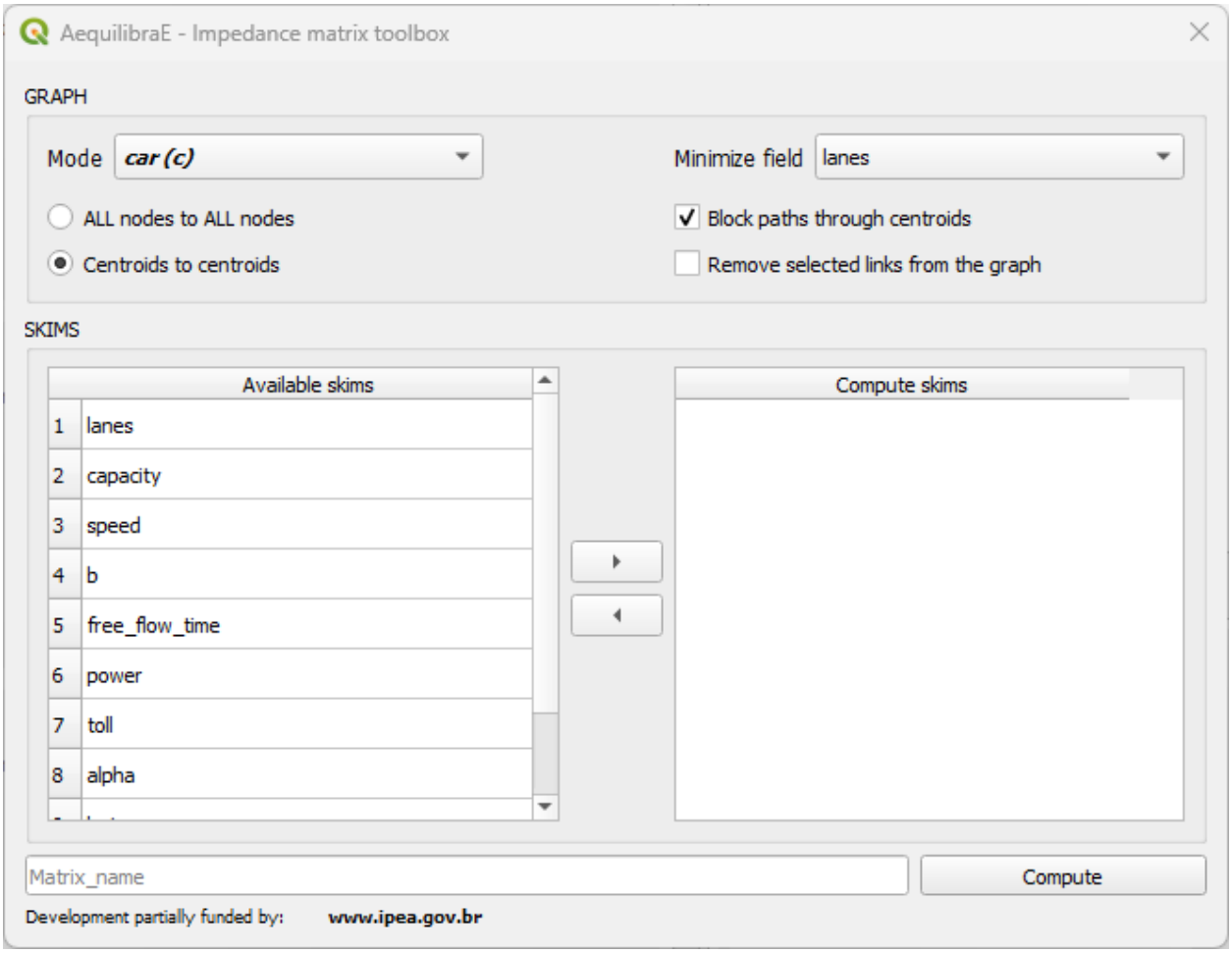


If we select that paths need to be in a separate layer, then every time you compute a path, a new layer with a copy of the links in that path will be created and formatted in a noticeable way. You can also select to have links selected in the layer, but only one path can be shown at time if you do so.



2.3.2 Impedance Matrix (aka Skimming Matrix)

We can also skim the network to look into general connectivity of the network.



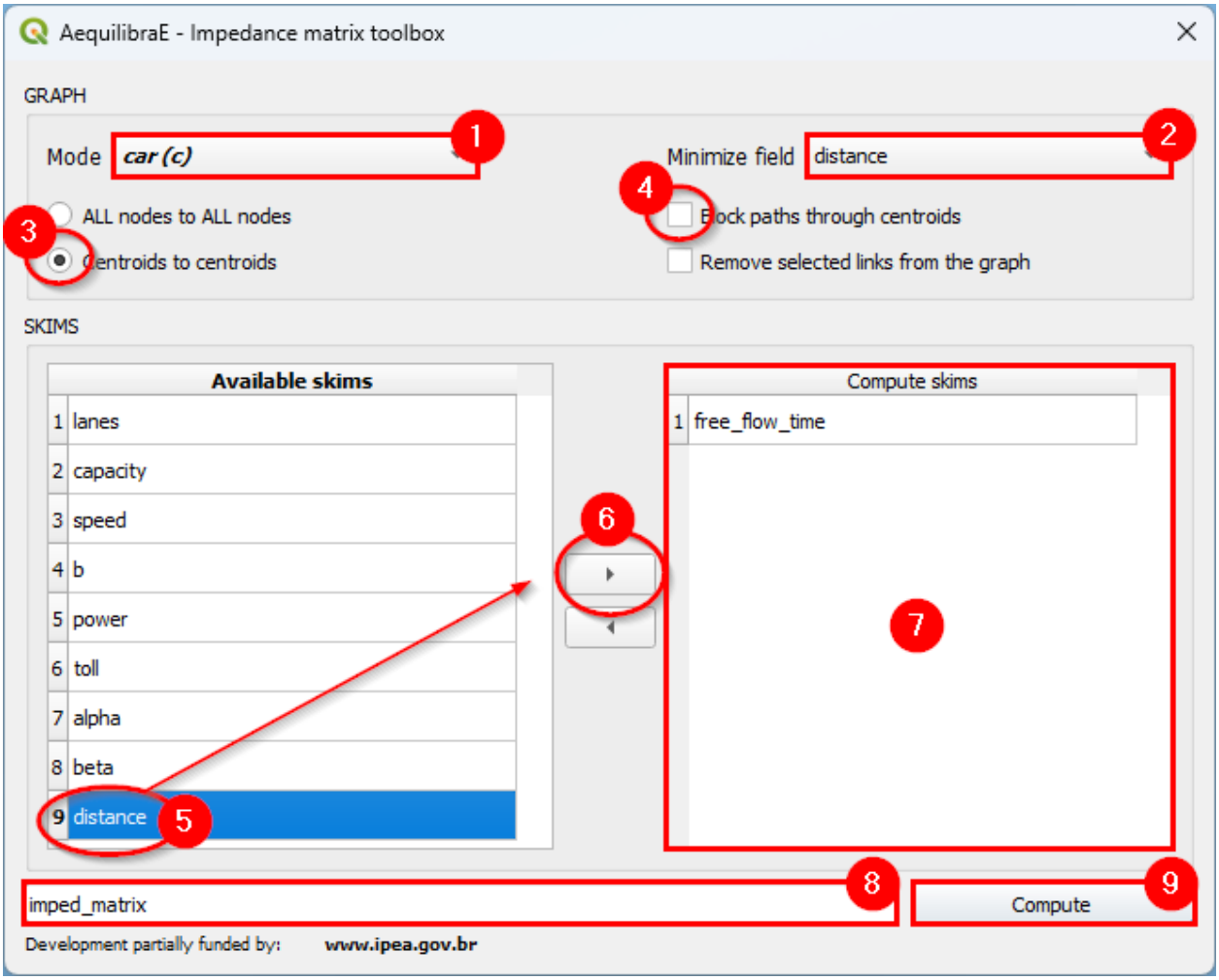
The "*Graph*" configurations at the top of the screen allow us to configure the graph: selecting the computation of a matrix from all nodes to all nodes, or from centroids to centroids, as well as to not allow flows through centroids. Besides chosing the mode to skimand the minimize field for computation.

The remaining controls stand at the "*Skims*" configuration, where we select the fields we should skim when computing the paths.

Basic workflow

We'll use Sioux Falls example to compute the impedance matrix. We begin chosing the mode (1) and the field to minimize (2). Let's compute skims for centroids to centroids (3), without blocking flows through centroids (4) because all nodes in Sioux Falls are also centroids.

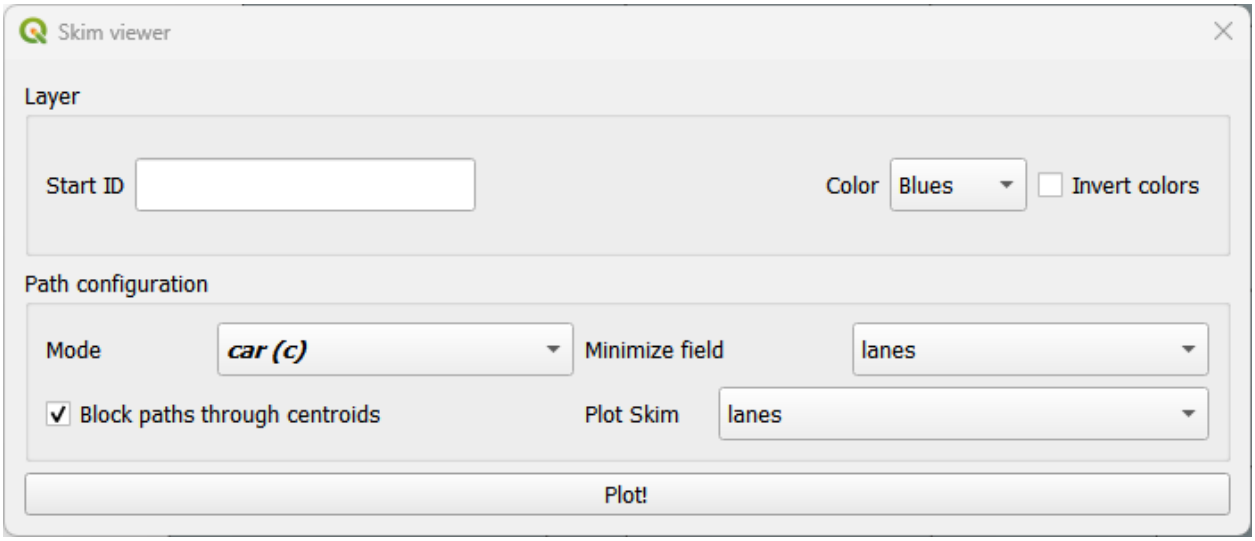
To select the skims for computation, select them at the available skims panel and click on the arrow to add it to computation panel (steps 5 to 7). Finally, create an output name (8) and compute the skims (9).



With the results computed, one can display them on the screen, loading the data from the *Matrix* tab, using the *Visualize data* tool in the Mapping menu.

2.3.3 Skim viewer

The skim viewer tool allows the user to easily visualize network costs, even for metrics that weren't skimmed yet. The skim viewer window looks like this:



In the *Layer* group, you can select the starting node/zone ID and configure the color map range for plotting. It is now possible for you to invert the colors in the color map, so feel free to customize your view as necessary. By default, every time Skim Viewer is initialized, a random value is assigned to the start ID. You can either use it or select a preferred start ID.

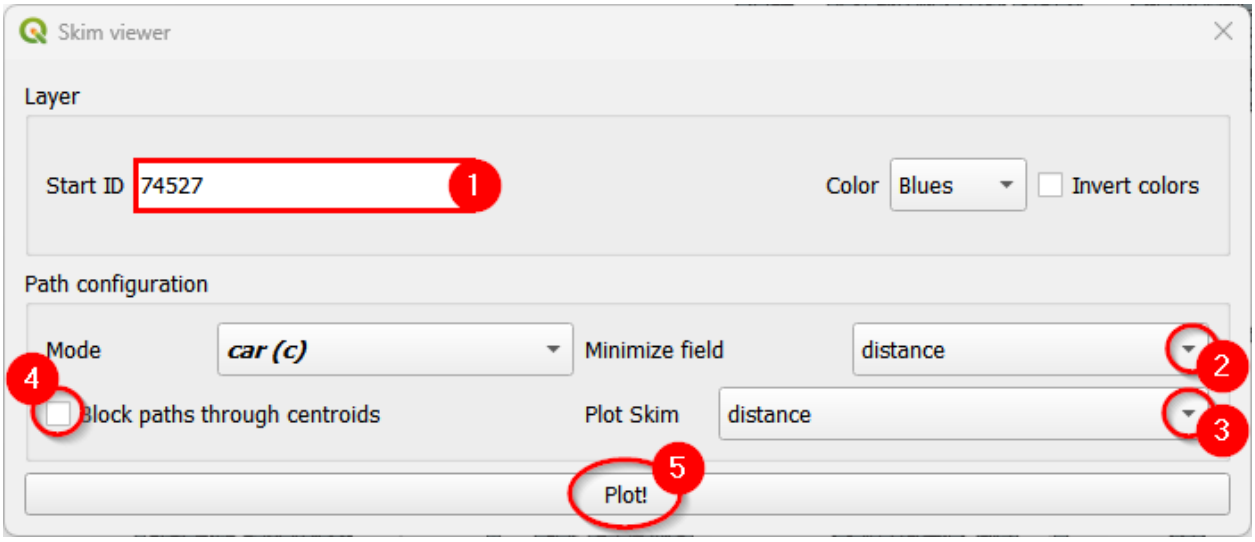
In the *Path configuration* group, you can set the graph configurations, such as the mode, the minimizing (cost) field, the choice to block or allow flows through the centroids. If the 'links' layer is at the layers' panel, Skim Viewer allows you to use its joined fields as minimizing or skimming fields.

To start Skim Viewer one of 'nodes' or 'zones' layer must be the active layer, otherwise an error is raised. To configure the start node/zone ID, it is possible to use the start ID box (pre-configured or customized) or select one feature in the active layer prior to the Skim Viewer initialization. If a feature is selected and a value for start ID is set, Skim Viewer is going to use the selected feature value for path calculation.

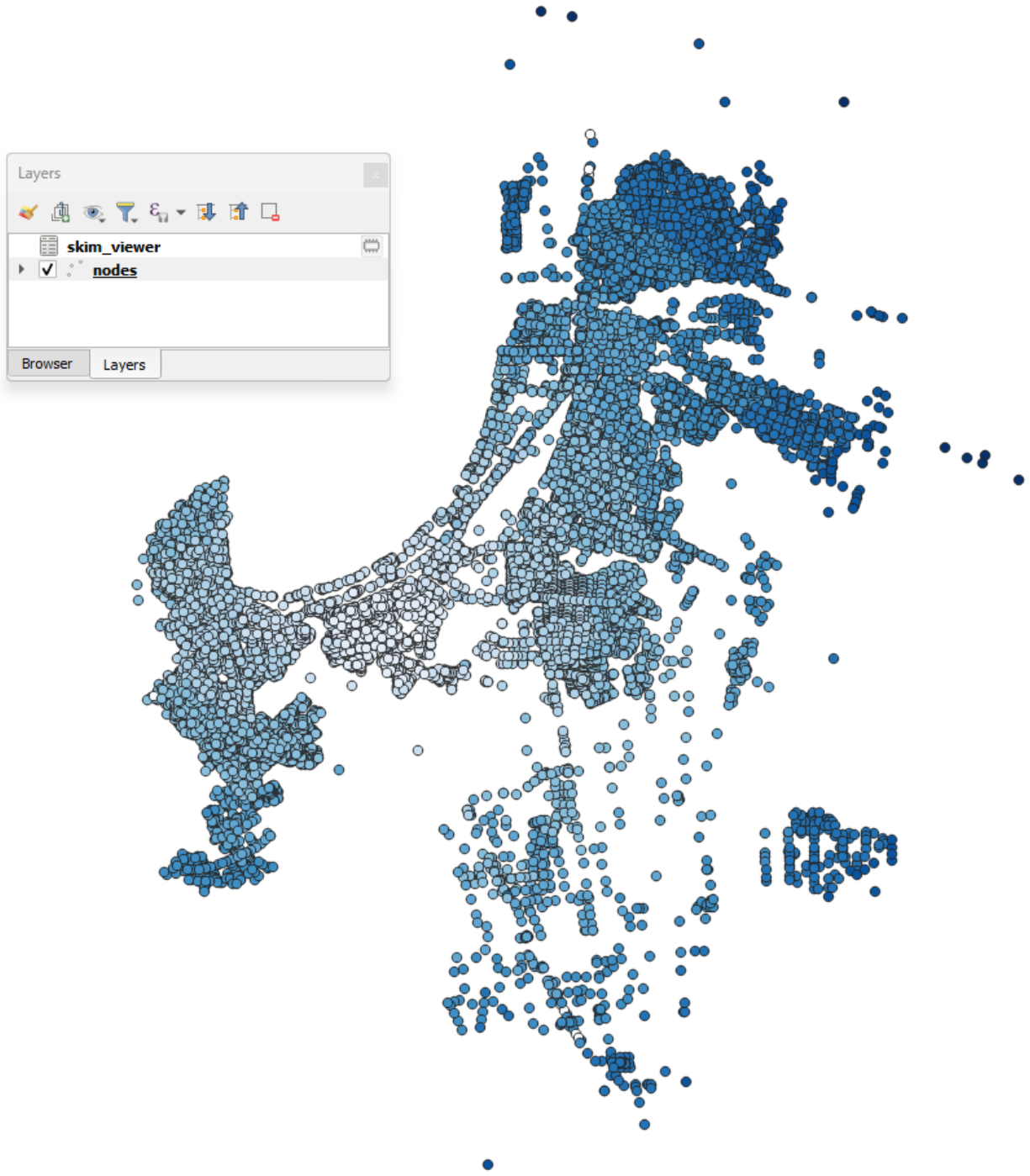
When visualizing the skims, you'll notice that a memory layer named 'skim_viewer' is created. It contains the node/zone ID for joining the nodes or zones layer and a data column that holds the data to be plotted. Whenever the selected node/zone changes, the values in the data column also change.

Skim view without joined layer

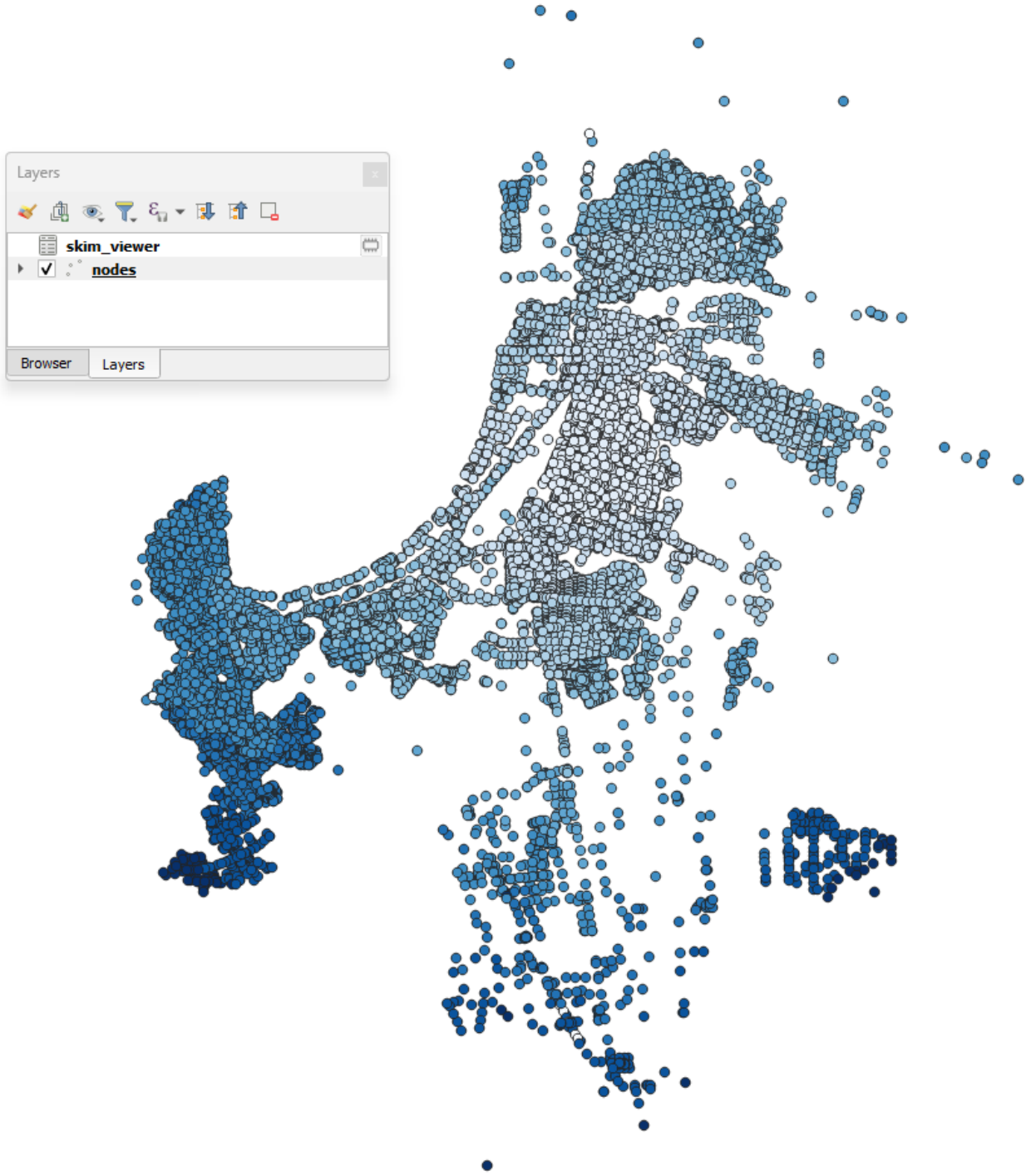
For demonstration purposes, we'll use the Coquimbo model for this example. You can go directly to the skim viewer and set the configuration, as presented below:



The output in the map canvas is:



If you select any other node with the skim viewer window open in the background, you will notice that the image displayed in the map canvas automatically changes.

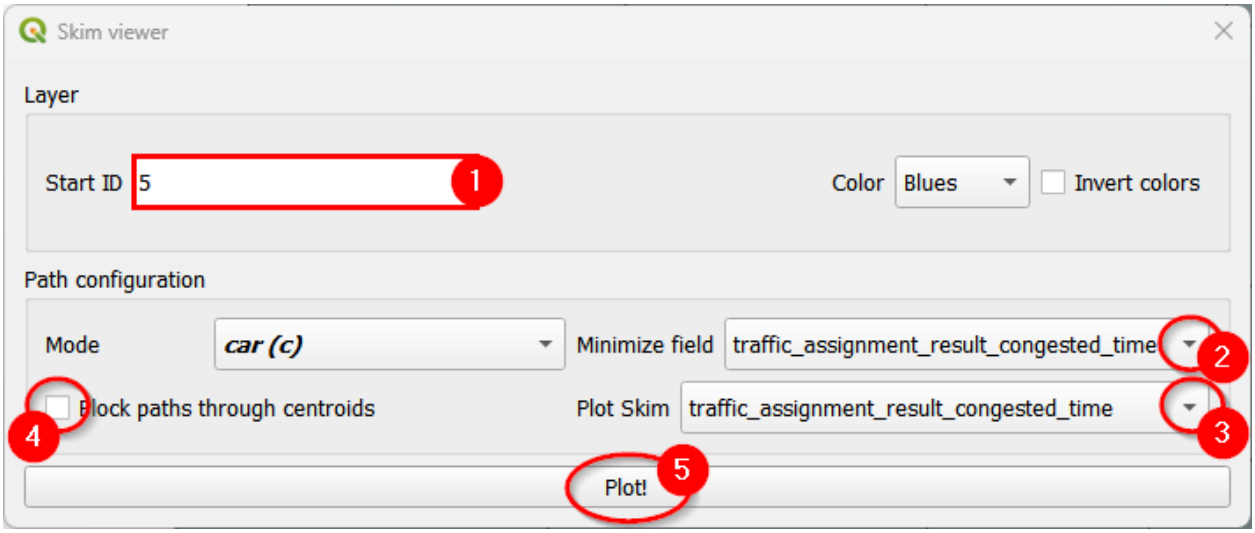


If you want to change either the minimizing field or skimming field (or both), you can modify your selection directly in the skim viewer window, and it will be automatically recomputed for display in the map canvas.

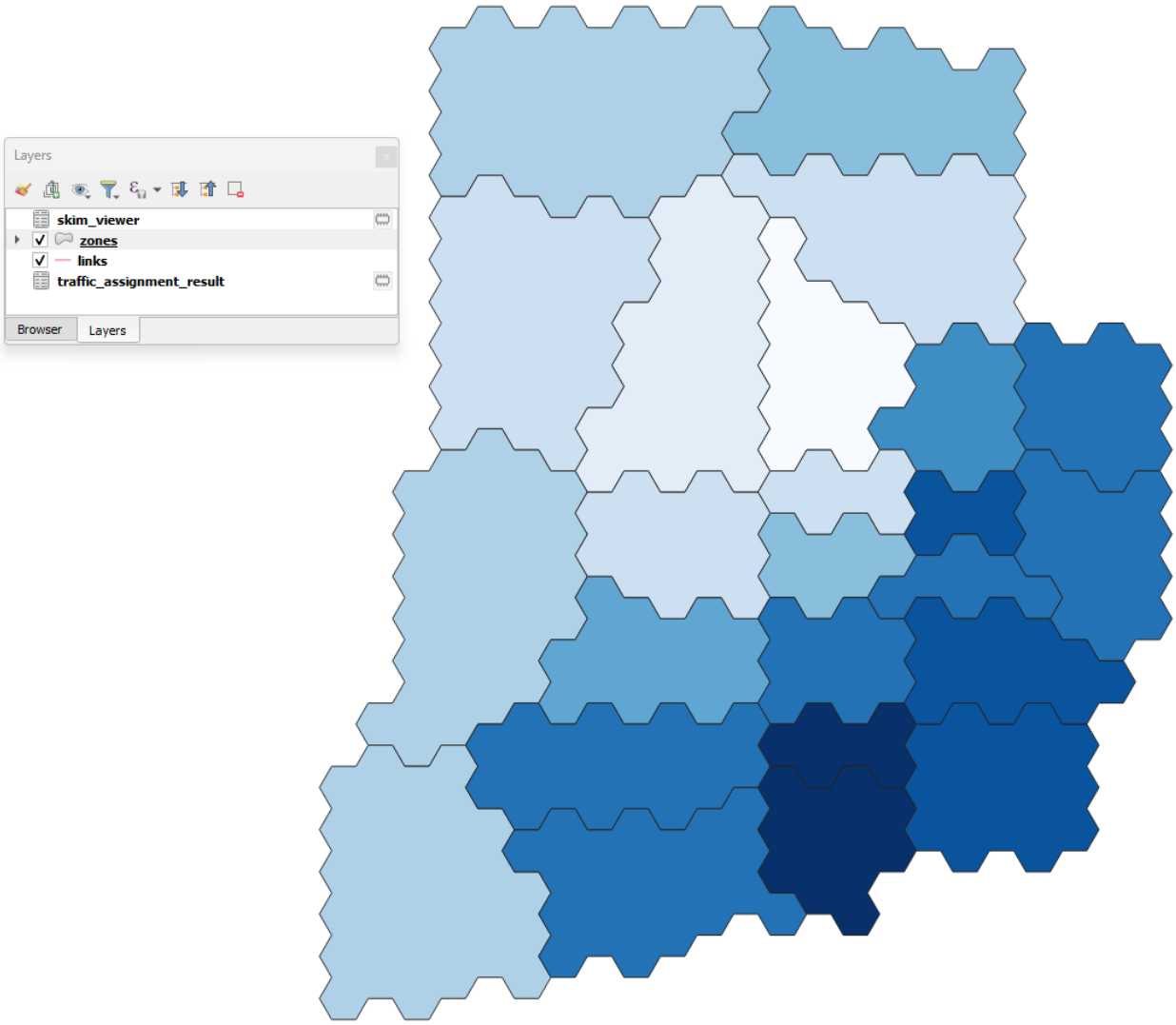
Skim viewer with joined layer

For this example, we'll use the Sioux Falls model. First, join the 'links' layer with the desired results table (see *Visualize data* for more information). Then, go to the skim viewer. When you see the window for the first time, you won't notice anything different, but when you click on the minimize field and available skims, you'll notice that the joined fields also appear here.

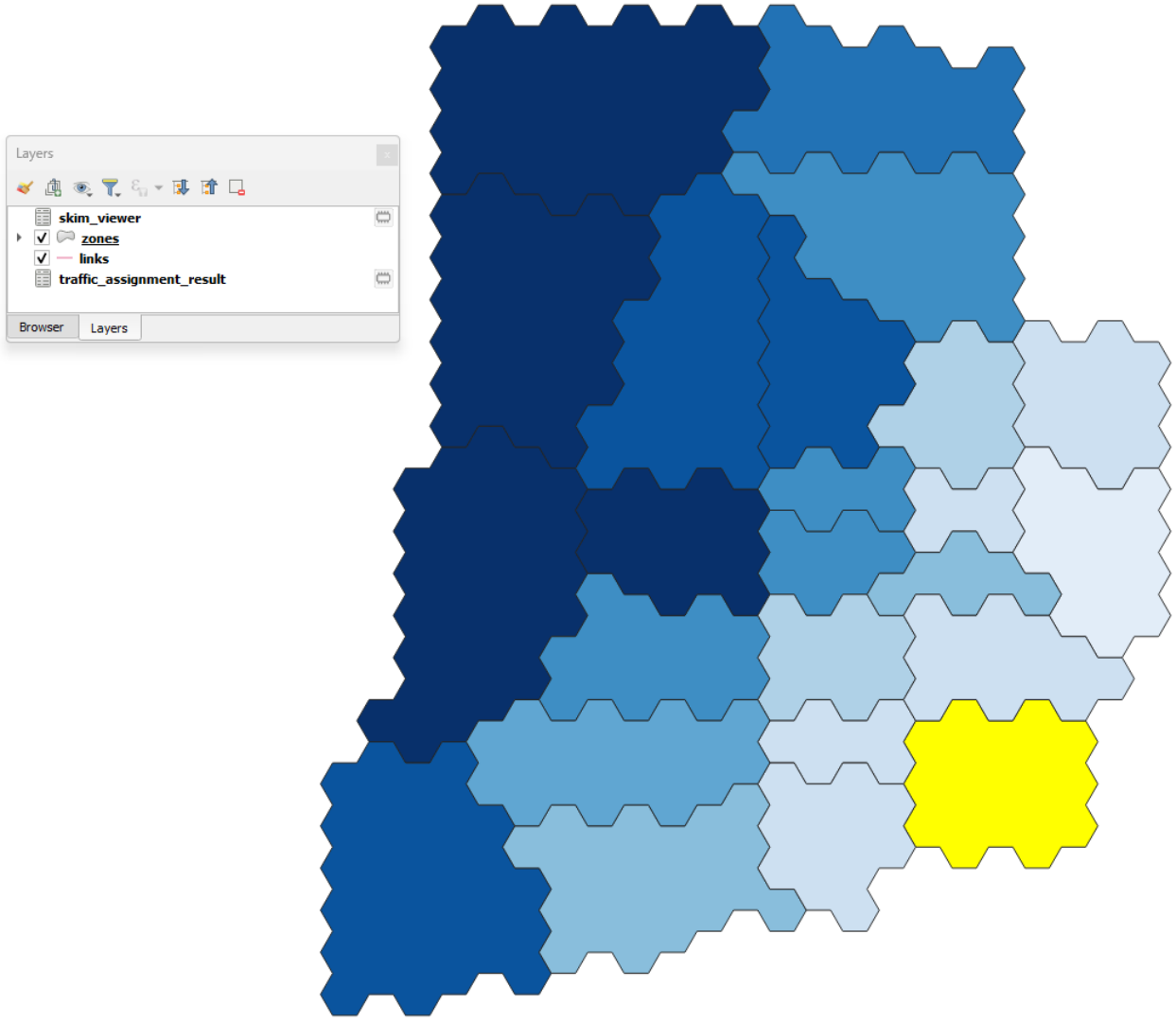
Let's plot the zones for Sioux Falls, starting at zone ID 5, and using *traffic_assignment_result_congested_time* for both the costs and skimming fields. The initial configuration looks like this:



The output in the map canvas will be:



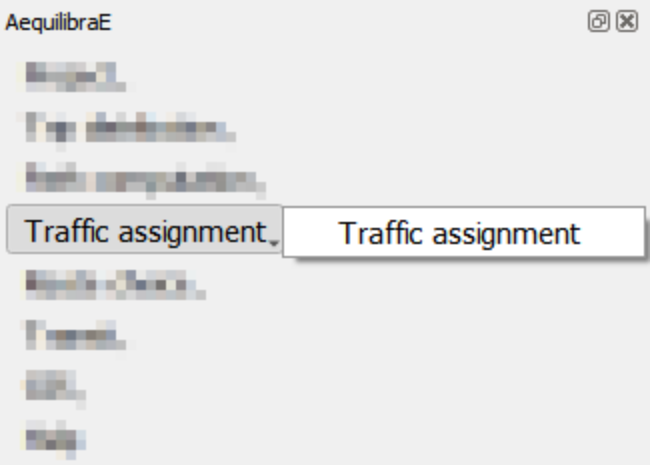
If your zone layer is active and you select another zone with the skim viewer window open in the background, you'll notice that the image in the map canvas automatically changes.



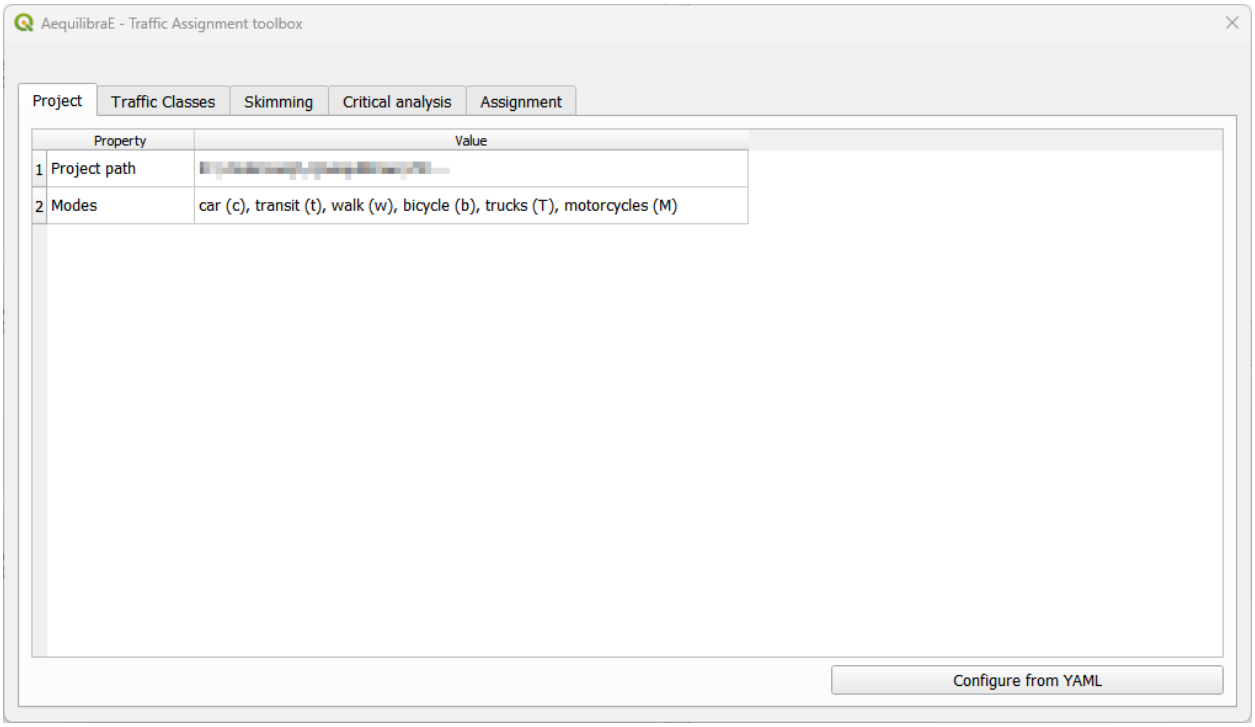
If you want to change either the minimizing field or skimming field (or both), you can modify your selection directly in the skim viewer window, and it will be automatically recomputed for display in the map canvas.

2.4 Traffic assignment

Having verified that the network seems to be in order, one can proceed to perform traffic assignment, since we have a demand matrix.



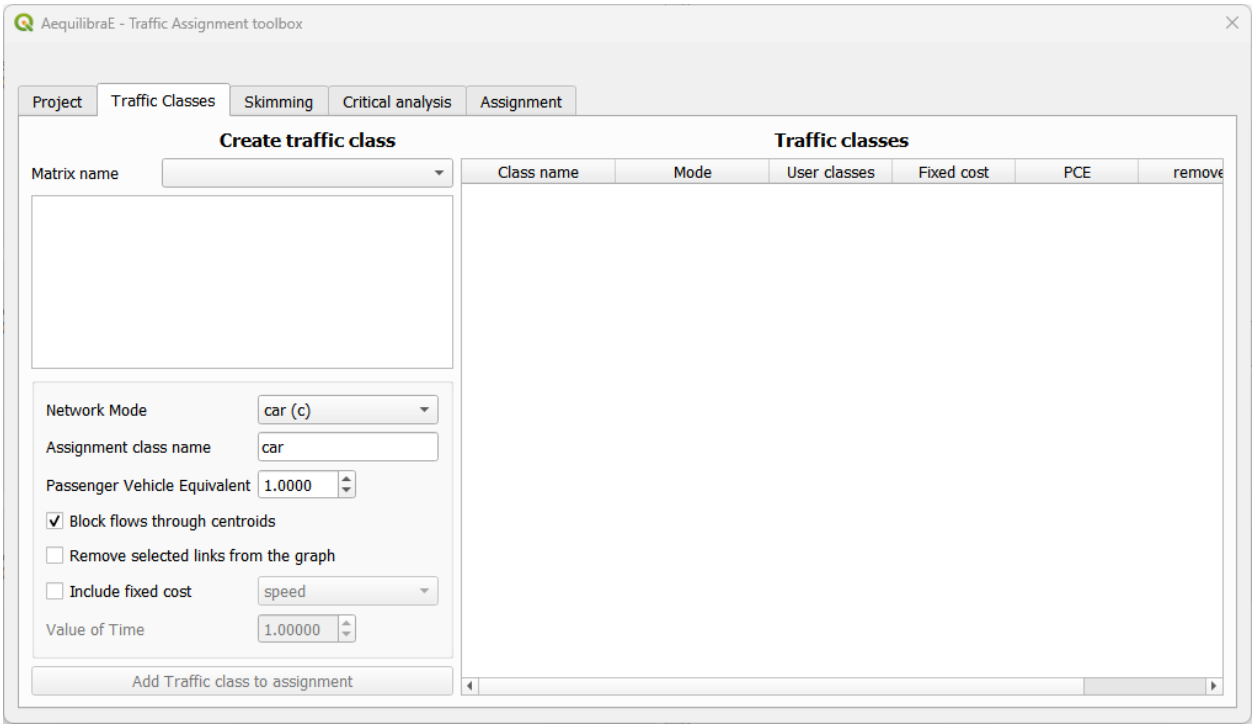
The Traffic Assignment window currently has five tabs. The first one is the *Project* tab where you can check the project path and the available modes. In the *Project* tab it is also possible to load configuration from a YAML file.



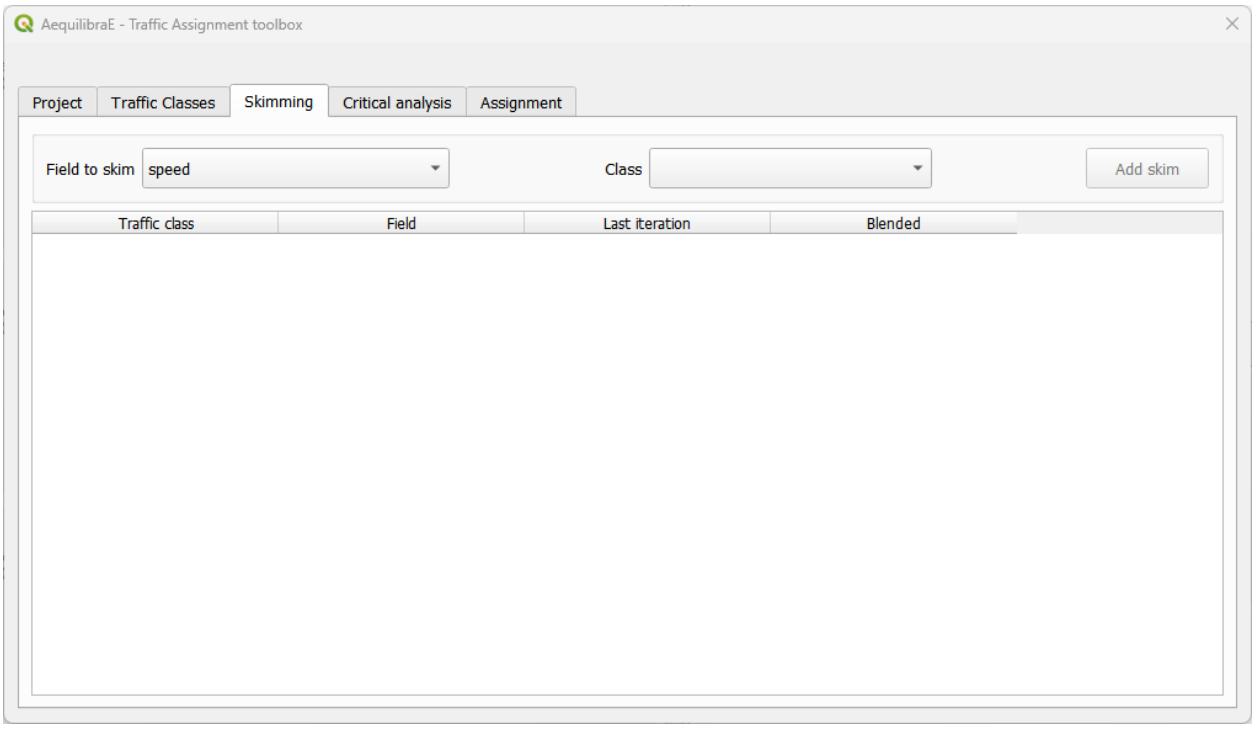
In the *Traffic Classes* tab you will create the traffic classes and graphs used in the project. This is where one of the available matrices and its core for computation are selected. We can modify the passenger car equivalent (PCE) for whatever value applies in your analysis, as well as include the value of time (VOT) as fixed cost. It is also possible to remove links from the analysis by selecting them before opening the Traffic Assignment window, and toggling the button *Remove selected links from the graph*. More about graph configuration can be found in the AequilibraE documentation.

Important

For setting a fixed cost, one **must** have a value fot the *vot* column in the project *modes* table.

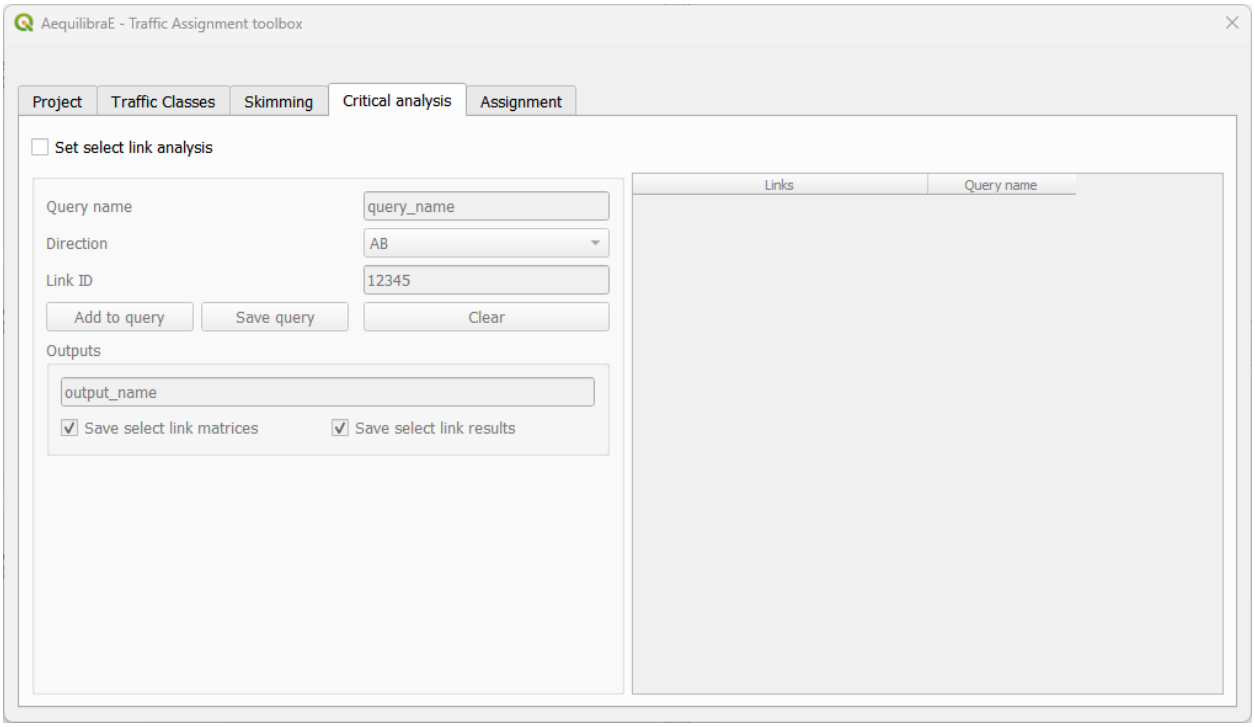


In the *Skimming* tab it is possible to select the skimming fields for each traffic class, and the results desired: final or blended. Final stands for the result of the last iteration, while blended represents the averaged results for all iterations. Just check the desired boxes. Notice that the Class field only is populated if the traffic classes were properly set in the previous step.

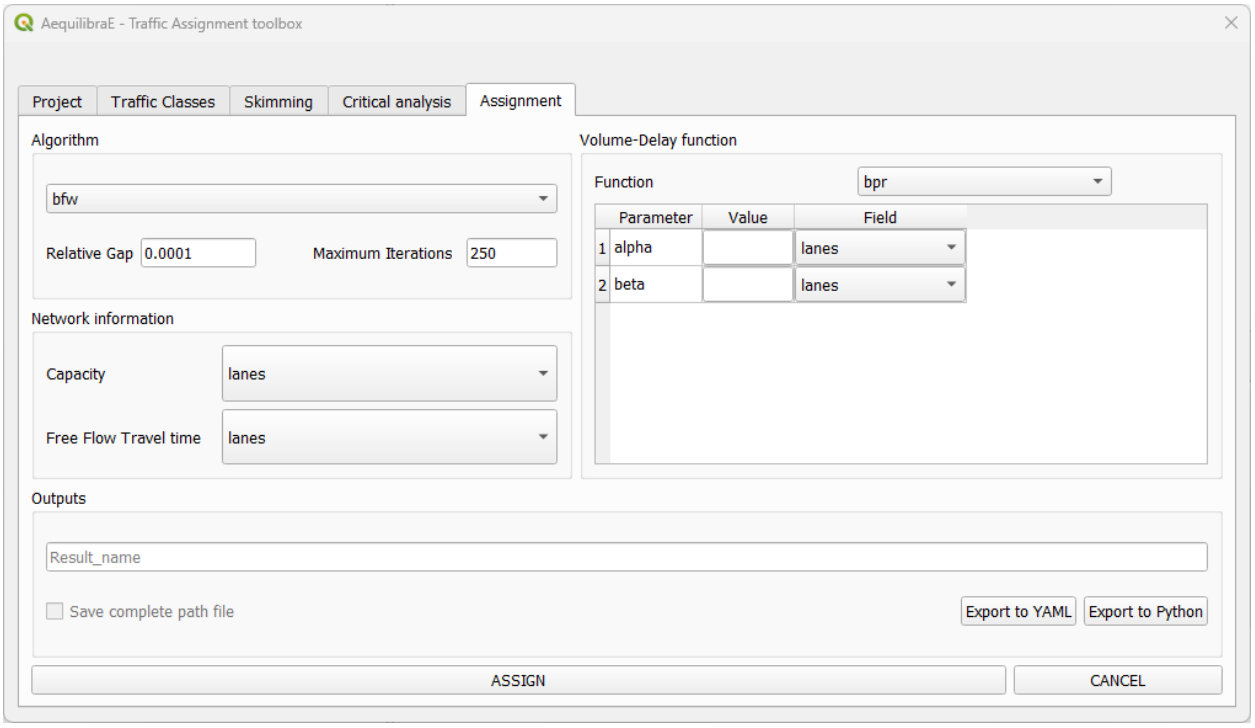


The *Critical analysis* tab allows running a select link analysis with the traffic assignment. QAequilibraE default configuration is NOT running this analysis, so you have to toggle its button to 'activate' the configuration interface. There, it is possible to set up a query name, a travel direction, and the link ID. Adding and removing links from queries, and

modifying queries should be straightforward. Saving the output matrices and results for select link analysis are also configured here.



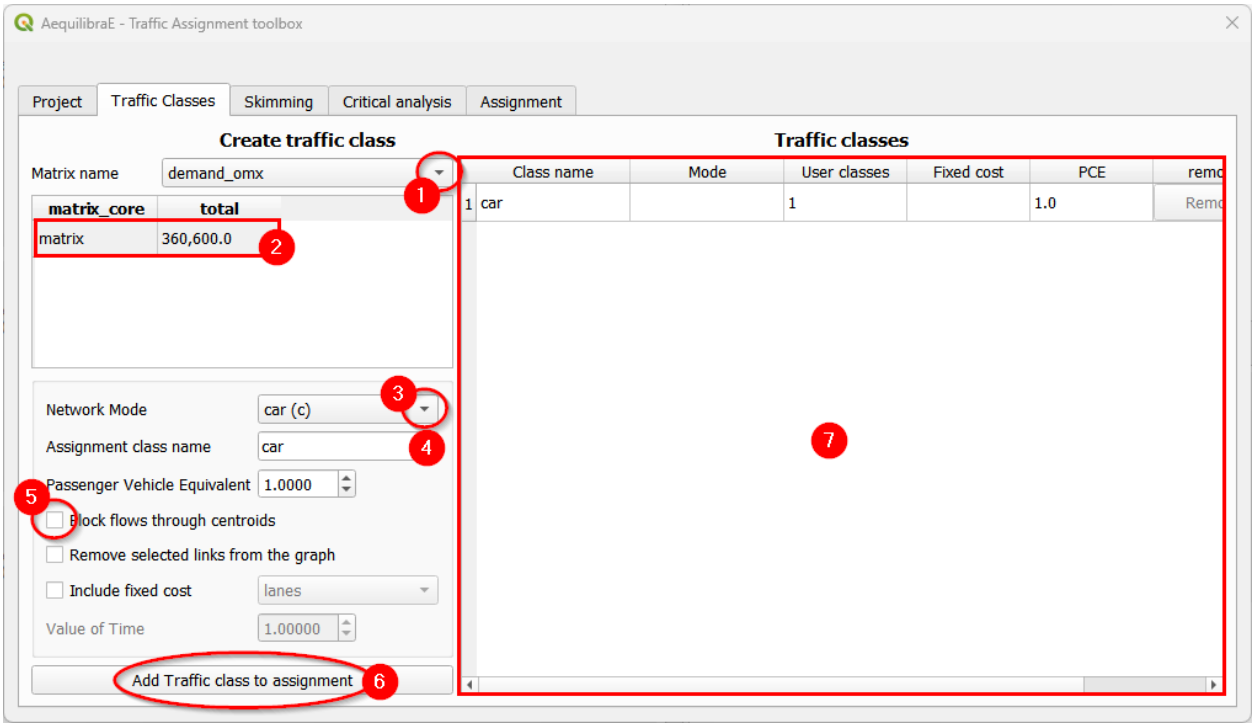
In the *Assignment* tab we select the assignment algorithm and configure the volume-delay function and its *alpha* and *beta* parameters. The fields for link capacity and free flow travel time are selected. We also confirm the relative gap and maximum number of iterations we want, and the output folder. Now it is possible to save your project configurations in either one Python or YAML file. While the Python file is stored inside the project run folder to be executed as part of 'Run procedures', the YAML file can be used as input for other model runs, as input in the *Project* tab.



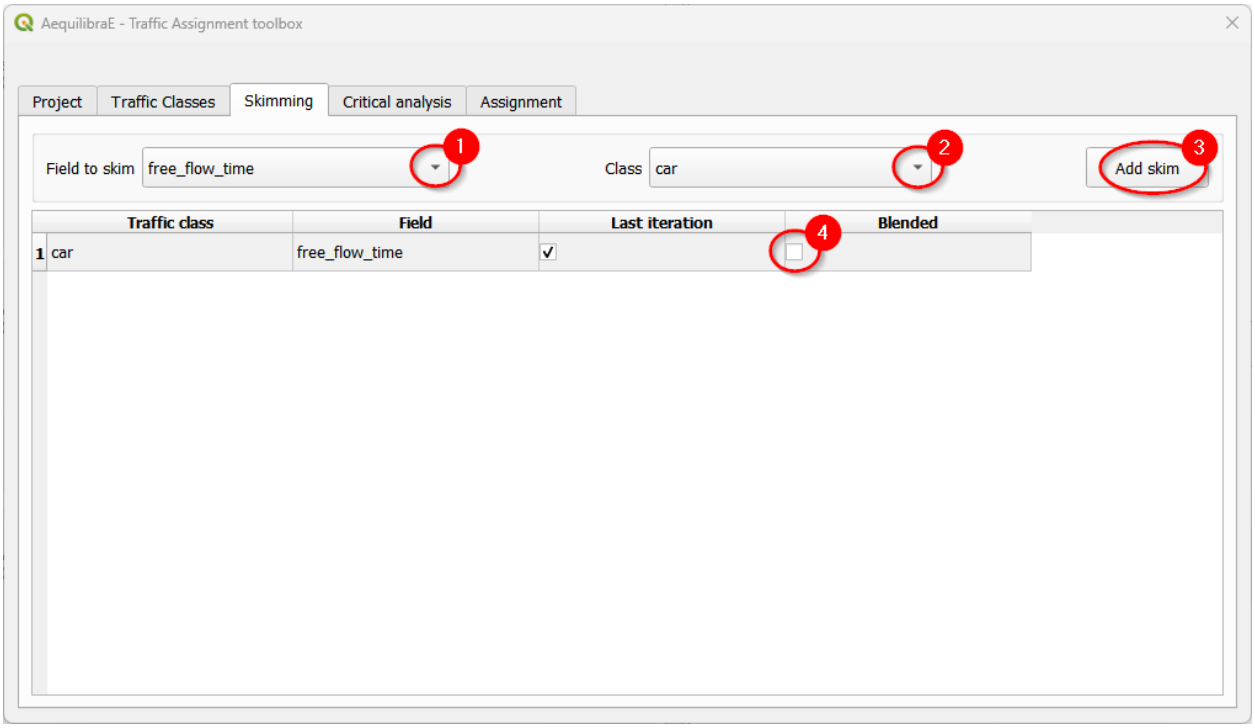
2.4.1 Basic workflow

Let's run a traffic assignment for Sioux Falls. You don't have to worry with demand matrices: we'll use one of the matrices already available in the model.

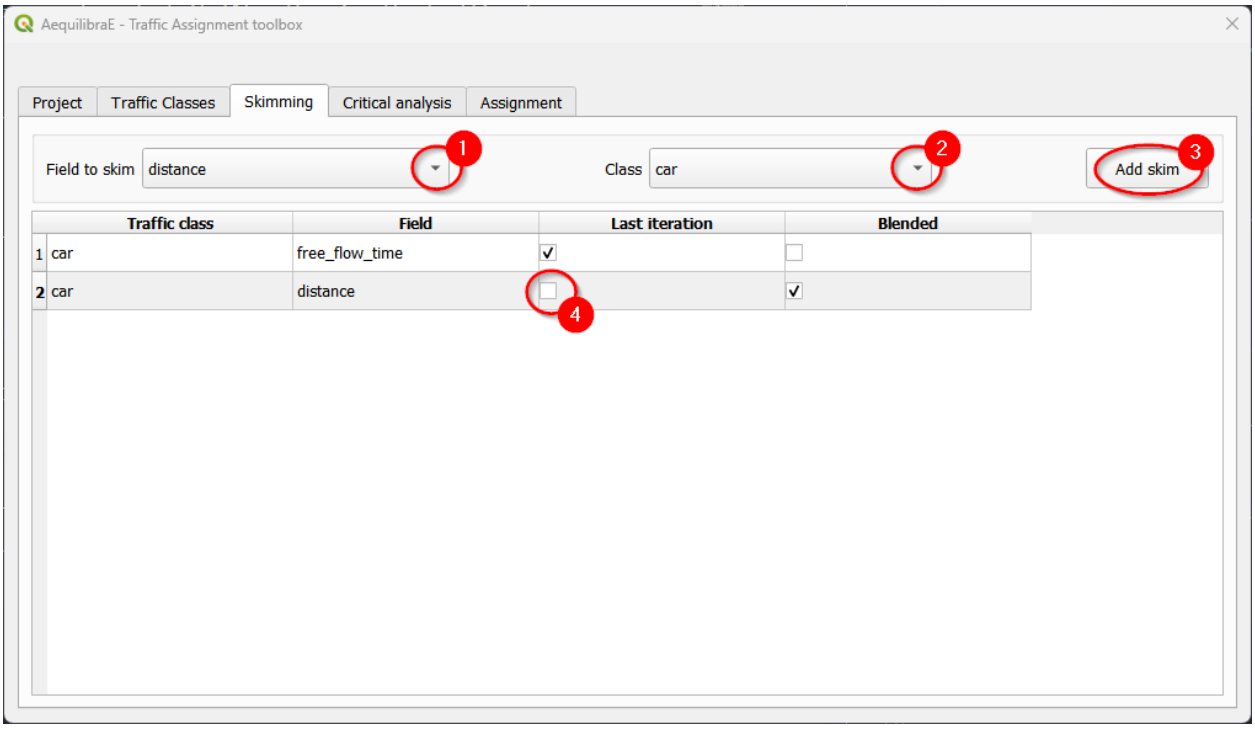
We start selecting one of the available matrices and the matrix core that will be used for computation. We must click on the matrix core to select it (2), otherwise the traffic class creation will fail. As this is a single class traffic assignment, we'll use the default configurations for network mode and assignment class name (3 and 4). For the Sioux Falls example, we don't want to block flow through centroids because regular nodes of the network are centroids (5). When you finish, just hit the *Add Traffic class to assignment* button (6). You will notice that your traffic class will automatically appear at the table at the right-hand side of the screen.



To select skims, we need to choose which fields/modes we will skim. If no mode appears at the class field, you should have missed any configuration on the traffic classes tab before. Let's continue with 'free_flow_time' and 'distance'. We select the field to skim (1) and the class (2) and add it (3) to the skimming table, at the bottom. Let's skim 'free_flow_time' only for the last iteration by untoggling the 'blended' check box (4).

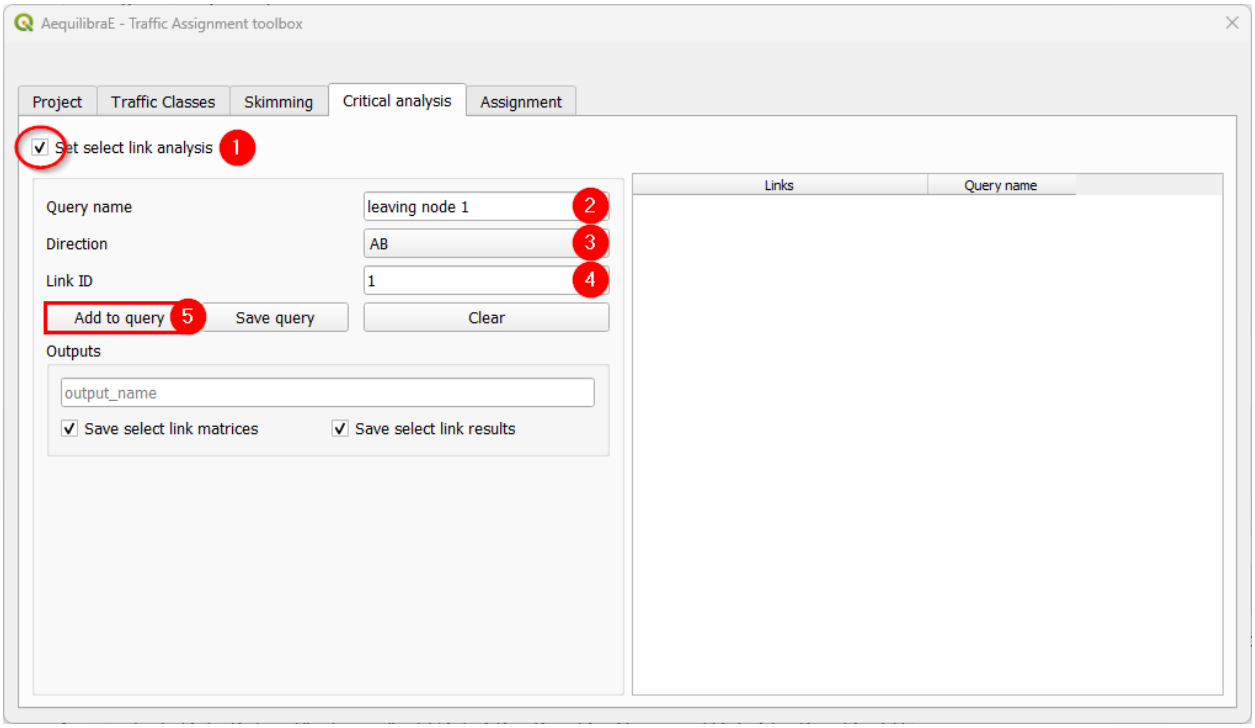


Let's repeat the skim selection process for 'distance', but now we'll use the blended results.

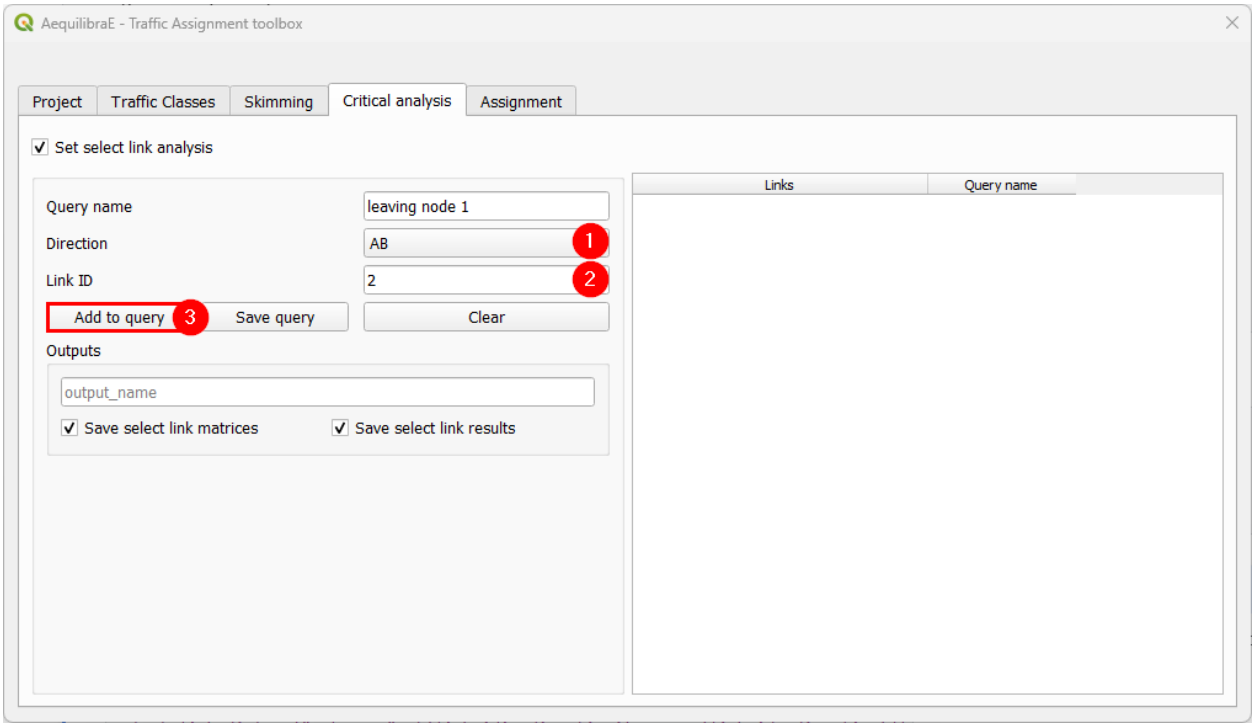


If you want to have skims for both final and blended, you don't have to untoggle any check boxes!

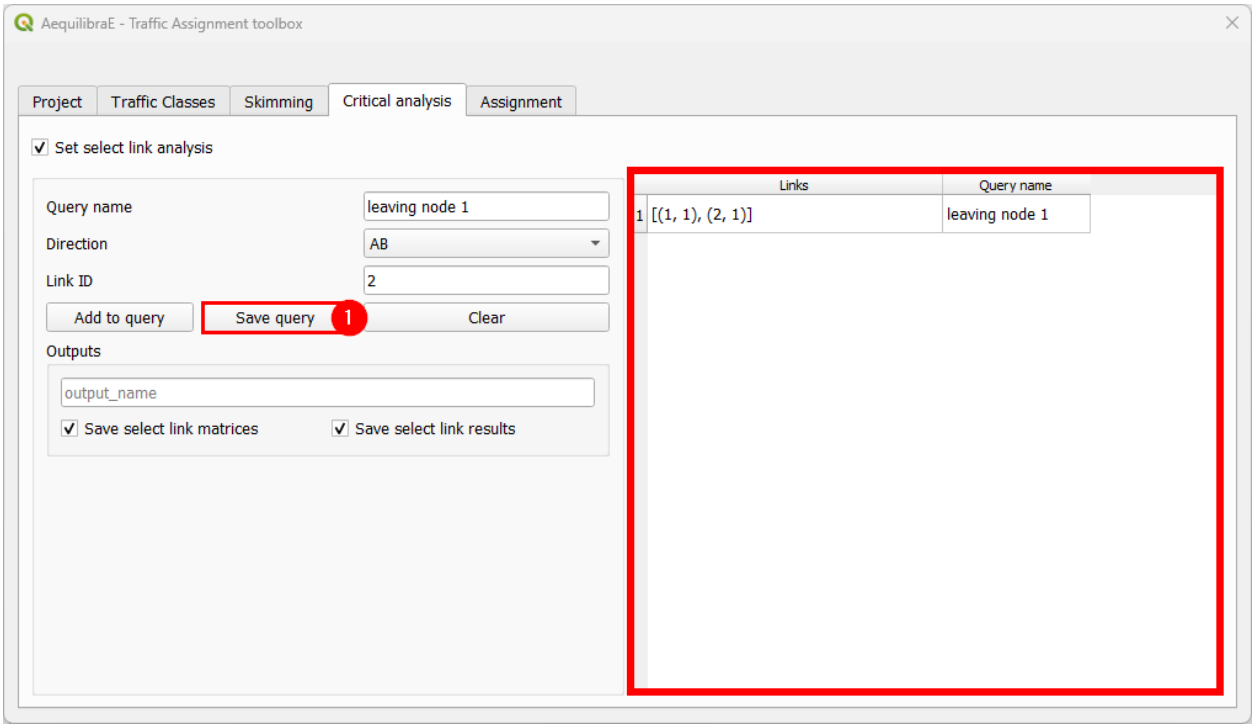
Then, we can choose to run a select link analysis. Its default configuration is not to select any links, so we have to toggle its *"Set select link analysis"* button (1). The creation of queries for analysis consists in: create a name for the query, select the travel direction, add the link ID, and click on *Add to query*, to temporarily save the data to the query.



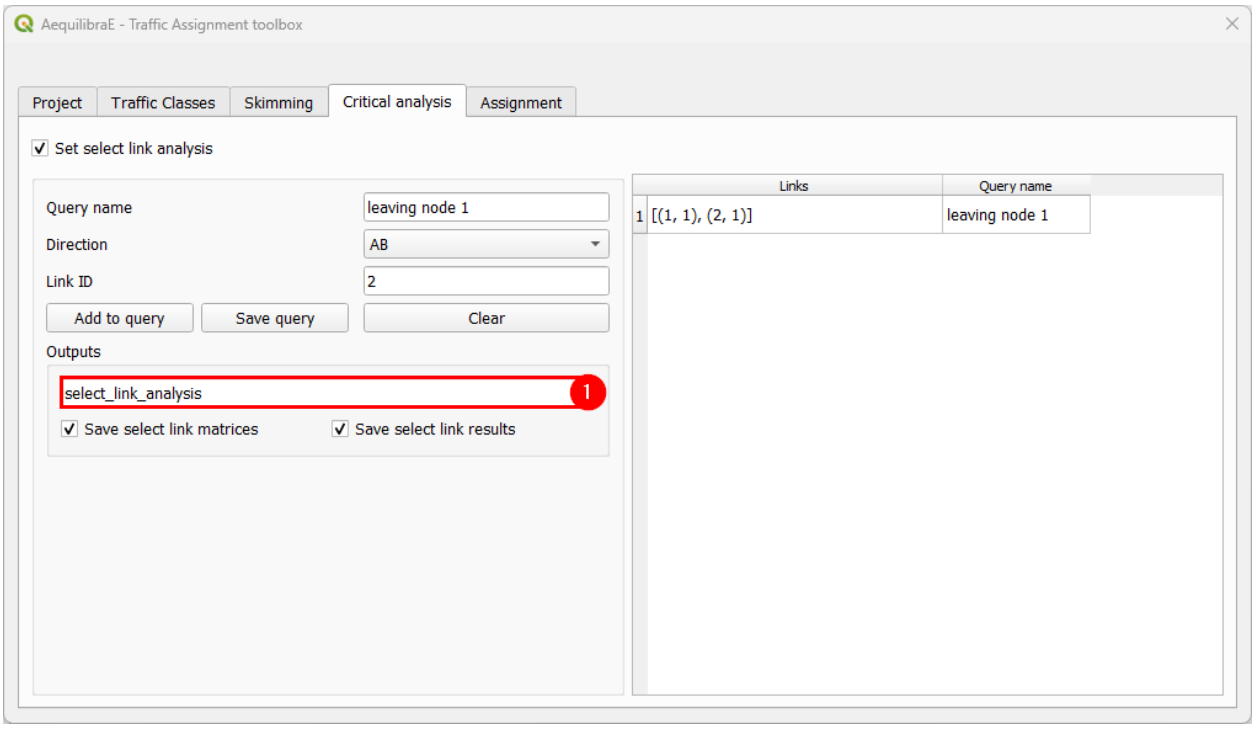
Adding more links to the previous query is straightforward. Select the direction and the link ID, and press *Add to query* once again.



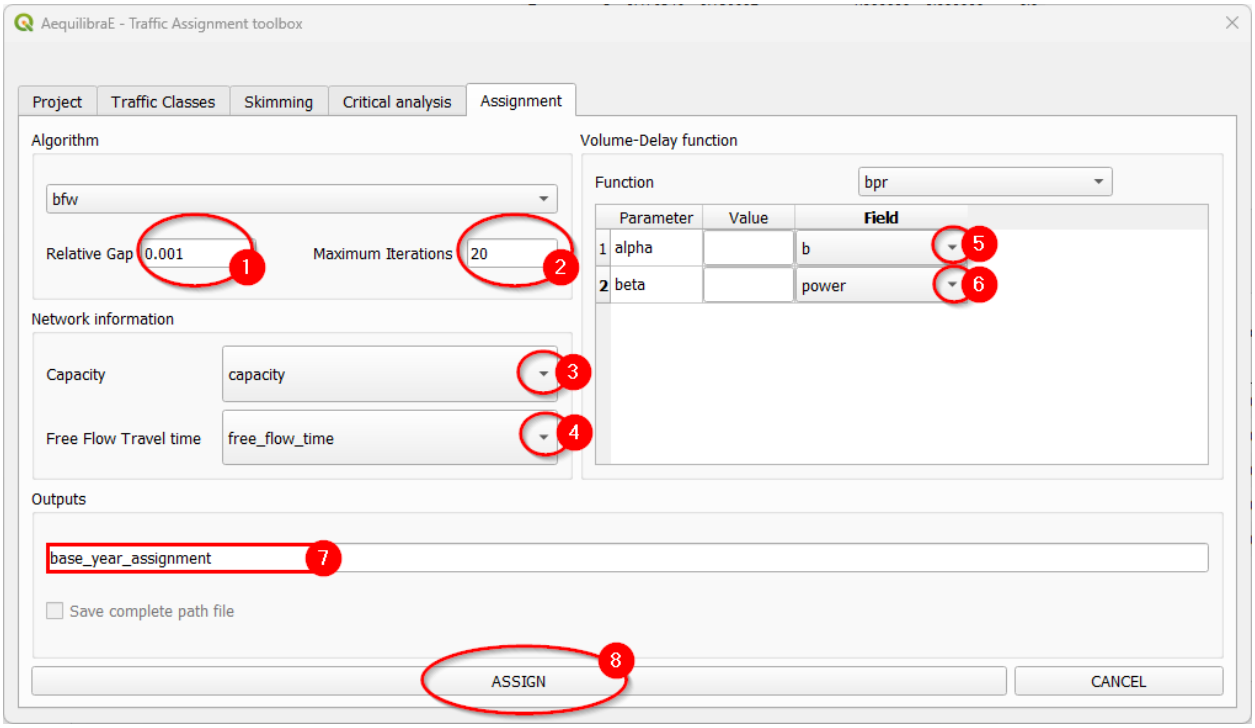
When we are done with the current query, we click on *Save query*, and notice that the query with the selected links is going to appear in the right-hand side queries table.



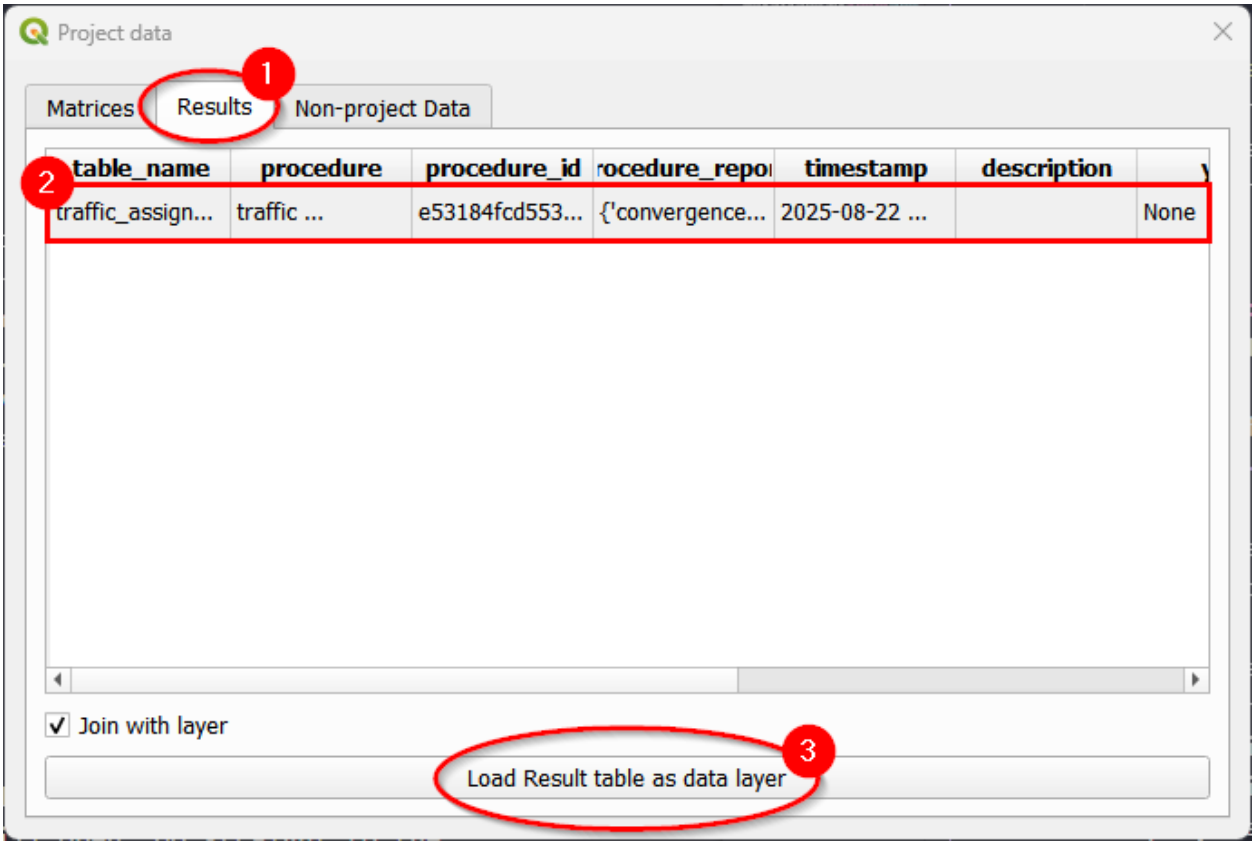
To finish the select link analysis step, we choose one name to save one or both of the matrix and results files.



The final step is to setup the assignment itself, by selecting the algorithms, setting up the relative gap and maximum number of iterations, volume-delay function, network information, and the assignment output name. When configuring the parameters of the VDF function, it is possible to use a value by typing it or an existing field. When configurations are done, just click on the *"Assign"* button and wait for the results. When QAequilibraE finishes the assignment procedure, the traffic assignment window automatically close.



The result of the traffic assignment we just performed is stored in the results database within the project folder. It can be easily accessed and loaded by clicking **Mapping > Visualize data**, and a project data window will open. Go straight to the *Results* tab (1), and select the desired result (2), let the *Join with layer* option checked, and click in the *Load Result table as data layer* button at the bottom (3). The result table layer will be automatically joined with the links layer and will appear at your QGIS mapping canvas area.



If you want to enhance your data visualization, now we can revisit the instructions for *Stacked Bandwidth*

2.4.2 Creating a YAML configuration file

One update in QAequilibraE's Traffic Assignment is allowing the user to load the assignment configurations from an YAML file. In this section we present how you can manually set up your YAML file.

1. The main file keys are 'traffic_classes' and 'assignment': they are mandatory and contain the basics of traffic assignment configuration.
2. When configuring a traffic class, use lists with dictionaries. The item corresponds to the dictionary key and is going to be your traffic class name. If you have more than one traffic class, as in multi-class assignment, each class is configured in the same way.
3. For each traffic class, the fields 'skims', 'fixed_cost', and 'vot' are optional. If you are going to use any of them, make sure to properly fill the values. For 'skims', add the skim and which results are desired: final (results of the final iteration) or blended (averaged results for all iterations).
4. The 'select_links' section is completely optional, however if you are using it the fields 'output_name' and 'selection' are required. 'save_matrix' and 'save_results' defaults to `True`. We encourage using them only if you don't want to save any of the outputs. The selection contains dictionaries whose keys are query names and the values are a list of lists containing the 'link_id' and 'link_direction' (check the direction section here).

Notice that all lines that are commented in the code below are optional.

Listing 0: Traffic assignment configuration

```
traffic_classes:
  - car:
```

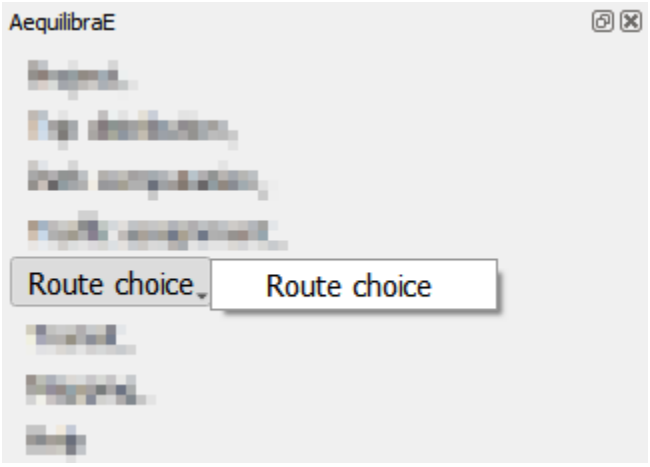
(continues on next page)

(continued from previous page)

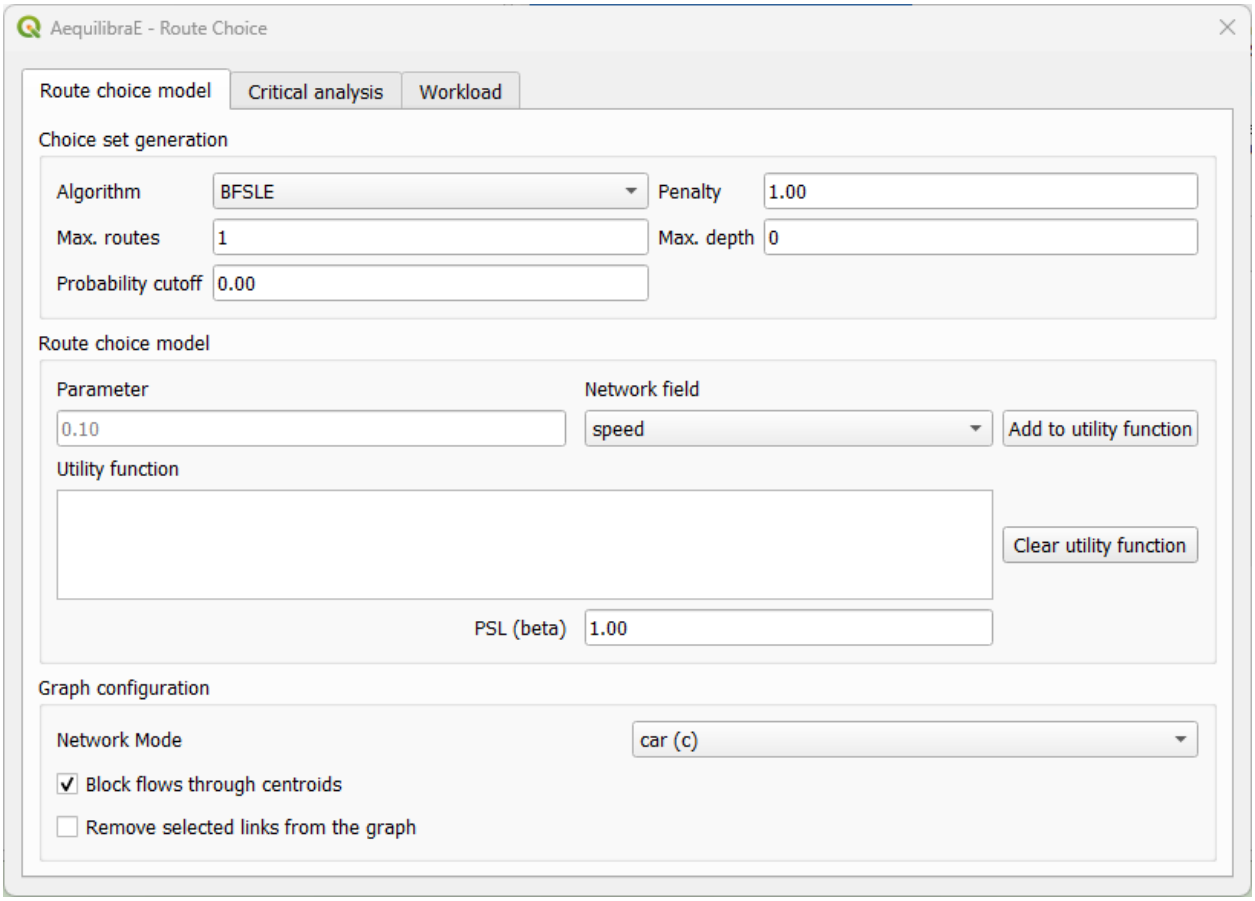
```
      matrix_name: demand_omx
      matrix_core: matrix
      network_mode: c
      pce: 1.0
      blocked_centroid_flows: False
      # skims:
      #   free_flow_time: [final]
      #   distance: [blended]
      # fixed_cost: vot
      # vot: 1.05
assignment:
  algorithm: bfw
  max_iter: 25
  rgap: 0.001
  vdf: BPR
  alpha: 0.15
  beta: 4.0
  capacity_field: capacity
  time_field: free_flow_time
  result_name: result_test_from_yaml
# select_links:
#   output_name: select_link_analysis_from_yaml
#   save_matrix: True  # optional
#   save_result: True  # optional
#   selection: # name with a list of lists as [[link_id, link_direction]]
#     from_node_1: [[1, 1], [2, 1]]
#     random_nodes: [[3, 1], [5, 1]]
```

2.5 Route choice

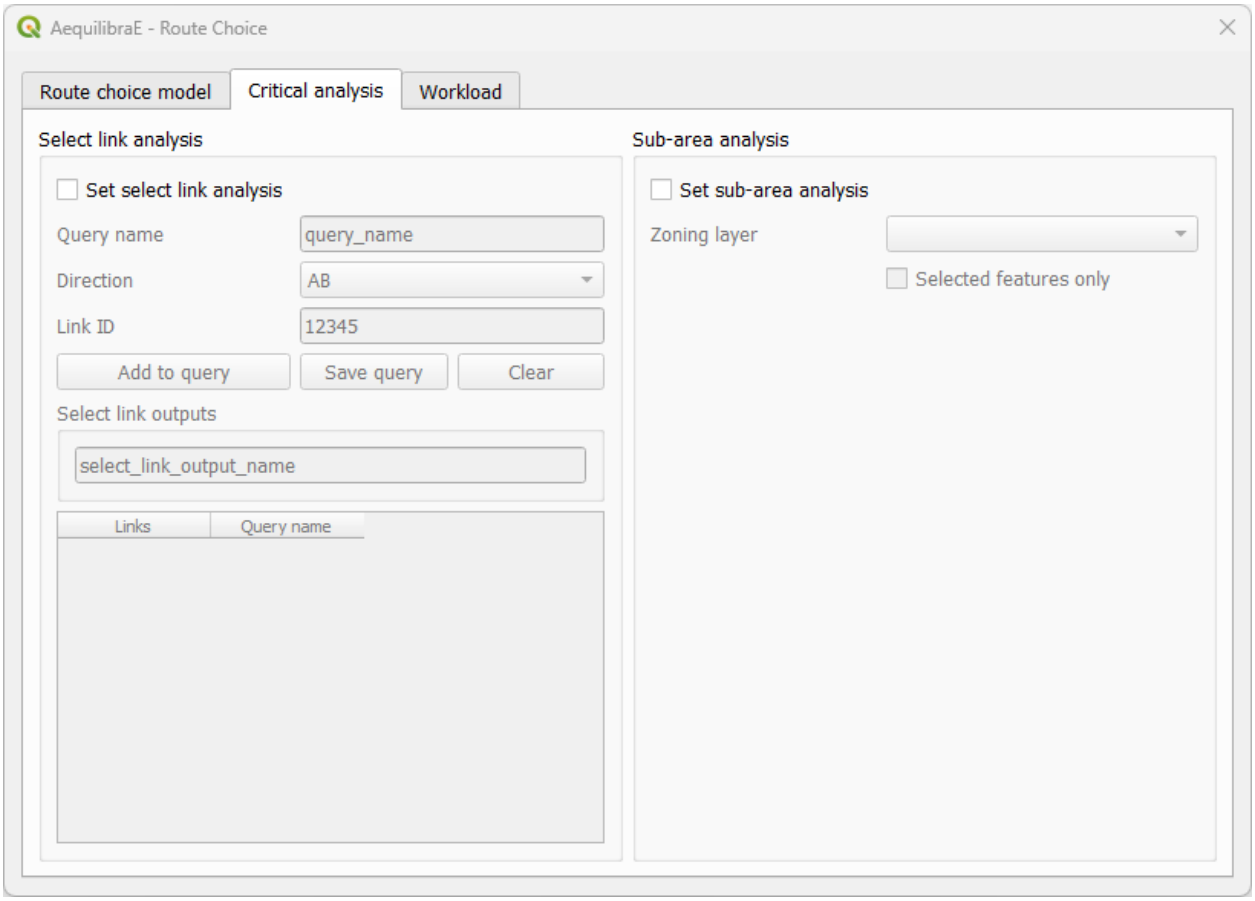
With the route choice sub-module, it is possible to create choice sets with three different algorithms as well as assign trips to the network using the traditional path-size logit. Using this module in QAequilibraE is trivial.



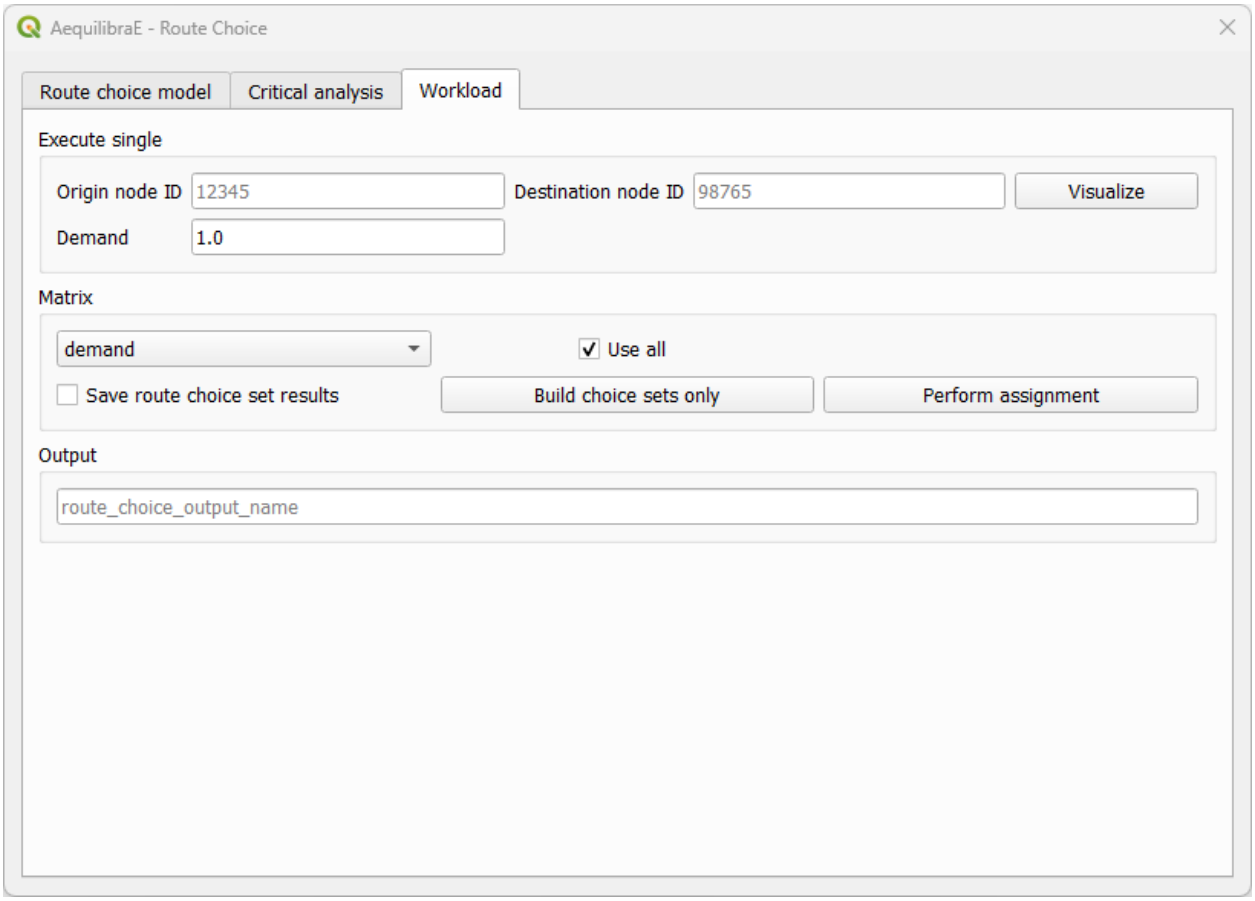
In the tab "Route choice model", we add the model configuration. It consists of three different boxes. In the first box "*Choice set generation*", we input parameters for the choice set construction. In the "*Route choice model"*, we add the parameters for the route choice model, such as the utility function and the path overlap parameter (PSL/beta) value. Finally, in "*Graph configuration*" we set up the graph used for computation.



In the tab "Critical analysis", we can select to run either a set of select link analysis or a sub-area analysis. These analyses cannot be run at the same time in QAequilibraE. If you choose to run a sub-area analysis, all OD pairs with demand are considered for computation. To select only a few pairs of interest, we encourage you to take a look at Route choice with sub-area analysis at AequilibraE's Python documentation and run this task outside QGIS.



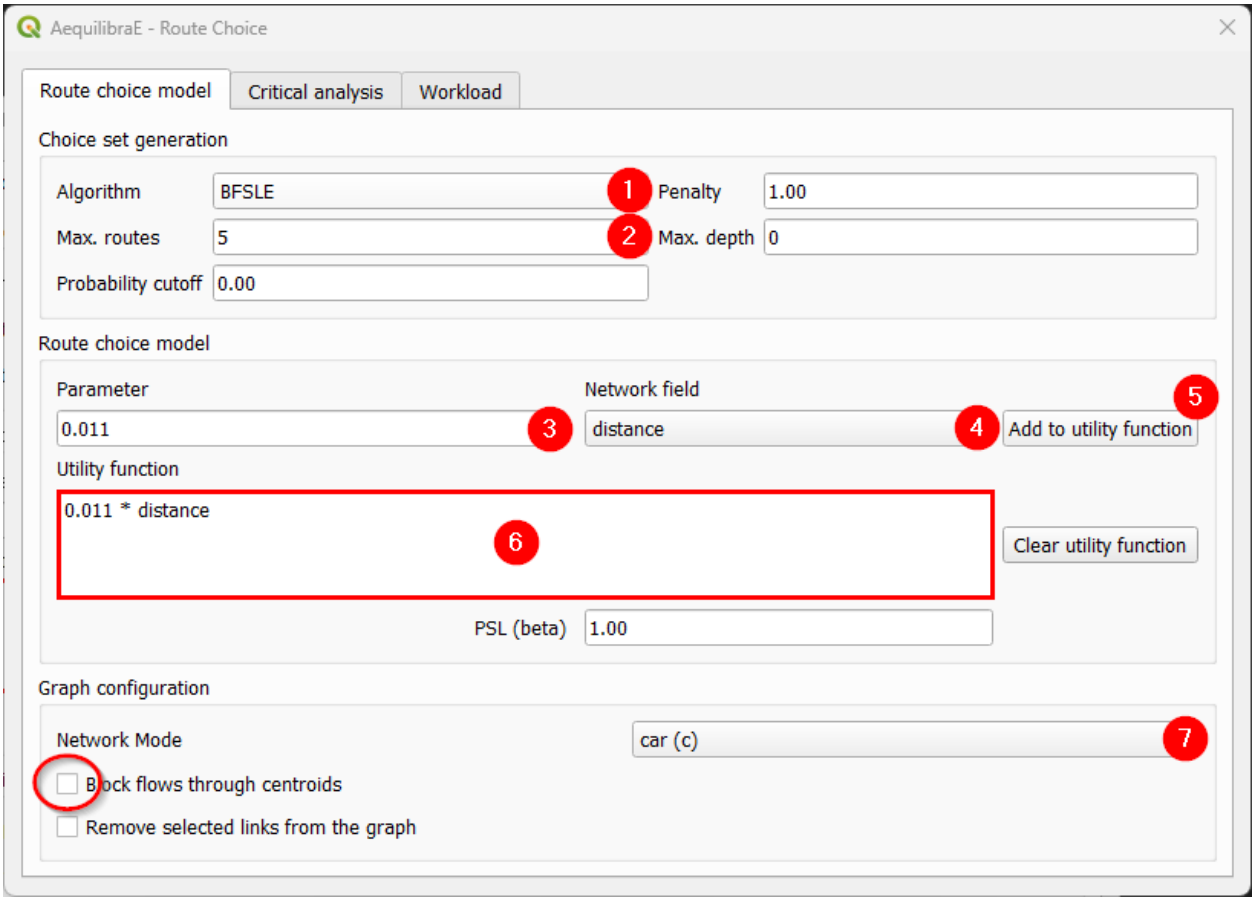
Lastly, the tab "Workload" allows users to choose between three tasks. The first box, "*Execute single*" consists of computing route choices between two different nodes and visualizing it, while the second box "*Matrix*" allows the selection of a travel demand matrix to be assigned using the route choice specified. This option also allows the user to save choice sets to disk while performing route choice.



2.5.1 Basic workflow

Basic route choice

In this example, we'll perform route choice for the Coquimbo example model for a single OD pair. As this example model does not ship with a demand matrix, we can manually create an open layer and use its data to import the matrix to the project, as shown in *Importing matrices*.

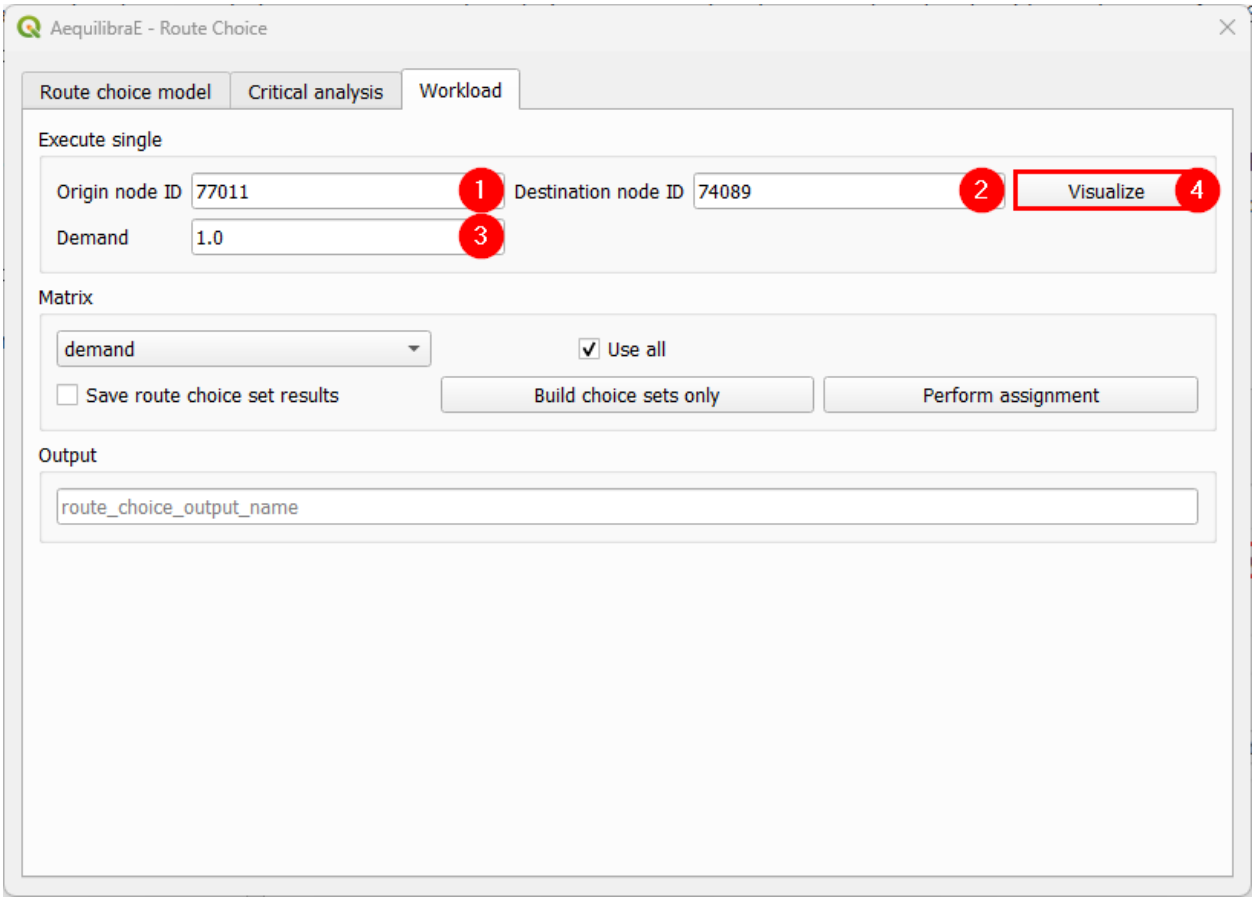


We start by setting the route choice parameters. In the "Choice set generation" box, we select the algorithm to be one of Link Penalization (LP), Breadth-First Link Search on Link Elimination (BFSLE), or BFSLE with LP, choose the values for probability cutoff and penalty, and choose a positive value for one of maximum number of routes (LP) or search depth (BFSLE and BFSLE + LP).

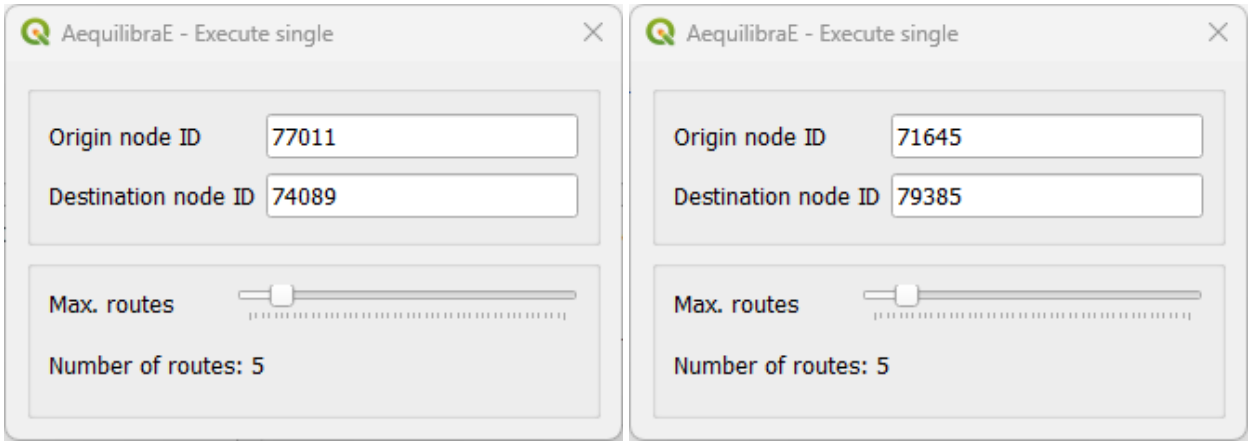
In the box "Route choice model" box, we configure our utility function. In this example, it is a function of distance, but could be any other numeric field, such as travel time or tolls. We then add the parameters to the utility function and it will appear in the utility function box. We can change the utility function by cleaning it and adding it one more time. To add more parameters to the utility function, just change the values and click in "Add to utility function" one more time.

Regarding "Graph configuration", we'll use the network for cars and allow flows through centroids.

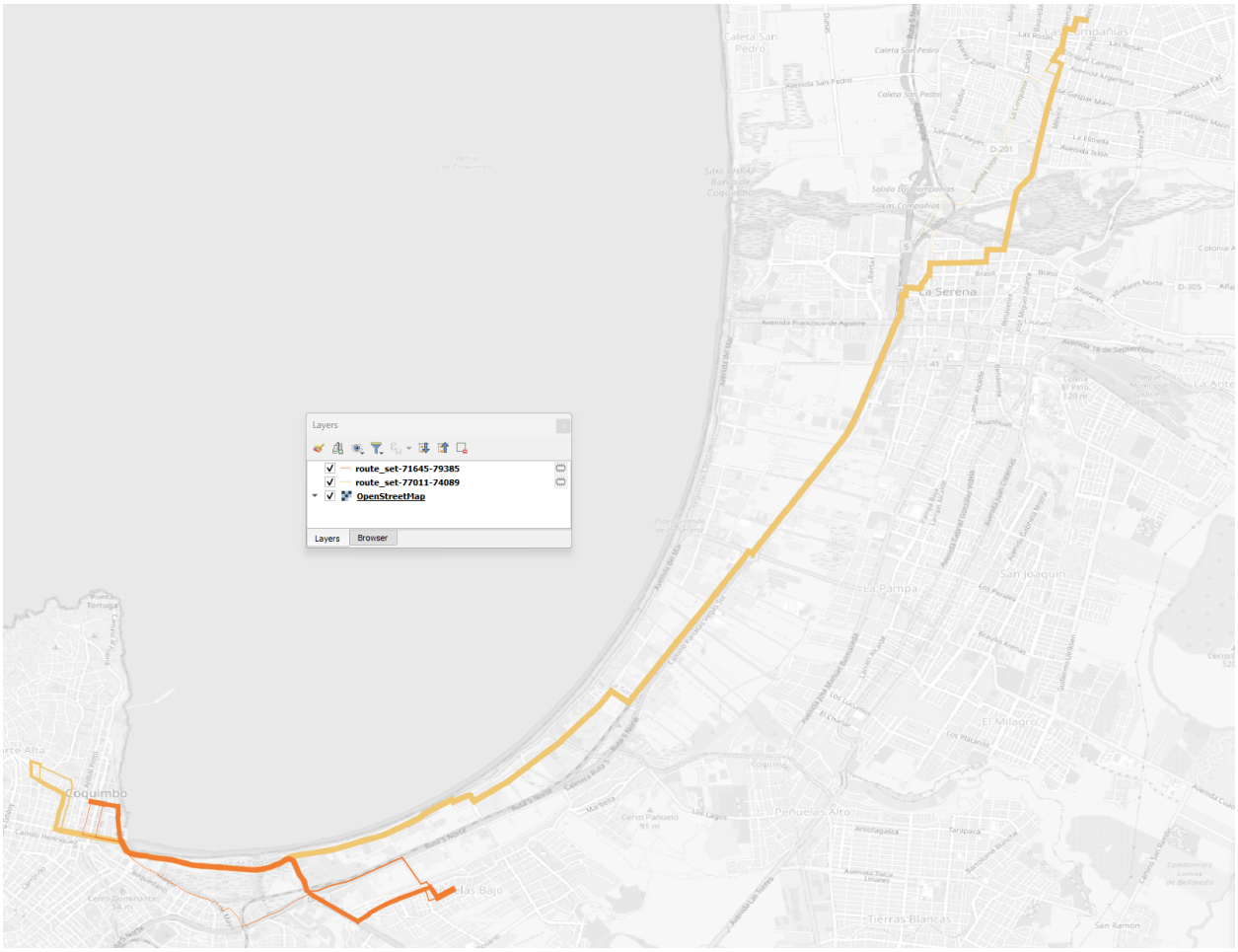
We can now move directly to the "Workload", select origin and destination nodes and click on the *visualize* button.



A new window named Execute Single will appear, loading the configuration we just used for the route choice set. If we are done with the choice set generation, we can close it, otherwise, we can generate the route choice set for another OD pair, also setting the desired number of routes.

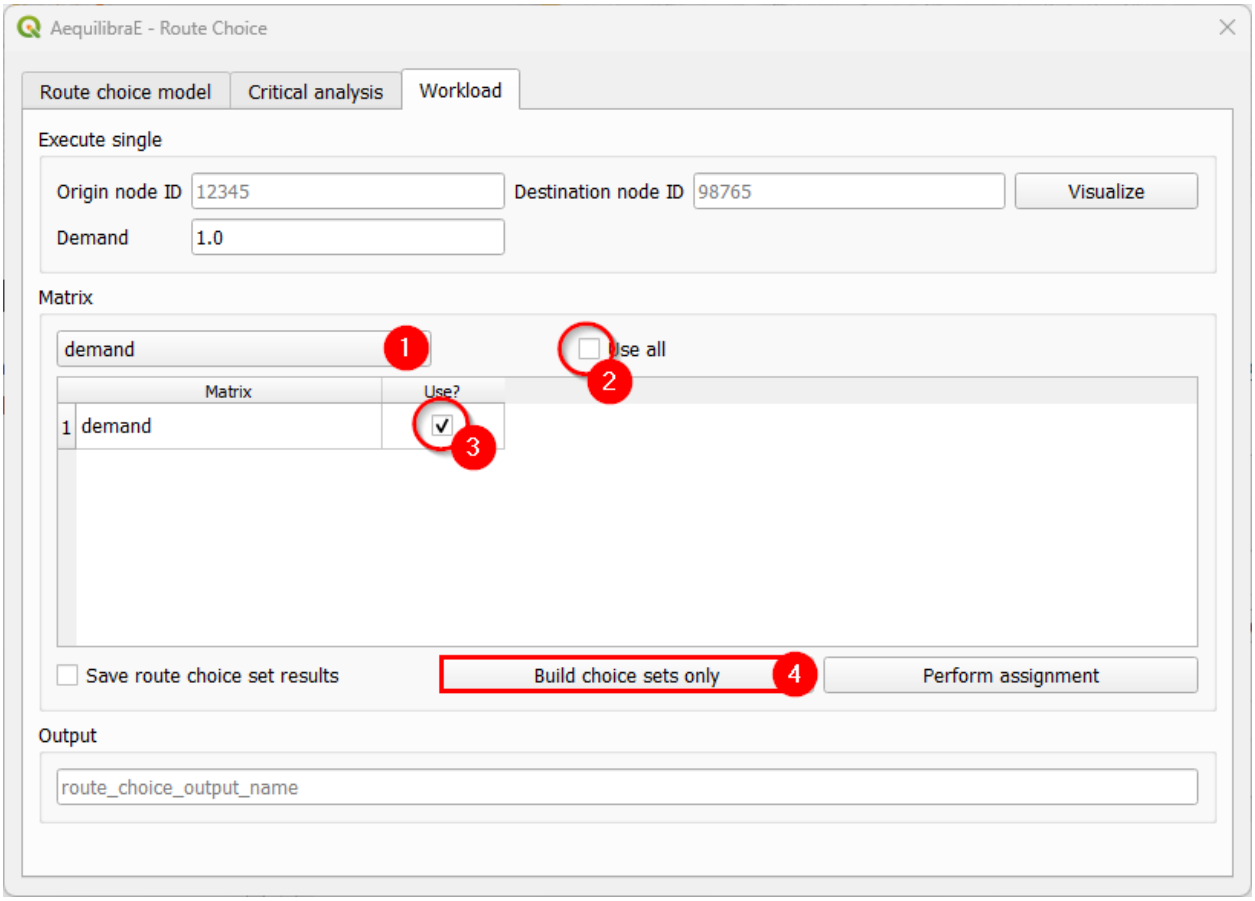


After a few seconds, the output visualization for the routes is shown in the map canvas and we can close the Execute Single window. The figure below presents the route choice sets, in which the line width corresponds to the probability of choosing each link.



Build choice sets

Within this workflow, we can build and save the choice sets without performing assignment. We start by *configuring the model parameters*, then go to the "Workload" tab and select our demand matrix and its cores for computation.



If you want to use all cores for computation, just let the "Use all" checkbox untoggled after choosing the matrix. Otherwise a table with the matrix cores and if they should be used is opened and we can select the cores we want.

Then all we need to do is hit the "Build choice sets only". Once the task is finished, our route choice window will automatically close. If you go to the project folder, you will notice that a folder named '*route choice*' containing folders with the choice sets for each centroid (index) in the matrix was created.

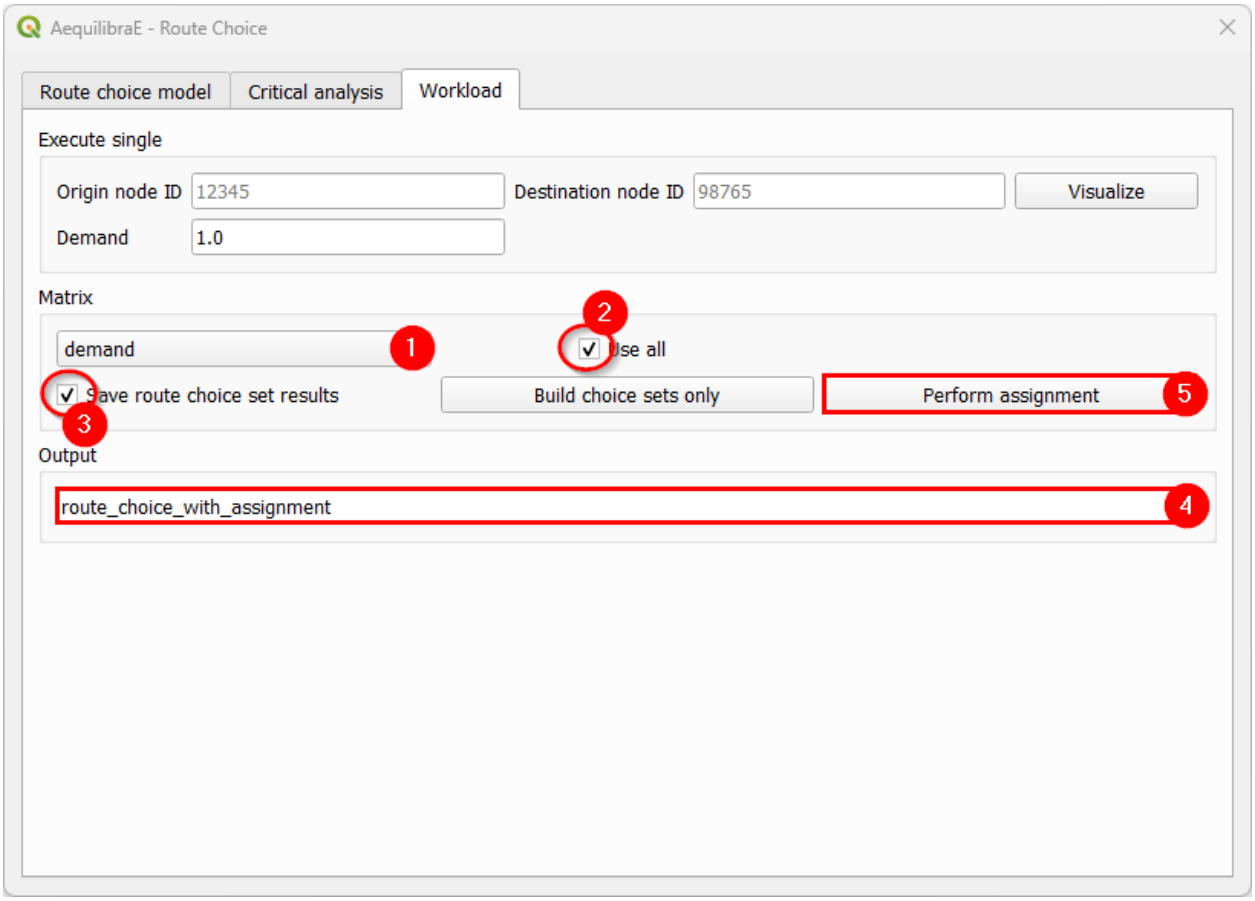
It should be noted that, although we are not performing assignment in this workflow, we use demand matrices to determine the OD pairs for which choice sets are needed, which are all of those with positive demand.

Perform assignment

This workflow runs a route choice assignment and allows the user to save the choice set generated while performing such. The set up is quite similar to the one above: After *setting the model parameters up*, we go straight to the "Workload" tab and select the demand matrix and its cores for computation.

In this example, we choose to also save the choice sets generated, by toggling the "Save route choice set results" button. If we leave this button untoggled, only link flows are saved into the results database.

We also choose a name for saving the results in the database. Pick up a name that you can easily find later. Then, just hit the button "Perform assignment" and wait until the window is closed and the process is finished.

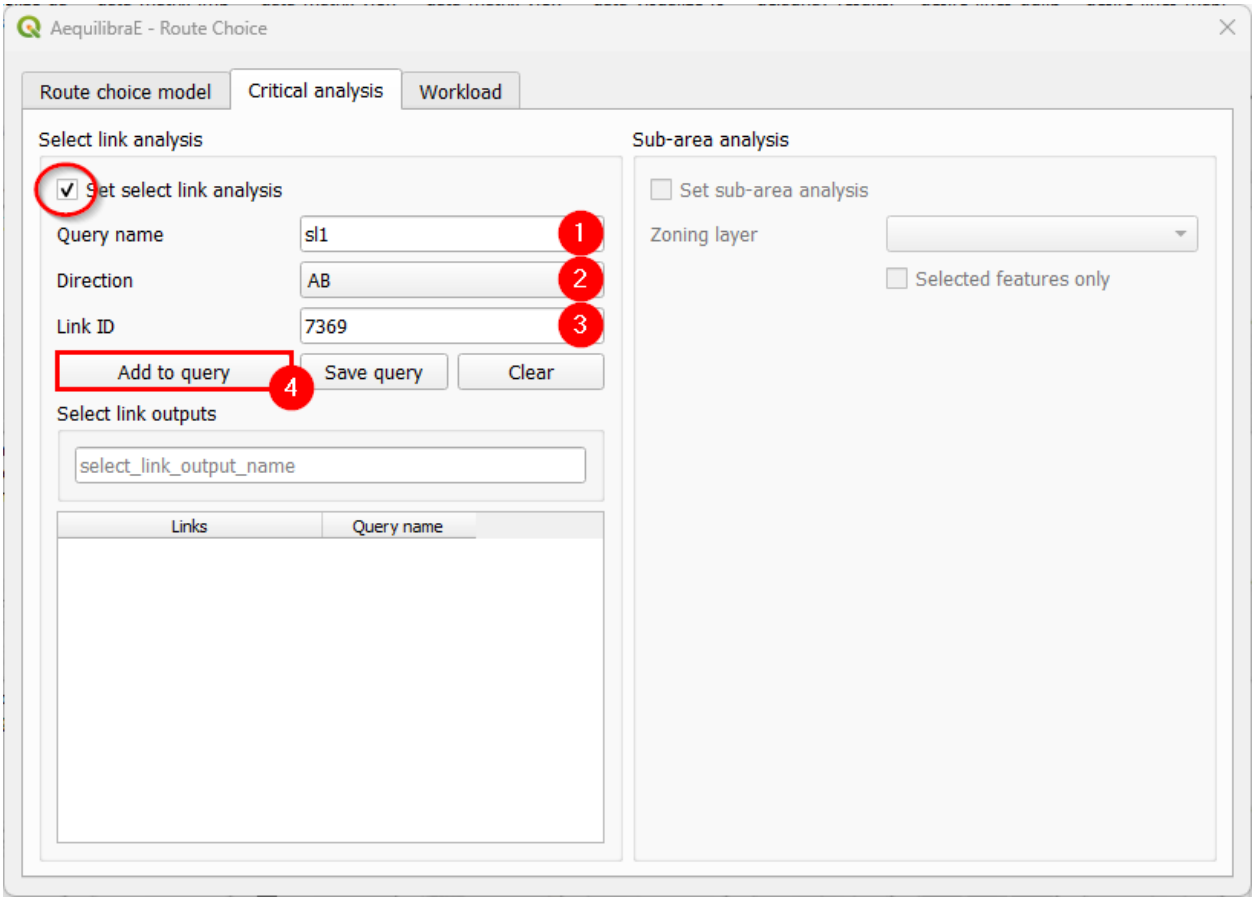


Select link analysis

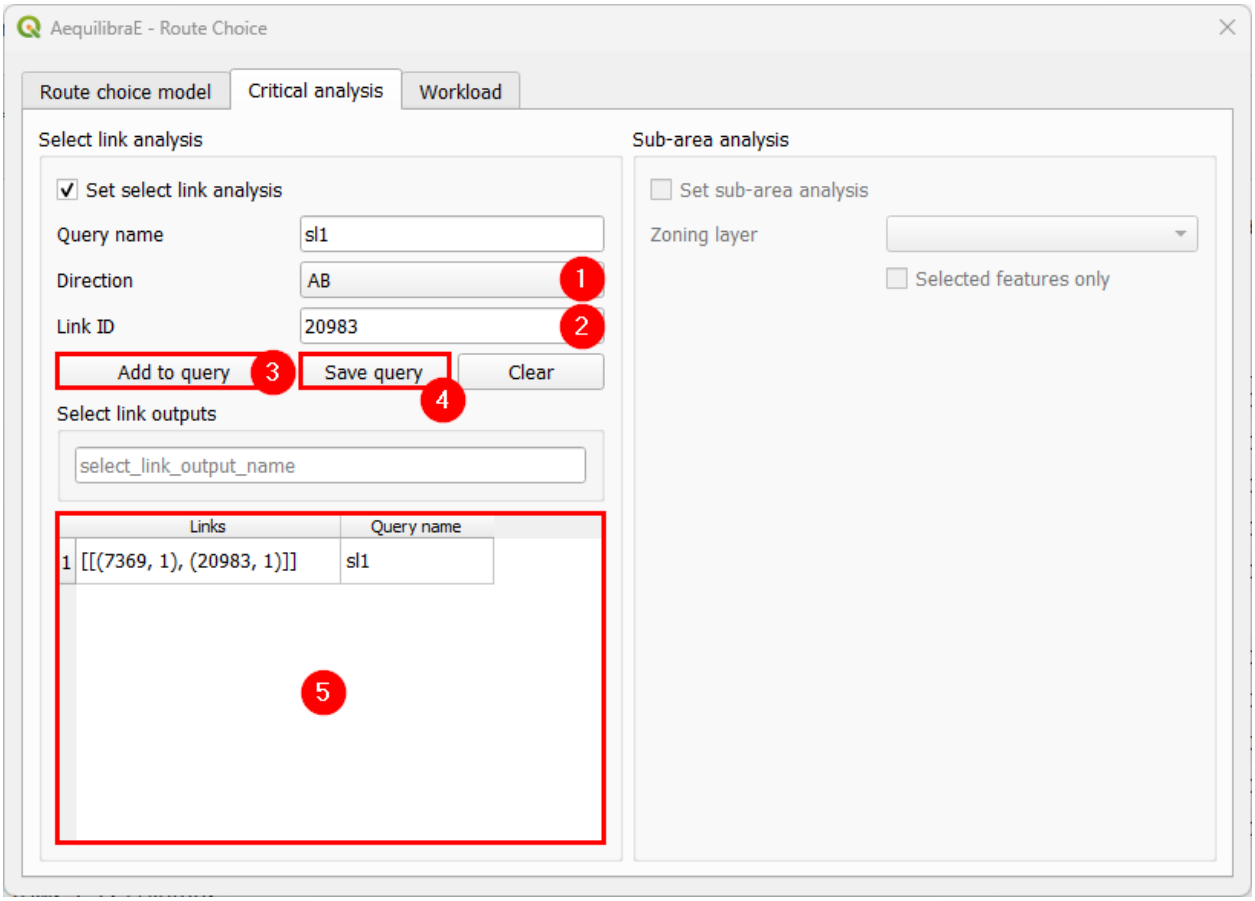
The left portion of the "Critical analysis" tab gives the user access to select link analysis. Its interface is quite similar to the one in Traffic Assignment, in which we can add and remove queries with selected links, and save both the matrix and the results in the databse.

We start by toggling the "Set select link analysis" checkbox and enabling the following menus.

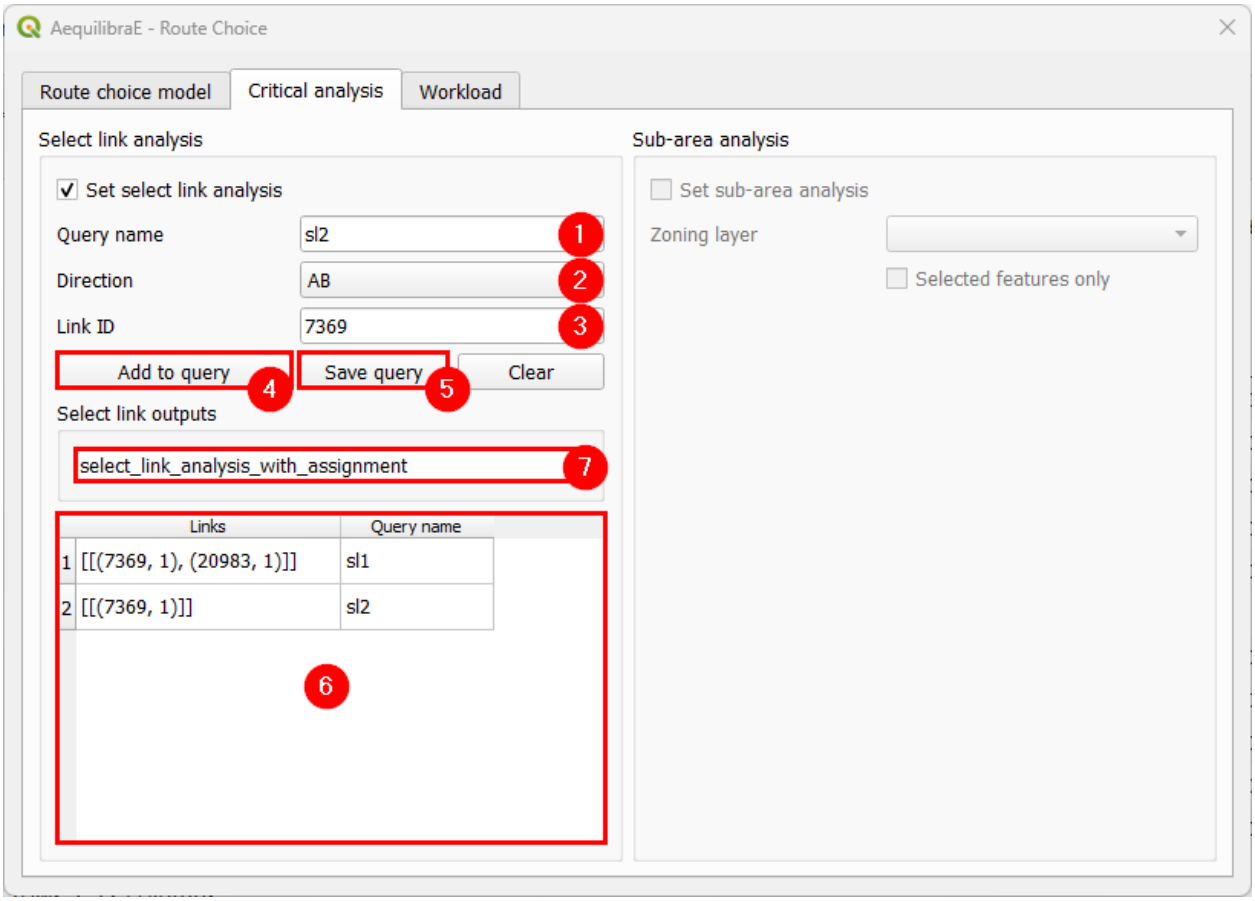
Let's add our first query. Create a name, set the link direction, add the link ID, and click on "Add to query".



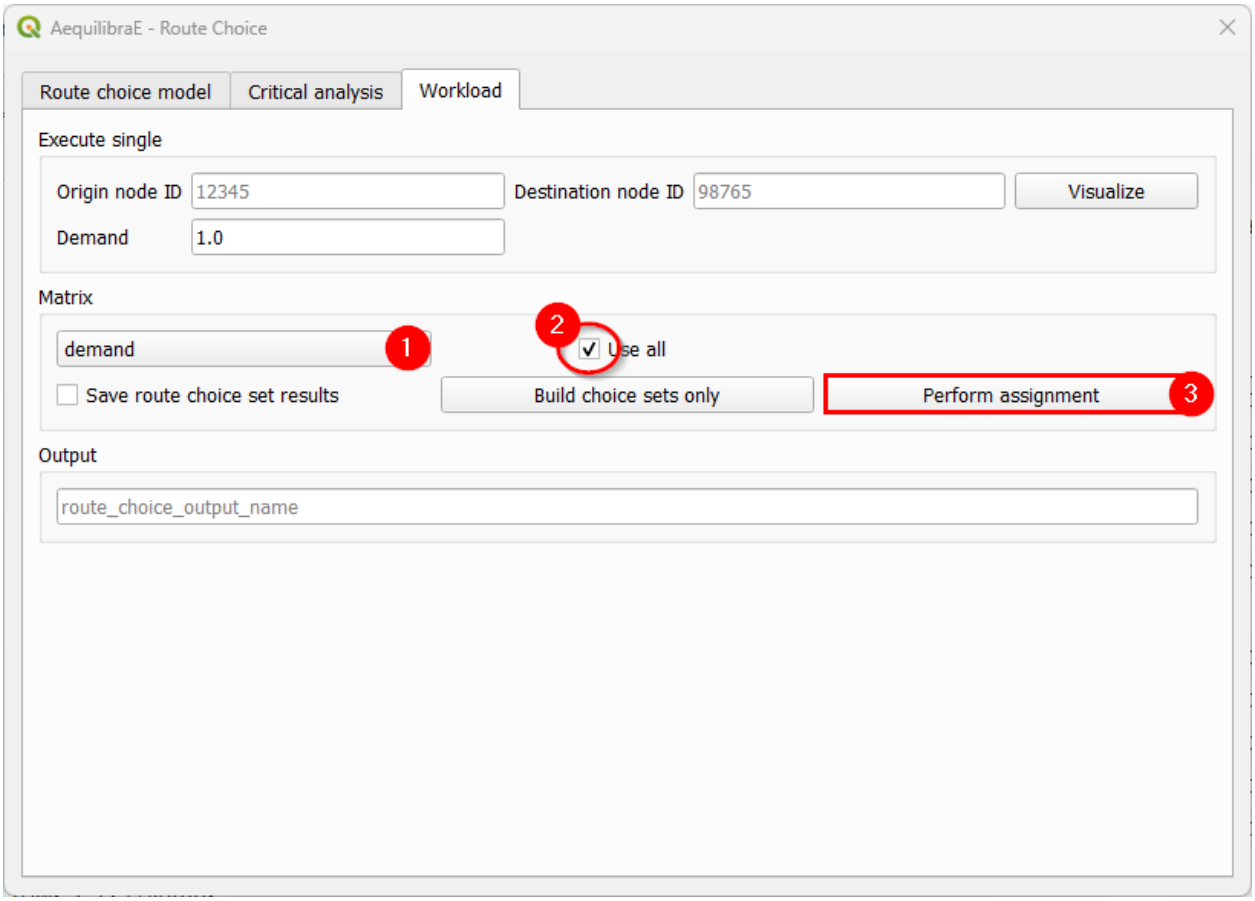
Let's add another link to our *SL1* query. Let's set the link direction and link ID, add to the existing query with "Add to query", and click on "Save query" (4). The *SL1* query will immediately appear in the table at the bottom of the window (5).



Just to make this example more interesting, let's create an *SL2* query. We repeat the process of creating a query name, setting the direction, selecting link ID, adding and saving the query. It will also appear at the bottom table (6). To remove any query from the query table, we can double-click the cell. Once this is our last query, we pick up a nice name to save our select link analysis results (7).

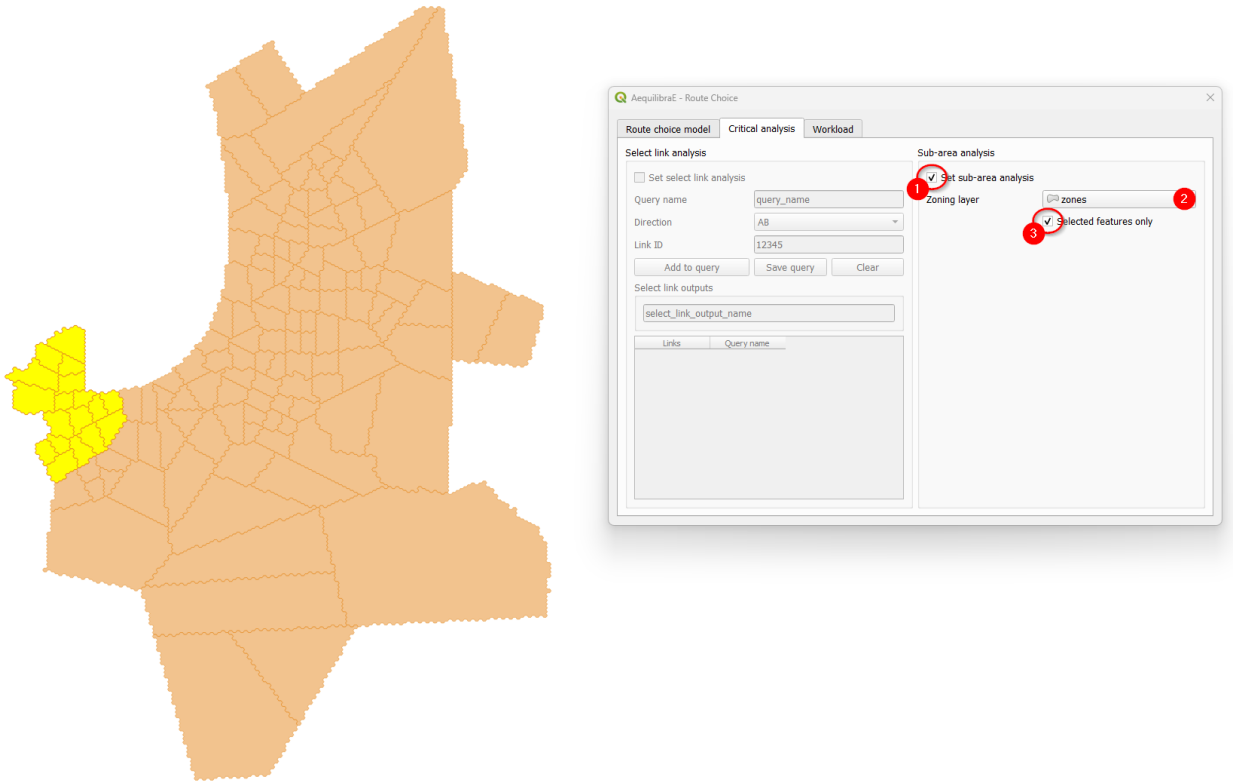


The last step consists in selecting the matrix and its cores for computation, and perform the assignment. It's not necessary to add a name to the route choice output, once we did it in the previous step.

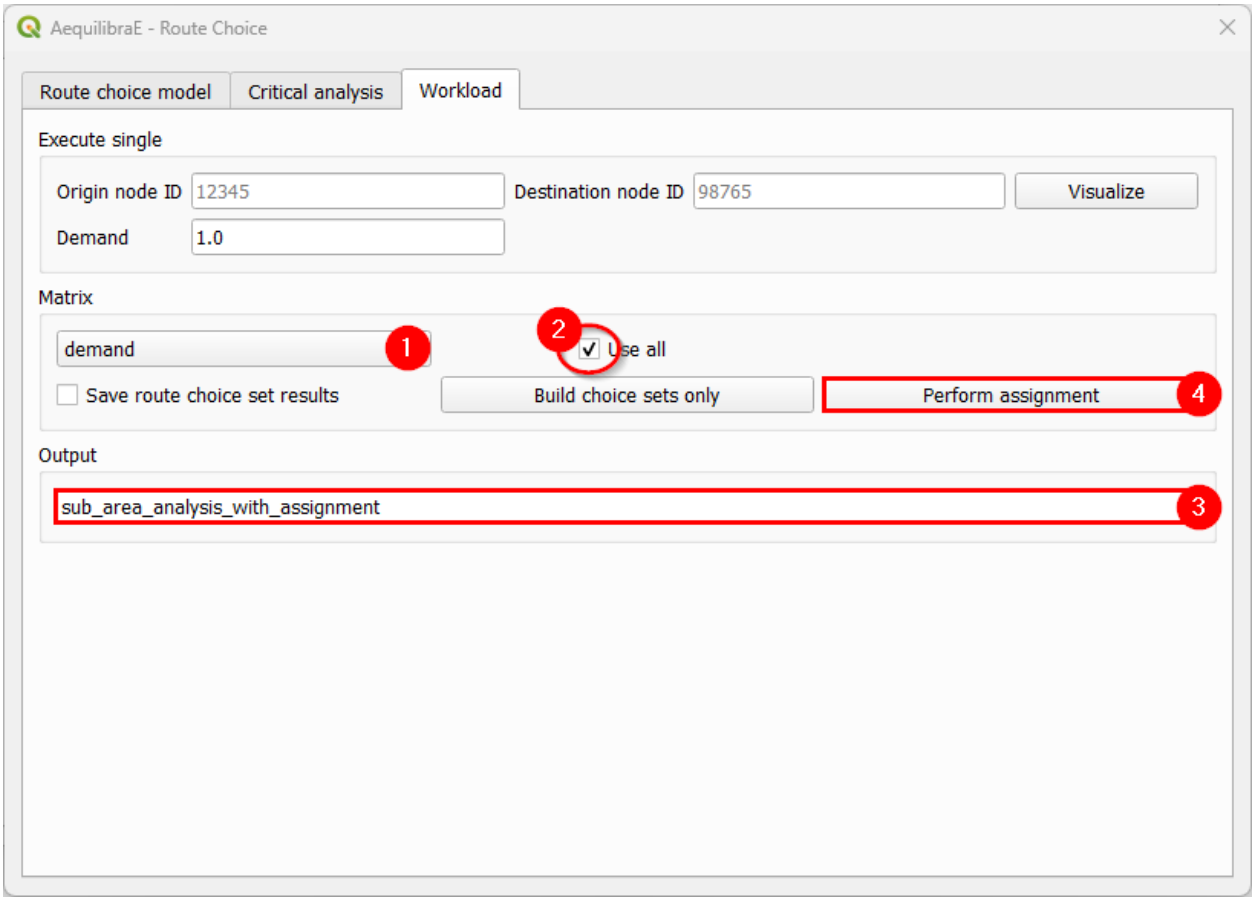


Sub-area analysis

To perform a sub-area analysis, we start by toggling the "Set sub-area analysis" checkbox, which enables us to choose a polygon layer that defines the sub-area of interest. In this example, we select a couple zones in Coquimbo, and toggle the checkbox "Selected features only". We could also use an external polygon layer with the desired region and use all the layer features rather than a part of it.



Finally, select all cores of our demand matrix for computation, don't forget to add a name for the output file, and hit the "Perform assignment" button. When the process is finished, the window is closed. If you go to the project folder, you will notice that a folder named '*route choice*' containing a `.parquet` file with the same output name you selected in (3) containing the sub-area demand matrix.

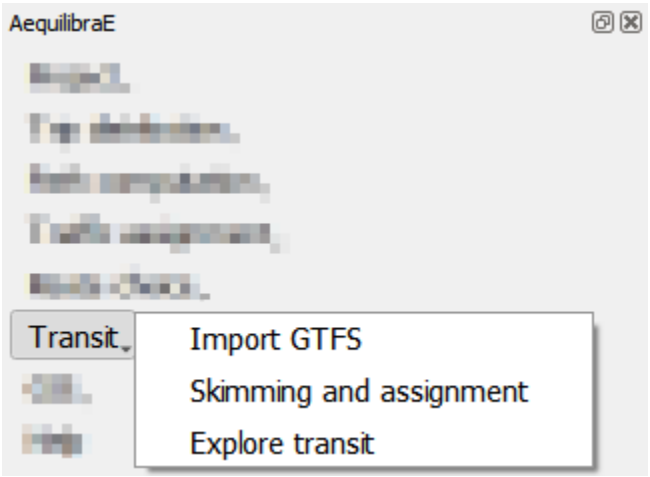


Tip

Try to reproduce AequilibraE's Route Choice examples in QGIS!

2.6 Transit

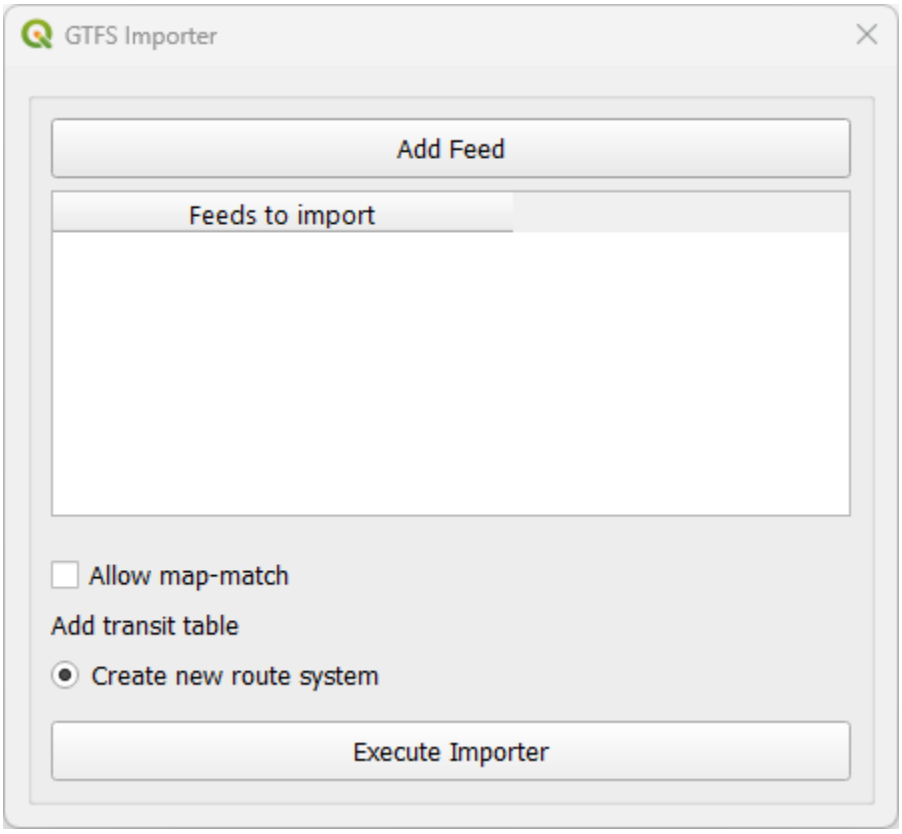
QAequilibraE is capable of importing a General Transit Feed Specification (GTFS) feed into its public transport database.



More details on the **public_transport.sqlite** are discussed on a *per-table* basis in AequilibraE's documentation, and we recommend understanding the role of each table before setting an AequilibraE model you intend to use. If you don't know much about GTFS, we strongly encourage you to take a look at the documentation provided by Mobility Data.

2.6.1 Import GTFS

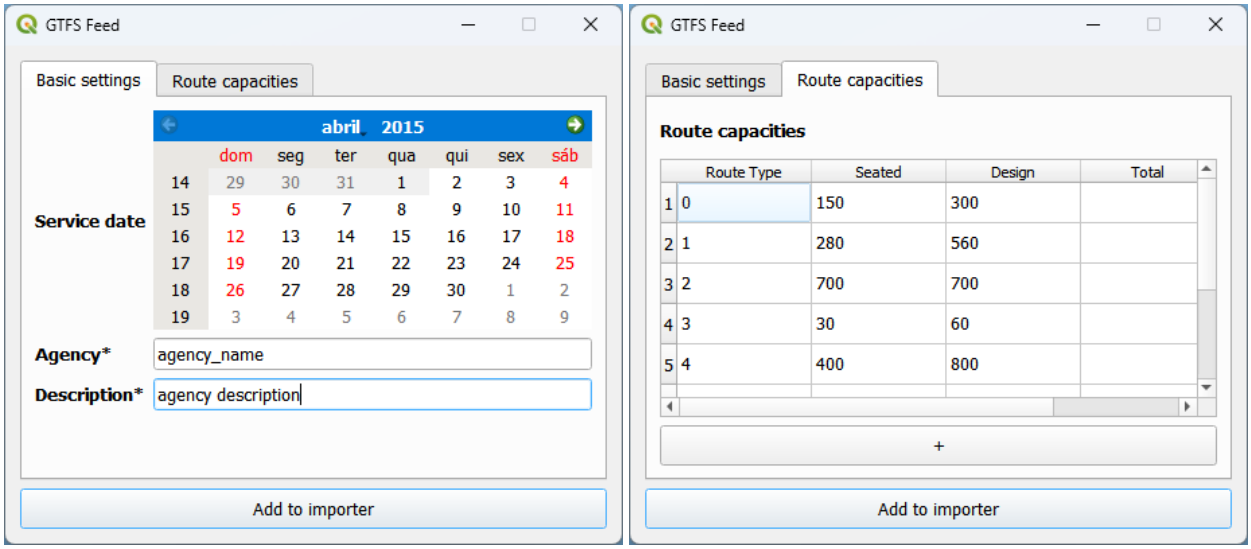
To import a GTFS feed, click **Public transport > Import GTFS**. A new window with the importer will open. If it is the first time you are creating a GTFS feed for your project, it may take a little while to create the public transport database in the project folder, and your QGIS screen might not be responsive until the database is created in the project folder. In the GTFS importer window, you can click on *Add Feed* and point to the location in your machine where the GTFS data is.

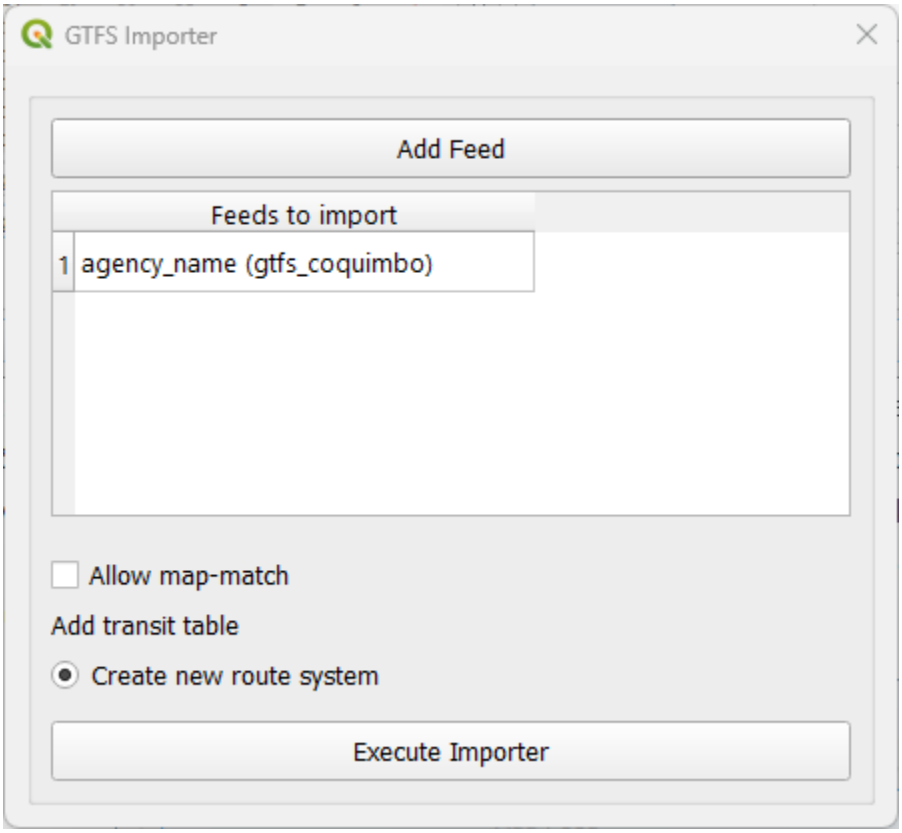


Once the feed is loaded, you can select the service date, the agency name, and write a description for it. It is also possible to add and/or modify the route capacities. When you're done, just click on **Add to importer** and you will return to the GTFS importer screen.

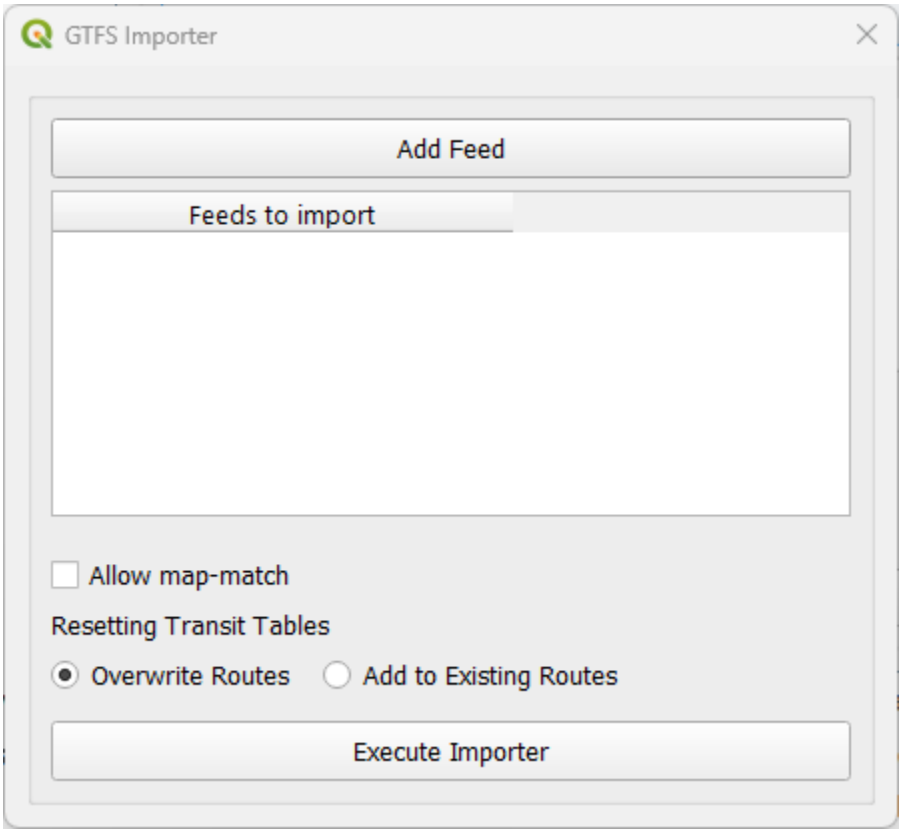
Notice that the feed information is now available at the *Feeds to import* table view. The first time you create a GTFS feed, the only option available is **Create new route system**, so you don't have to click on it. If you want to map-match the existing transit routes, you can select **Allow map-match**. Then, you can import your GTFS feed to your project by clicking on **Execute Importer**.

A window with a progress bar will open and once it is finished, you can check out the GTFS feed data you just imported in your project folder.





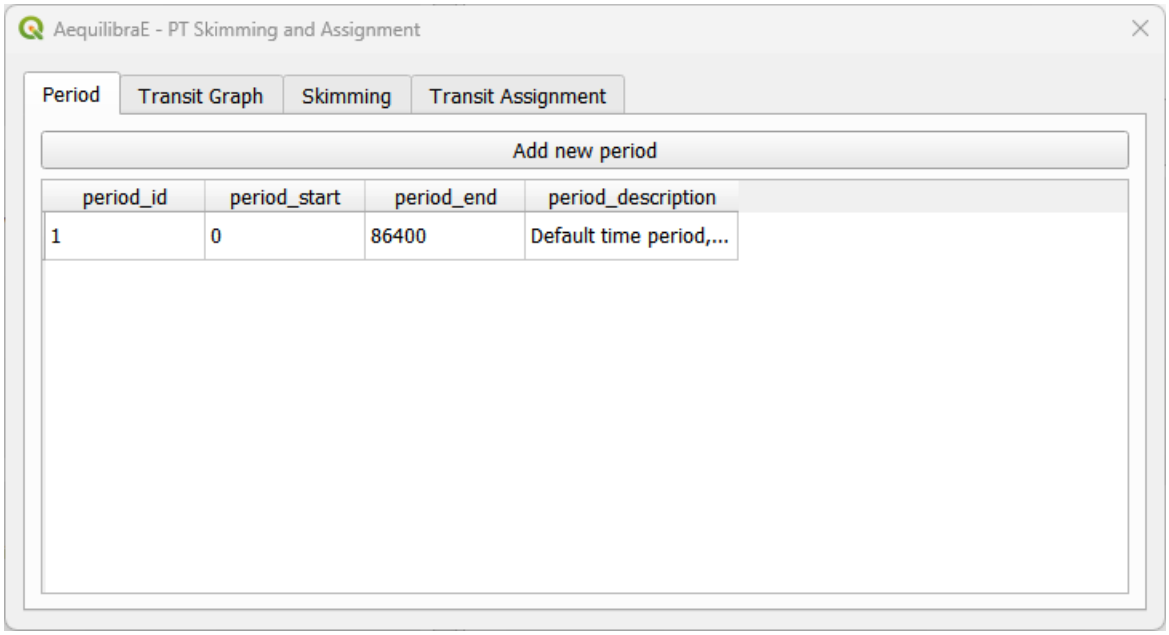
In case you want to add or rewrite information on your public transport database, you can click on **Public Transport > Import GTFS**. You will notice a difference in the clickable buttons at the bottom of the page, and it is now possible to **Overwrite routes** or **Add to Existing Routes**. For any of these options, you follow the same steps previously presented to add feed data and load it into the project.



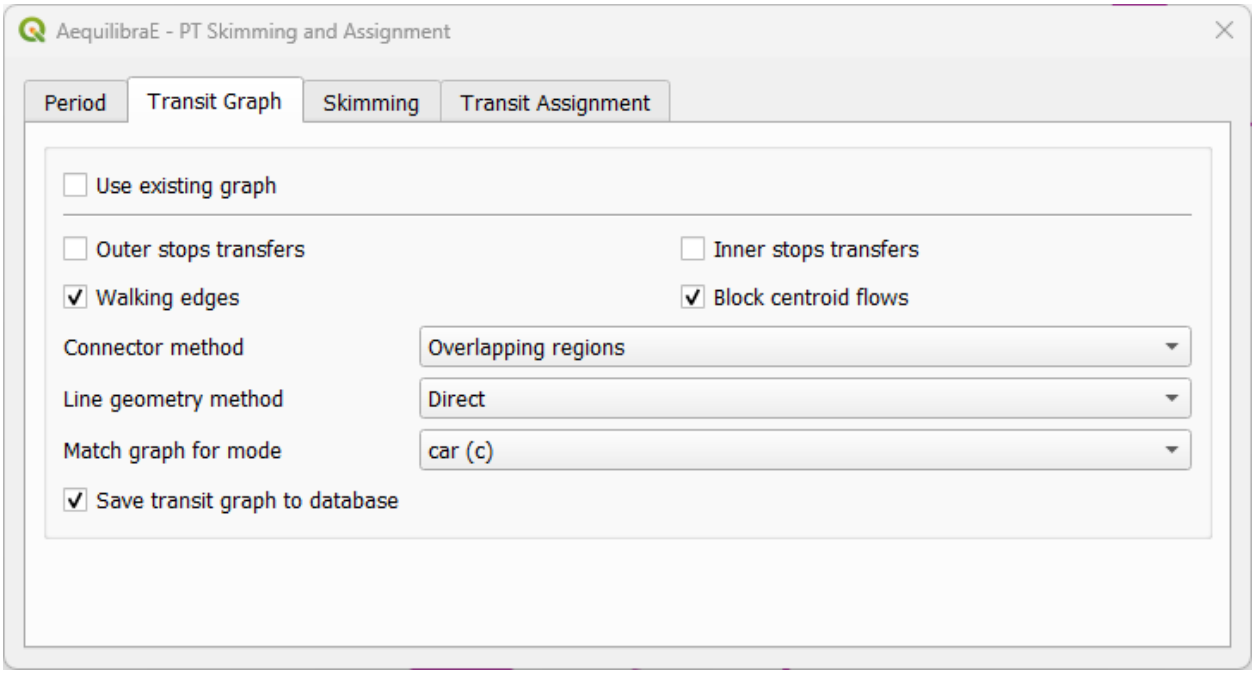
2.6.2 Transit skimming and assignment

QAequilibraE incorporates two of AequilibraE's transit features: skimming and assignment. In this section, we'll replicate AequilibraE's Python examples and show you how to add a new Period to your transit model. To open the menu, click on **Public Transport > Skimming and Assignment**.

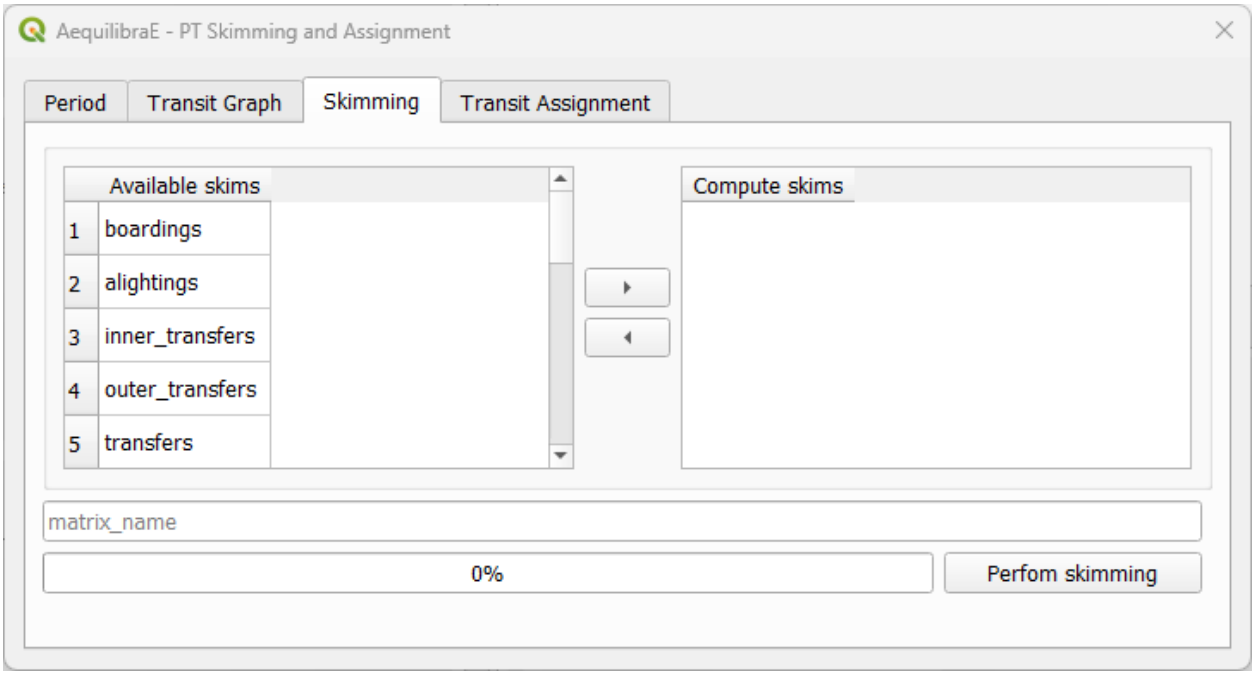
The Transit skimming and assignment module consists in four different tabs. "*Periods*" is the first tab and it displays a visualization of the periods in the project. It also has a clickable button for you to add a custom period as desired. Notice that, a period representing all day-long (`period_id == 1`) exists by default.



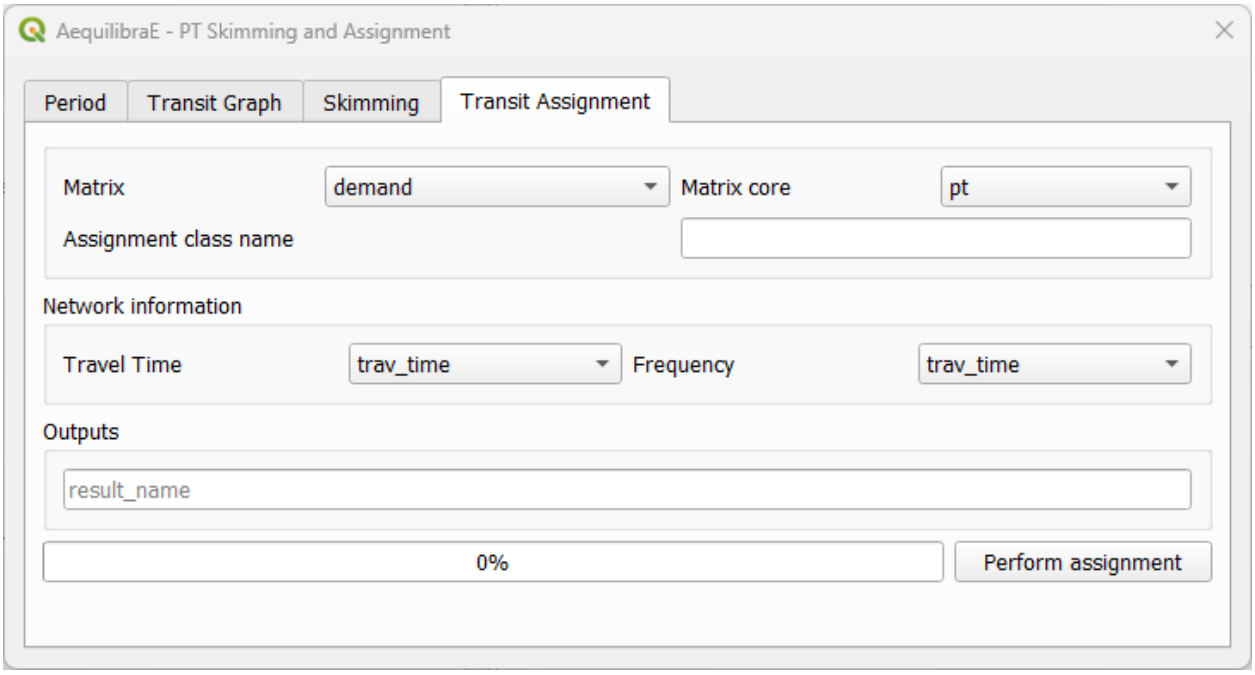
The second tab is "*Transit Graph*", in which you will add the configuration of the graph that will be created. The four checkboxes at the top of the tab indicate some characteristics of the network and you can select all that apply. The three drop-down buttons configure, respectively, the connector method (which creates the connector edges between each stops and ODs), the line geometry method (which creates a LineString for each edge), and the match graph for mode. The last checkbox indicates weather you want to save the assignment result in the database or not.



In the "*Skimming*" tab it is possible to select the fields we want to create skims for, perform the actual skimming , and save the result as an *.OMX file.



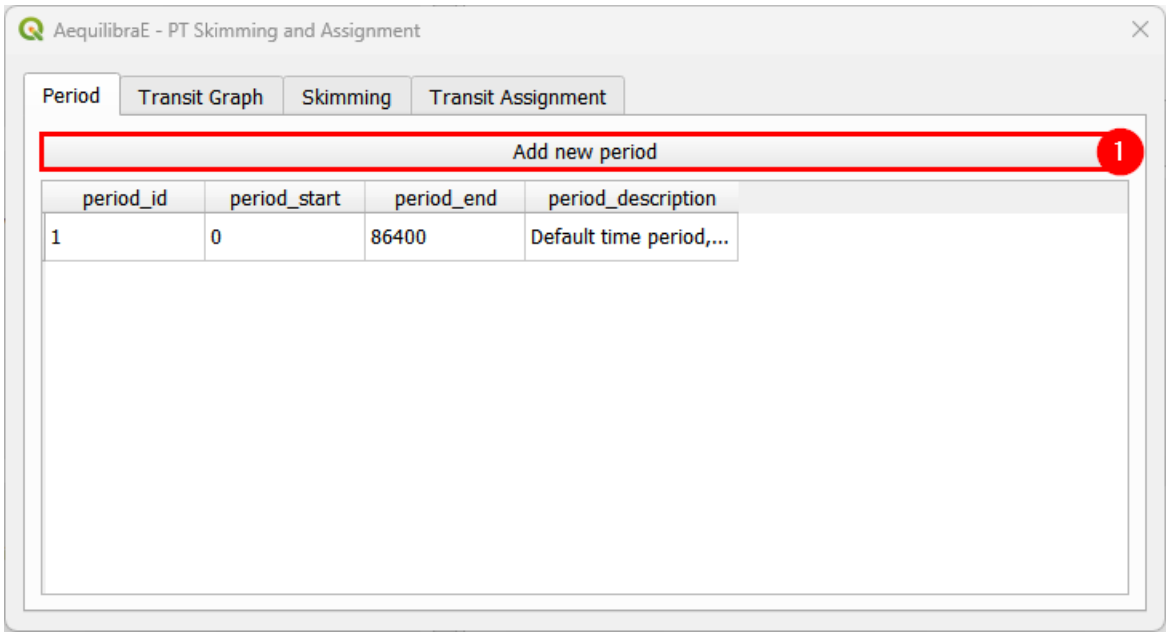
Finally, in the "*Transit Assignment*" tab, we select the demand matrix and its core that will be set for computation, the name of the assignment class, the fields corrresponding to the travel time and frequency, and the name we want to save the results table.



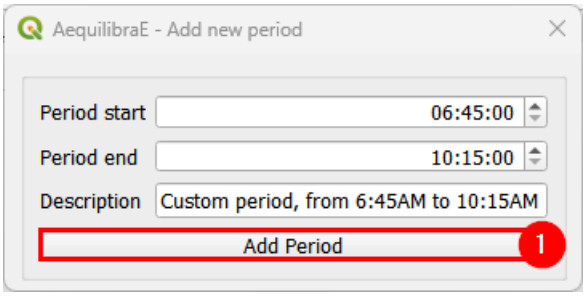
In the next sub-sections, we'll present two different workflows, one performing skimming with a custom period and the other performing assignment for the period of one day.

Skimming with custom period

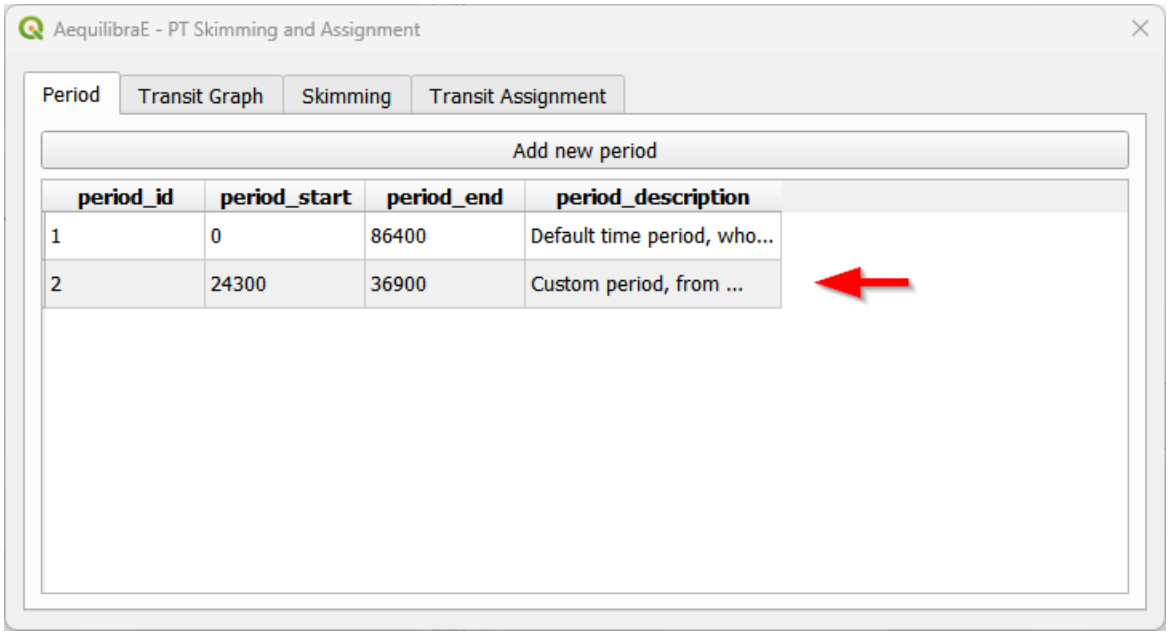
In this example we'll create a custom period and its related skimming. We start at the tab "*Periods*" clicking on the *Add new period* button.



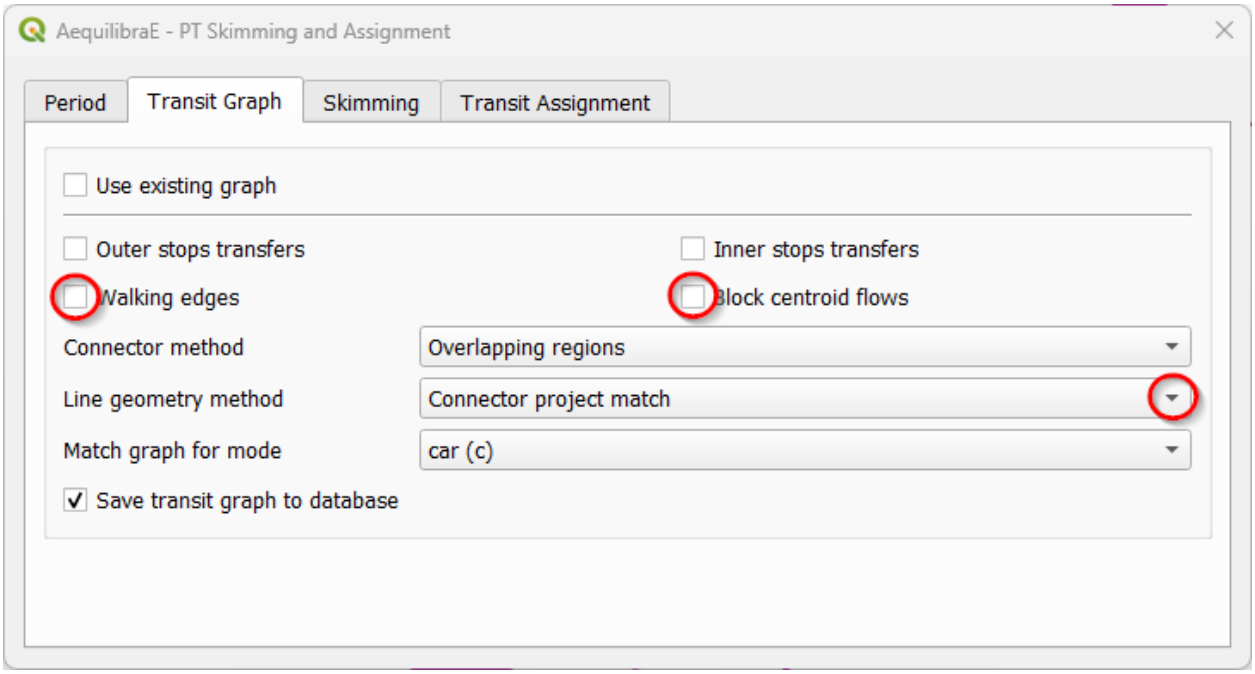
A new window containing the fields period start, end, and description will open. Add the appropriate time and description and hit the *Add period* button at the bottom.



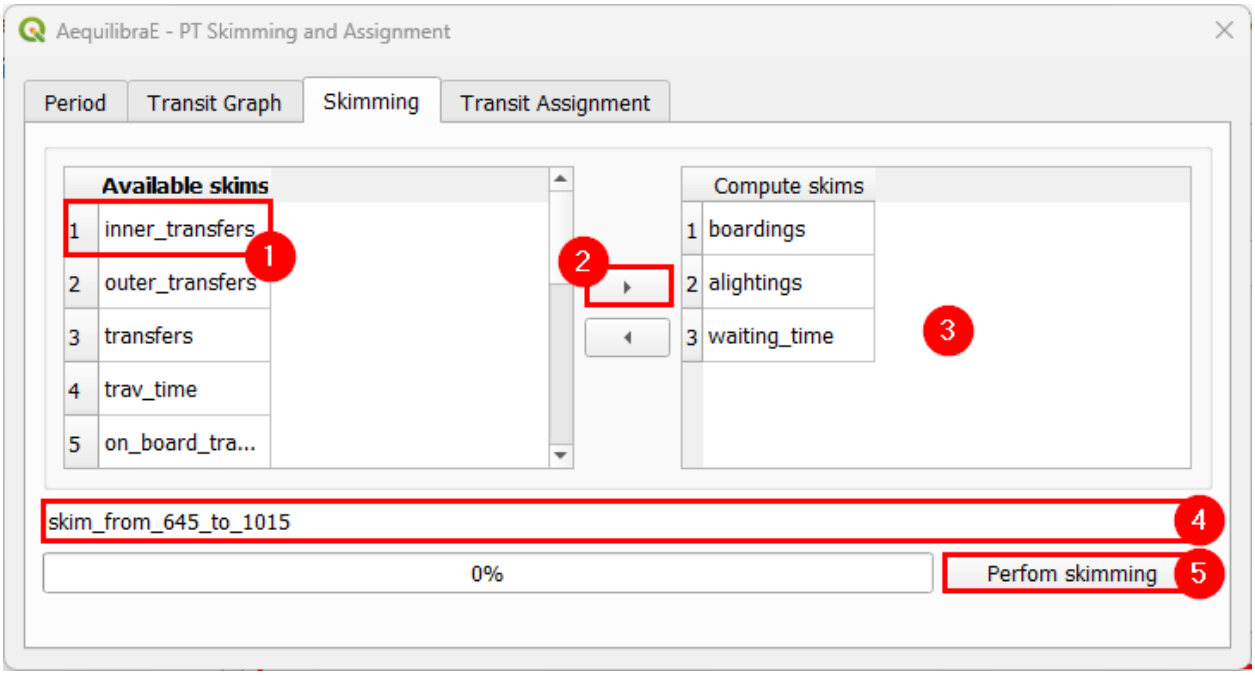
The window will close and the period will be automatically shown in the Periods table view. By default, the periods are numbered in an ascending order based on the number of the last period added. Notice that the start/end periods we added before are displayed as seconds at the table. Before continuing, select the desired period by clicking on it, otherwise an error will be thrown when skimming/assigning.



At the tab "*Transit Graph*", we'll set up the configurations of the graph. For this example, we'll uncheck the boxes for "walking edges", and "block centroid flows". For the purpose of this example, we'll let the box "save transit graph to database" checked so we can reuse the graph for assignment. Let's change the value of line geometry method to "connector project match" because graphs should be created using this method. Finally, as Coquimbo doesn't have many walking edges, we'll match the graph for cars.



Moving to the "*Skimming*" tab, we can select the skims we want to compute, as well as select a name to our matrices file. To add a skim to computation, we select the fields one by one at the "Available skims" column and add them to the "Compute skims" column by clicking on the right-arrow button (see steps 1, 2, and 3). Let's create a name for our output (step 4) and click on the *Perform skimming* button. It will perform the skimming for a unit matrix, and store the result at the project matrices' folder.

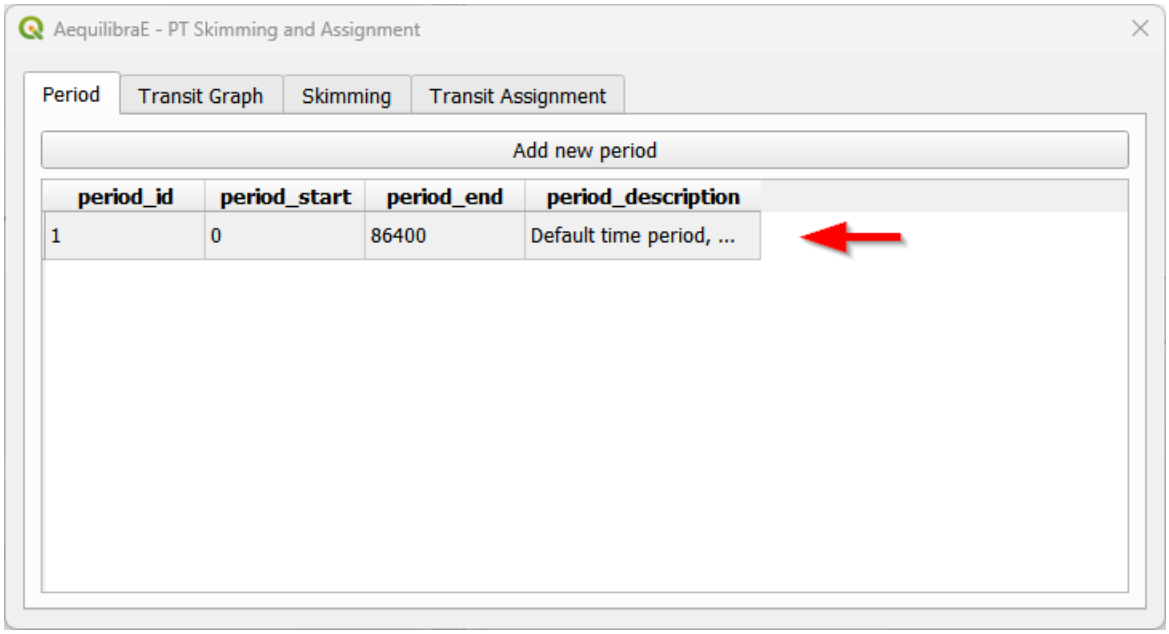


When the process is finished, the PT Skimming and Assignment window will automatically close and you can check the outputs at the matrices folder.

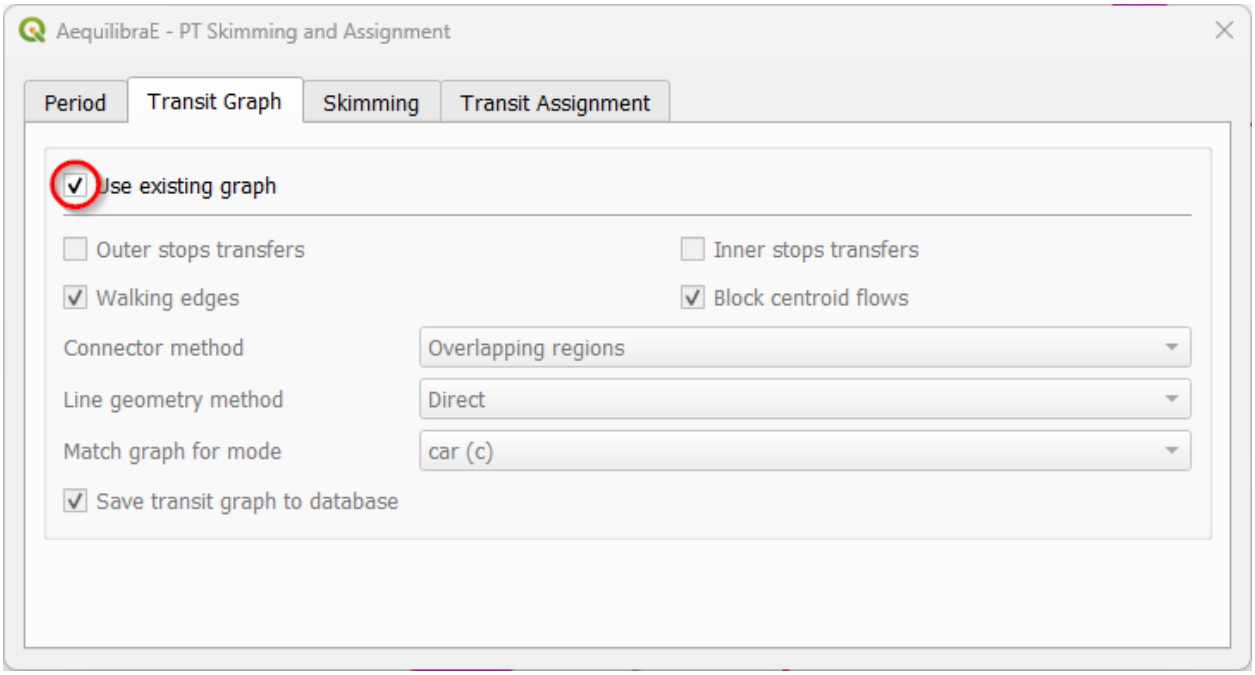
Transit assignment

In this example, we'll perform the assignment for all day-long also for Coquimbo. This is a reproduction of an AequilibraE's example.

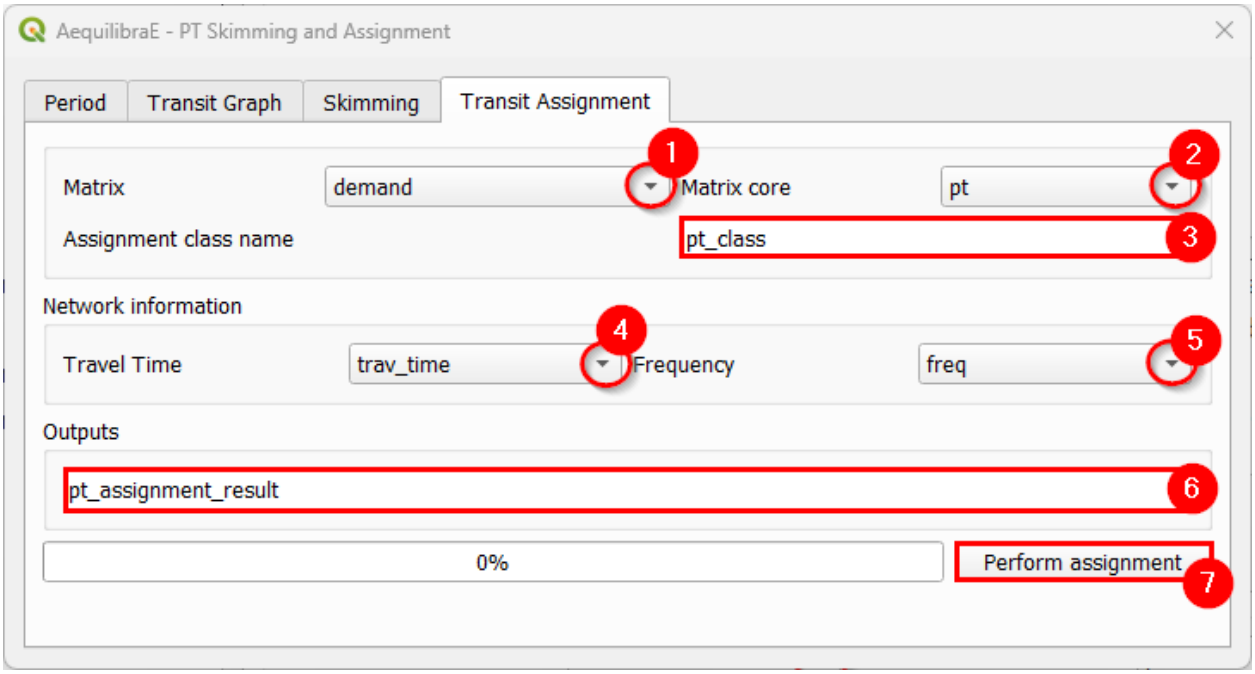
Let's start the example selecting the default period at the periods table.



Instead of setting up the graph configurations again, you can reuse the graph currently in memory. To do this, select the "Use existing graph" checkbox. The lower section of the "*Transit Graph*" tab will then be disabled. If you change your mind, simply clear the checkbox and configure the graph settings as needed.

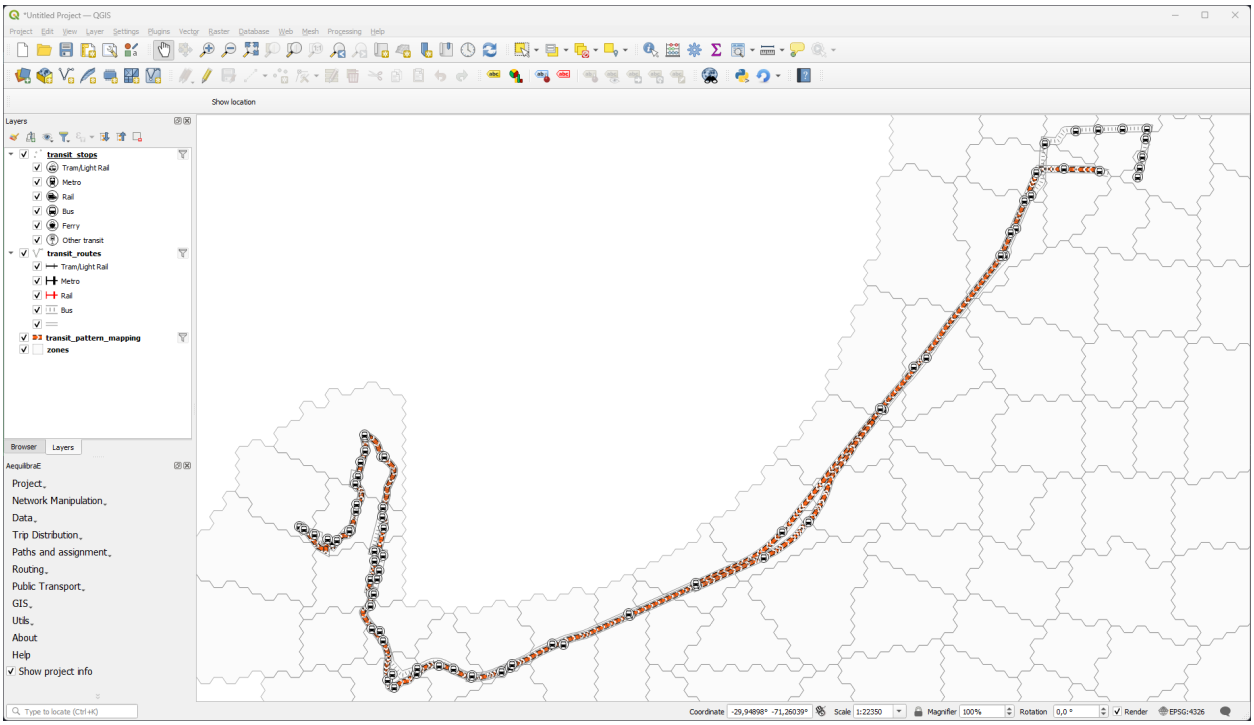


As we're running an assignment, we'll skip the Skimming tab and move directly to "*Transit Assignment*". Let's select a demand matrix and its core for computation (steps 1 and 2). As Coquimbo doesn't have any matrix in its matrices folder, you'll have to create one open layer and *import it to the project*. Then, select an appropriate name for the transit assignment class (setp 3), and the variables that corresponds to the travel time and frequency (steps 4 and 5). Lastly, select an appropriate name for the output that will be stored in the results database (step 6) and just hit the *Perform Assignment* button at the bottom.

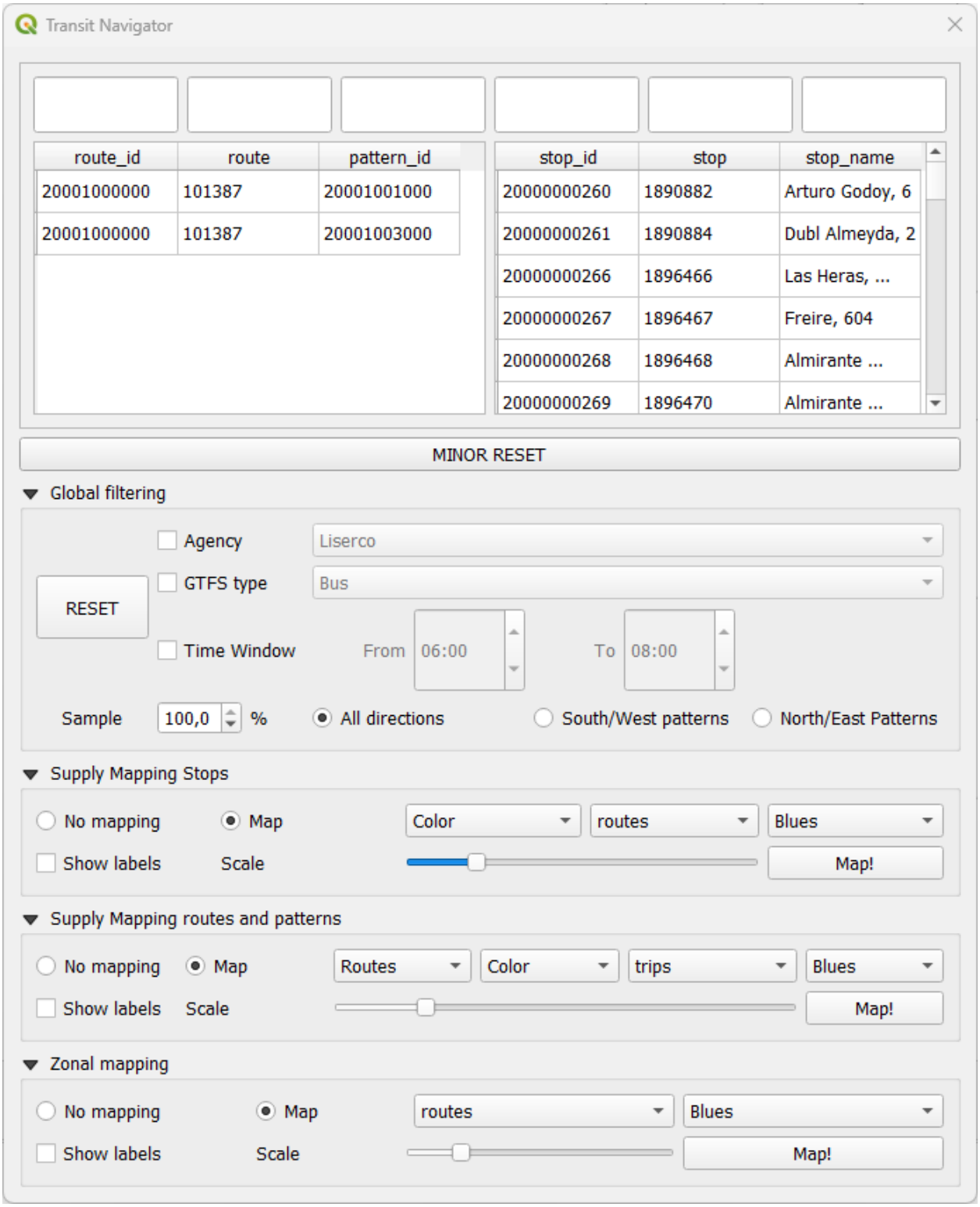


2.6.3 Explore transit network

Case you have already imported a GTFS feed into your project or you want to open a feed from an AequilibraE project created with Python, you can click on **Public Transport > Explore Transit** to visualize the Transit routes. While opening the Transit Navigator, you will notice that the layers *patterns*, *routes*, *stops* and *zones* from the GTFS file are going to be displayed in your map canva, and appear in the layers list.



The navigator window has five different tabs you can explore.

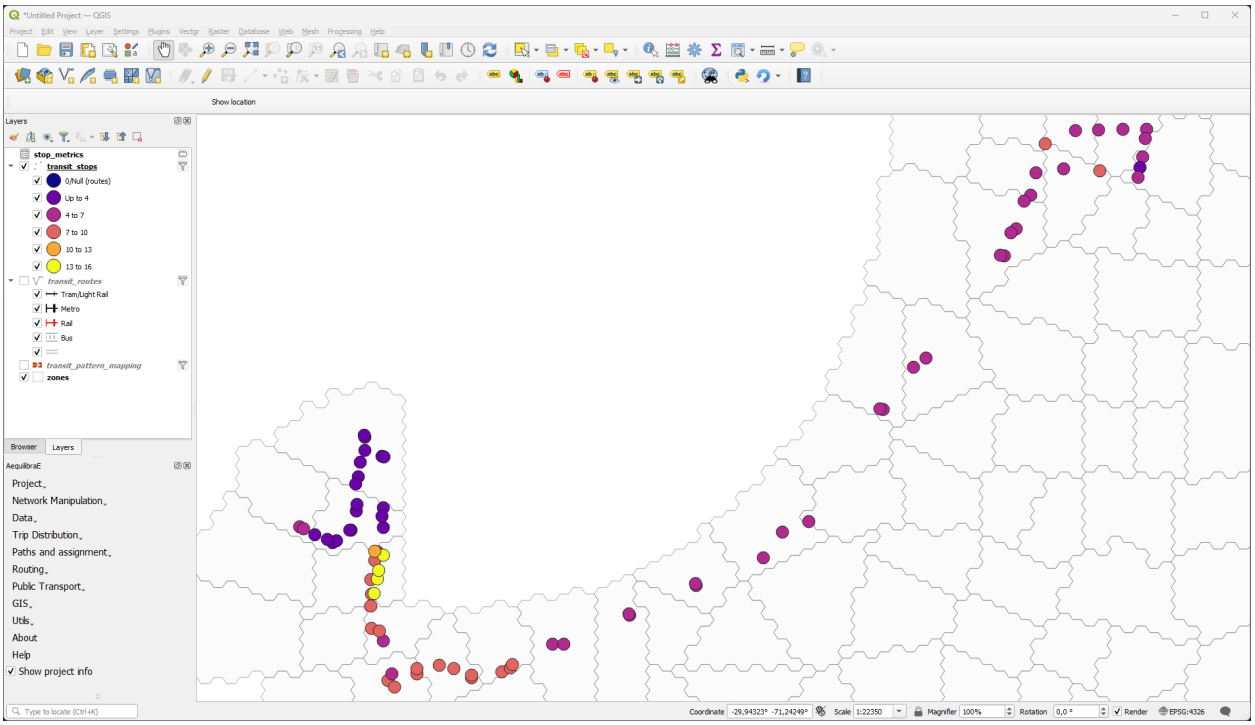


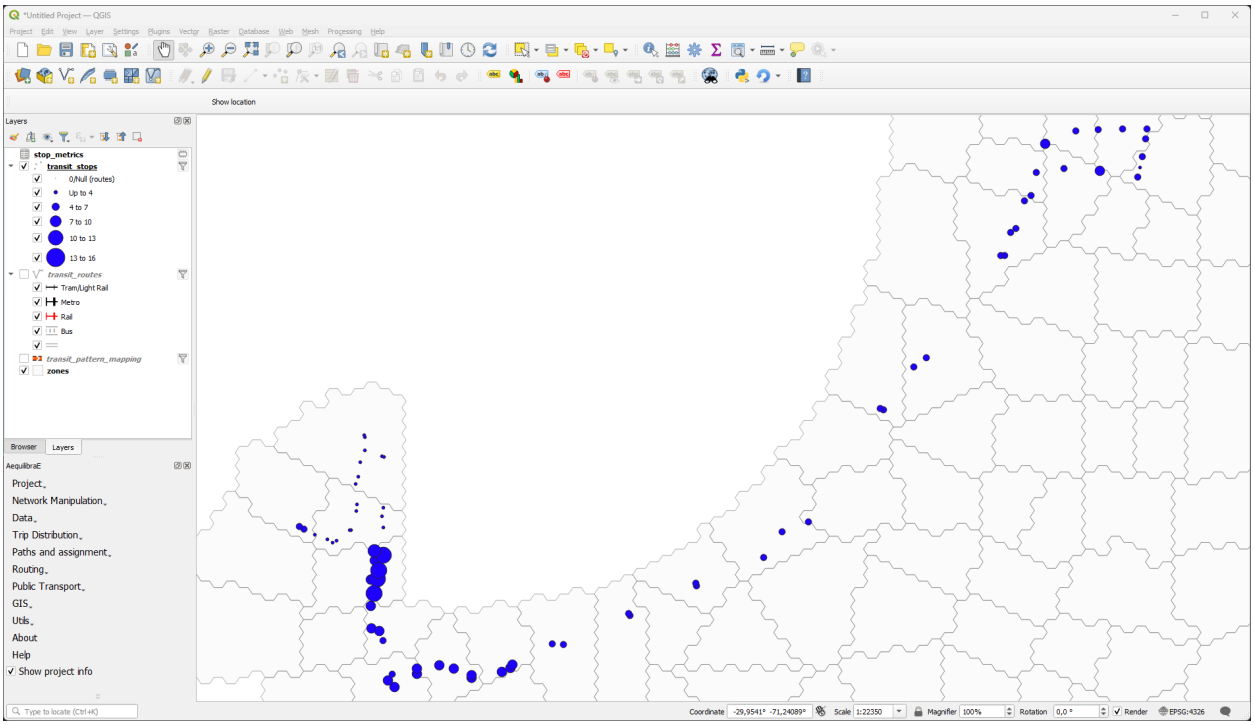
In the top, there are the three boxes one can select and filter routes, patterns, and stops. You will notice that whenever an element is selected or filtered, this selection is automatically displayed in the map canva. After filtering data, if you want to restore the original layers, you can click on **Minor reset**, and your layers are restored.

In the *Global filtering* tab, it is possible to filter your GTFS by *Agency*, *GTFS type*, *Time window*, and *directions*. It is also possible to select a sample from the GTFS data to analyze. The fitering performed in this tab is automatically

displayed in the map canva. To restore the original layers, just click on **Reset**.

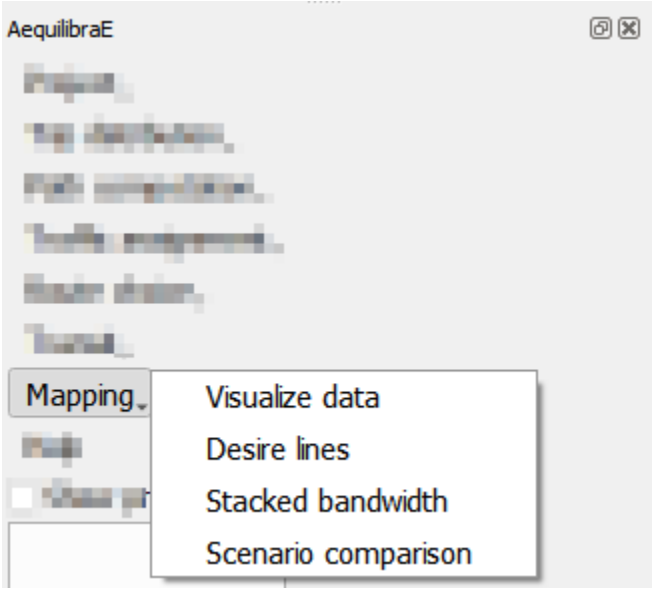
In the last three tabs, one can display useful information about stops, routes, patterns, and zones. For instance, you can find out *how many routes stop at a specific stop location*, *what is the total capacity of a specific route*, or *which zones have more stops or routes across them*. Within these tabs, it is possible to configure how one wants to display the information, by selecting the object color, or thickness (size). It is also possible to display labels, by selecting the **Show labels** option. The figures below show the number of routes across the stops displaying the information with different symbol colors and sizes. Notice that in the layers list, the variable scale for number of routes is shown, as well as a data layer named *stops_metrics*, which contains the available metrics for the existing stops.





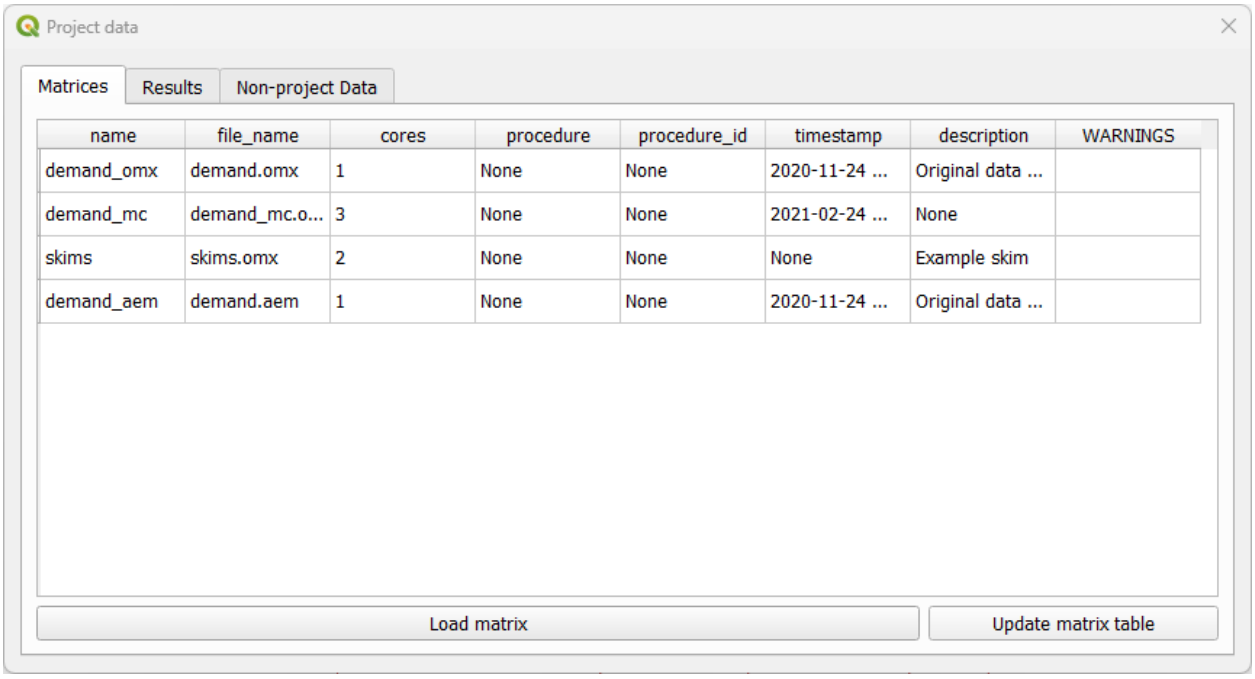
2.7 Mapping Tools

QAequilibraE has some tools to allow the user to visualize the data.

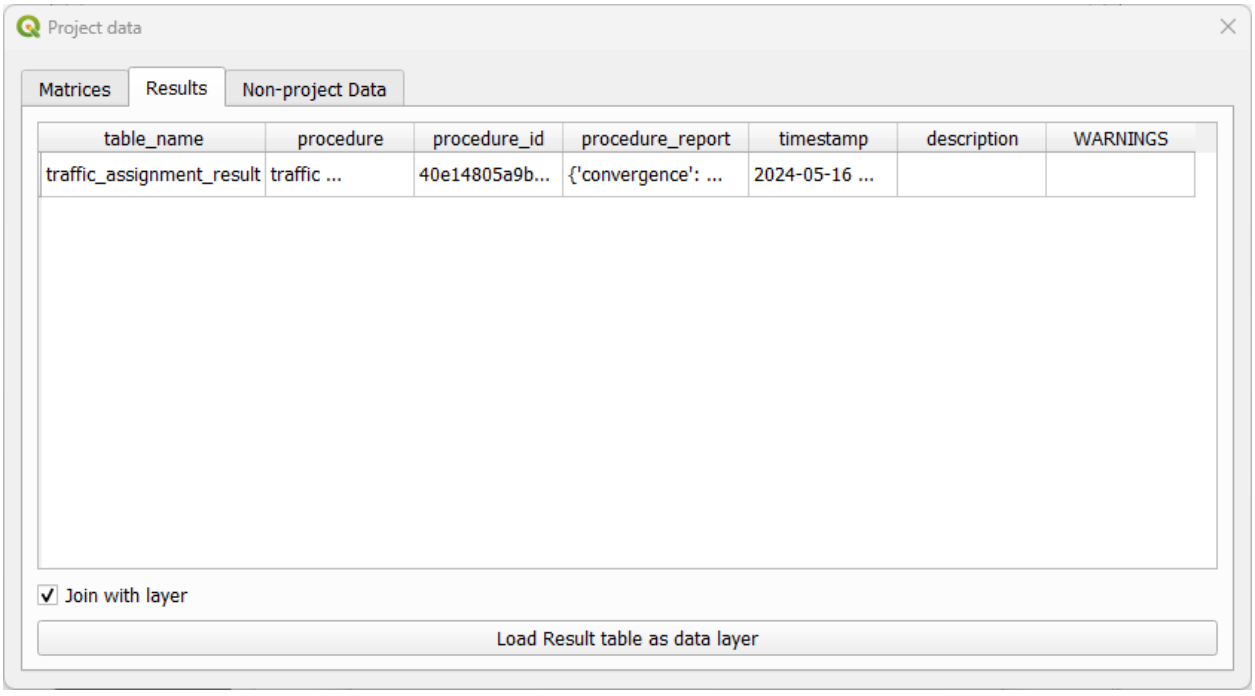


2.7.1 Visualize data

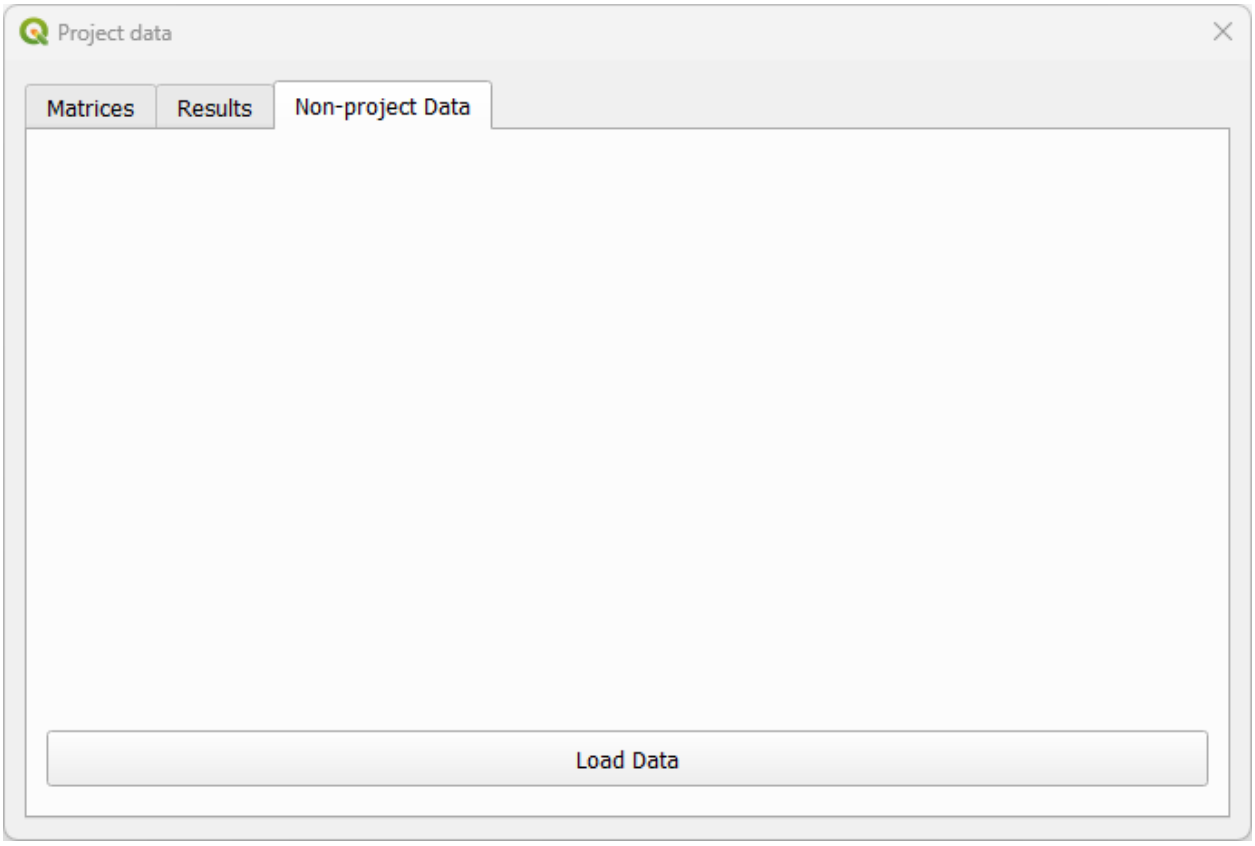
When clicking **Mapping > Visualize data**, a new window with three different tabs opens. The tab *Matrices* present all matrices available for the current project.



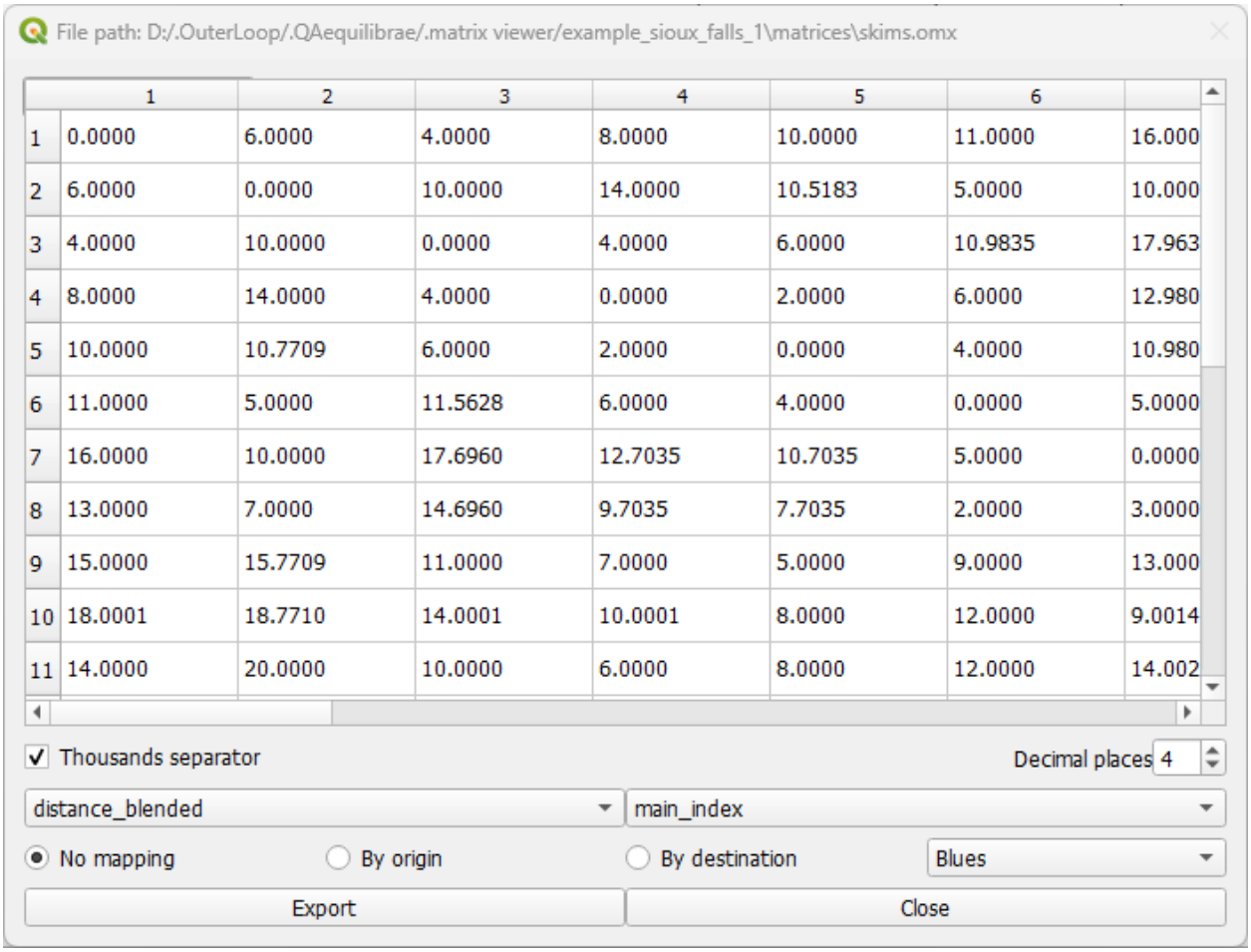
As for the tab *Results* it displays the results of procedures that took place and that are saved in the project 'results_database.sqlite' file.



The tab *Non-project data* allows you to open and visualize matrices stored in *.omx format. **This is the only tab available if no AequilibraE project is open**. Suppose you want to check a skim matrix from a previous project. When clicking the **Load data** button, you can point AequilibraE the location of the file and its visualization is displayed.

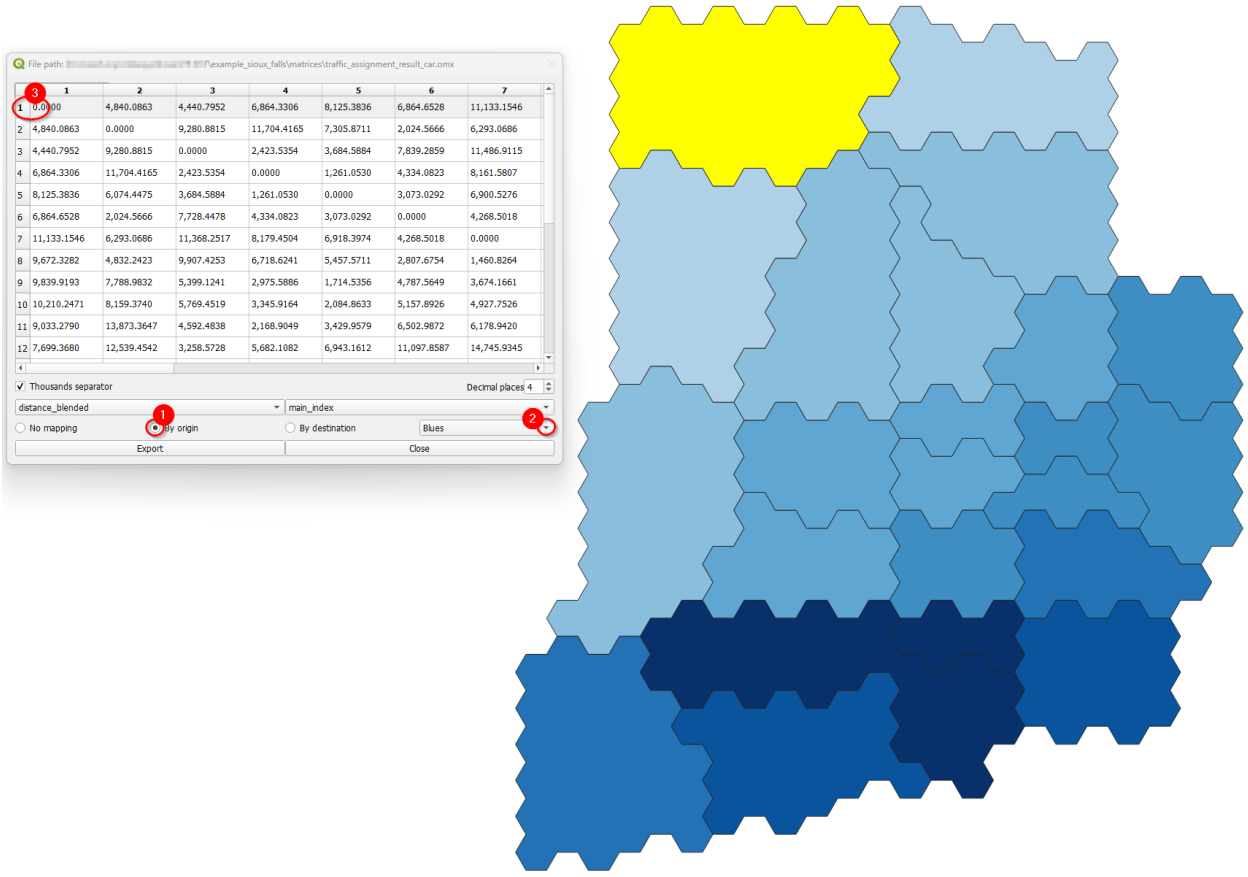


Check the figure below to see how the visualization window looks like! General configurations for data displaying such as the number of decimal places and the usage of thousand separator are available. In case your file has more than one view, you can select the desired view using the dropdown buttons at the bottom of the page. In our figure, they are represented by the dropdowns containing *distance_blended* and *main_index*. To save your current matrix into *.csv format, just click in the *export* button in the lower left corner of the window.

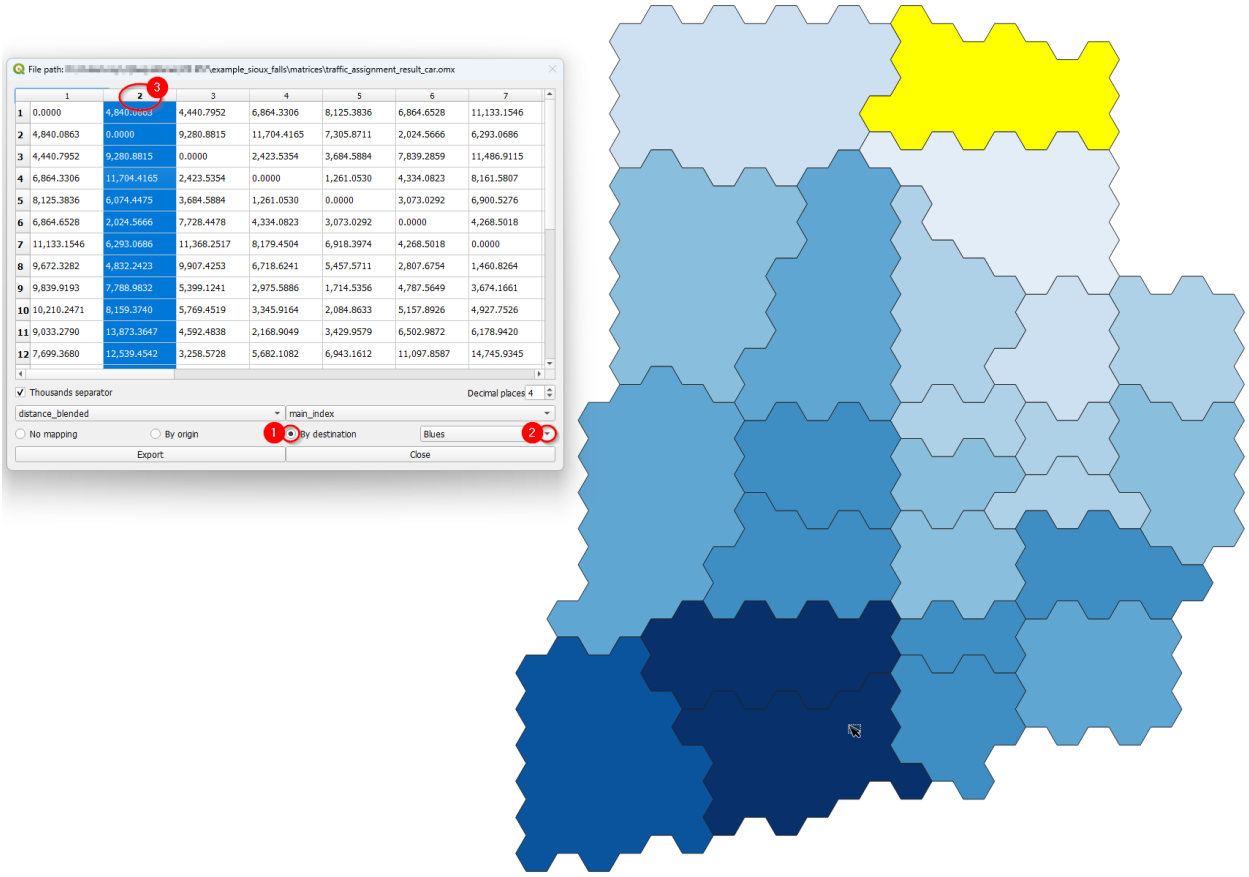


	1	2	3	4	5	6	
1	0.0000	6.0000	4.0000	8.0000	10.0000	11.0000	16.000
2	6.0000	0.0000	10.0000	14.0000	10.5183	5.0000	10.000
3	4.0000	10.0000	0.0000	4.0000	6.0000	10.9835	17.963
4	8.0000	14.0000	4.0000	0.0000	2.0000	6.0000	12.980
5	10.0000	10.7709	6.0000	2.0000	0.0000	4.0000	10.980
6	11.0000	5.0000	11.5628	6.0000	4.0000	0.0000	5.0000
7	16.0000	10.0000	17.6960	12.7035	10.7035	5.0000	0.0000
8	13.0000	7.0000	14.6960	9.7035	7.7035	2.0000	3.0000
9	15.0000	15.7709	11.0000	7.0000	5.0000	9.0000	13.000
10	18.0001	18.7710	14.0001	10.0001	8.0000	12.0000	9.0014
11	14.0000	20.0000	10.0000	6.0000	8.0000	12.0000	14.002

Additionally, we can visualize how the matrices look like in the map! Using the buttons *By origin* and *By destination*, it is possible to select the traffic zone by its origin or destination. If one select *By origin*, then click on the desired row, and notice that is going to be highlighted. The *zones* layer (if it exists) is going to be loaded and the corresponding zones are going to receive a different color shade, according to the color palette selected in the dropdown menu. One other possibility to select the zone for displaying is directly into the map canvas: with the *Select features* button enabled, just click on the desired zone in the layer and you'll notice that the color shades change accordingly, as well as the row selection in the matrix.

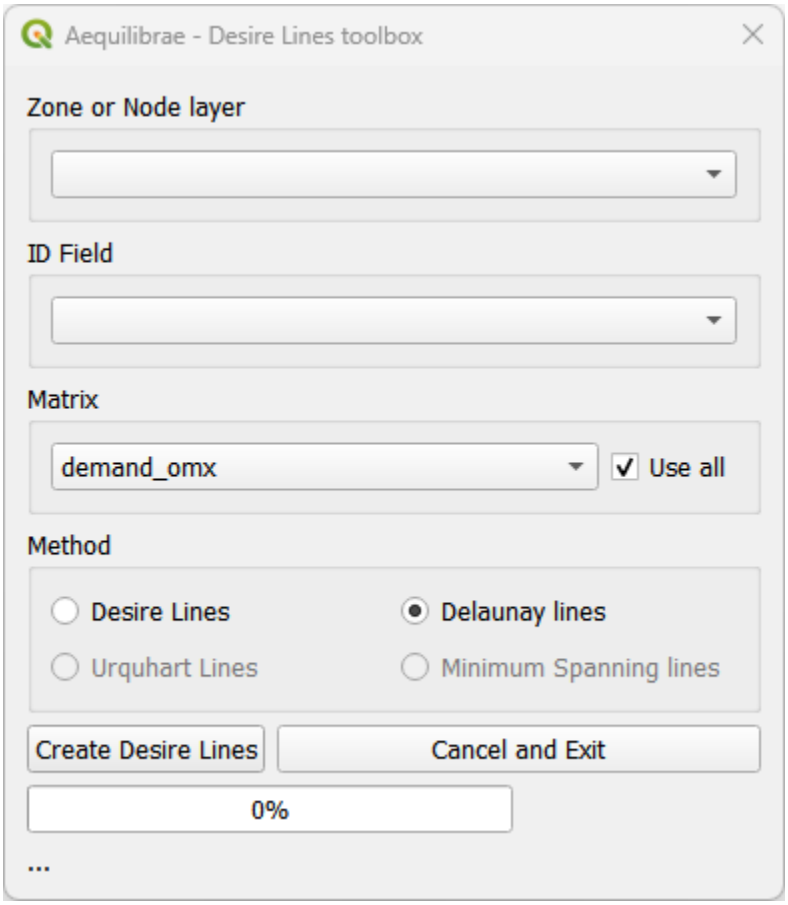


The step-by-step when selecting *By destination*, is identical to the one before. Select the desired column (destination), notice that it will be highlighted, and the *zones* layer is going to present a color shade according to the color palette selected. The selection of zones for displaying is also available for destinations, and the steps are the same as presented above.



2.7.2 Desire Lines

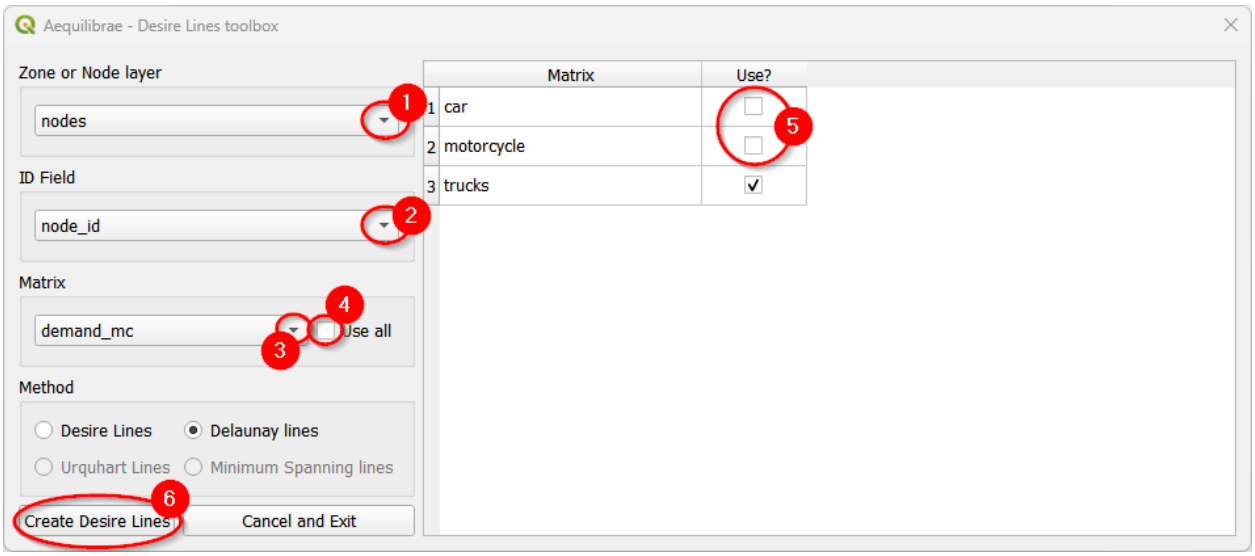
QAequilibraE is capable of doing two types of desire lines from a zone or a node layer: 'regular' desire lines or Delaunay lines for the demand matrix provided.



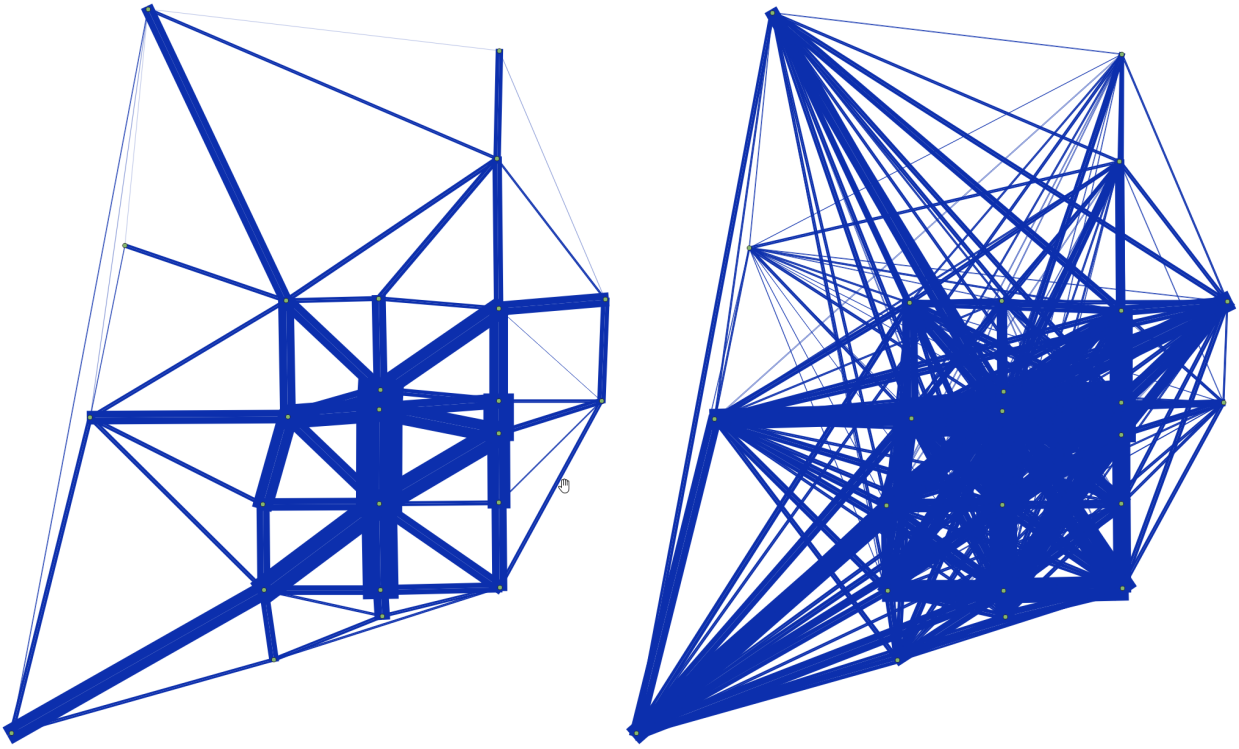
Basic workflow

Let's use the Sioux Falls example and one of its default matrices. Make sure one of 'nodes' or 'zones' layer is active in the layers list.

We start selecting the input layer (1) and the field we'll use to create our desire lines (2). We'll use 'node_id'. Then select a matrix (3). We'll use 'demand_mc', which has more than one core, but we won't use all cores. Un-check the *"Use all"* box, and notice that a matrix core table will open at the right-hand side of the window. Un-check the cores you want to remove from computation (5) and click on *"Create Desire Lines"*. If you want the usual desire lines, change the selected field in the *Method* box.



Make sure to select a zone/node layer and node ID that is compatible with your matrix.

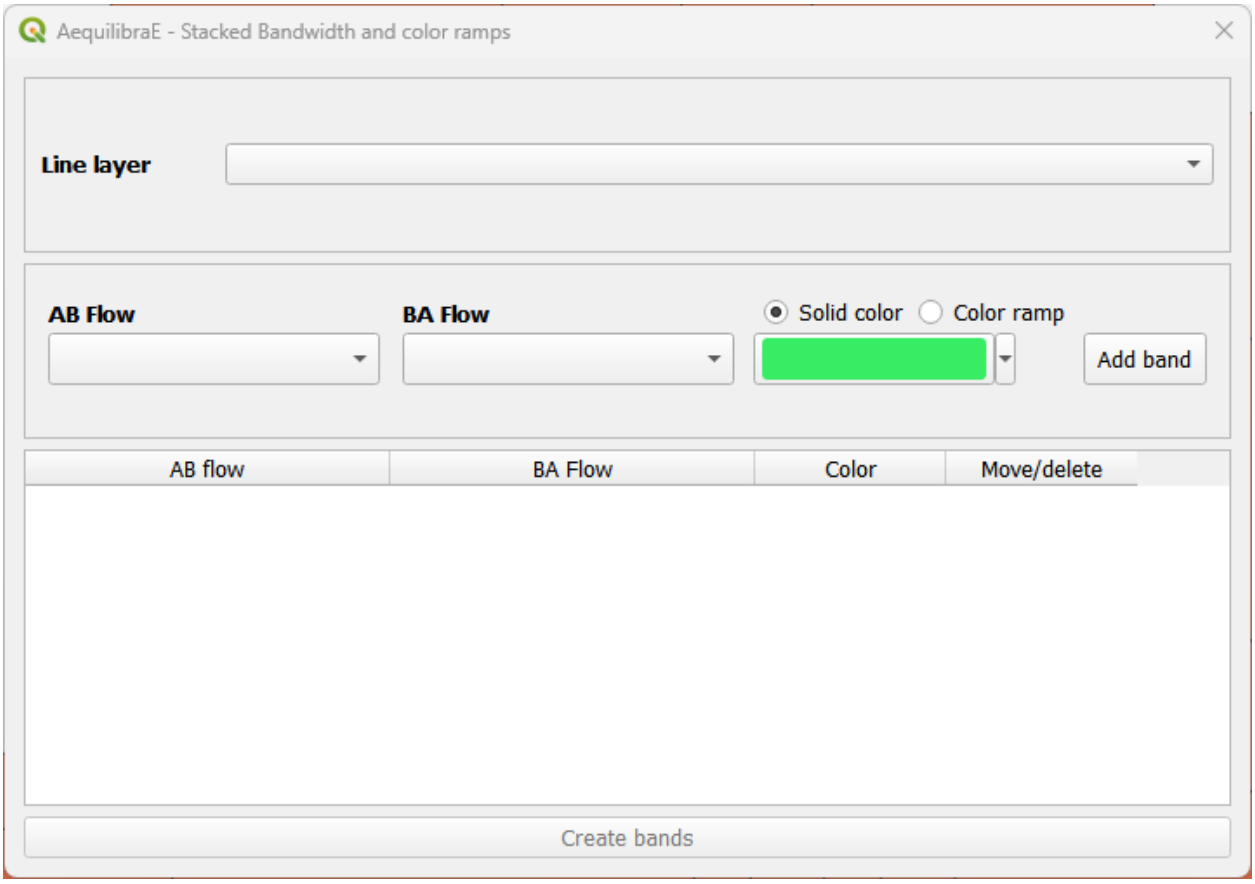


(a) Delaunay lines

(b) Desire lines

2.7.3 Stacked Bandwidth

This is a tool for plotting link flows. It uses a link layer, including Delaunay lines or desire lines. It is also possible to choose between solid or gradient colors.

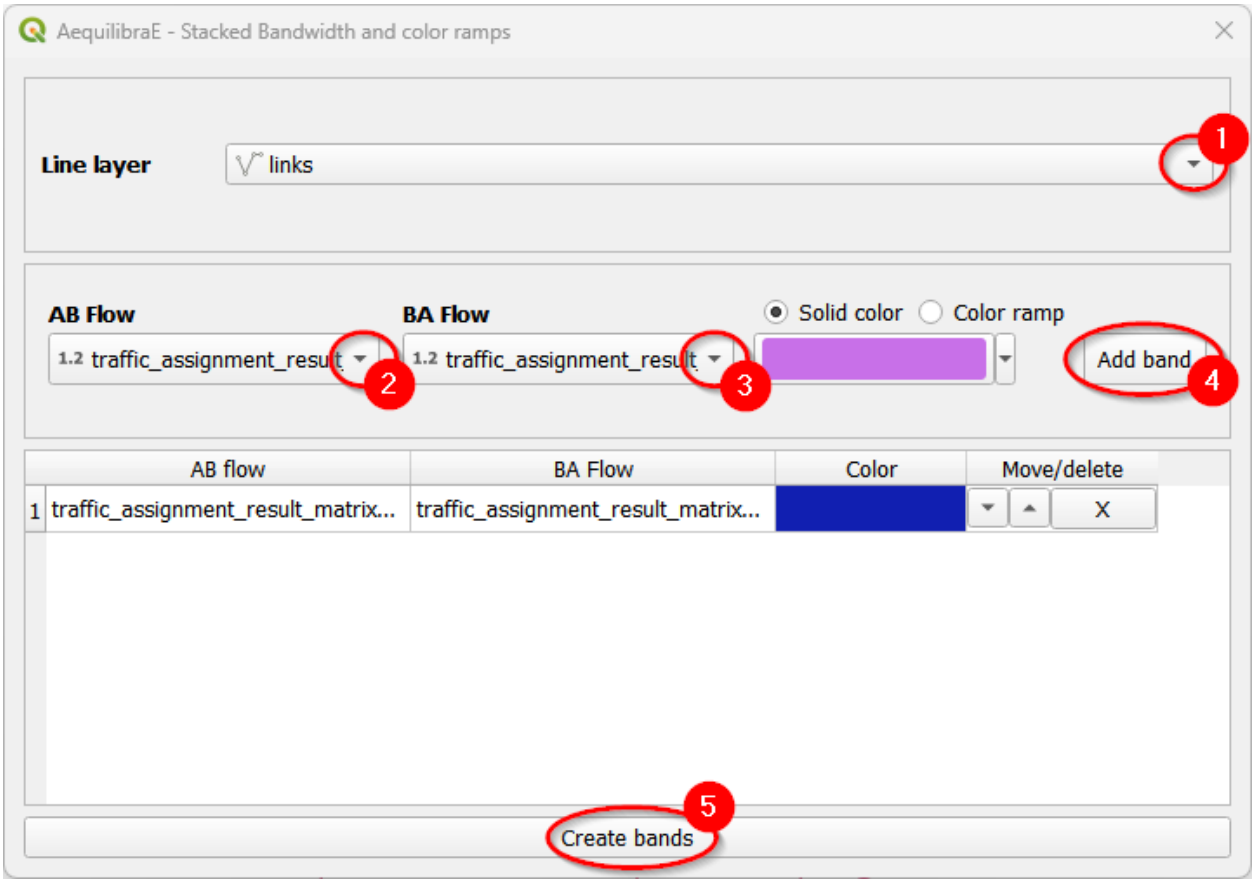


Basic workflow

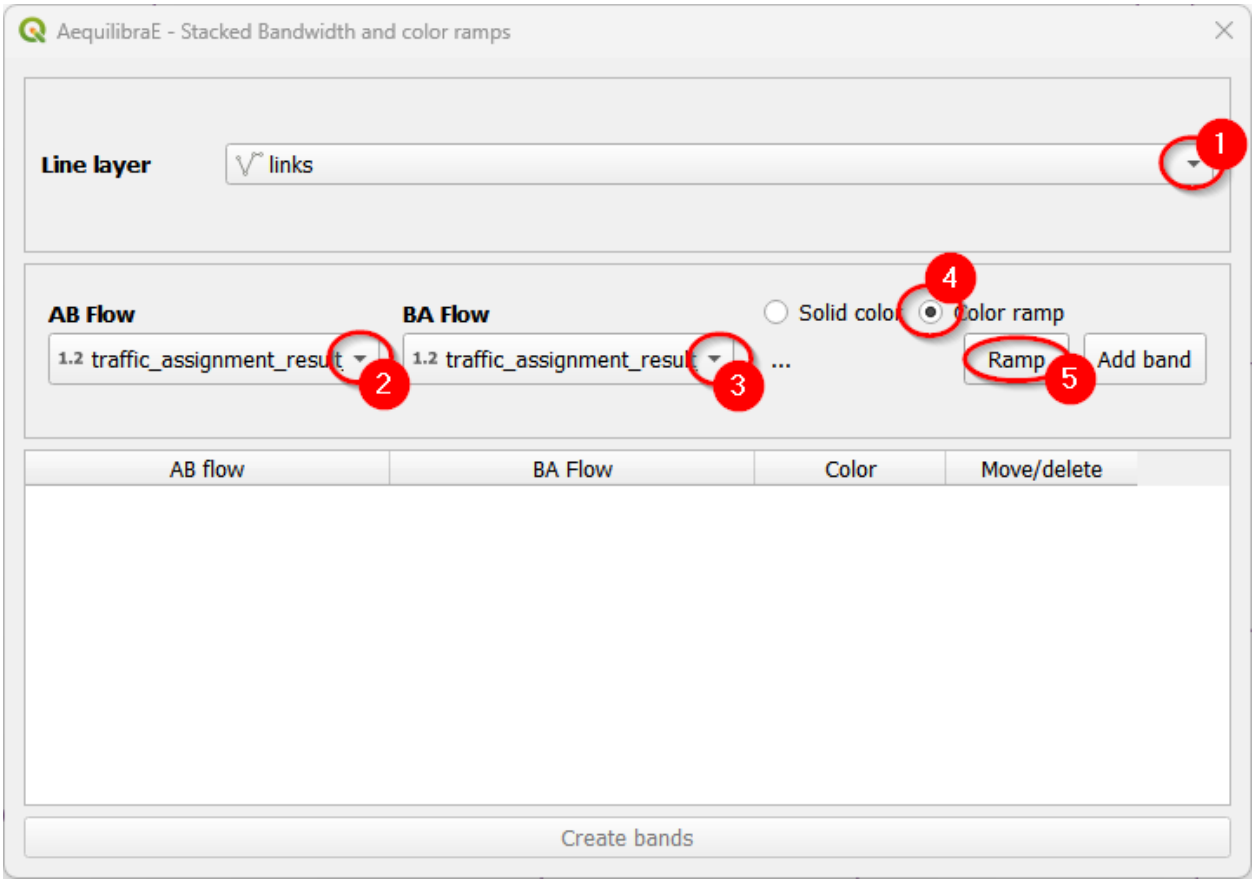
We'll use the traffic assignment result for Sioux Falls in this example. Don't worry if you haven't done the assignment: you can use any other line layers and flows you want!

Before set up the bandwidth configuration, make sure you have the 'links' and 'traffic_assignment_result' layers active in the layers list. If you open the links' layer attribute table, you'll see that the fields of 'traffic_assignment_result' are joined.

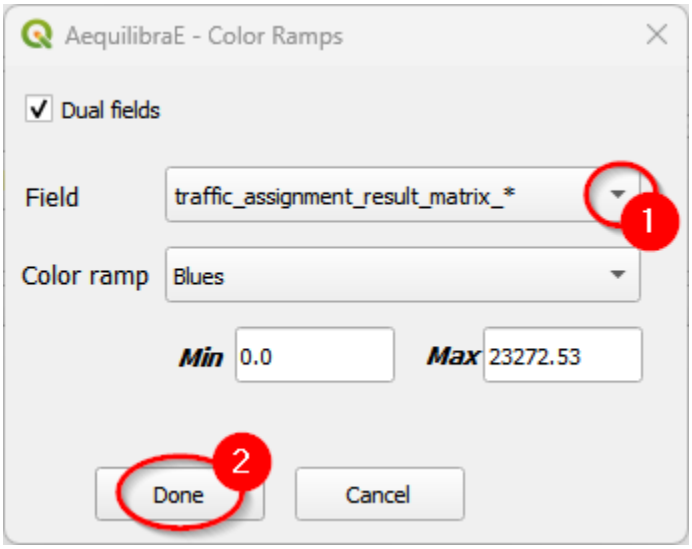
Let's proceed with a solid band first. First, we select the line layer (1) and the AB/BA flow variables (2 and 3). Regarding the color, you can use a random color selected by QAequilibraE or choose the one you want. Just click on the dropdown button at the right-hand side of the color box. To add the band, we click on the *"Add band"* button (4). You'll notice that the band configuration is now available at the table and the *"Create bands"* button is enabled. Click on it (5) and the links layer in the canvas will automatically be updated.



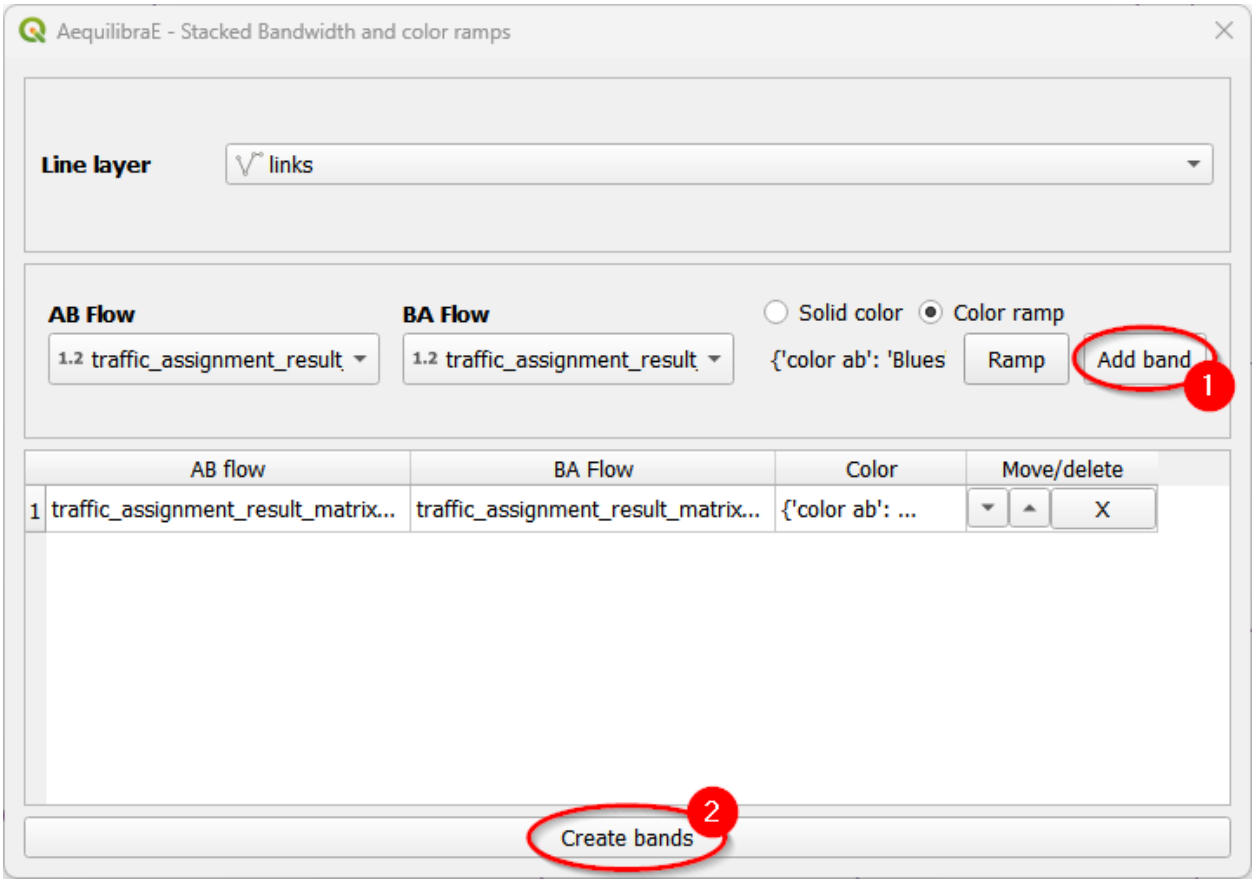
It is also possible to use color ramps instead of solid colors for plotting your data. Select the line layer (1) and the flow variables (2 and 3). Let's select the *"Color ramp"* option (4) and configure the ramp (5).



When the color ramp window opens, configure once again the fields (1) and the color ramp you want to use. We'll use the default 'Blues' in this example. Click *"Done"* (2) when you finished.



Finally, add the configured band to the project table (1) and click on the *"Create bands"* button (2).



You can also control the overall look of these bands (thickness and separation between AB and BA flows) in the project properties. Go to the properties box in the Project menu and click on the *Variables* tab. We'll edit the `aeq_band_width` variable.

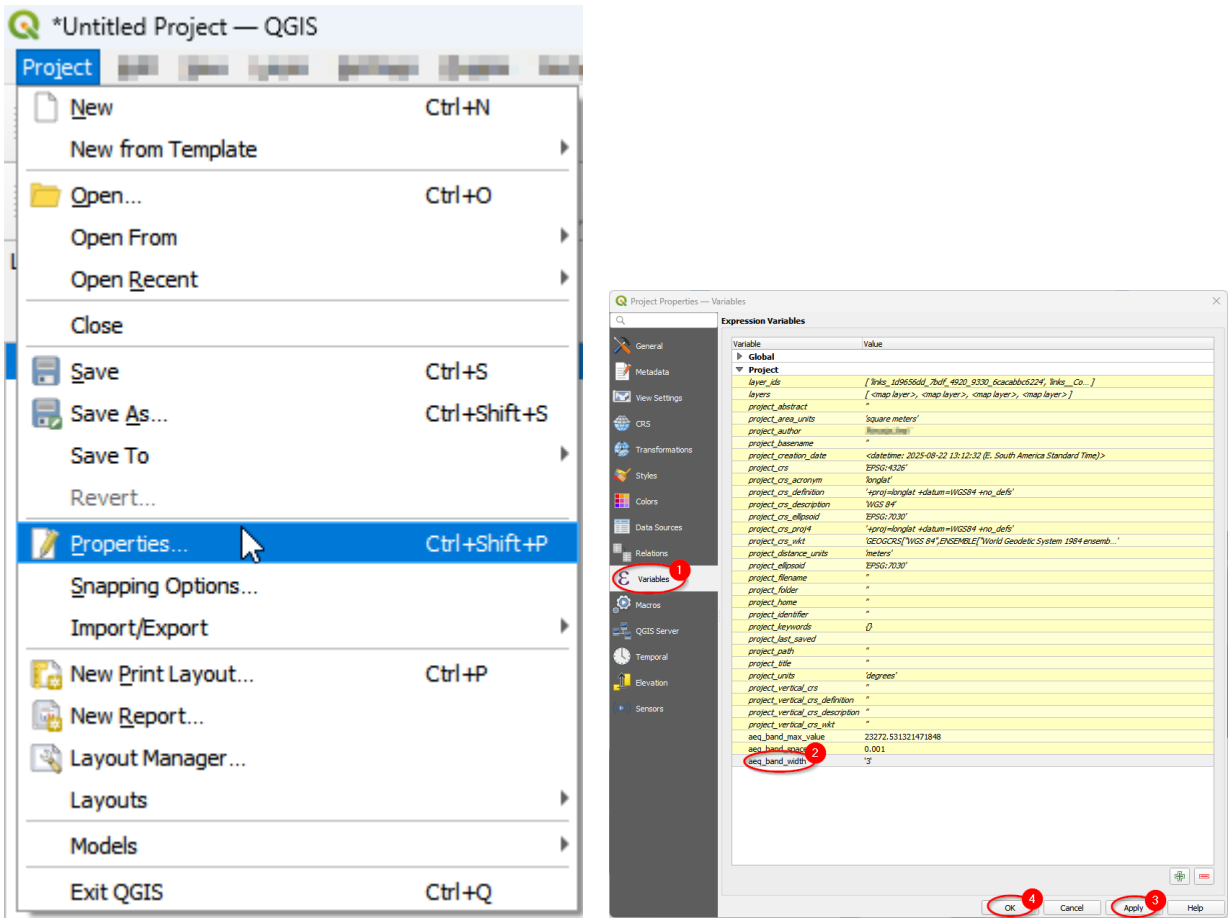
And we're all set! You might need to refresh or pan the map for it to redraw after changing the project variables.

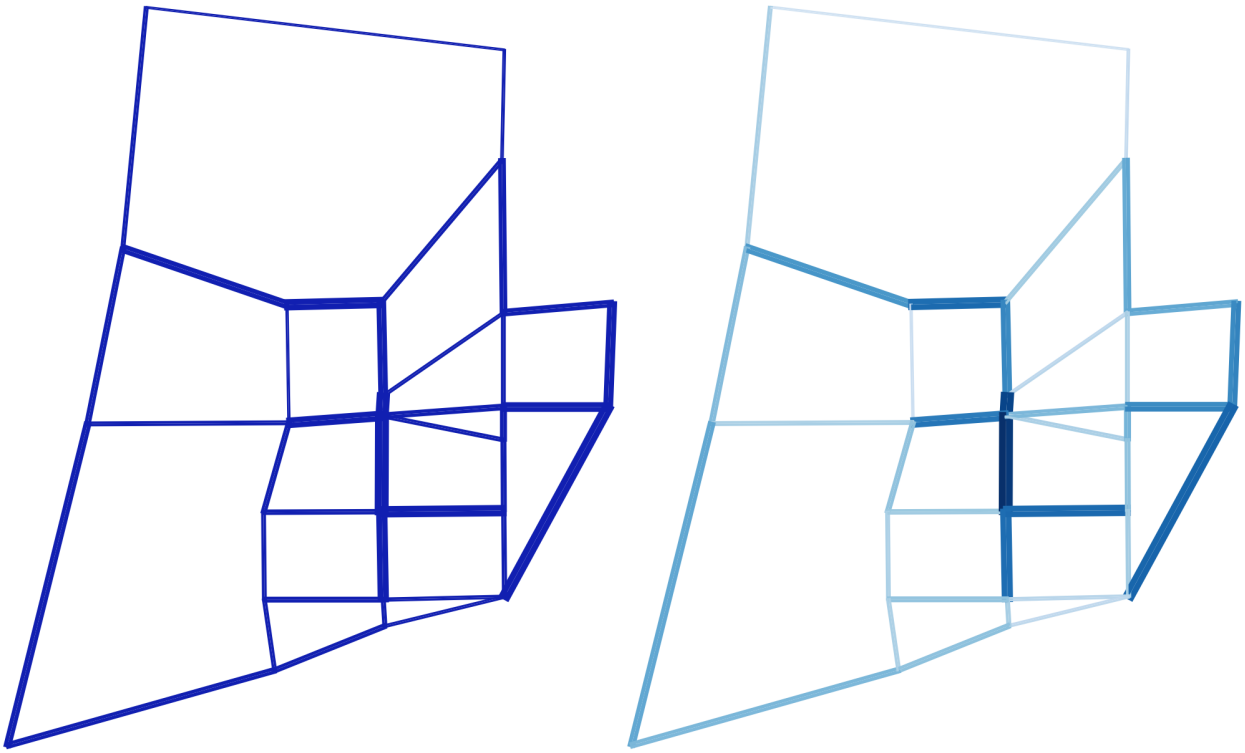
2.7.4 Scenario Comparison

To compare scenarios, we need to have two different assignnment results (the original one and another from forecast). If you don't know how to run a forecast, take a look at the *trip distribution workflow*.

The scenario configuration requires the user to set AB/BA flows for the two sets of link flows being compared, as well as the space between AB/BA flows, and band width.

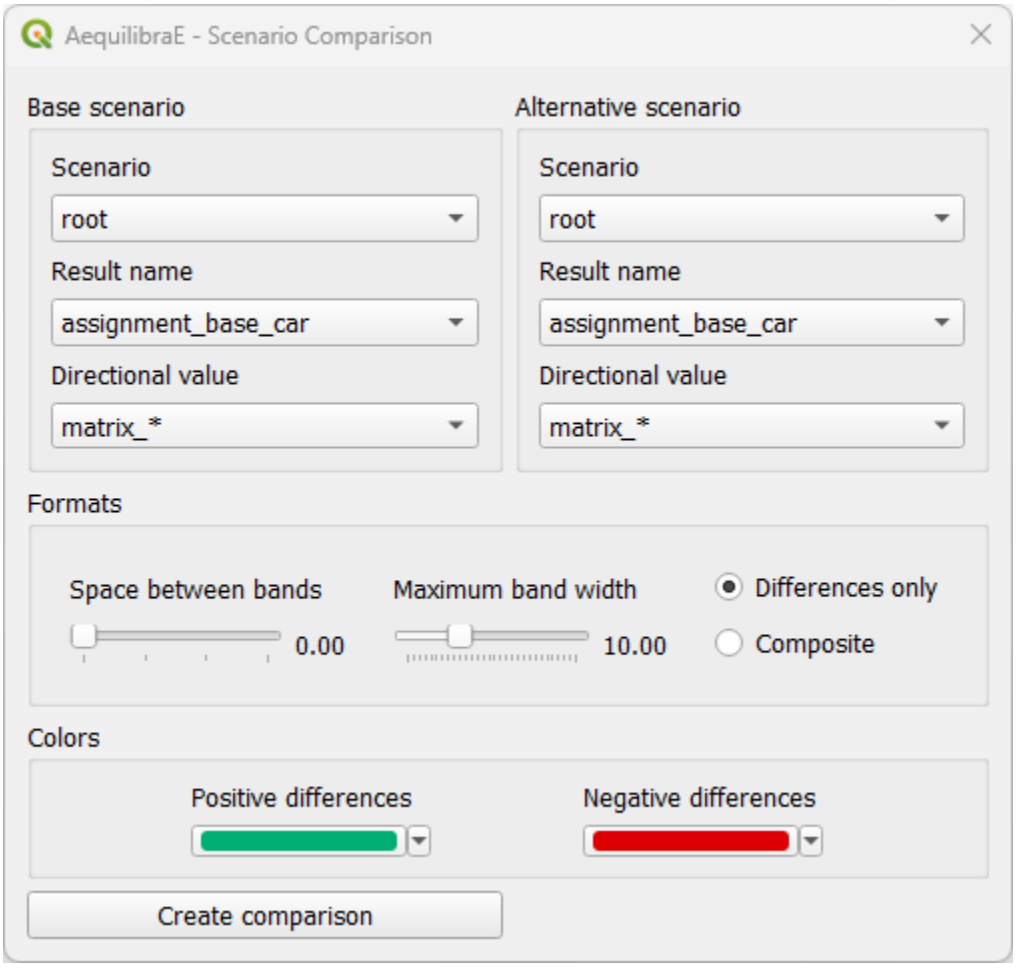
The user can also select to show a composite flow comparison, where common flows are also shown on top of the positive and negative differences, which gives a proper sense of how significative the differences are when compared to the base flows.





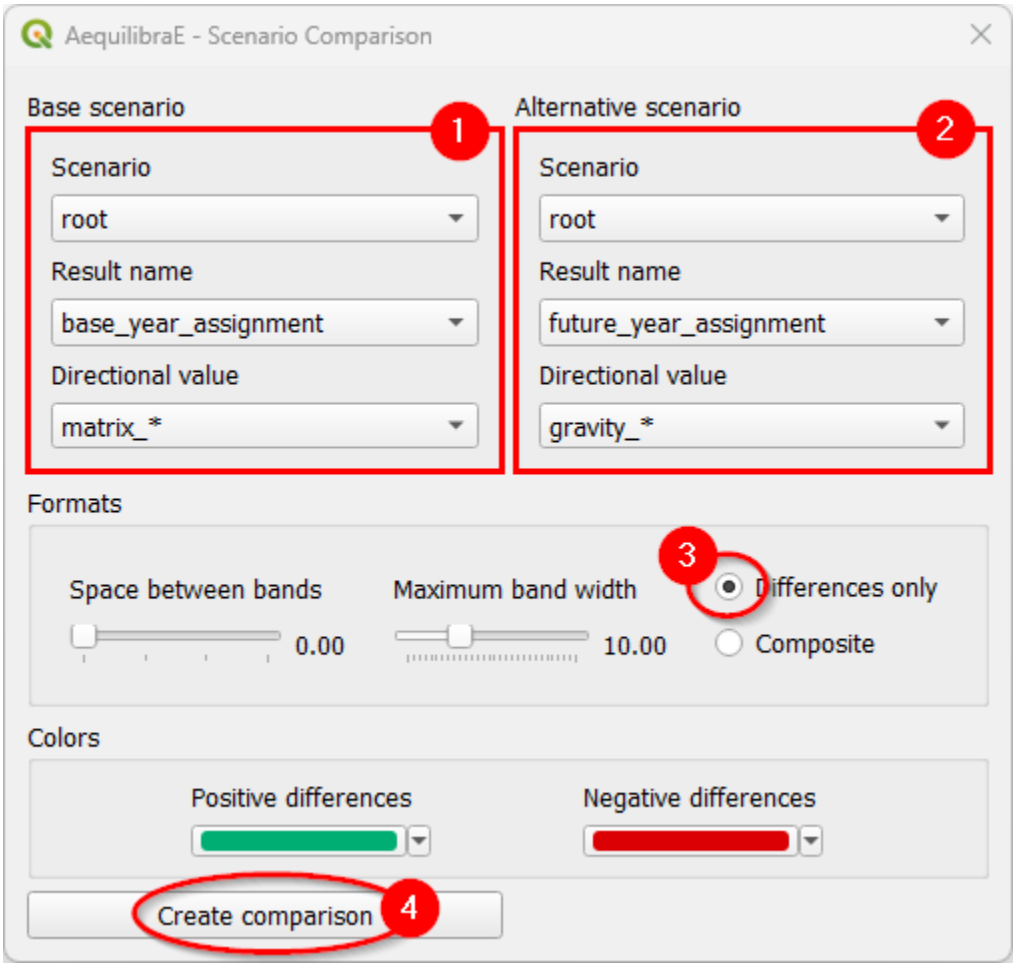
(a) Solid color

(b) Color ramp

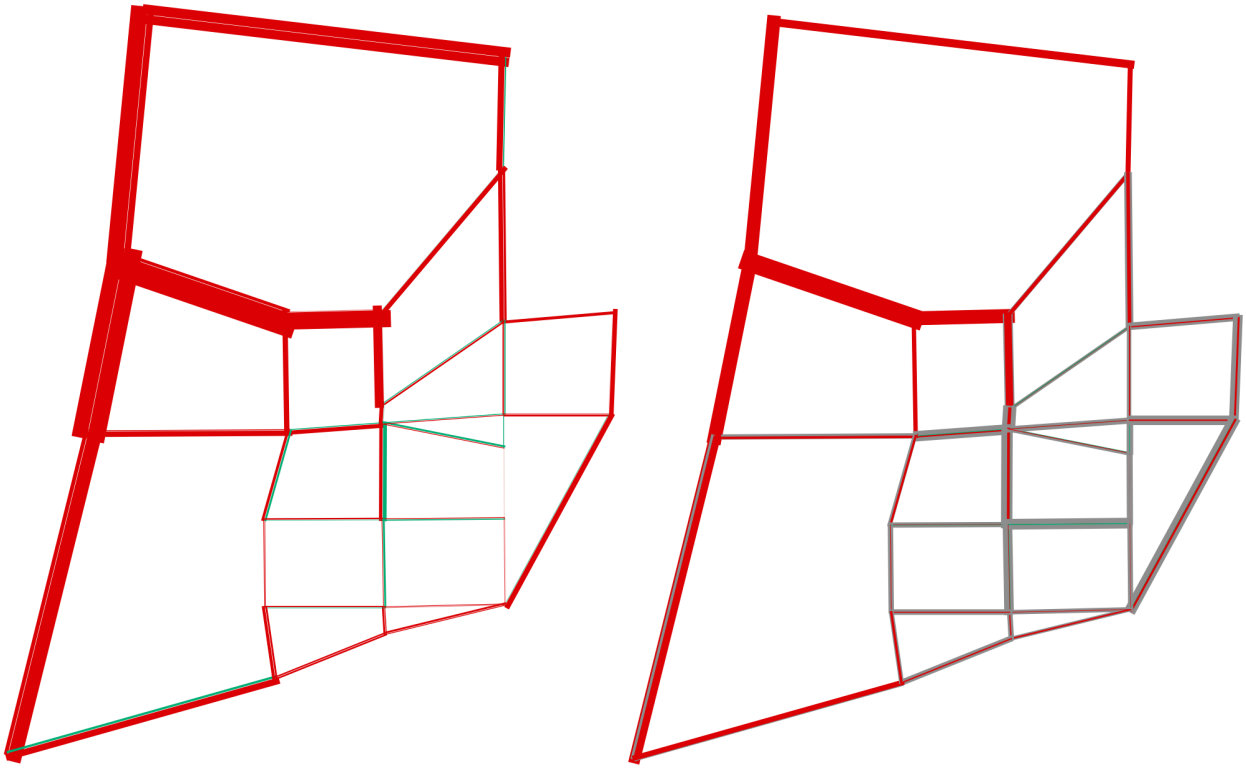


Basic workflow

Create a mapping for scenario comparison is pretty straightforward: select the result name and directional value for the base scenario (1) and repeat the process for the alternative scenario (2). If you want, you can edit the space between bands and the maximum band width: it all depends on how you want your results to look like. Select your displaying method (3) - we'll use differences only, and just click on "*Create Comparison*" (4) to run the procedure.



And this is what it looks like!



(a) Differences only

(b) Composite lines

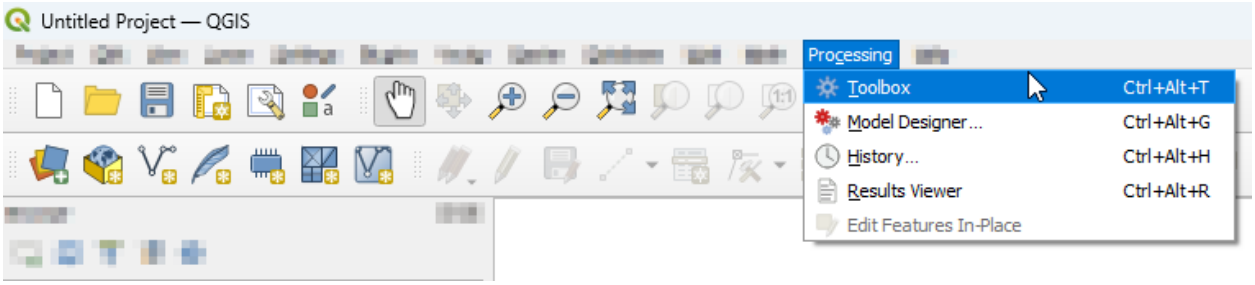
CHAPTER

THREE

PROCESSING TOOLS

AequilibraE's plugin functionalities are also available in a processing plugin. The processing plugin is automatically installed with QAequilibraE and allows you to perform several tasks, such as creating project from links, exporting matrices, and much more.

To find AequilibraE's processing plugin, click on the **Processing** panel and select **Toolbox**. You can also use the available QGIS shortcut to open the Toolbox window.



At the bottom of the window, you'll find the AequilibraE logo and the available functions. The functions are divided into groups, following the same logic as the AequilibraE widget menu. Notice that all AequilibraE functionalities are available for processing, but not all processing tools exist at the main AequilibraE menu.

In the following subsections, we'll go over all menus and its functionalities. As the provider menus are ordered alphabetically, we'll display them in the same order.

3.1 Data

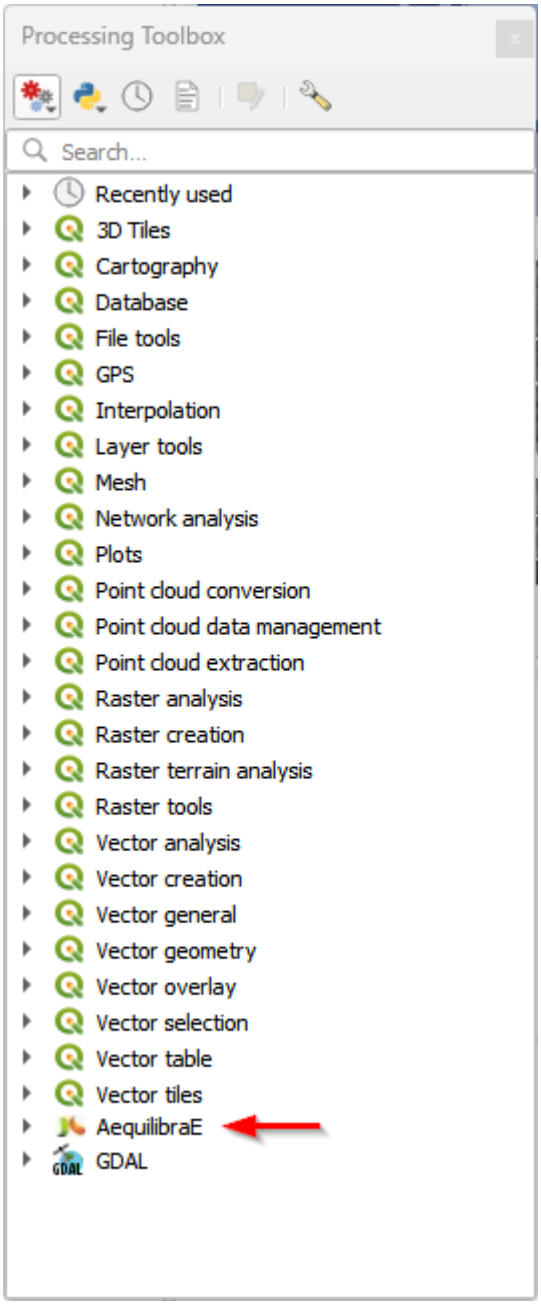
With Data tools, it is possible to import/export matrices to/from the project, as well as perform matrix calculations and generate a trip length distribution output usig project data.

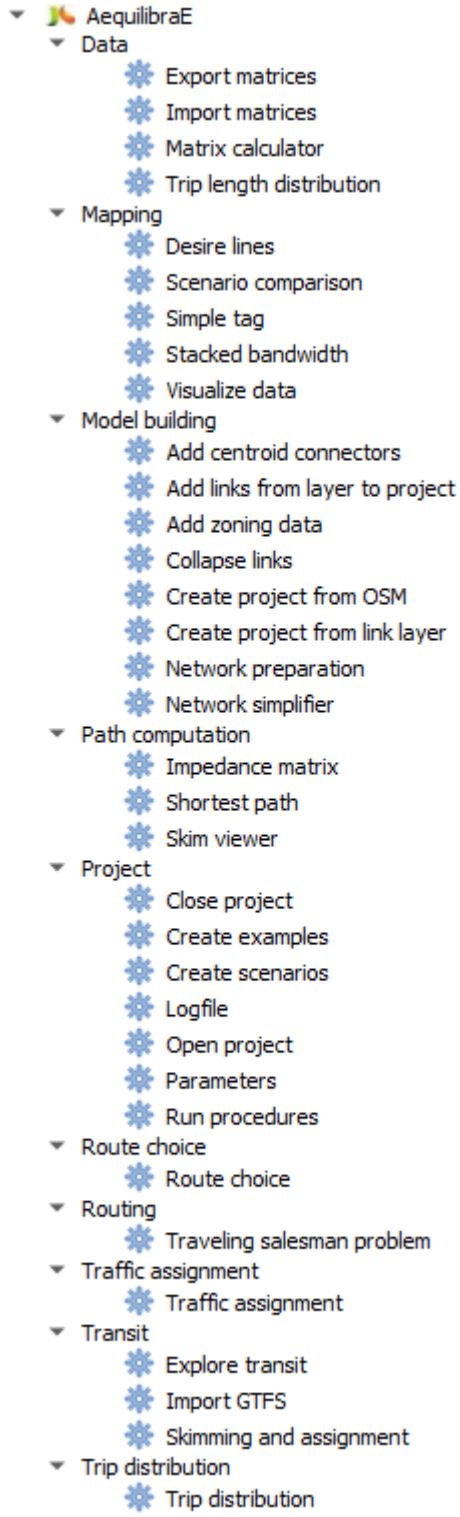
Warning

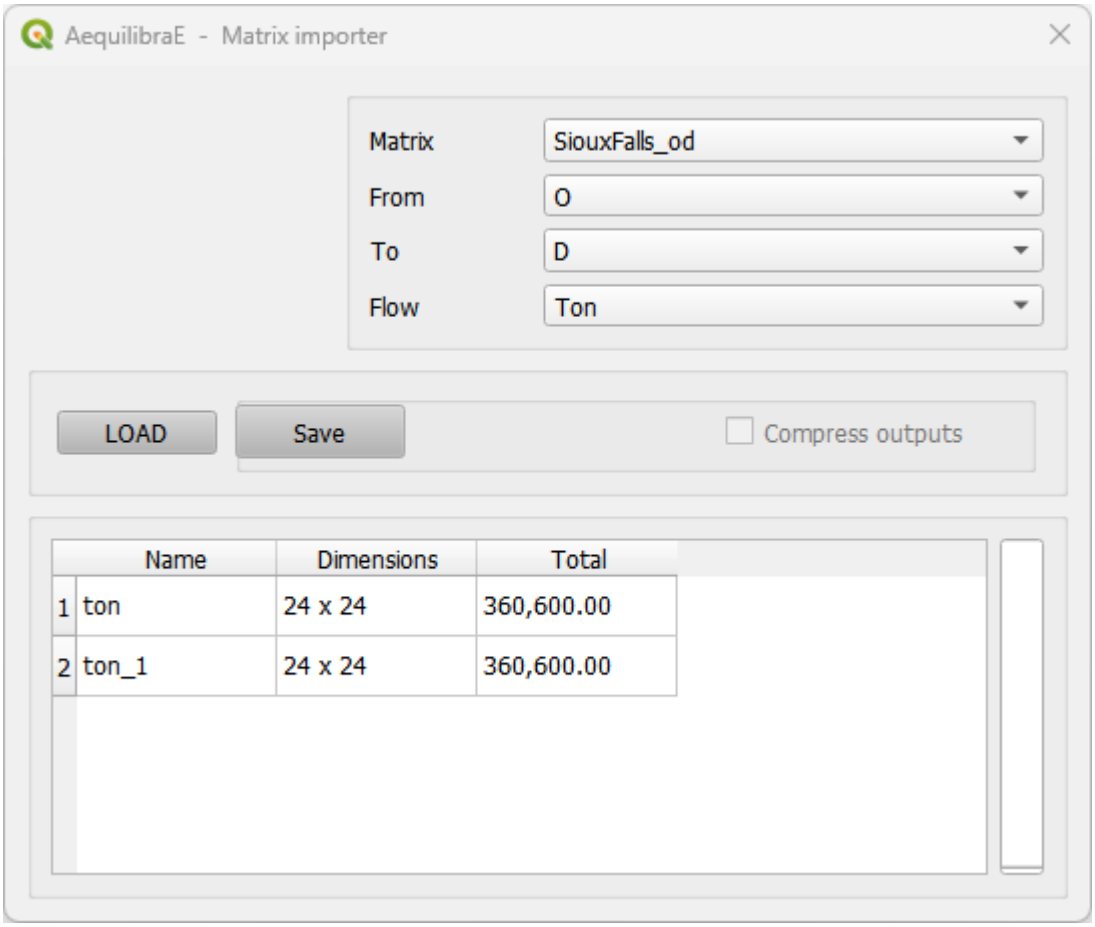
Support for AequilibraE Matrix (AEM) files will be removed in a future version.

3.1.1 Importing matrices

It is also possible for the user to import matrices from an open layer to a project. This can be done by clicking **Data > Import Matrices** and properly indicating the fields in the new window. First click *Load* and then *Save*. A new window will open and you can point to the project matrices folder. To take a look in the matrix you just imported, you can upload the matrix table and display it as shown in the last topic.

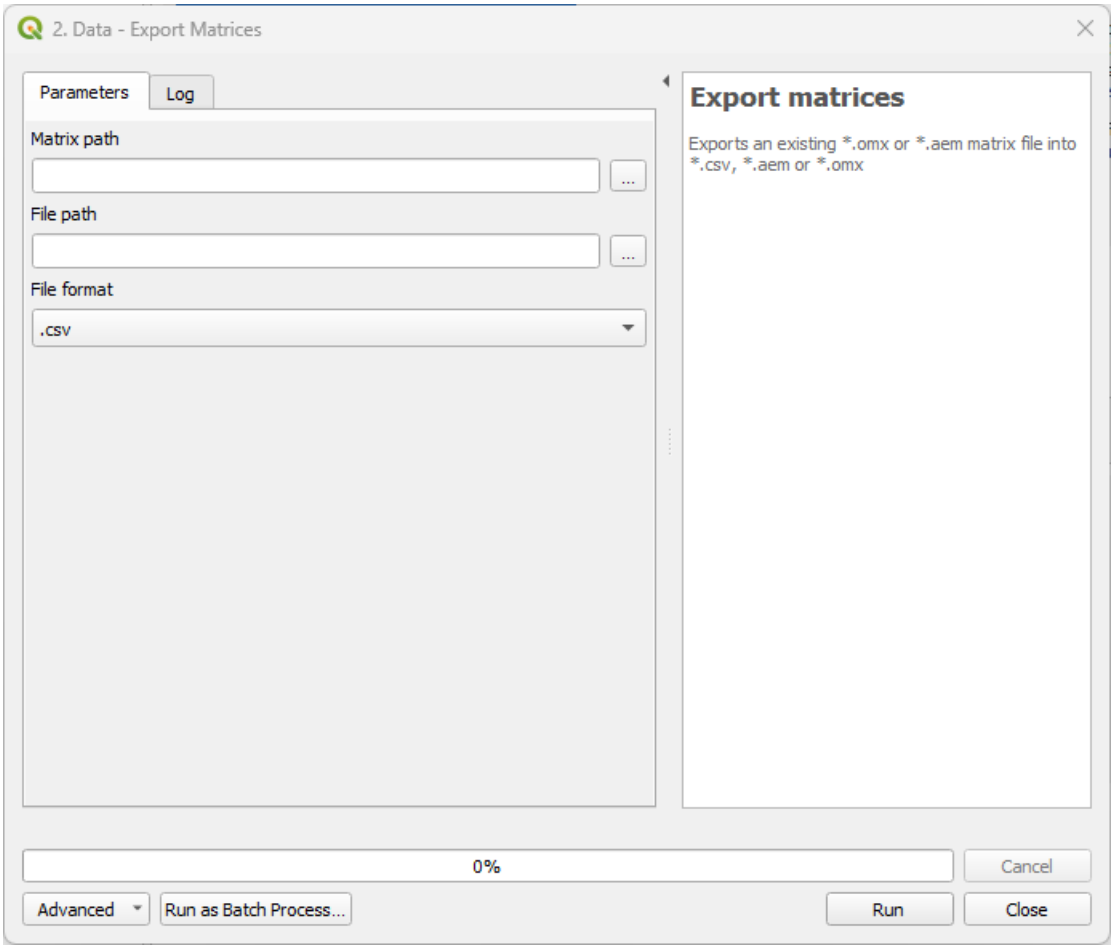

Processing Toolbox
Search...
Recently used
3D Tiles
Cartography
Database
File tools
GPS
Interpolation
Layer tools
Mesh
Network analysis
Plots
Point cloud conversion
Point cloud data management
Point cloud extraction
Raster analysis
Raster creation
Raster terrain analysis
Raster tools
Vector analysis
Vector creation
Vector general
Vector geometry
Vector overlay
Vector selection
Vector table
Vector tiles
AequilibraE
GDAL


AequilibraE
Data
Export matrices
Import matrices
Matrix calculator
Trip length distribution
Mapping
Desire lines
Scenario comparison
Simple tag
Stacked bandwidth
Visualize data
Model building
Add centroid connectors
Add links from layer to project
Add zoning data
Collapse links
Create project from OSM
Create project from link layer
Network preparation
Network simplifier
Path computation
Impedance matrix
Shortest path
Skim viewer
Project
Close project
Create examples
Create scenarios
Logfile
Open project
Parameters
Run procedures
Route choice
Route choice
Routing
Traveling salesman problem
Traffic assignment
Traffic assignment
Transit
Explore transit
Import GTFS
Skimming and assignment
Trip distribution
Trip distribution



3.1.2 Export matrices

The *Export matrices* tool is analogous to the *Export* button in the matrix viewer (see: *this figure* for more details). Its usage is straightforward: select the matrix you want to export, specify the path on your machine to store the file, and select its output format. Only *.aem and *.omx files can be used as input, and the output format can be either one of *.aem, *.omx, or *.csv.

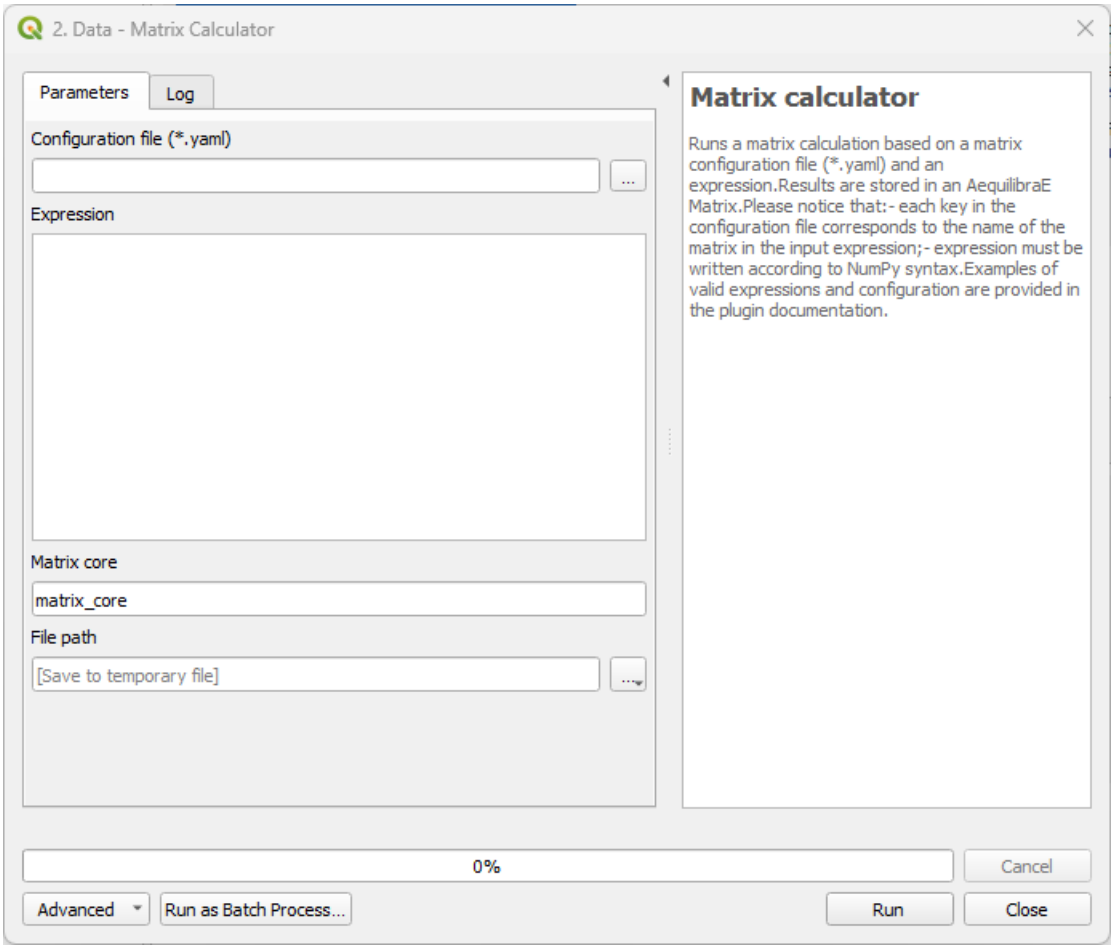


3.1.3 Matrix calculator

Under the hood, this tool performs several matrix calculations using NumPy. Its output is an AequilibraE matrix stored in the file path you provide. Notice that not all matrices operations available in NumPy are also available here. We currently handle the following operations.

- `+`, `-`, `*`, `/`
- `min`, `max`, `abs`
- `ln`, `exp`, `power`
- `null_diag`, `T`

To be more effective in your calculation, please use the brackets to separate the operations in the desired order of execution.



The following code blocks present, respectively, examples of a matrix input configuration for the YAML file and an expression that can be used for calculation.

Listing 0: Matrix configuration

```
# For each matrix used for calculation
- matrix_name1:
    matrix_path: path to file
    matrix_core: specifiy the core name
```

Listing 1: Expression

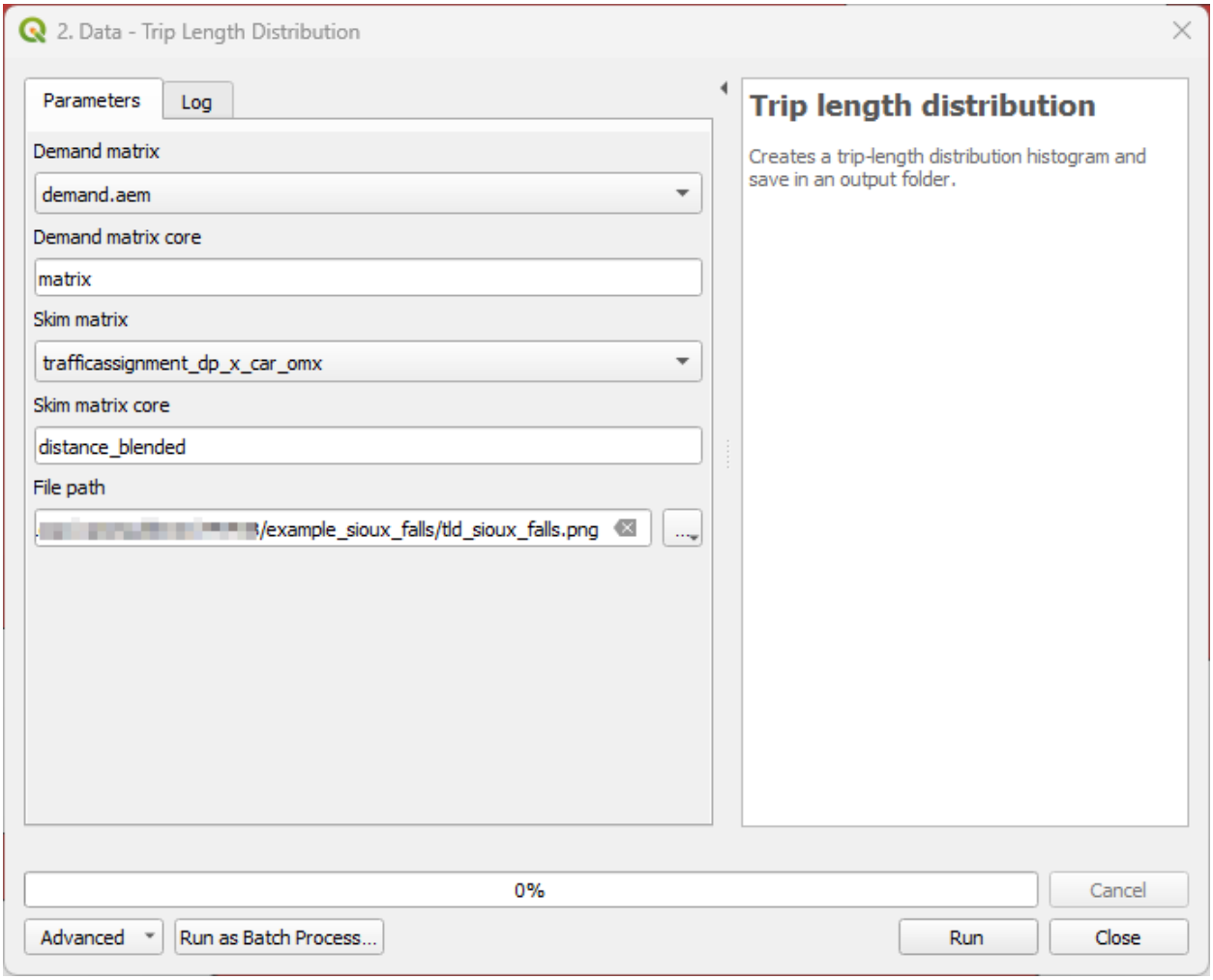
```
(matrix_name1 - matrix_name2).T
```

3.1.4 Trip length distribution

This tool generates a Trip Length Distribution (TLD) plot for a pair of demand and skim matrices and their selected cores.

Important

An open AequilibraE project is required for this tool to work.



3.2 Mapping

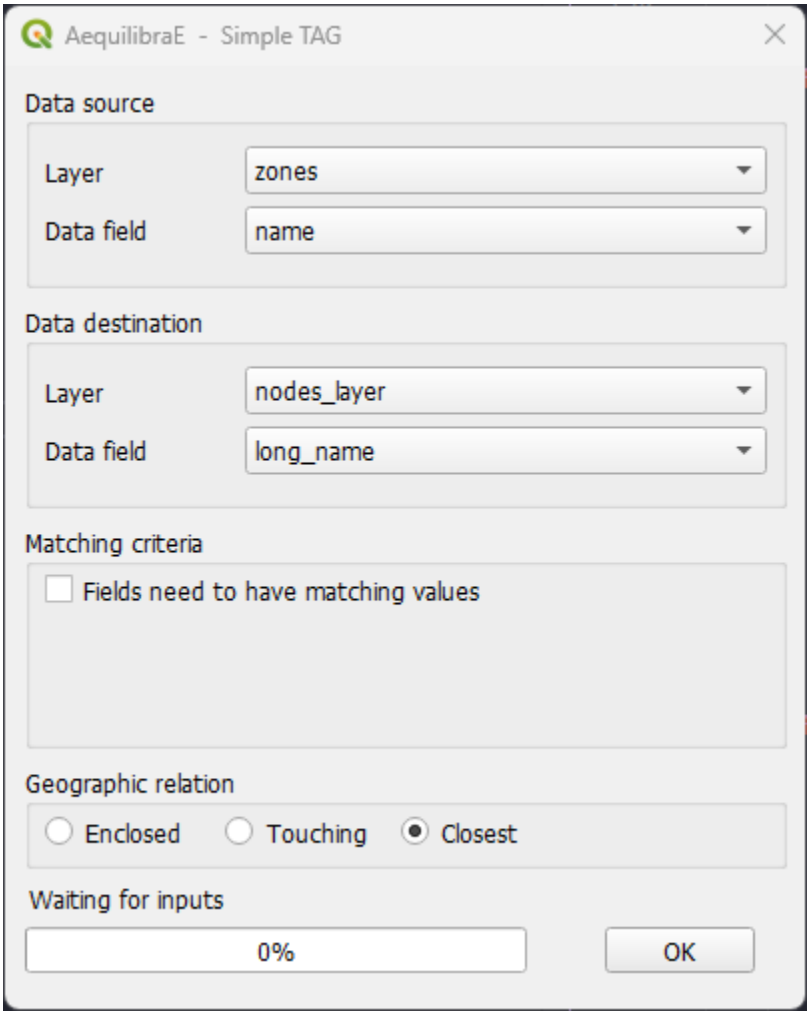
With Mapping tools, the user can easily visualize project data. For the tools not presented here, please refer to the *Mapping tools module* documentation.

3.2.1 Simple tag

Mapping > Simple tag works as a spatial join tool in AequilibraE that allows you to join useful information between layers.

Suppose you have a nodes layer with a 'name' column only with `NULL` values, and a zoning layer with an analogous column 'name' but filled with actual names. We can join the information from the zoning layer into the nodes layer using Simple tag.

We start selecting the layer and the field from which we want to import the data, and then selecting the layer and the field we want to 'paste' the data. Notice that depending on the operation one want to perform, not all methods are available.



Be aware that the existence of triggers in the project database might affect the performance of Simple tag.

3.3 Model Building

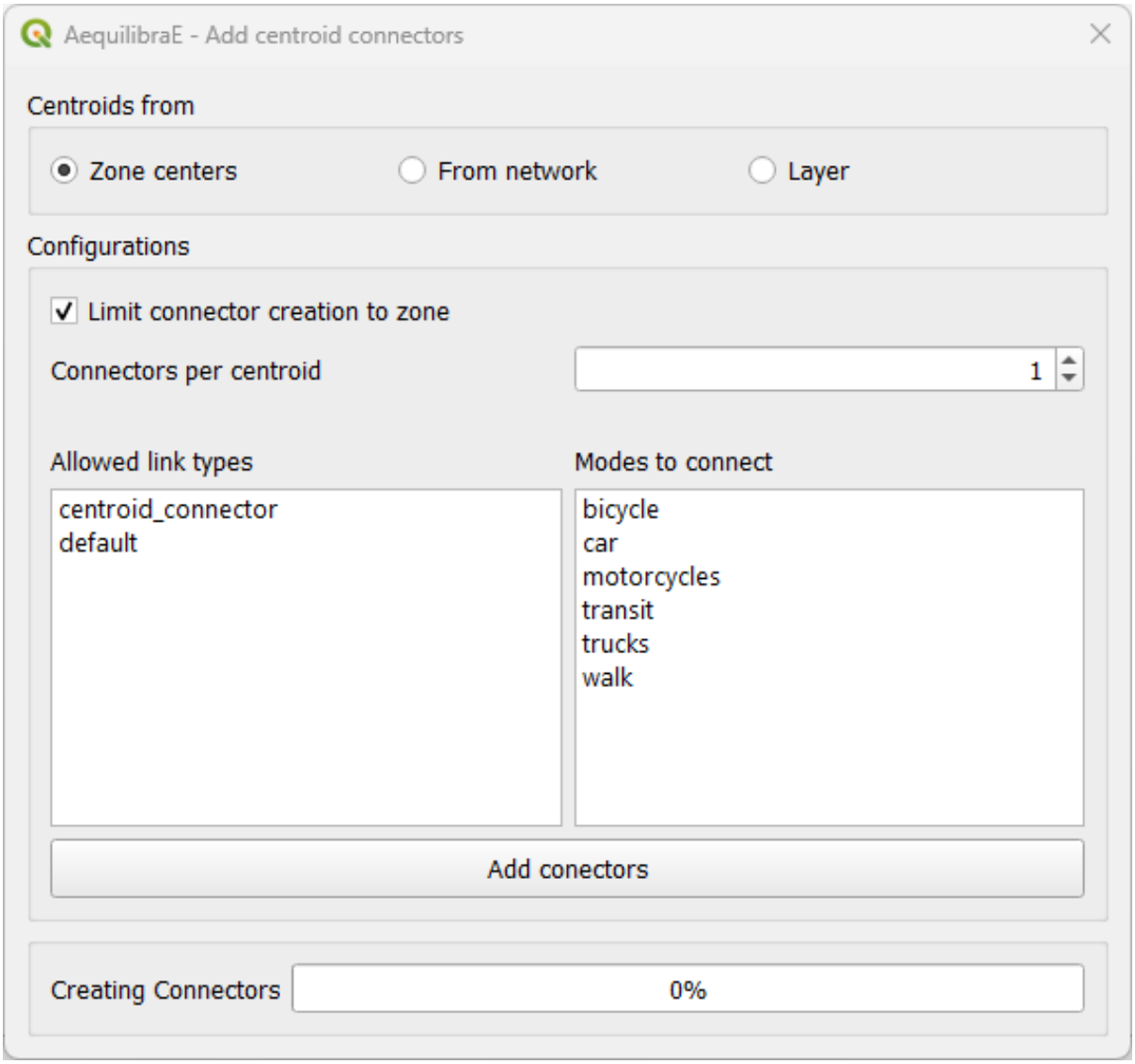
With the Model Building tools, it is possible to effectively build an AequilibraE model, and to do so, there are some options, such as creating project from Open Street Maps or using your existing layers. Model Building also provides options for editing the model's network.

3.3.1 Add centroid connectors

Starting in version 0.6 of AequilibraE, centroid connectors can now only be added to AequilibraE projects, and no longer generates new layers during the process.

Before we describe what this tool can do for you, however, let's just remember that there is a virtually unlimited number of things that can go awfully wrong when we edit networks with automated procedures, and we highly recommend that you **BACKUP YOUR DATA** prior to running this procedure and that you inspect the results of this tool **CAREFULLY**.

The *GUI* for this procedure is fairly straightforward, as shown below.



When creating centroids from zone centers, one can choose to limit the connector to the zone or not. Plase notice if one choose to limit the connector creation to a zone that has fewer nodes connected to links of the required types than the number of connectors will result in fewer connectors being created than desired.

One would notice that nowhere in the *GUI* one can indicate which modes they want to see the network connected for or how to control how many connectors per mode will be created. Although it could be implemented, such a solution would be convoluted and there is probably no good reason to do so.

Instead, we have chosen to develop the procedure with the following criteria:

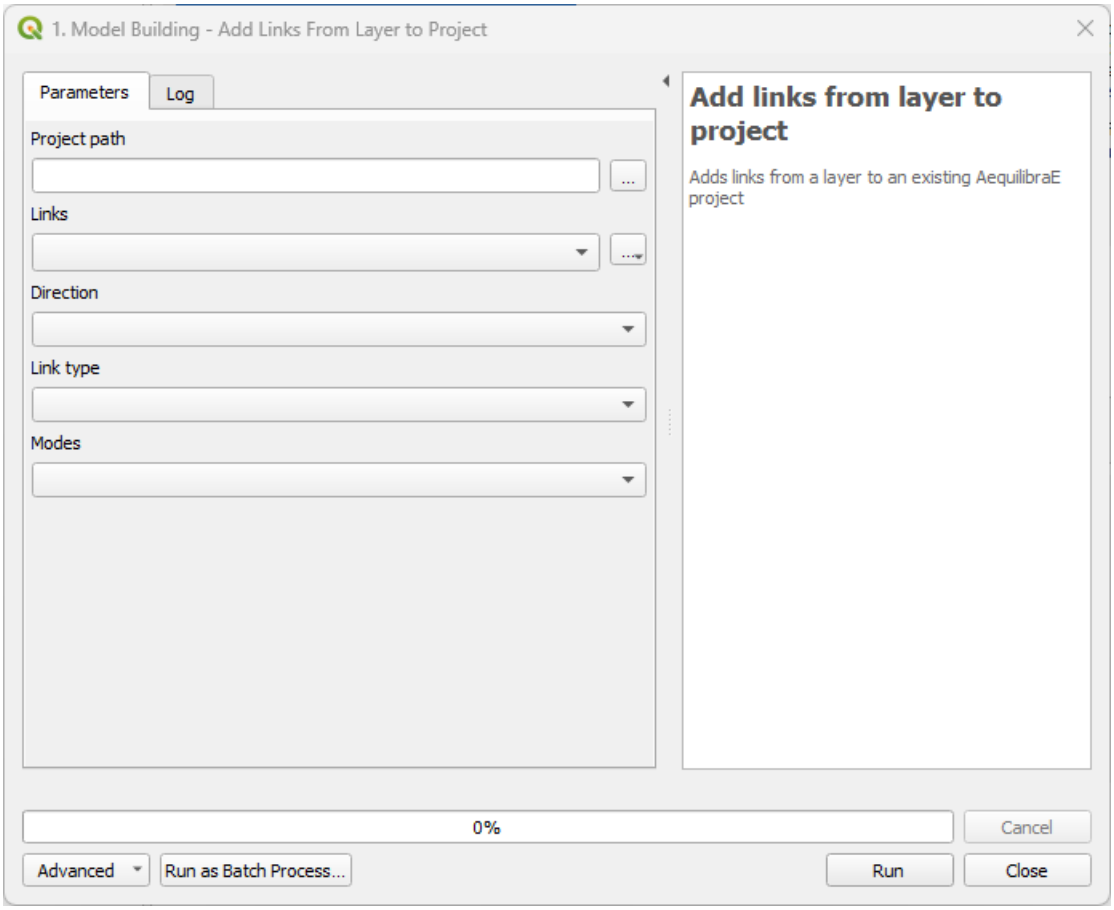
- All modes will be connected to links where those modes are allowed.
- When considering number of connectors per centroid, there is no guarantee that each and every mode will have that number of connectors. If a particular mode is only available rather far from the centroid, it is likely that a single connector to that mode will be created for that centroid
- When considering the maximum length of connectors, the *GUI* returns to the user the list of centroids/modes that could not be connected.

Notice that in order to add centroids and their connectors to the network, we need to create the set of centroids we want to add to the network in a separate layer and to have a field that contains unique centroid IDs. These IDs also cannot

exist in the set of node IDs that are already part of the map.

3.3.2 Add links from layer to project

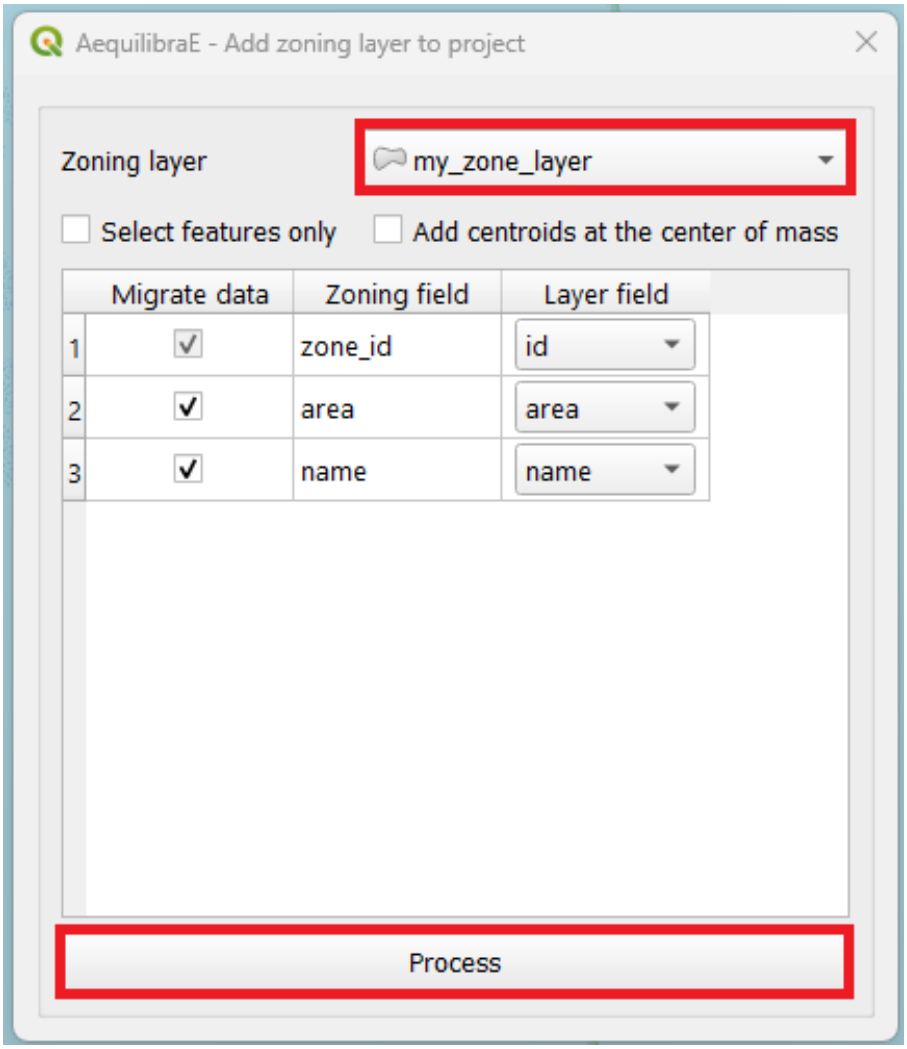
This tool allows you to add links from a vector layer to your existing project network. The fields usage is straightforward: in *Project path*, you add the project's path in your machine, then select a vector layer that corresponds to the new links you want to add to your project, and indicate the layer fields that correspond to the link type, direction, and modes. Notice that this tool doesn't require a node layer, nor does it require fields such as `a_node` or `b_node`, as it will use the existing numbering in the project.



3.3.3 Add zoning data

It is possible to import to AequilibraE project your own zoning system in case you already have one. Currently, AequilibraE only supports one projection system, which is the EPSG:4326 (WGS84), so make sure your zone layer is in this projection.

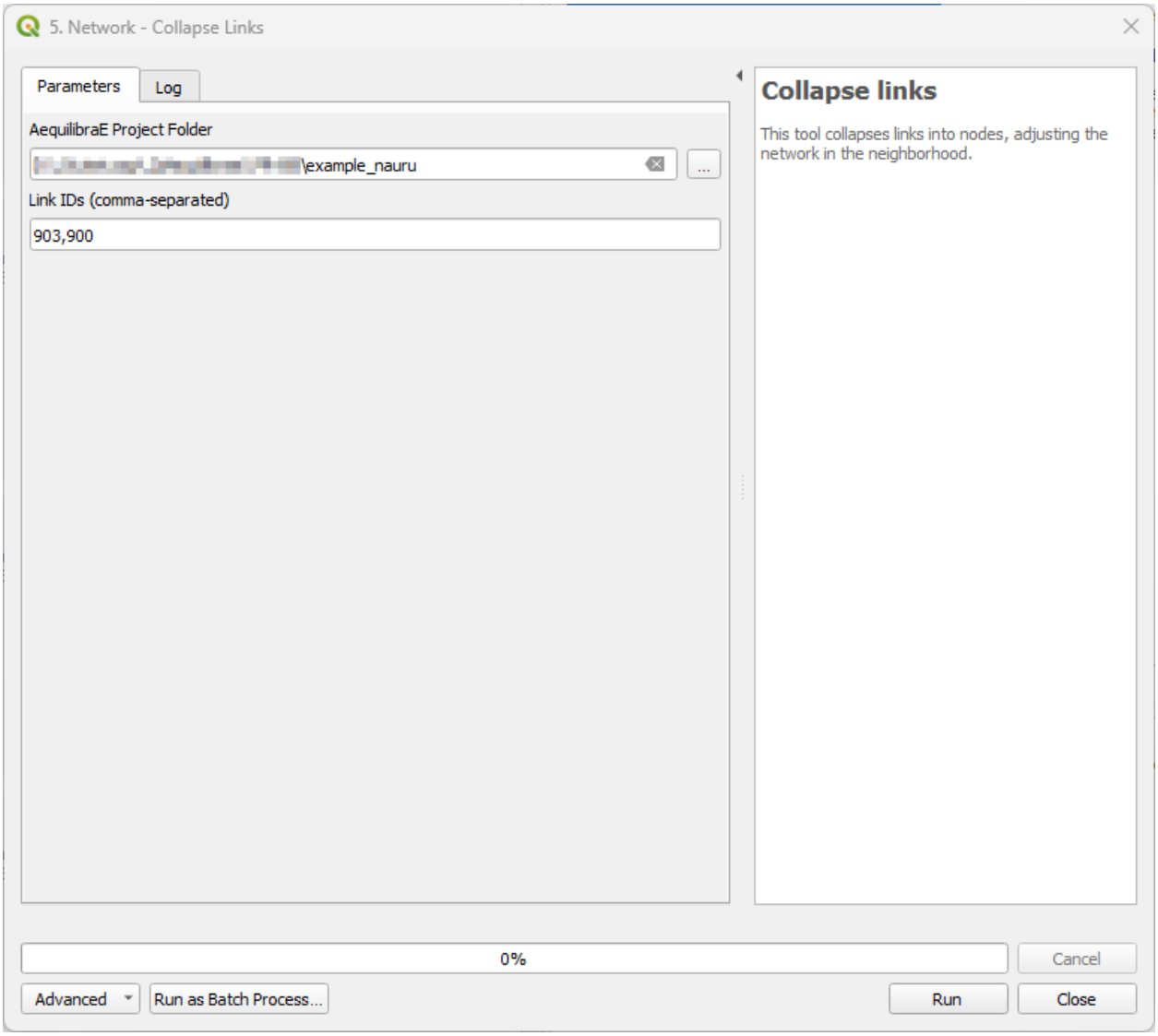
To add your zones to the active project, go to **Model building > Add zoning data**, select the zoning layer you want to add to the project, select weather you want to migrate the data and the respective layer field in the zoning layer, and finally click on process.



3.3.4 Collapse links

This tool allows you to collapse one or more links into nodes, adjusting the network in the neighborhood if necessary.

The input for the tool consists in a folder containing an AequilibraE project and a the link IDs of the links you want to collapse separated by a comma.



3.3.5 Create project from OSM

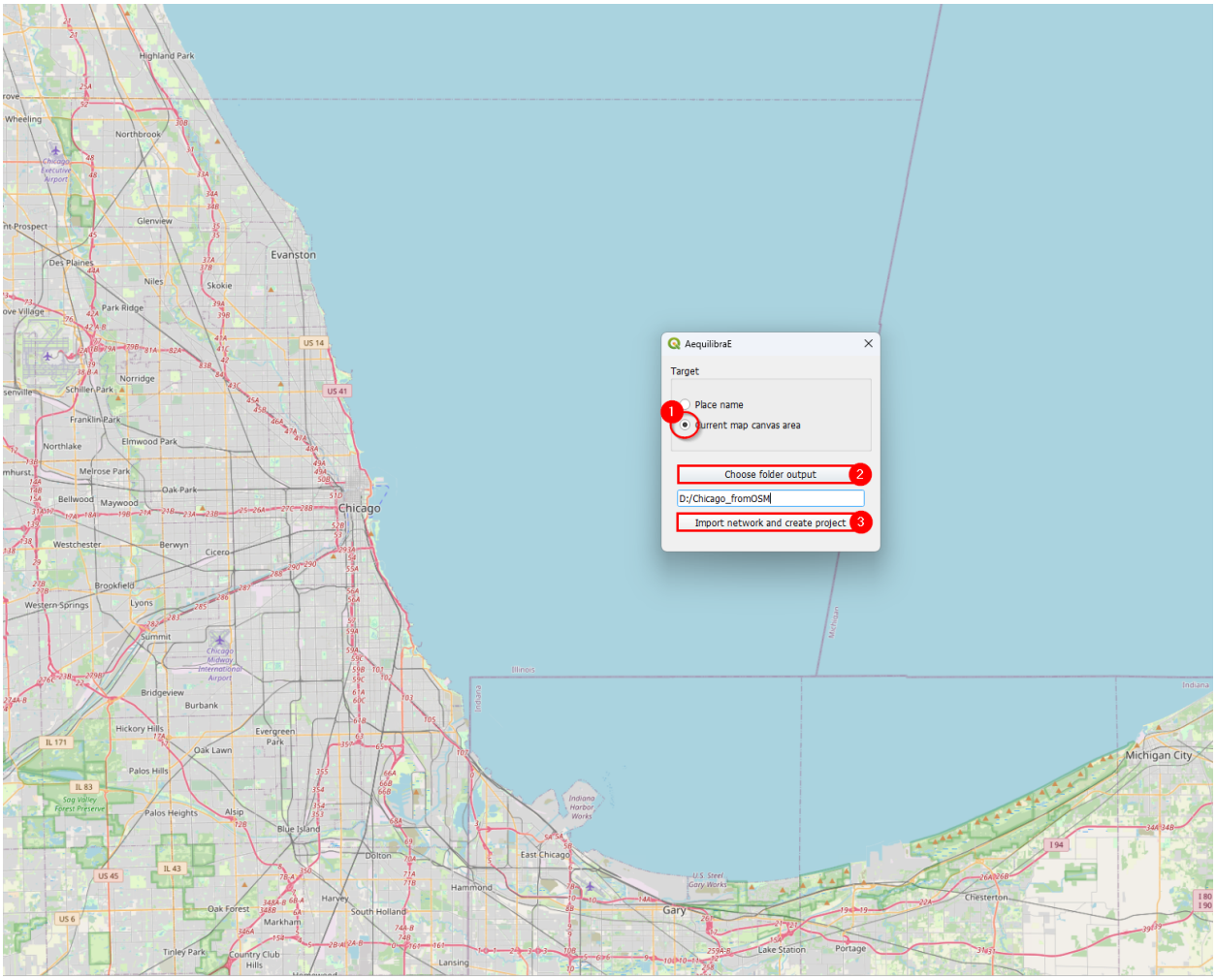
The first feature is the capability of importing networks directly from Open Street Maps into AequilibraE's efficient TranspoNet format. This is also time to give a HUGE shout out to Geoff Boeing, creator of the widely used Python package OSMNx. For several weeks I worked with Geoff in refactoring the entire OSMNx code base so I could include it as a submodule or dependency for AequilibraE, but its deep integration with GeoPandas and all the packages it depends on (Pandas, Shapely, Fiona, RTree, etc.), means that we would have to rebuild OSMNx from the ground up in order to use it with AequilibraE within QGIS, since its Windows distribution does not include all those dependencies.

For this reason, I have ported some of Geoff's code into AequilibraE (modifications were quite heavy, however), and was ultimately able to bring this feature to life.

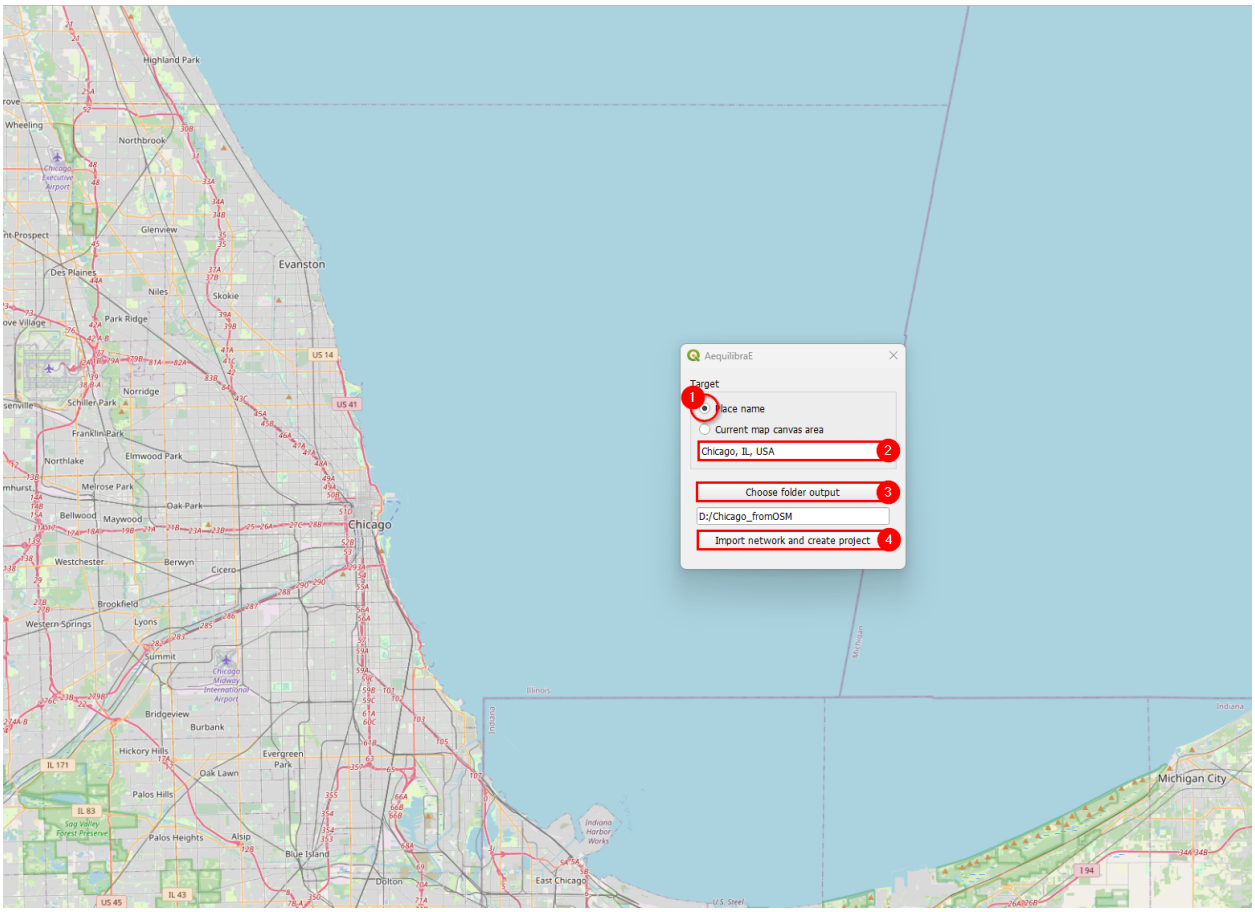
Note

Importing networks from OSM is a rather slow process, so we recommend that you carefully choose the area you are downloading it for. We have also inserted small pauses between successive downloads to not put too much pressure on the OSM servers. So be patient!!

Importing networks from OSM can be done by choosing an area for download, defined as the current map canvas on QGIS…



… or for a named place.



3.3.6 Project from layers

The AequilibraE project can also be bootstrapped from existing line and node layers obtained from any other source, as long as they contain the following required field for the conversion:

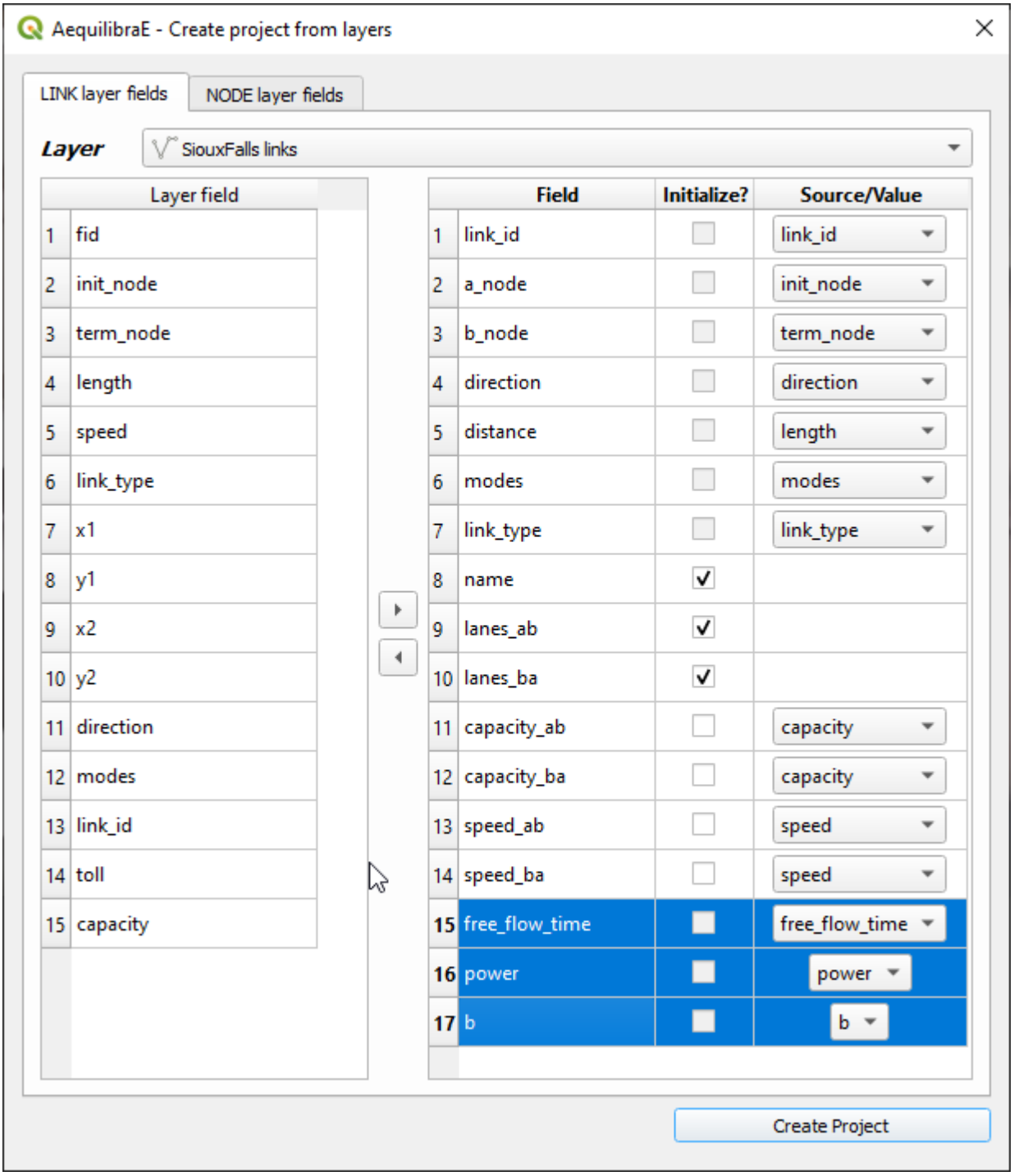
- link_id
- a_node
- b_node
- link direction
- length
- speed
- allowed modes
- link type

These requirements often create quite a bit of manual work, as most networks available do not have complete (or reliable) information. Manually editing the networks might be necessary, which is common practice in transport modelling.

Before creating a project from the layer, you can understand how to prepare the layers for this task on the page Preparing a network.

Basic workflow

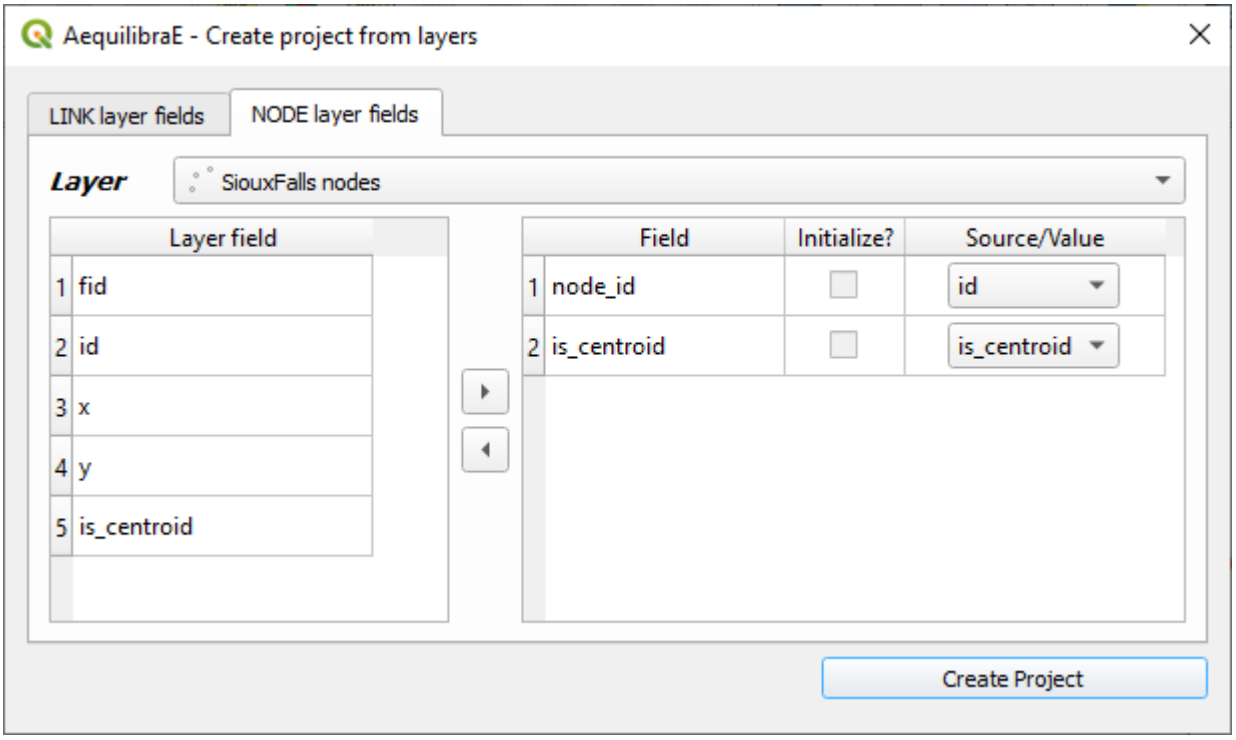
Accessing **Model building > Create Project from Layers**, the user is presented with the following screen.



The first 7 fields for links are mandatory, and one needs to associate the corresponding layer fields to the network fields.

The other fields that will be listed on the left side come from the parameters file (see the manual for that portion for more details), but the user can add more fields from the layer, as all of them are listed on the left side of the screen

In the case of the nodes layer, only two fields are mandatory.



After filling all fields, it is just a matter of saving it!

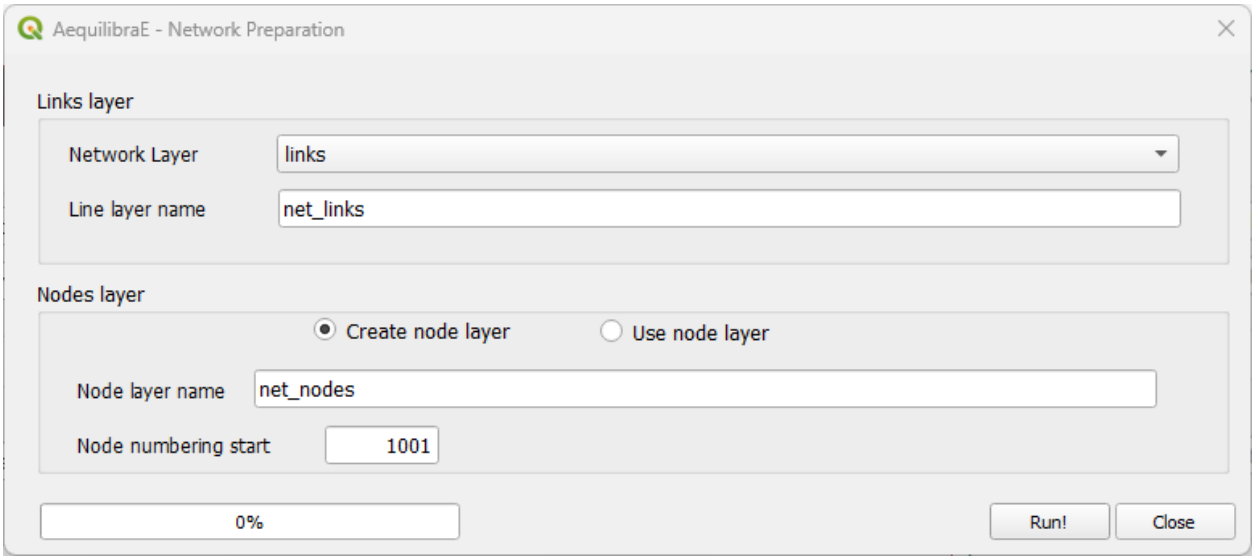
After running this tool a sqlite file (spatialite enabled) will be created and you can edit the network (create, move or delete links and nodes) and both layers (including node *ID* and *A_Node*/*B_Node* fields) will remain consistent with each other.

3.3.7 Network preparation

When preparing your project network, you might face there are two distinct situations:

1. **User has only the network links**: This is the case when one exports only links from a transportation package or downloads a link layer from Open Street Maps or a government open data portal and want to use such network for path computation. This tool then does the following:
   - Duplicates the pre-existing network in order to edit it without risk of data corruption
   - Creates nodes at the extremities of all links in the network (no duplicate nodes at the same latitude/longitude)
   - Adds the fields *a_node* and *b_node* to the new link layer, and populate them with the *IDs* generated for the nodes layer
2. **User has the network links and nodes but no database field linking them**: In case one has both the complete sets of nodes and links and nodes for a certain network (commercial packages would allow you to export them separately), you can use this tool to associate those links and nodes (if that information was not exported from the package). In that case, the steps would be the following:
   - Duplicates the pre-existing network in order to edit it without risk of data corruption
   - Checks if the nodes provided cover both extremities of all links from the layer provided. Node IDs are also checked for uniqueness
   - Adds the fields *a_node* and *b_node* to the new link layer, and populate them with the *IDs* chosen among the fields from the nodes layer

The *GUI* for these two processes can be accessed in the AequilibraE menu **Model Building > Network Preparation**, and it looks like this:

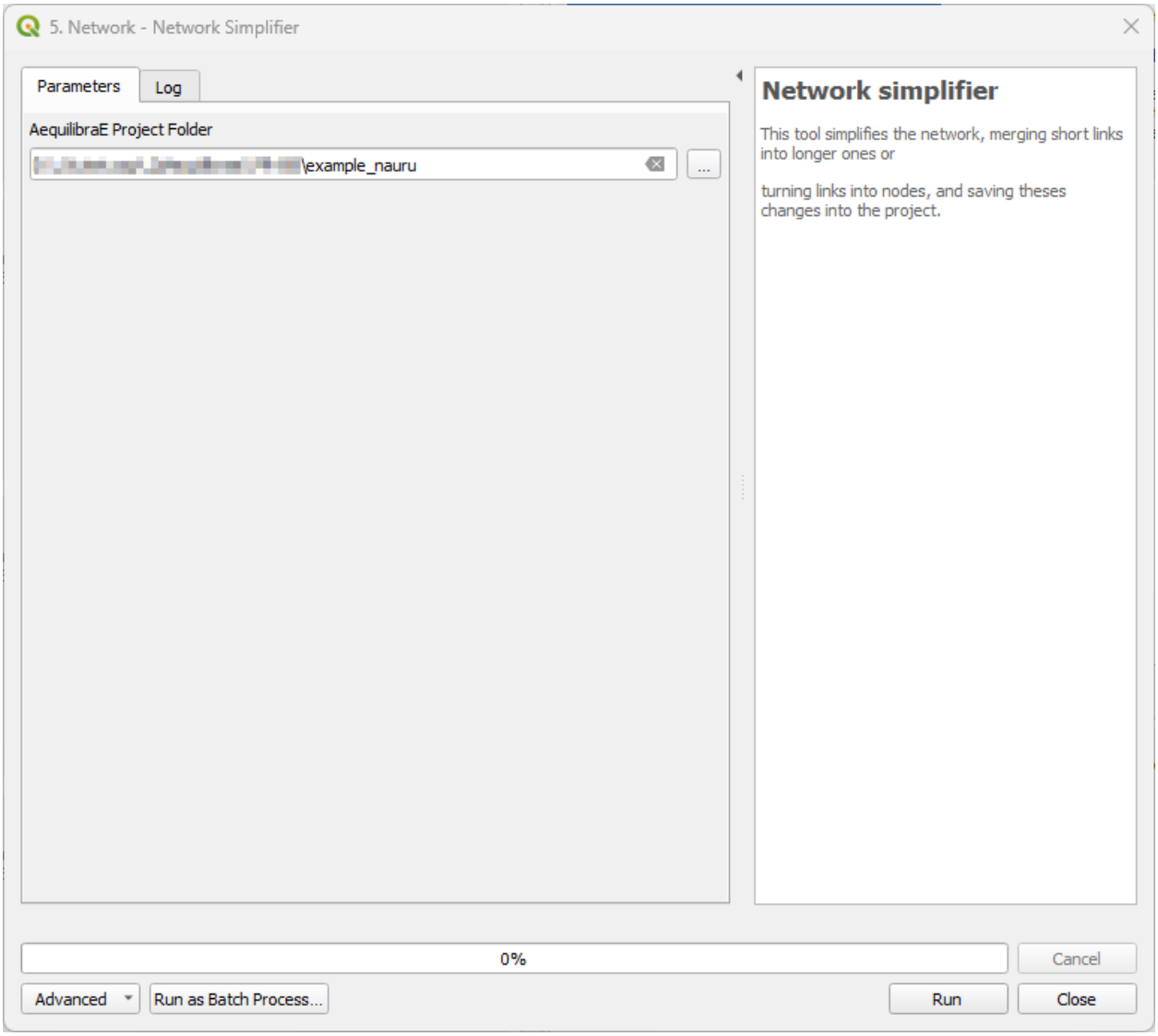


In this case we chose to add nodes with IDs starting in 1,001, as we will reserve all nodes from 1 to 1,000 for centroids, external stations and other special uses (we are not planning to use all that range and that is not necessary, but the numbering gets quite neat that way).

3.3.8 Network simplifier

This tool allows you to simplify the network, merging short links into larger ones or turning links into nodes, and save these changes into the project.

The input for the tool consists in a folder containing an AequilibraE project.



3.4 Path computation

Please refer to the *Path computation module* documentation.

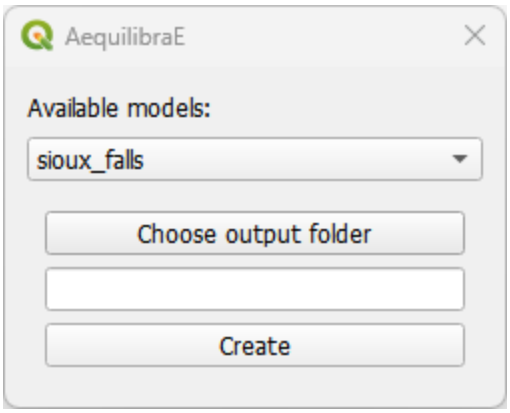
3.5 Project

In the project menu, the user can perform actions such as open/close project, create examples, run procedures, or check parameter and log files. For the tools not presented here, please refer to the *Project* documentation.

3.5.1 Create example

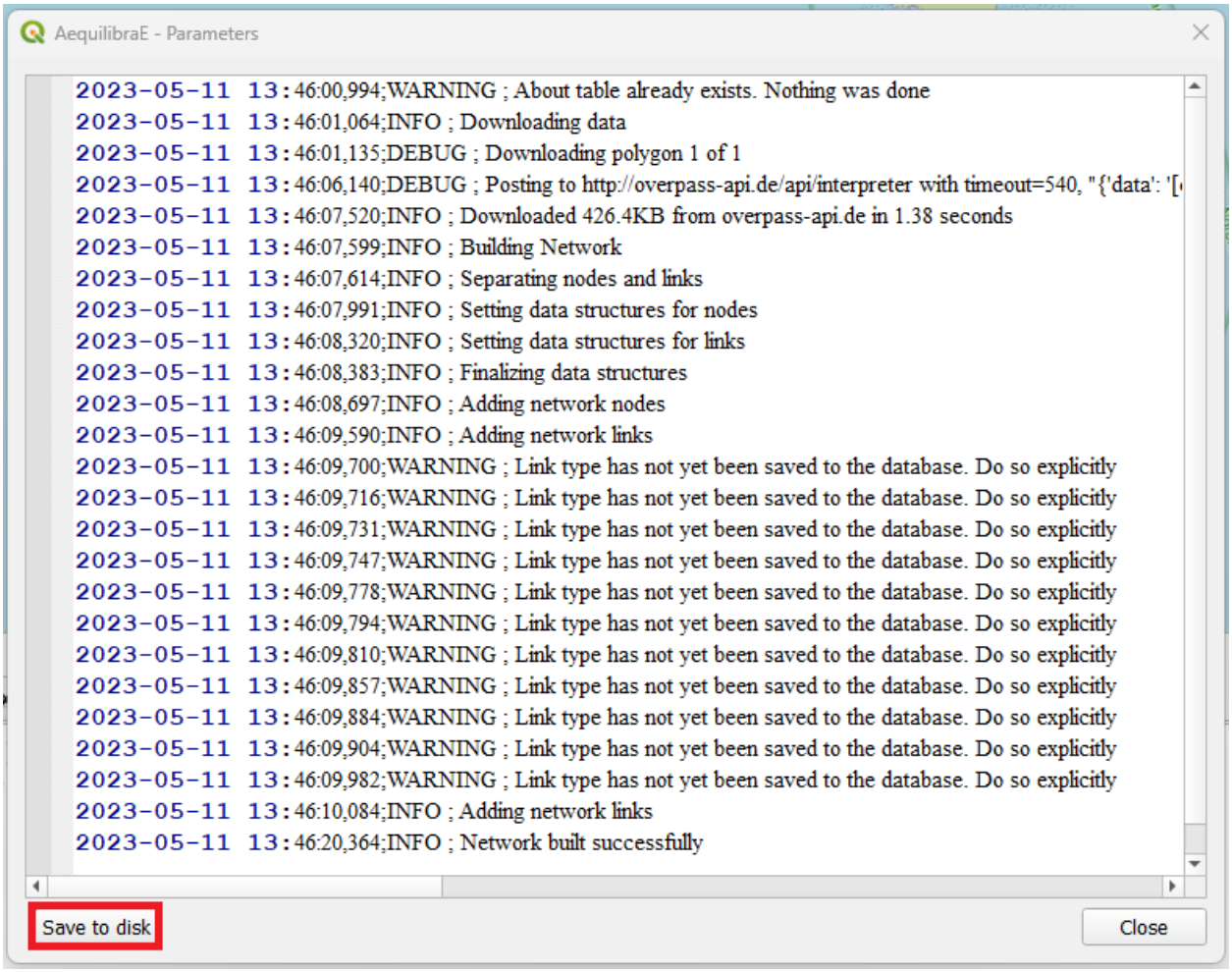
AequilibraE has three different example sets one can use as learning tool, and they were all made available within the QGIS ecosystem.

Within **Project > Create example**, select one of the available models, the desired location of the output folder, and just press *Create*. The window will close automatically and you can open the project folder in the Project tab.



3.5.2 Log file

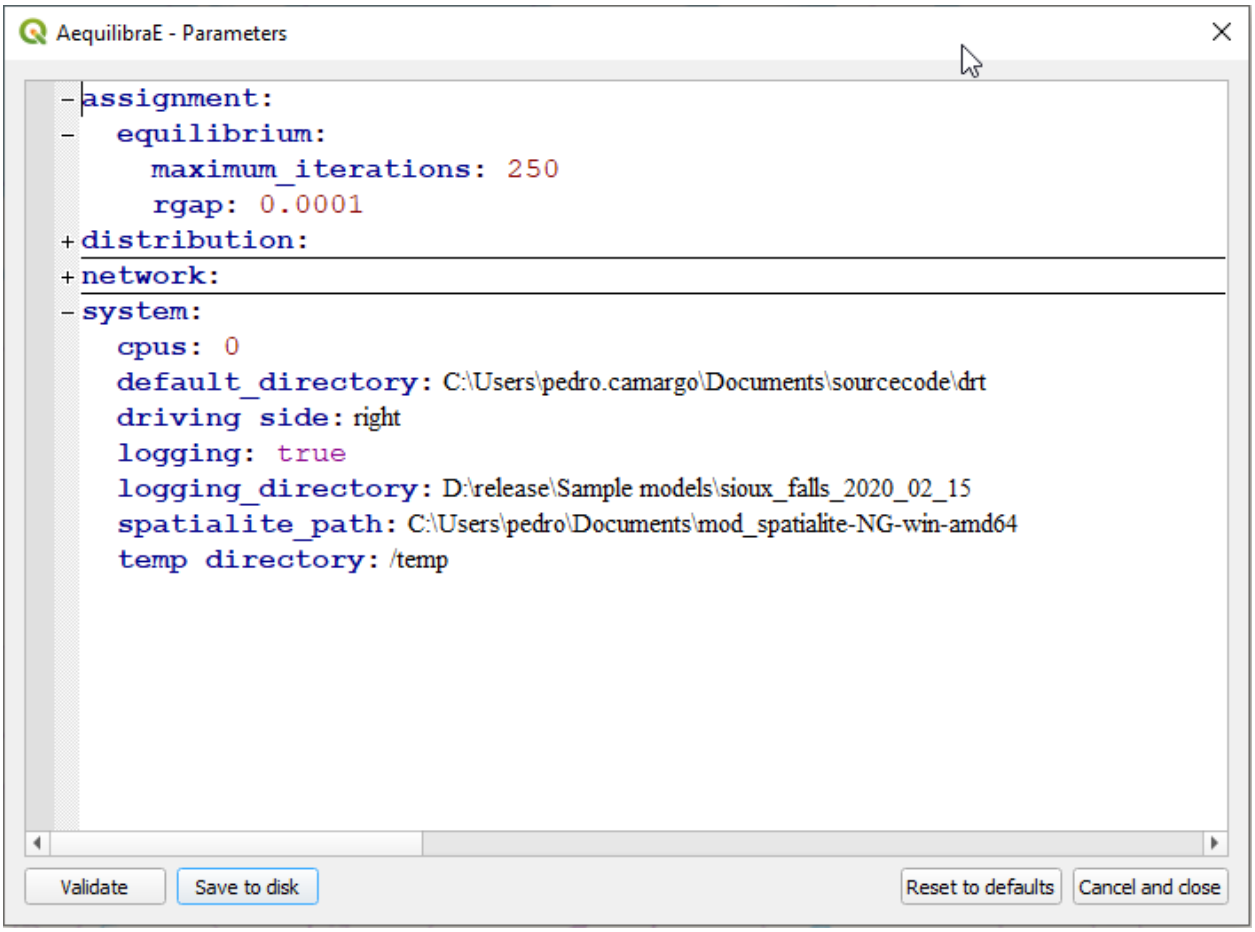
The log file contains information about which actions took place and when they happened. For example, after you *create a project from OSM*, if you access the log file, you are going to see something like the figure below, containing the sequence of steps followed to import the OSM network. If you wish to access this file later on, it is also possible to save this log file locally in your machine, using the **save to disk** button in the lower left corner of the log file box.



3.5.3 Parameters

The parameters file is part of the AequilibraE package for Python, so all the reference documentation for this section can be found in its Python companion page.

The QGIS plugin, however, has a nice interface to view and edit the parameters file, which can be accessed through **Project > Parameters**. This interface, depicted below, allows one to edit and validate parameters before submitting them as the new parameter file for all AequilibraE procedures.



3.6 Route choice

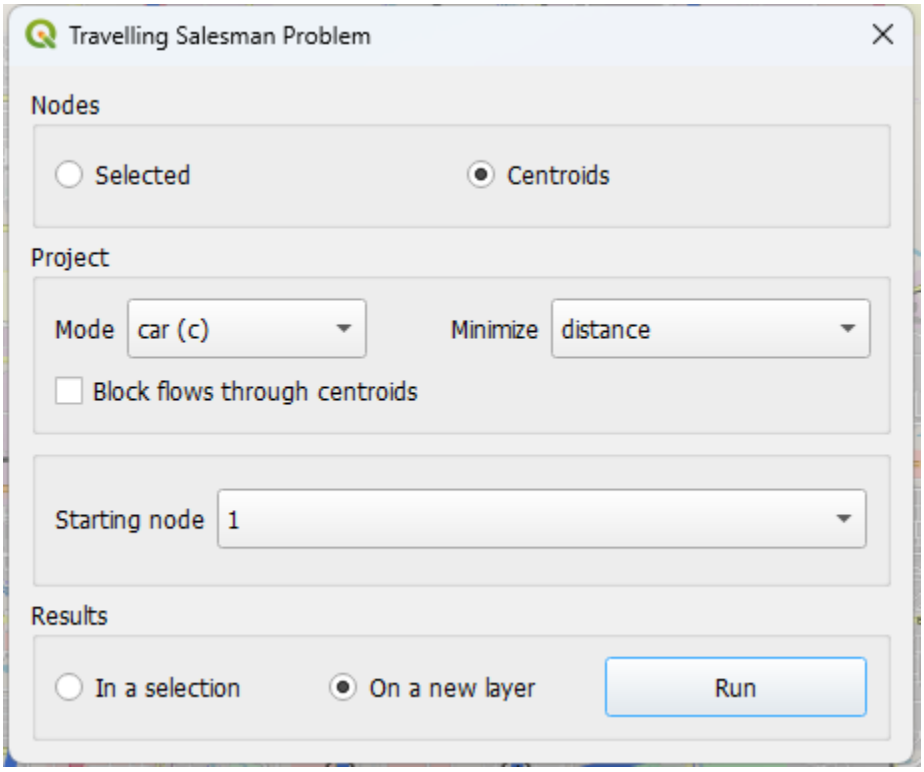
Please refer to the *Route choice* documentation.

3.7 Routing

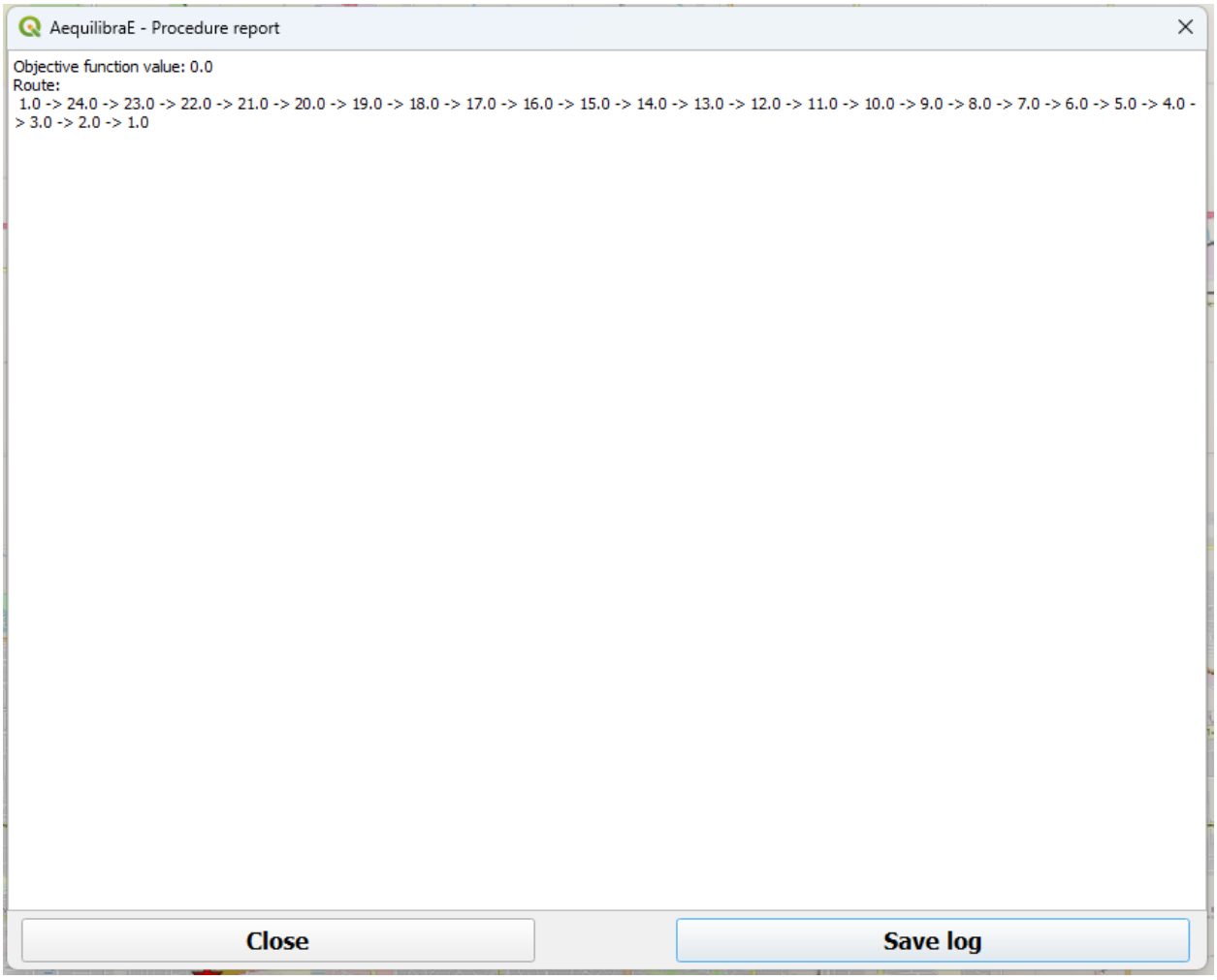
AequilibraE's routing allows the user to run a Travelling Salesman Problem (TSP), using a selected set of nodes or the centroids of a network.

Its usage is straightforward. For Sioux Falls, for example, we would select the centroids of the network, and minimize the distance travelled by car. It is also possible to choose the start node of our TSP (we'll let node_id 1 to be the starting node, but it could be any available node), and indicate we want to see the result in a new layer.

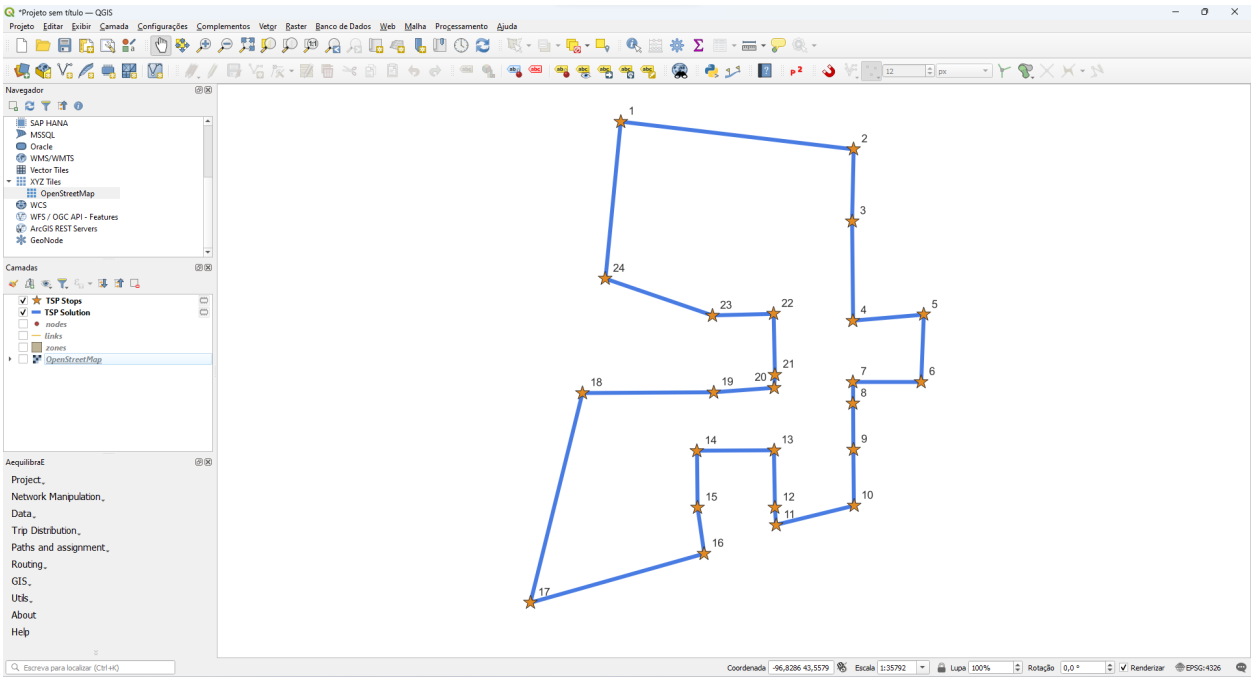
Our prompt box would look like this:



When AequilibraE is done solving the TSP, it provides a procedure report, like the one in the figure below. You can export the procedure report in a .txt file if you wish, by clicking on the lower right button in the window. Otherwise, you can just close this window (the TSP sequence can be found in the TSP stops layer).



And as we chose to display the result in a new layer, it would look like the figure below. Please note that the TSP stops are labeled according their sequence.



Note

TSP is a well-known optimization problem and it has already been implemented in several different software and programming languages. However, the main problem related to TSP is related to its size (hence its complexity). This means that as we increase the number of stops we want to travel to, the software will take much longer to provide you with an answer, and in some cases, it might also crash.

3.8 Traffic Assignment

Please refer to the *Traffic Assignment module* documentation.

3.9 Transit

Please refer to the *Transit module* documentation.

3.10 Trip distribution

Please refer to the *Trip distribution module* documentation.

CHAPTER

FOUR

FREQUENTLY ASKED QUESTIONS (FAQS)

4.1 Installation

The installation fails when using a firewall

When operating behind a firewall, the pip dependency installation may fail and result in an import error when next starting QGIS. One way to ensure the installation, is using the `DownloadAll` class in a script, run in the Python terminal.

```
# This script is to be executed in the QGIS Python console. Open it via the menu bar ->
# Plugins -> Python console, or use Ctrl + Alt + P. Then use the "Show Editor" button,
# paste this script, edit, and press execute. Pasting into the console directly will
# result in an exception.
import os
from qaequilibrae.download_extra_packages_class import DownloadAll

# Set the environment variables required for pip here
# os.environ["variable name"] = "variable value"

# Create an instance of the our downloader class, this handles all the dependencies that
# qaequilibrae needs and removes packages that conflict with the global QGIS packages.
installer = DownloadAll()

# The downloader uses two files to mark when dependencies have need installed, we'll
# remove those to get it to retry the downloads.
installer.retry_pkg_install()
```

4.2 User issues

I've found a problem, how can I report it?

After reviewing the documentation and the discussions on our Google Group, and ensuring that there is an actual bug or documentation issue, you can report it in the following ways:

- Filling a bug report in the issue tracker;
- Creating a post in our Google Group.

APPENDIX

A

CONTRIBUTING TO AEQUILIBRAE FOR QGIS

This page presents some initial instructions on how to setup your system to start contributing to QAequilibraE and lists the requirements for all pull-requests to be merged into main.

A.1 Software Design and requirements

QAequilibraE is built on top of AequilibraE's main features, and the most important piece of AequilibraE's backend is, without a doubt, NumPy.

The user might not see or know, but whenever vectorization is not possible through the use of NumPy functions, compiled code written in Cython is developed in order to accelerate computation.

QAequilibraE also observes a strong requirement of only using libraries that are available in the Python installation used by QGIS on Windows.

We have not yet found an ideal source of recommendations for developing QAequilibraE, but a good initial take can be found in this article.

Please notice that QAequilibraE installation MUST WORK at least in the most recent long-term release (LTR).

A.2 Developing QAequilibraE

We recommend using a dedicated virtual environment to develop QAequilibraE, using the version of Python related to the most recent QGIS long-term release. When this section was updated (October/2025),LTR 3.40.12 was coming with a default 3.12.11 Python environment.

We also assume you are using one of PyCharm or VSCode, which are good IDEs for Python. If you are using a different IDE, we would welcome if you could contribute with instructions to set that up.

(For us,) The easiest way of developing a QGIS plugin is using a Docker container to build an image containing a QGIS installation. When cloning QAequilibraE repository into your local machine you will find a `Dockerfile` with this recipe.

```
git clone https://github.com/AequilibraE/qaequilibrae.git
```

Then all you have to do is activate the virtual environment and adding the environmental variables. Without adding these variables, your installation of AequilibraE in QGIS is goint to be useless.

We understood that the creation of a virtual development environment within a container would be redundant, however after facing some developing issues related to PEP 668, we believe that using a virtual environment would be a good practice.

```
. .venv/bin/activate
export PYTHONPATH=$(pwd)/qaequilibrae/packages:$PYTHONPATH
export QT_QPA_PLATFORM=offscreen
```

If you have to test changes in QAequilibraE after its installed in QGIS, we strongly recommend using the Plugin Reloader, a plugin to reload another plugins.

A.2.1 Developing QAequilibraE and AequilibraE simultaneously

This is a very specific case for features that are being developed simultaneously in the Python package and in the QGIS interface. Here, we need to create a symbolic link that reflects the changes in AequilibraE within QGIS. The following step-by-step instructions are for a Windows operating system (if you are using a different operating system, contributions to this documentation are welcome).

First, let's create a virtual environment for AequilibraE.

```
python -m venv .venv
. .venv/bin/activate
pip install -U pip uv

# Check the branch you are going to install
git status
git pull

# Install AequilibraE in an editable version
uv pip install -e .
```

Open PowerShell as administrator.

```
# Navigate to where your QGIS plugins are
cd C:\Users\renat\AppData\Roaming\QGIS\QGIS3\profiles\default\python\plugins

# Create the symbolic link
New-Item -Path ./qaequilibrae -ItemType SymbolicLink -Value C:\Users\renat\Documents\
↪GitHub\qaequilibrae\qaequilibrae
```

Proceed with the QAequilibraE installation in QGIS normally. Then, navigate to the folder where your plugin was installed and delete the AequilibraE folders. Return to the AequilibraE virtual environment.

```
# Install AequilibraE in QAequilibraE
uv pip install . --target C:\Users\renat\Documents\GitHub\qaequilibrae\qaequilibrae\
↪packages --no-deps
```

This approach for installing AequilibraE in QAequilibraE has a major disadvantage: whenever there is a change in AequilibraE, it is necessary to reinstall it, however this is the simplest configuration for this case.

A.2.2 Developing QAequilibraE with AequilibraE's develop

There are two different scenarios: 1) you will develop updates in QAequilibraE based on AequilibraE's develop branch, or 2) you will test in QGIS (software) whether what you did in the development environment actually works.

The first case is very simple: we install the develop branch in the qaequilibrae/packages folder and clean up the installation of redundant packages in QGIS. "But wouldn't it be easier to just install AequilibraE directly in the virtual environment and call it a day?" Yes, but this way, we wouldn't be developing and testing the plugin in the way it is actually used.

```
python -m uv pip install "git+https://github.com/AequilibraE/aequilibrae@develop" --
↪target qaequilibrae/packages
python -c "from qaequilibrae.download_extra_packages_class import DownloadAll;␣
↪DownloadAll().clean_packages('qaequilibrae/packages')"
```

For the second case, I'm assuming you will test the installation from the QAequilibraE ZIP file. If I'm not mistaken, these git installation operations are not permitted in QGIS, so an alternative is to install the AequilibraE binary, available as an artifact of the Build workflow. Look for the one that corresponds to develop and matches your operating system.

And how do we install this in QGIS? The alternative is to install QAequilibraE from a ZIP file and, initially, cancel the installation of additional packages. An error message reporting that QAequilibraE will not work is displayed, but we can ignore it for now. Go ahead and close QGIS as well. The next operations are performed in the OS4GEO shell.

```
# Check the QGIS python version to be sure which wheel is going to be installed
python --version

# Navigate to where your wheels are stored
cd C:\Users\renat\Downloads\aequilibrae_wheels

# And install it at the 'packages' folder inside QAequilibraE, just like we did before.
python -m pip install aequilibrae-1.5.0-cp312-cp312-win_amd64.whl --target "C:\Users\
↪renat\AppData\Roaming\QGIS\QGIS3\profiles\default\python\plugins\qaequilibrae\packages"
```

Reopen QGIS. QAequilibraE will ask you again if you want to install the additional packages. This time answer yes and let QAequilibraE automatically remove the installation of duplicate packages. If your installation runs without errors, the plugin will be available for use containing the develop version of AequilibraE; otherwise, check the error message on the screen.

A.3 Development Guidelines

QAequilibraE development (tries) to follow a few standards. A huge effort is being undertaken by the development team to update several portions of the code are still not up to such standards.

We try as much as possible to use built-in QGIS tools to develop QAequilibraE. If you need a guide to develop, try the QGIS testing developer cookbook or the QGIS Python API documentation. These two are going to be your development life jackets.

A.3.1 Style

- Python code should follow (mostly) the pycodestyle style guide.
- Python docstrings should follow the reStructuredText Docstring Format.
- We are big fans of auto-code formatting. For that, we use Black.
- Negating some of what we have said so far, we use maximum line length of 120 characters.

A.3.2 Imports

- Imports should be one per line.
- Imports should be grouped into standard library, third-party, and intra-library imports.
- Imports of NumPy should follow the following convention:

```
import numpy as np
```

A.3.3 Translatable Strings

If you are adding or modifying any piece of QAequilibraE's code that includes translatable strings, which are the strings displayed in the widget windows, please ensure you use the `tr` function to locate the strings. This will guarantee that the strings are included in our future translations. Currently, only classes that have a `self` method support the localization of strings.

```
# Indicates that the message "You need at least three centroids to route." will be
# set for translation.
self.iface_error_message(self.tr("You need at least three centroids to route."))

# In case you have to insert any text into a string, the best way is to use string format
self.error = self.tr("ID {} is non unique in your selected field").format(str(id))
```

Strings in QAequilibraE Processing Provider can also be translated. To indicate the strings, import the translation function and configure it to return the context and the message.

```
from qaequilibrae.i18n.translate import trlt

class YourClassHere():
   ...
   # YourClassHere functions
   ...
   def processAlgorithm(self, parameters, context, model_feedback):
     ...
     feedback.pushInfo(self.tr("Running assignment"))  # indicates the translatable␣
↪string
     ...

   def tr(self, message):
     return trlt("TrafficAssignYAML", message)
```

QAequilibraE's translations are all hosted in Transifex. If you want to contribute to QAequilibraE by translating the plugin to other languages or reviewing the existing translations, please let us know in our AequilibraE Google Group. Feel free to request another languages for translation!

In the *plugin internationalization* page, you can find more information on creating your account and start translating QAequilibraE.

A.3.4 Contributing to AequilibraE for QGIS

GitHub has a nice visual explanation on how collaboration is done GitHub Flow. (For us,) The most important points there are:

- The main branch contains the latest working/release version of QAequilibraE
- Work is done in an issue/feature branch (or a fork) and then pushed to a new branch
- Automated testing is run using Github Actions. All tests must pass:
  - Unit testing
  - Build/packaging tests
  - Documentation building test
- If the tests pass, then a manual pull request can be approved to merge into main
- The main branch is protected and therefore can only be written to after the code has been reviewed and approved

- No individual has the privileges to push to the main branch

A.3.5 Release versions

For the past few years, QAequilibraE's release versioning was related to the major and minor releases in AequilibraE. If you frequently update your plugin, you might have noticed that we recently jumped from v1.3.1 to v1.4.3, without any micro releases in between, just because these were the most recent AequilibraE version when the releases happened.

To add the *Plugin Repository* feature, the development team decided to change how the QAequilibraE versioning is done. We'll move from version tags based on AequilibraE, for time-based tags when the release is made (so don't be scared if you see a version such as 25.192.23).

We'll continue using the de-facto Python standard for versioning, but with a different version scheme.

```
MAJOR.MINOR[.MICRO]
```

- MAJOR designates the year of the release
- MINOR designates the number of the day in the year
- MICRO designates the hour of the day the release was made

A.3.6 Testing

QAequilibraE testing is done with some tools:

- Black, the uncompromising code formatter
- Ruff, a linter and code formatter
- pytest, a Python testing tool
- pytest-cov, a tool for measuring test code coverage
- pytest-qt, a tool for testing PyQt5 applications
- pytest-qgis, a tool for writing QGIS tests

To run the tests locally, you will need to figure out what to do…

These same tests are run by GitHub Actions with each push to the repository. These tests need to pass in order to somebody manually review the code before merging it into main (or returning for corrections).

In some cases, test targets need to be updated to match the new results produced by the code since these are now the correct results. In order to update the test targets, first determine which tests are failing and then review the failing lines in the source files. These are easy to identify since each test ultimately comes down to one of Python's various types of `assert` statements. Once you identify which `assert` is failing, you can work your way back through the code that creates the test targets in order to update it. After updating the test targets, re-run the tests to confirm the new code passes all the tests.

Tip

If you want to check if the test values are at the right place in the UI, *qtbot* can help you. Add *qtbot* in the function definition and take a screenshot of the UI. To visualize it, don't forget to use a print statement.

```
path = qtbot.screenshot(dialog)
print(path)
```

A.3.7 Documentation

All the QAequilibraE documentation is (unfortunately) written in reStructuredText and built with Sphinx. Although reStructuredText is often unnecessarily convoluted to write, Sphinx is capable of converting it to standard-looking HTML pages, while also bringing the docstring documentation along for the ride.

Build the documentation in HTML format with the following commands run from the `root` folder. The Dockerfile has already installed the documentation packages in your virtual environment.

```
. .venv/bin/activate

# Replace the variables if needed
export LANG=C.UTF-8
export LC_ALL=C.UTF-8

cd docs
make html
```

A.3.8 Finally

A LOT of the structure around the documentation was borrowed (copied) from the excellent project ActivitySim.

APPENDIX

B

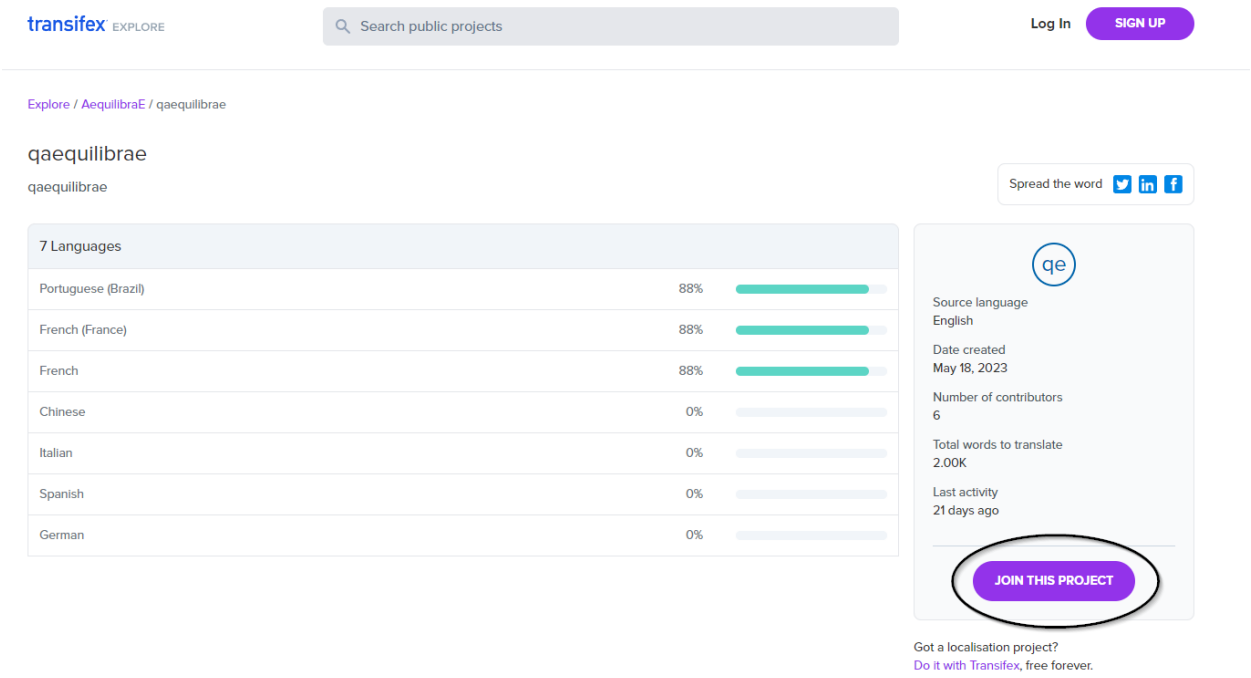
PLUGIN INTERNATIONALIZATION (I18N)

In this section, we briefly explain how you can set up your Transifex account and start contributing to QAquilibraE translation.

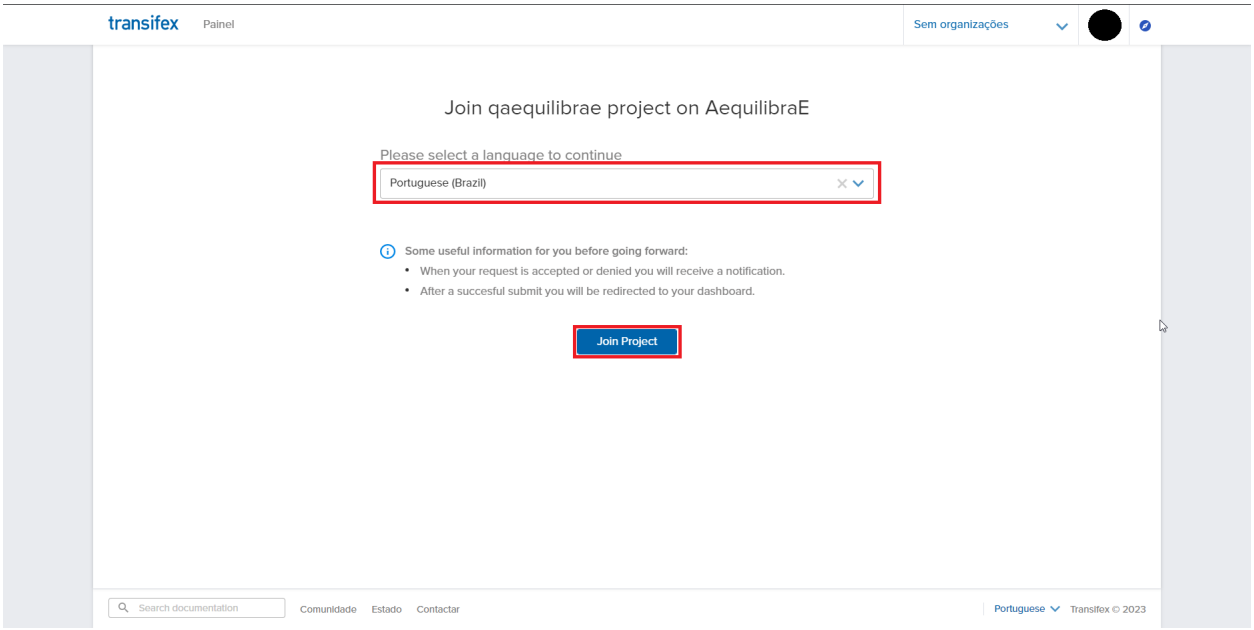
B.1 Creating account

To translate QAequilibraE's QGIS plugin, we use Transifex, a web-based platform that hosts translations. Before joining our project, you must create an account. Direct yourself to Transifex login page and follow the instructions. The process is straightforward and does not take a lot of time.

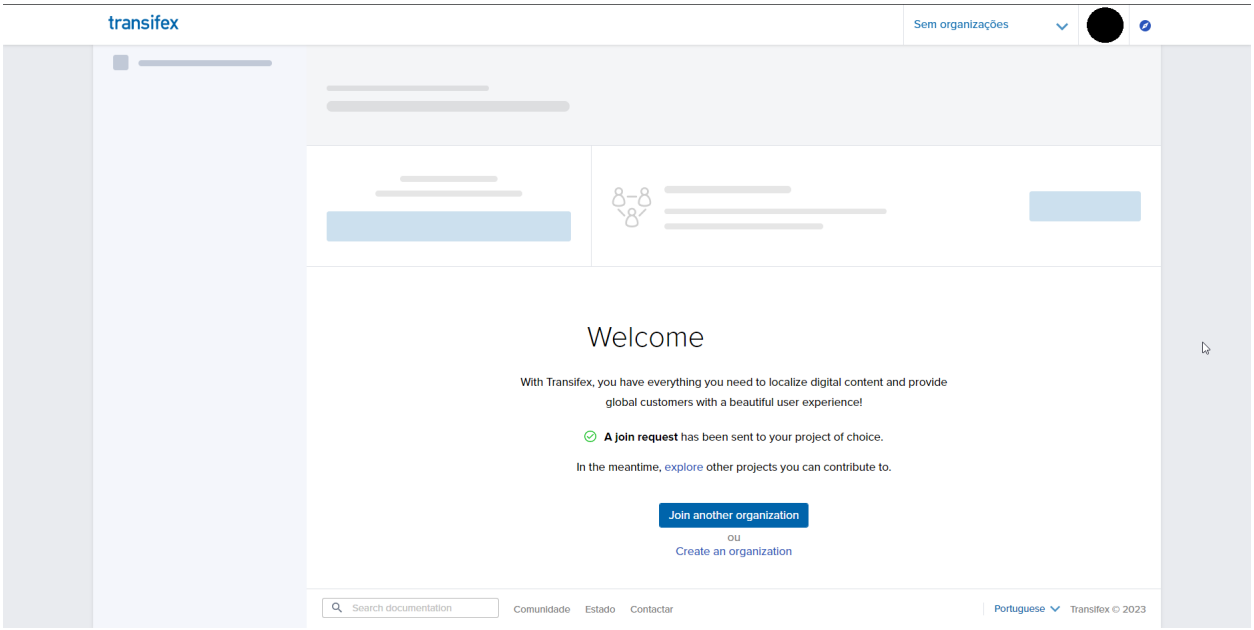
After creating your account, direct yourself to team's page and request your access to the project. Approving your request might take a couple of days (we are a small team of developers!), so don't worry.



You'll be redirected to this page to enter your translation languages of preference.

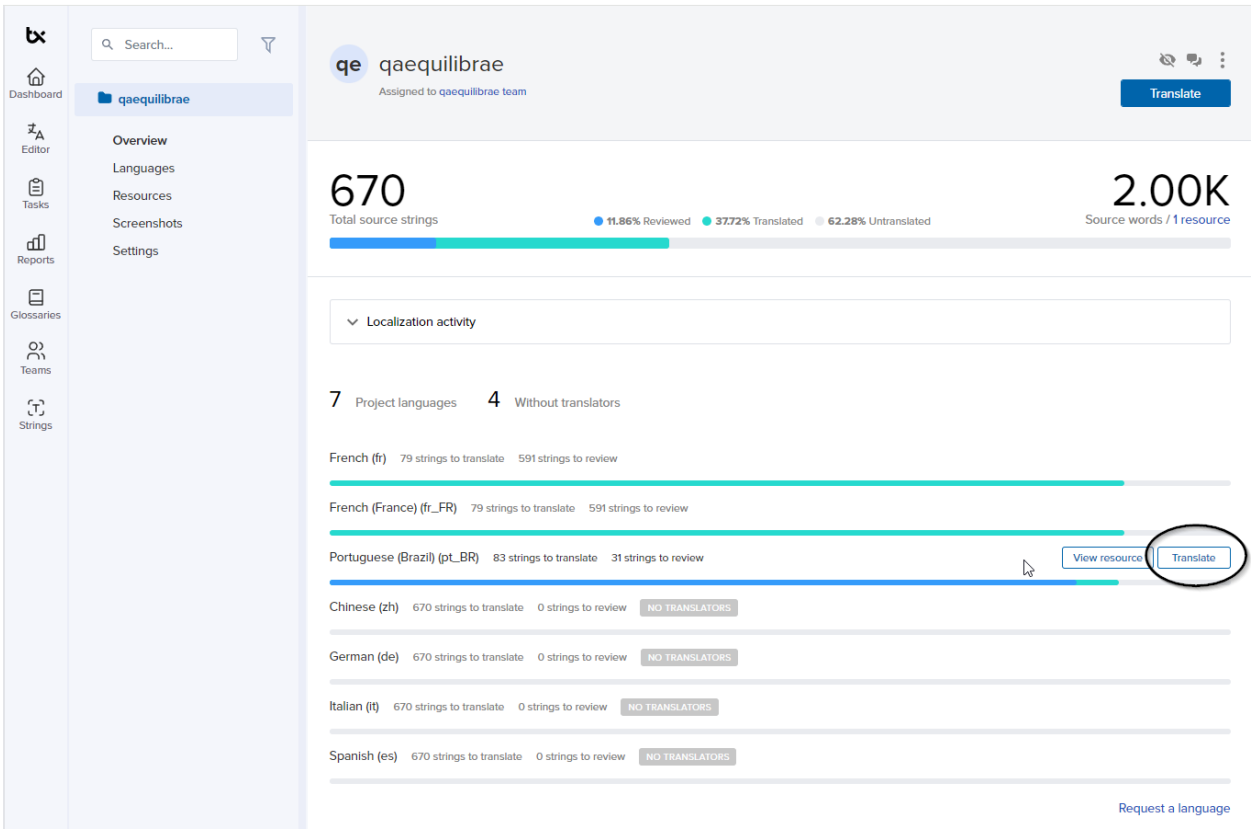


As soon as your request is approved, you can log in again into Transifex and begin translating.

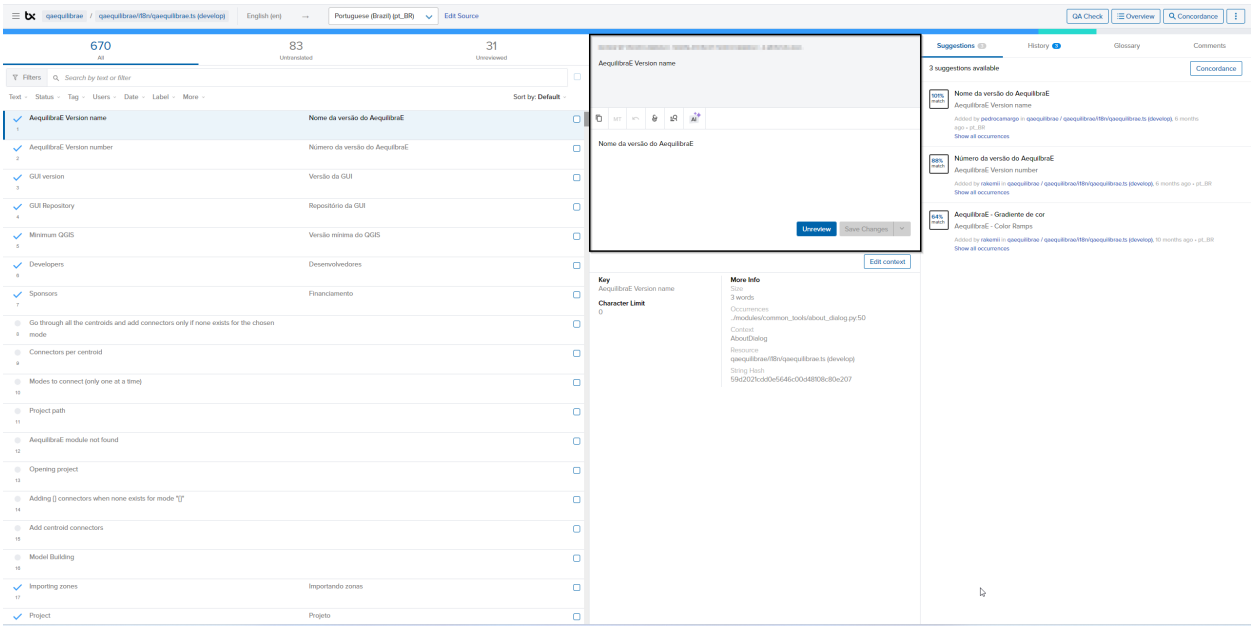


B.2 Translating into your language

Your project screen should look like the one in the figure below. If you feel comfortable translating into any of these idioms, just drag your cursor to the right side of each translatable idiom and click on the **Translate** button.



You'll be redirected to a new page.



You can check the translation status in the left column (*Text*), where the green buttons indicate that there is an existing translation for that string. You can insert their translation in the box and save it for all strings If your string happens to appear multiple times, Transifex will give you translation suggestions, that can match whatever you want to translate. This can spare you some time.

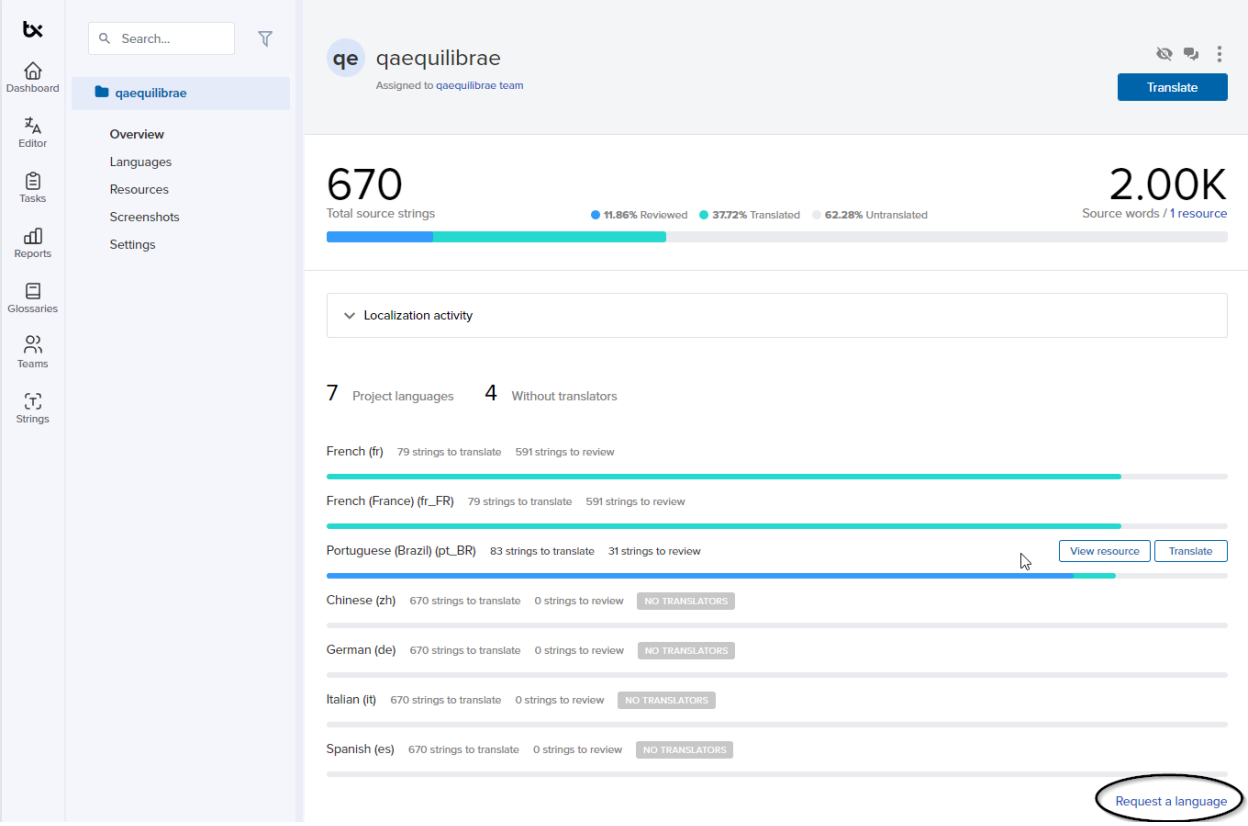
When you finish your work, direct yourself to the **Unreviewed** field, to check all strings you have translated. Here's

your opportunity to look out for typos or incorrect translations.

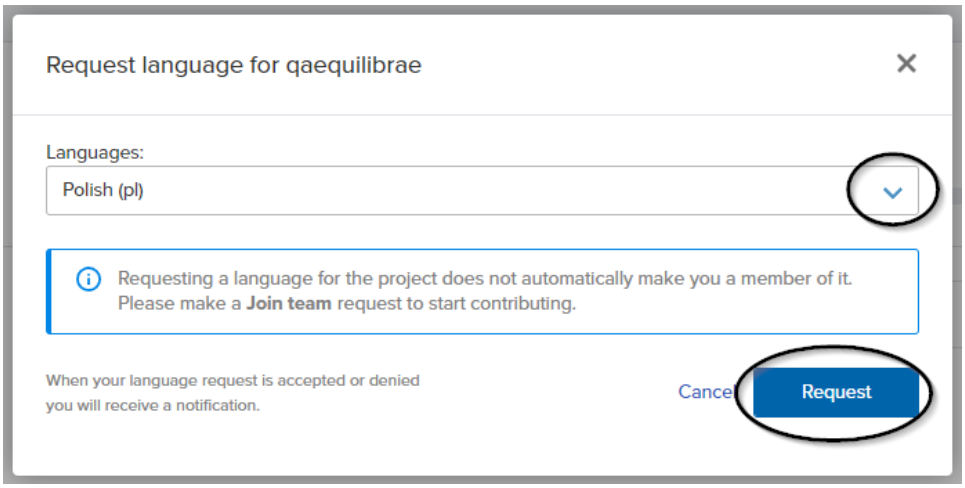
For more information, we strongly encourage you to check Transifex official documentation in their website.

B.3 Suggesting a new translation language

If your language is unavailable and you want to contribute, you can request it directly to the AequilibraE Team! In your project screen, look for the **Request a language** button on the page.



Select your language and create a request. As soon as possible, we'll approve the creation of the language, and you can start translating!



B.4 Translation tips!

Last but not least, here are some recommendations for translation, many of them borrowed / adapted from QGIS documentation.

1. Be aware to use exactly the same (number of) special characters of the source text such as `, ``, *, **, ::, {}. These contribute to the cosmetics of the information provided.
2. Do not begin nor end the text hold by special characters or tags with a space
3. Do not end the translated strings with a new paragraph, otherwise the text will not be translated during the HTML generation.
4. Prefer using `format` over `f`-strings.

APPENDIX

C

SUPPORT & SPONSORS

C.1 Support

AequilibraE is developed by a small but dedicated team of professionals with limited funding.

C.1.1 Paid support

Paid support for AequilibraE is offered by Outer Loop Consulting, an Australia-based consulting company, with support available in English, Portuguese, German & Spanish.

All support is offered in prepaid packages of a minimum of 10h of consulting by phone, e-mail or Microsoft Teams at a fixed rate of USD 150/h.

To acquire a paid support package for AequilibraE, please e-mail aequilibrae@outerloop.io

C.1.2 Free support

There are two mechanisms to obtain free support that also allow you to check whether your question was already asked in the past, with the first option being the most often used as of December/2023:

1. Joining the AequilibraE Google Group and sending your question there.
2. Posting your question to GIS StackOverflow using the `#aequilibrae` tag;

Please note that all questions and answers in both forums are public.

When sending your question, be as specific as possible, providing screenshots, details on the QGIS version, and as much detail as you can!

C.2 Sponsors

Sponsoring AequilibraE's maintenance and new features is the best way to guarantee that the bugs you need fixed and the features your projects require are incorporated quickly into the software. Please contact aequilibrae@outerloop.io for sponsoring opportunities.

Maintenance Sponsors	
2024	ADEME
2023	Outer Loop Consulting - La Fabrique des Mobilités - EGIS France
2022	Outer Loop Consulting
2021	Outer Loop Consulting
2016	Instituto de Pesquisa Econômica Aplicada

Feature Sponsors	
2023	GTFS import - La Fabrique des Mobilités
	Transit assignment - EGIS France
	Select Link Analysis - Outer Loop Consulting

APPENDIX

D

CITE US!

If you're using QAequilibraE in a scientific publication, we would appreciate citations to the following paper:

Camargo, P. (2015) AequilibraE: a free QGIS add-on for transportation modeling. Foss4g North America

INDEX

I