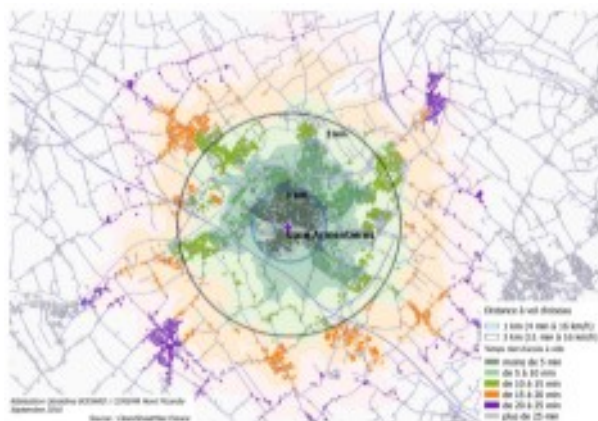


# Multimodal transport analysis with Qgis plugin « networks »



# Sommaire

A- Create an accessibility map (walking, cycling, car or public transport) from a multimodal network.....	4
Warning.....	4
1. Install « Networks » plugin.....	5
Part 1 : Data preparation – Multimodal network development.....	6
2. Get network data from OSM.....	6
3. Clean/cut OSM data for routing.....	8
4. Add and update fields to the cleaned road layer.....	9
5. Make motorways impassable.....	11
6. Opposite direction preparation.....	12
7. Build road graph.....	14
8. Generation of the Musliw individual network file.....	15
9. Suppress isolated nodes (nodes which are unreachable/ non connected to the main graph).....	16
10. Prepare GTFS – From a raw GTFS (not prepared for Musliw).....	18
11. Generation of a PT network.....	19
12. Import PT stops in QGIS.....	20
13. Generation of the connectors.....	21
14. Generation of the multi-modal network.....	22
Part 2 : generation of the territorial accessibility map.....	24
15. Matrix generation for Musliw.....	24
16. Musliw parameters file.....	24
17. Musliw computation.....	25
18. Update ti tj.....	25
19. Linear Interpolation.....	27
20. Isovalue polygons.....	28
Partie 3 : To go further.....	29
21. Compute population inside iso-value polygons.....	29
B- Production of a PT services offer (example : french regional railways offer).....	30
1. Get the theoretical offer on the SNCF open data website.....	30
2. GTFS pre-processing.....	32
3. GTFS import in Qgis.....	33
4. View stops.....	34
5. Show stations names.....	35
6. Produce a proportional symbol size map.....	37
7. View the number of services by link.....	37
8. Widths proportional to the number of circulations.....	39
9. Label the number of services.....	41
C- Production of a traffic map.....	44
1. Get traffic volumes.....	45
2. Define proportional widths.....	47
3. Set stroke width with an expression.....	47
4. Show traffic volumes flows.....	49
D- Create a territorial accessibility map on a time period and/or from/to several points.....	53
1. Create the accessibility map and the area of influence map.....	53
1.1. Matrix creation.....	53
1.2. Customize Musliw computation parameters.....	55
1.3. Musliw computation.....	56
1.4. Compute travel times aggregation.....	56
1.5. Update aggregated ti tj.....	58
1.6. Update of the links pole.....	58
1.7. Linear interpolation.....	59
1.8. Drawing of isovalue polygons and areas of influence.....	60

E- Simplified accessibility map.....	62
1. Installation.....	62
2. Settings.....	62
3. Making the map.....	64
4. Tips for setting parameters.....	65
F- Create an accessibility map based on a grid.....	68
1. Advantages and drawbacks.....	68
2. Creating the grid.....	68
3. Cut the grid according to the outline of the study area.....	69
4. Create a zone identifier.....	70
5. Make a copy of the grid in centroids.....	70
6. Create the grid - road network connectors.....	71
7. Insert connectors into the multi-modal network.....	73
8. Create an adapted parameter set and launch the calculation.....	74
9. Visualize the results.....	74
G- Creation of gravity accessibility maps with or without spatial and/or modal competition.....	78
1. Presentation.....	78
1.1. Introduction.....	78
1.2. Principle.....	78
1.3. Function used for accessibility by gravity.....	78
1.4. Required data.....	79
2. Gravity indicators with or without spatial competition.....	79
3. Gravity indicators with spatial and modal competition.....	81
H- Using processing models.....	85
1. Introduction.....	85
2. Installation.....	85
3. Running the model.....	86
4. Description of model outputs.....	87
I- Musliw documentation.....	89
1. The « network » file.....	89
1.1.1 Possible use of the type of link.....	91
1.1.2 Example of coding of individual LINKS taking into account time periods and calendar.....	91
2. The penalties and transfers definition file.....	93
3. The «Matrix» file.....	94
3.1.1 Standard specifications.....	94
3.1.2 Advanced specifications.....	95
4. Computation time optimization.....	97
5. Computation procedure.....	97
5.1. Basic case.....	97
5.2. Advanced settings.....	100
5.3. The “results” files.....	100
6. <FILENAME>_OD.TXT.....	101
7. <FILENAME>_TEMPS.TXT.....	103
8. <FILENAME>_AFF.txt.....	105
9. <FILENAME>_CHEMINS.TXT.....	106
10. <FILENAME>_NOEUDS.TXT.....	108
11. <NOM_FICHIER>_LOG.TXT.....	108
12. <FILENAME>_SERVICES.TXT.....	110
13. <FILENAME>_TURNS.TXT.....	111

# A- Create an accessibility map (walking, cycling, car or public transport) from a multimodal network

## Warning



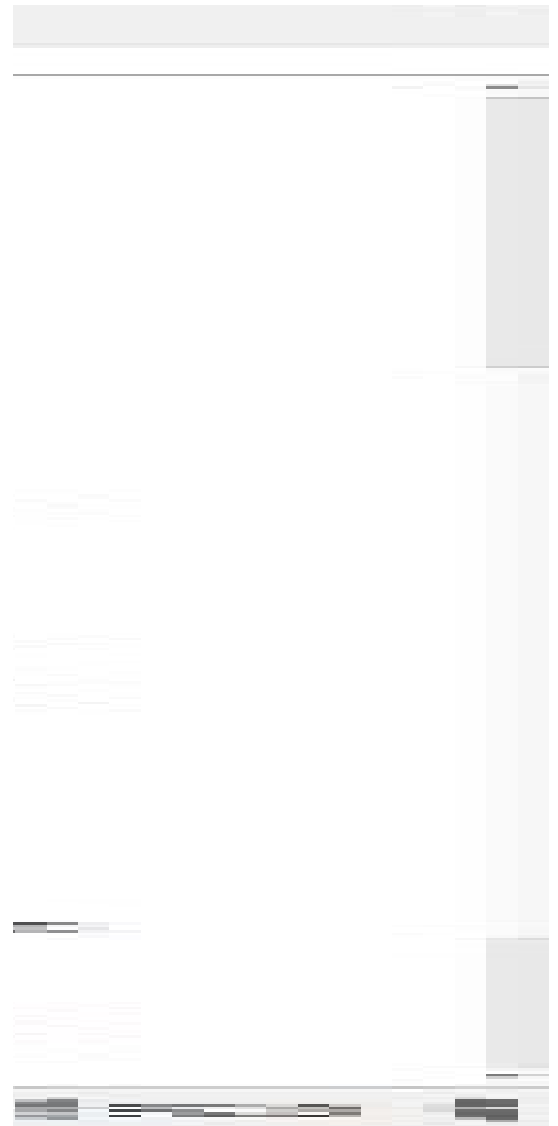
**Do not put blank or accent for directory and / or table names. Example: Armentieres / voirie\_decoupe\_2senss**

Most scripts used for:

- modify the different road layers, nodes, ...
- create different networks (cycling, walking, transportation)

can be found in the processing toolbox, under the Networks tab.

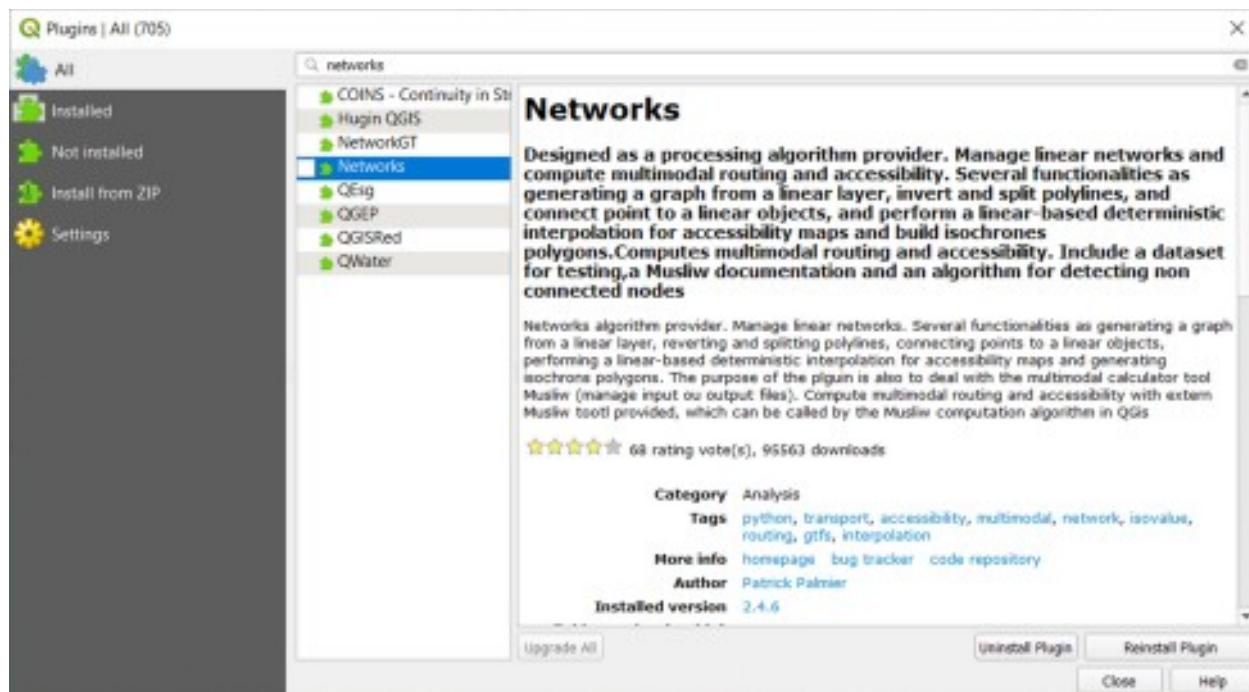
The plugin networks is installed from the Qgis  
« Plugins/Manage and install,,, » Menu



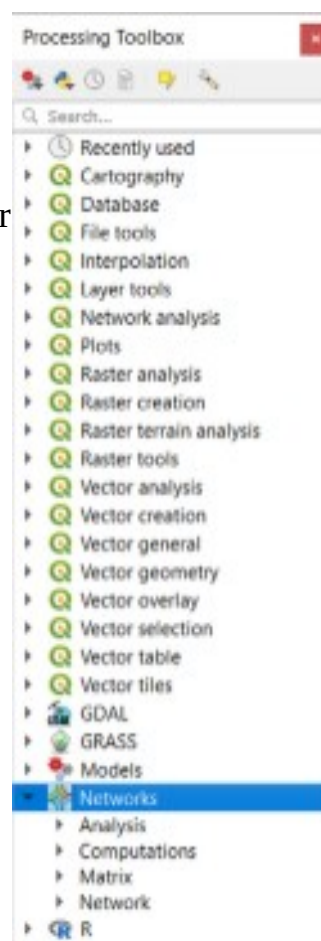
## 1. Install « Networks » plugin

In Qgis, **download and activate the “networks”** plugin which add a new algorithm provider in your processing toolbox.

In Plugins/Manage and install Plugins menu, select Networks and install it.



The “networks” algorithm provider is added to your processing toolbox



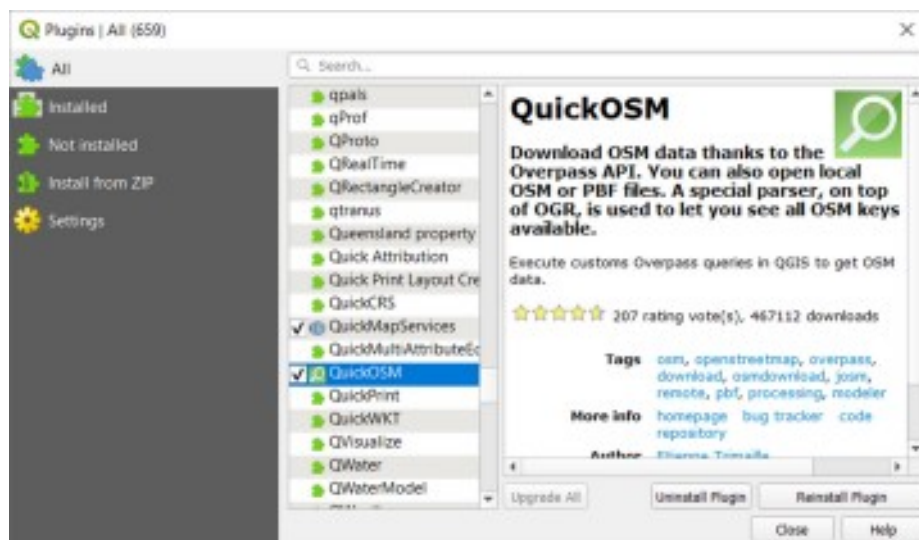
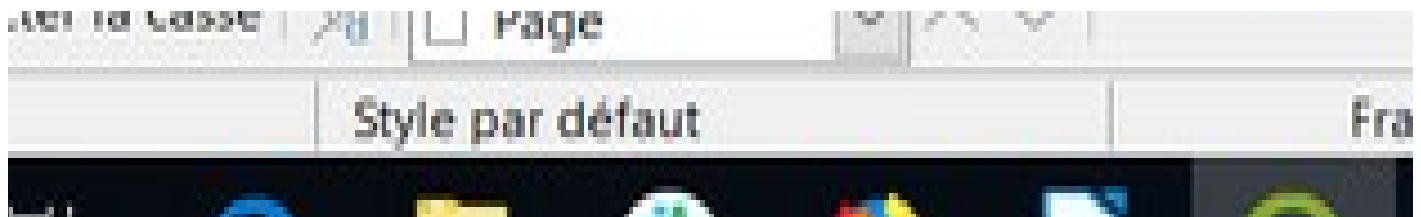
# Part 1 : Data preparation – Multimodal network development

chapters 1 and 2 are only for users who do not have a road network and who wish to generate it from OSM. Otherwise you can go directly to 3

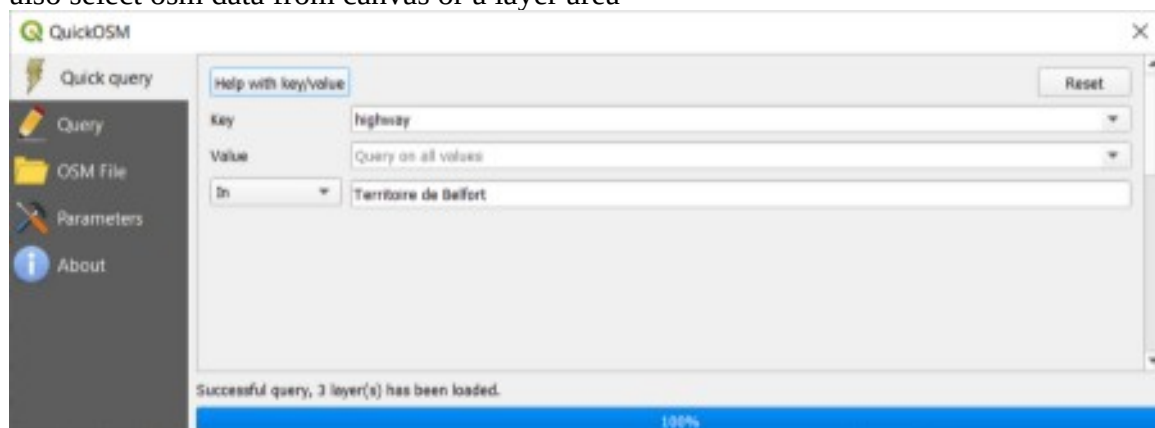
## 2. Get network data from OSM

First, you can install the QuickOsm plugin

Menu : **Plugins** / Manage and Install Plugins : **Install QuickOSM**

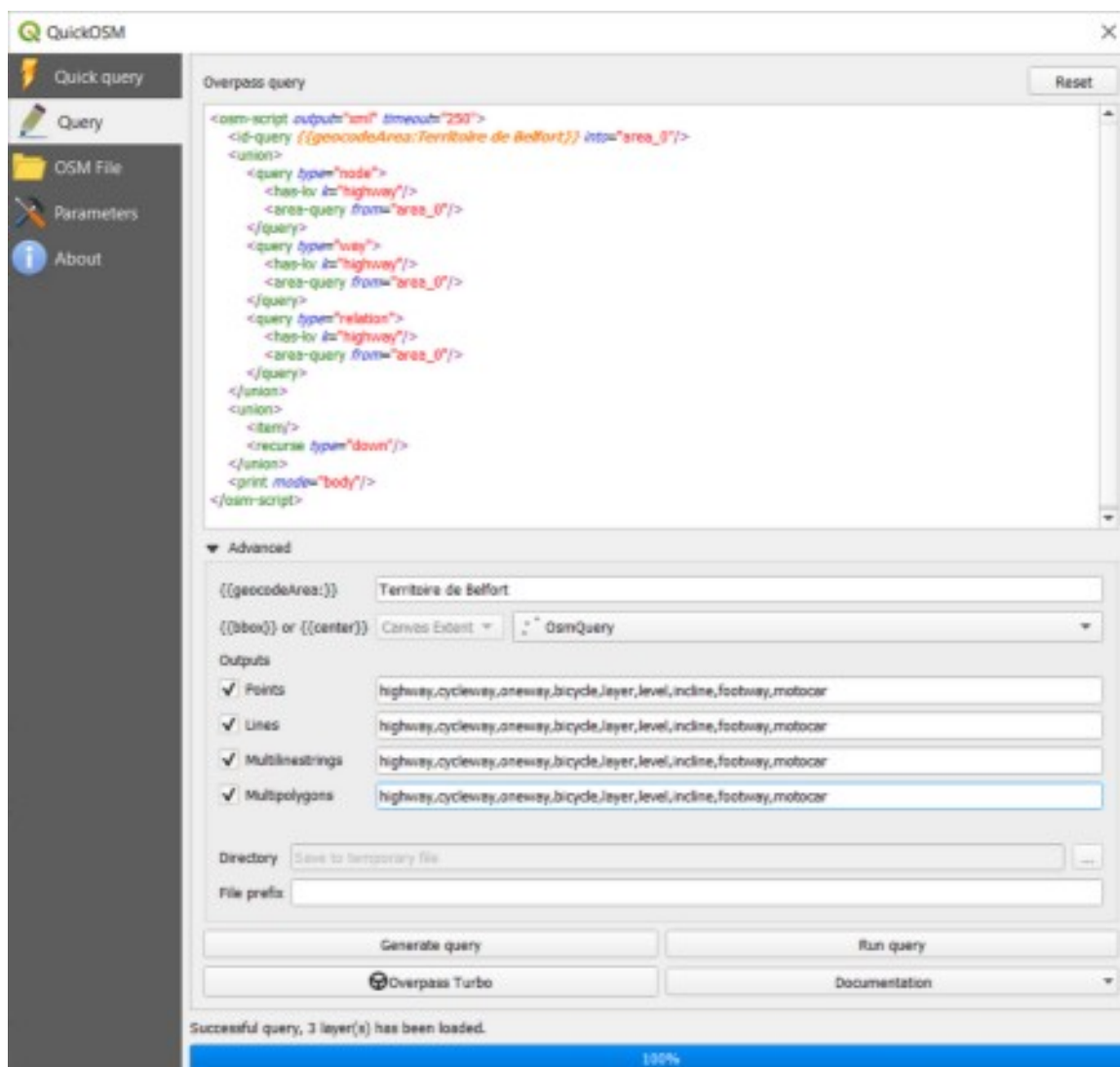


For road network, you can select highway feature, and in this example data from Territory of Belfort, You can also select osm data from canvas or a layer area



You can also select in « Advanced » only « outputs » OSM tags you need to have in your layers, Otherwise you'll have as many fields as different tags in your OSM query data,





You can use either XML query language or OQL query as below. With the OQL query below, you can select roads that are at the border of the area too,  
Don't forget to increase « timeout » and « maxsize » for huge files

### Query example to obtain data for Belfort Territory :

```
[timeout:2500]
[maxsize:20000000000];
area[admin_level='6'][ref='90']->.area;
rel(pivot.area)->.rel;
(
node[highway](area.area);
node[highway](r.rel);
way[highway](area.area);
way[highway](r.rel);
rel[highway](area.area);
rel[highway](r.rel);
);
(._;>.);
out;
```

**Then :**

Run Query

Close

Save « Osmquery » :**Change SCR (ex :EPSG 2154 for France)**

Example : DEPT90\_voirie

**Save the query to load other networks later.**

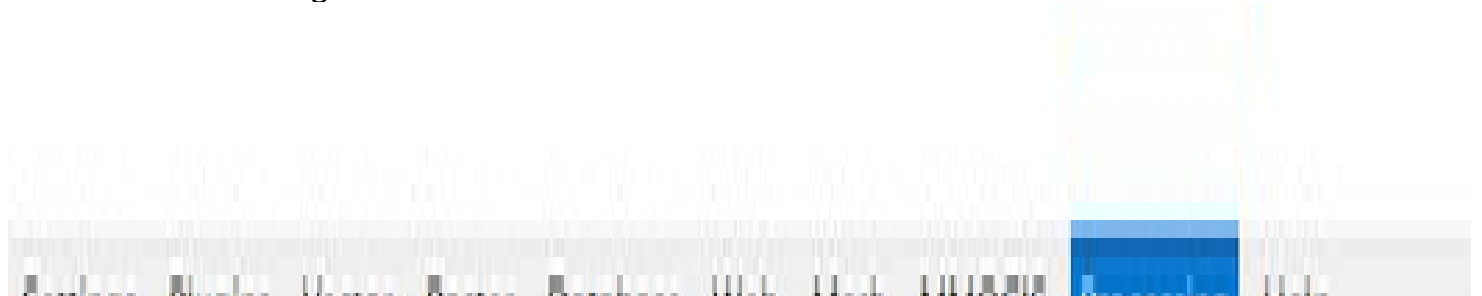
### **3. Clean/cut OSM data for routing**

In order to use OSM data for routing you need to do pre-processing operations. For example you need to cut arcs at intersections.

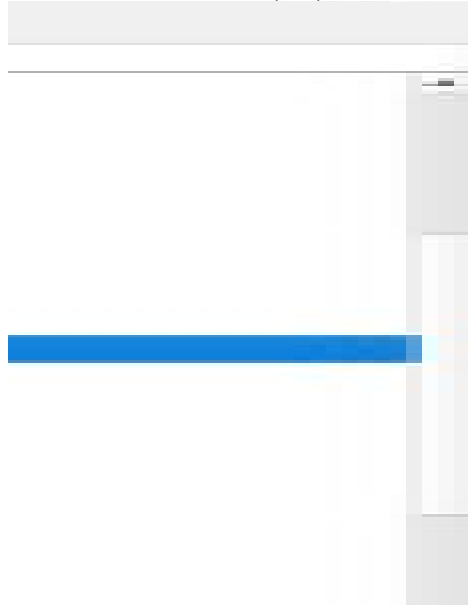
For such tasks, you can use Grass Processing Provider

**Show the processing toolbox (if not shown)**

**Tab : Processing/ToolBox**



**Use Grass / Vector (v.\*) / v.clean algorithm**





Grass / Vector (v.\*) / v.clean Commande :

Layer to clean : select the layer to clean : **DEPT01\_Voirie**

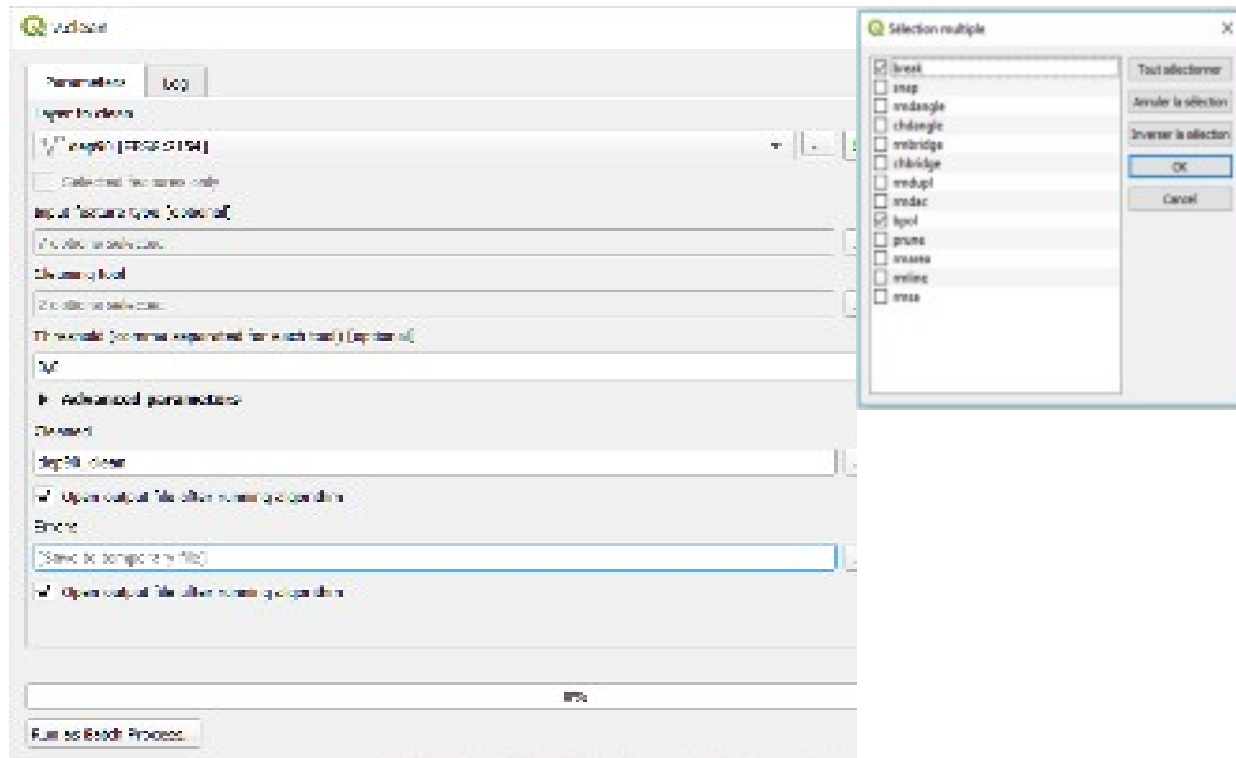
Cleaning tools : choose the tools : **break**

Threshold (comma separated for each tool) : **0**

Cleaned : save result in a layer : **dep90\_clean**

Errors : save errors file :

Run



Save the clean table (cleaned road layer) and add in Qgis : **change SCR (ex : EPSG 2154)**

Exemple : dep90\_clean

## 4. Add and update fields to the cleaned road layer

From now, all algorithms used are those from the « networks » provider

Algorithm : **Update field**

*Update field algorithm is like « field calculator » in the Standard toolbox, but you can also put a filter for update,*

As we need to create and update several fields, we can run the script in « batch mode »

**Right Click on « update field » algorithm and select « Execute as batch process » (save query if you want to use it again later)**

Batch Processing - Update field

Parameters Log

	Layer	Field	Type	Size	Precision	Filter	Formula
1	AutoFill...	AutoFill...	AutoFill...	AutoFill...	AutoFill...	AutoFill...	AutoFill...
2	...	*	String	20	15	*	*
3	...	*	String	20	15	*	*
4	...	*	String	20	15	*	*
5	...	*	String	20	15	*	*
6	...	*	String	20	15	*	*
7	...	*	String	20	15	*	*

☐ Load layers on completion

0%

Run as Single Process...

Run Cancel Close

**Layer :** select the road layer to update

**Fields to create and/or update :**

Field : **Sens**

Type : **String**

Size : **1**

Precision : **0**

Filter :

Formula : **'1'**

Field : **Longueur**

Type : **Double**

Size : **15**

Precision : **5**

Filter :

Formula : **\$length**

Field : **Diffusion**

Type : **String**

Size : **1**

Precision : **0**

Filter :

Formule : **'3'** (allow spread in both direction)

Champ : **Impasse**

Type : **String**

Taille: **1**

Précision : **0**

Filtre :

Formula : **'0'** (passable)

Field : **Cycling\_time or Walk\_time**

Type : **Double**

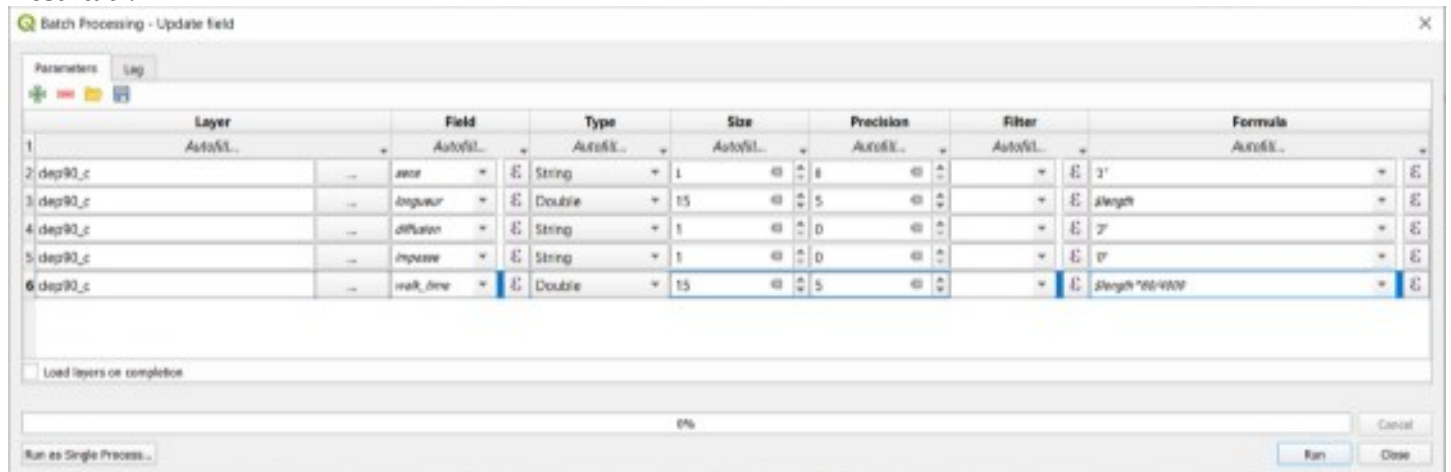
Taille : **15**

Precision : 5

Filter :

Formula :  $\$length * 60 / 16000$  (cycling : 16 km/h) ou  $\$length * 60 / 4000$  (walking : 4 km/h)

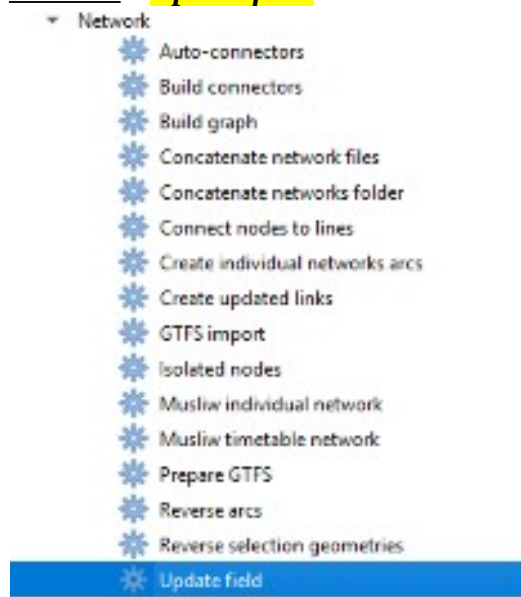
### Résultat :



The field « **filter** » is used to update a subset of an existing field

## 5. Make motorways impassable

### SCRIPT : **Update field**



*Right Click on « update field » algorithm and select « Execute as batch process » (save query if you want to use it again later)*

#### Fields to update

Field : **Sens** (choose a field name for flow direction)

Type : **keep default value**

Size : **keep default value**

Precision : **keep default value**

Filter : **highway in**

**('motorway','motorway\_link','trunk','trunk\_link')**

Formula : **'0'**

Field : **Diffusion**

Type : **laisser le champ affiché par défaut**

Size : **laisser le champ affiché par défaut**

Precision : **laisser le champ affiché par défaut**

Filter : **highway in ('motorway','motorway\_link','trunk','trunk\_link')**

Formula : **'0'**

Field : **Impasse**

Type : **keep default value**

Size : **keep default value**

Precision : **keep default value**

Filter : **highway in ('motorway','motorway\_link','trunk','trunk\_link') and "layer" is Null**

Formula : **'3'** (allowed from both sides of the motorway)

Field : **Impasse**  
 Type : **keep default value**  
 Size : **keep default value**  
 Precision : **keep default value**  
 Filter : **highway in ('motorway','motorway\_link','trunk','trunk\_link') and "layer" in ('0')**  
 Formula : '3'

Field : **Sens**  
 Type : **keep default value**  
 Size : **keep default value**  
 Precision : **keep default value**  
 Filter : **foot in ('no')**  
 Formula : '0'

Field : **Diffusion**  
 Type : **keep default value**  
 Size : **keep default value**  
 Precision : **keep default value**  
 Filter : **foot in ('no')**  
 Formula : '0'

Field : **Impasse**  
 Type : **keep default value**  
 Size : **keep default value**  
 Precision : **keep default value**  
 Filter : **foot in ('no')**  
 Formula : '3'

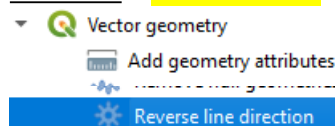
**Result :**

Champs	Chang	Type	Table	Precision	Formule	Formule
DEPT_04_Claire_Viville_Quatre	str lane	10	10	11	highway in ('motorway','motorway_link','trunk','trunk_link')	3
DEPT_04_Claire_Viville_Quatre	str Diffusion	10	10	11	highway in ('motorway','motorway_link','trunk','trunk_link')	3
DEPT_04_Claire_Viville_Quatre	str Impasse	10	10	11	highway in ('motorway','motorway_link','trunk','trunk_link') and 'layer' in ('0')	3
DEPT_04_Claire_Viville_Quatre	str Sens	10	10	11	highway in ('motorway','motorway_link','trunk','trunk_link') and 'layer' in ('0')	3
DEPT_04_Claire_Viville_Quatre	str lane	10	10	11	foot in ('no')	0
DEPT_04_Claire_Viville_Quatre	str Diffusion	10	10	11	foot in ('no')	0
DEPT_04_Claire_Viville_Quatre	str Impasse	10	10	11	foot in ('no')	3

## 6. Opposite direction preparation

It is better for modeling to have only one-way arcs, as travel times and characteristics could be different on each direction. For example travel times for cycling are very different downhill and uphill. For cars drivers, many streets are only one-way. To do this, it is necessary to reverse and duplicate arcs

## SCRIPT : **Reverse arcs**



network : **Open the network layer to reverse**

Add to network (**if checked : reverse arcs are added directly to the initial layer**)

reverted network: name of the reverse arcs layer

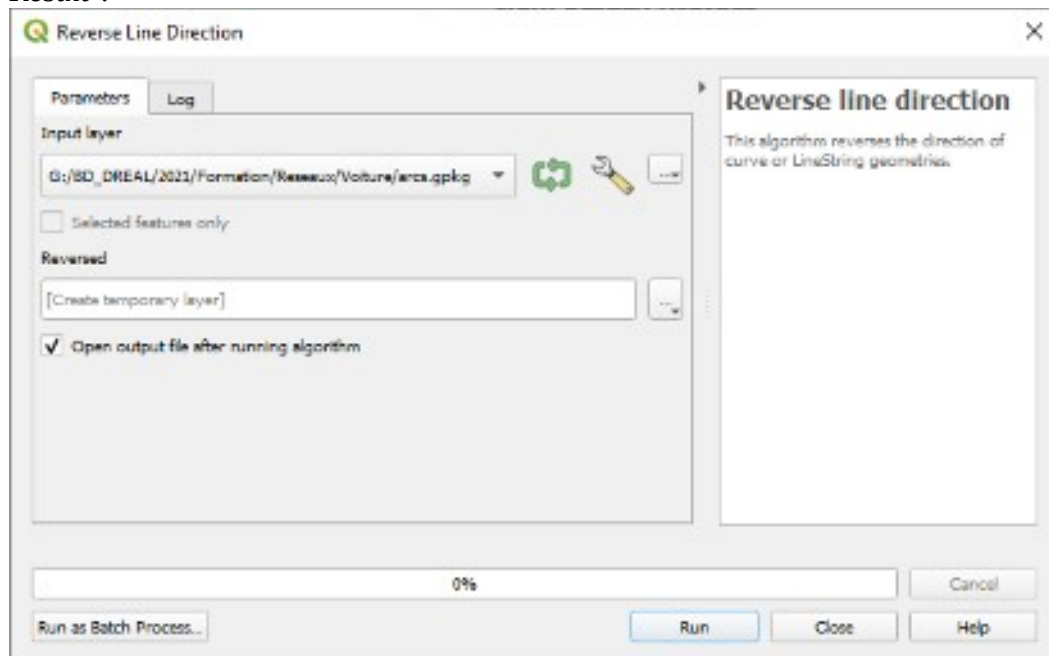
Open output file after running algorithm Qgis.

I advise users to create in first two separate layers, one for original arcs and another for the reserve arcs layer, in order to manage easily flow directions with “oneway” tag for cars and “cycleway” tag for bikes in particular.

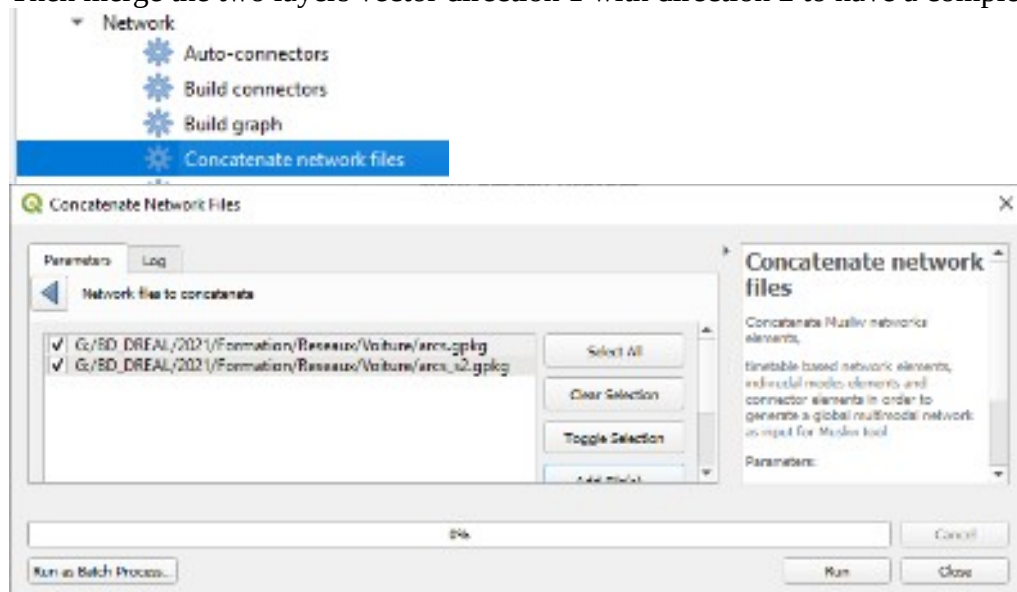
For example, an arc with oneway=’yes’ in the original layer should have “direction”=’1’ (allowed) in the original layer and “direction”=’0’ (prohibited) in the reverse layer

Please look at <https://wiki.openstreetmap.org/wiki/Key:oneway> for more details on one wayslinks and <https://wiki.openstreetmap.org/wiki/Key:cycleway> for cycling facilities.


Result :



Then merge the two layers vector direction 1 with direction 2 to have a complete network



Only for bike network« **need to be adapted depending on territories** »

 oneway= 'yes' and highway in ('primary', 'secondary', 'tertiary', 'primary link', 'secondary link', 'tertiary link') and cycleway is Null  
update field "Sens" (or direction) : '0'

From update fields : 

example for reverting cycling slopes.

Update field « incline » to revert slopes (*values to adapt depending on slopes values*)

Case when « incline »= 'up' then 'down'

when « incline »= 'down' then 'up'

when « incline »= '-8%' then '8%'

when « incline »= '10%' then '-10%'

when « incline »= '8%' then '-8%'

when « incline »= '-10%' then '10%'

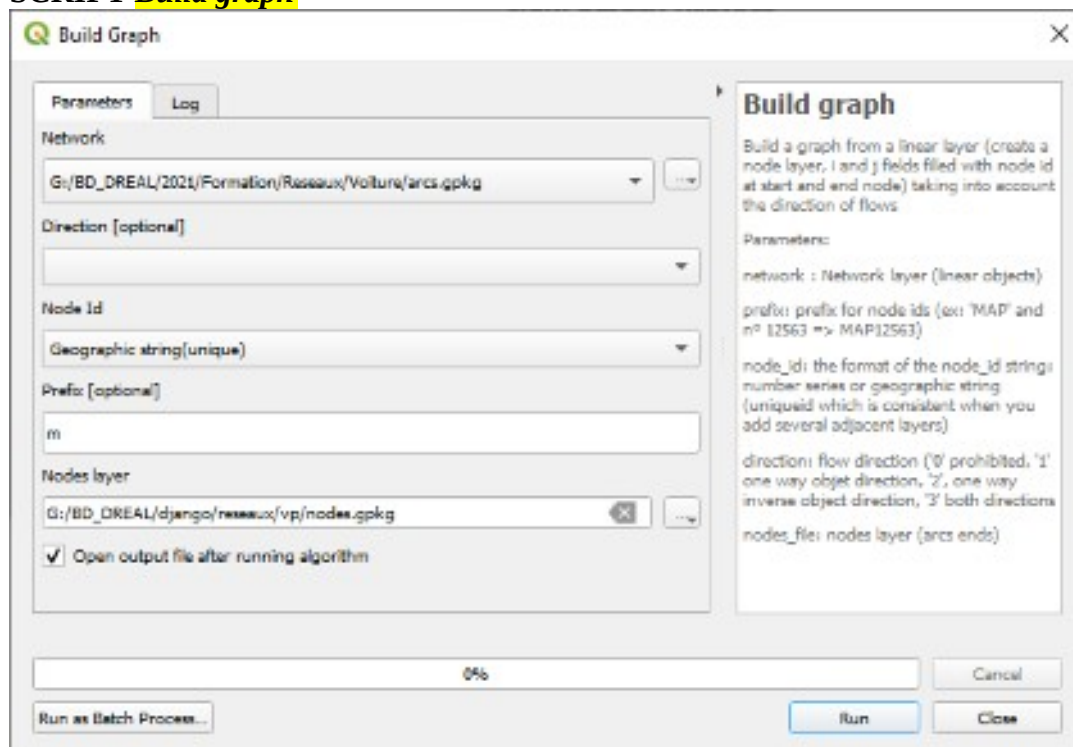
else Null

End

## 7. Build road graph

For shortest paths computing, we need to build a road graph from the network layer. The graph generation consists in the generation of a node layer and the addition for each arc of a "from node" and a "to node" field corresponding to the nodeId of the node layer

### SCRIPT **Build graph**



Network : **Network layer (arc layer original + reverse arcs layer)**

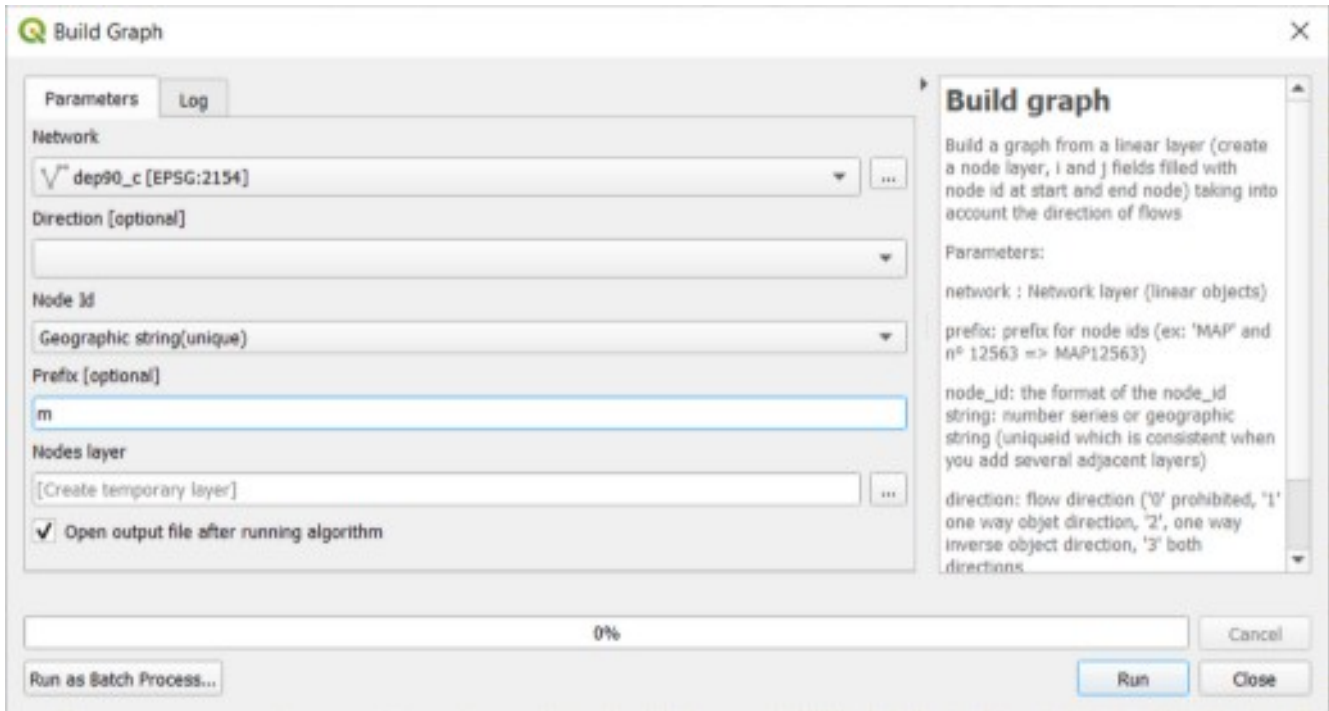
Node id : **geographic string(unique)**

Prefix : **m** (in order to differentiate arcs from for different modes)

Nodes layer : **Name of the generated nodes layer**

Run

**Result :**



## 8. Generation of the Musliw individual network file

The shortest path algorithm is working with networks and matrices ascii files in the Musliw file format (semi-column separated text files).

### SCRIPT **Musliw individual network**

Allow to generate a Musliw individual network from a Qgis road network layer

Load road network layer

Run “Musliw individual network” script

Road network : **road network layer**

Sens : **“sens” of direction field**

Time : **walk-time**

Length : **longueur**

i-node : **i**

j-node : **j**

Time category id : **keep default value**

Time period id : **keep default value**

Start time : **keep default value**

End time : **keep default value**

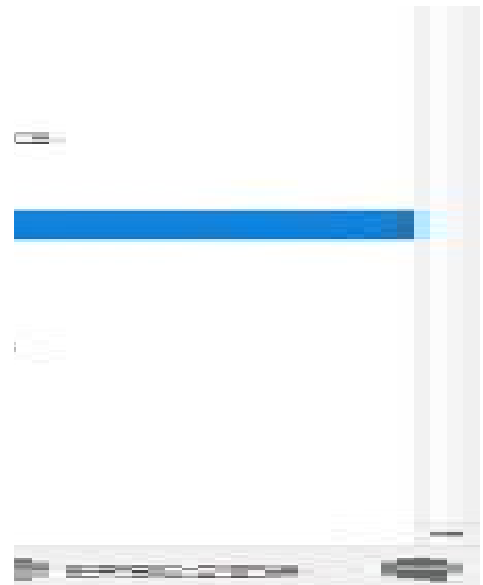
Calendar : **keep default value**

Arc label : **highway**

Mode : **m** (si marche)

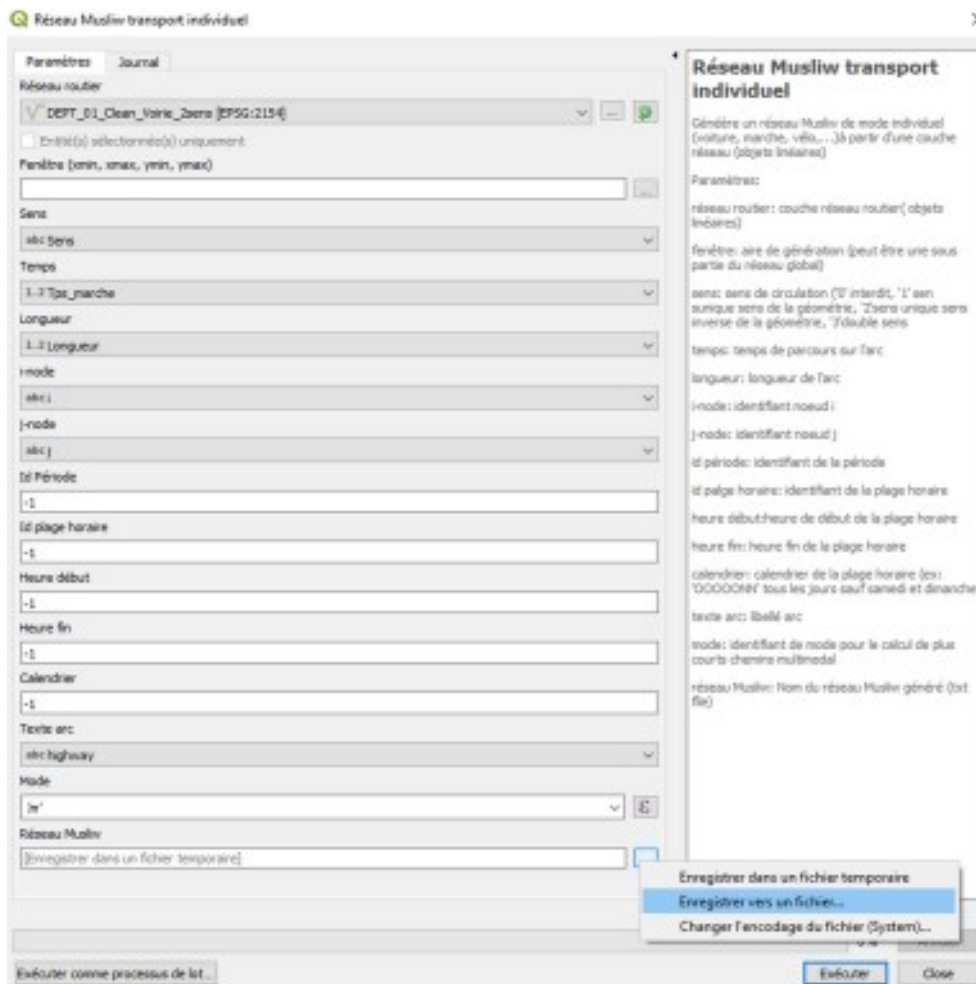
Musliw network: **save file (with .txt extension)**

**Run**



### Result





## 9. Suppress isolated nodes (nodes which are unreachable/ non connected to the main graph))

### SCRIPT **Isolated nodes**

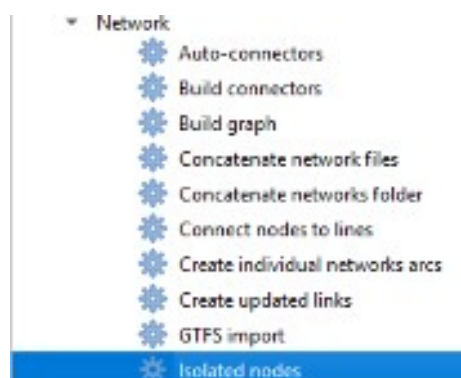
**Nodes: open nodes layer**

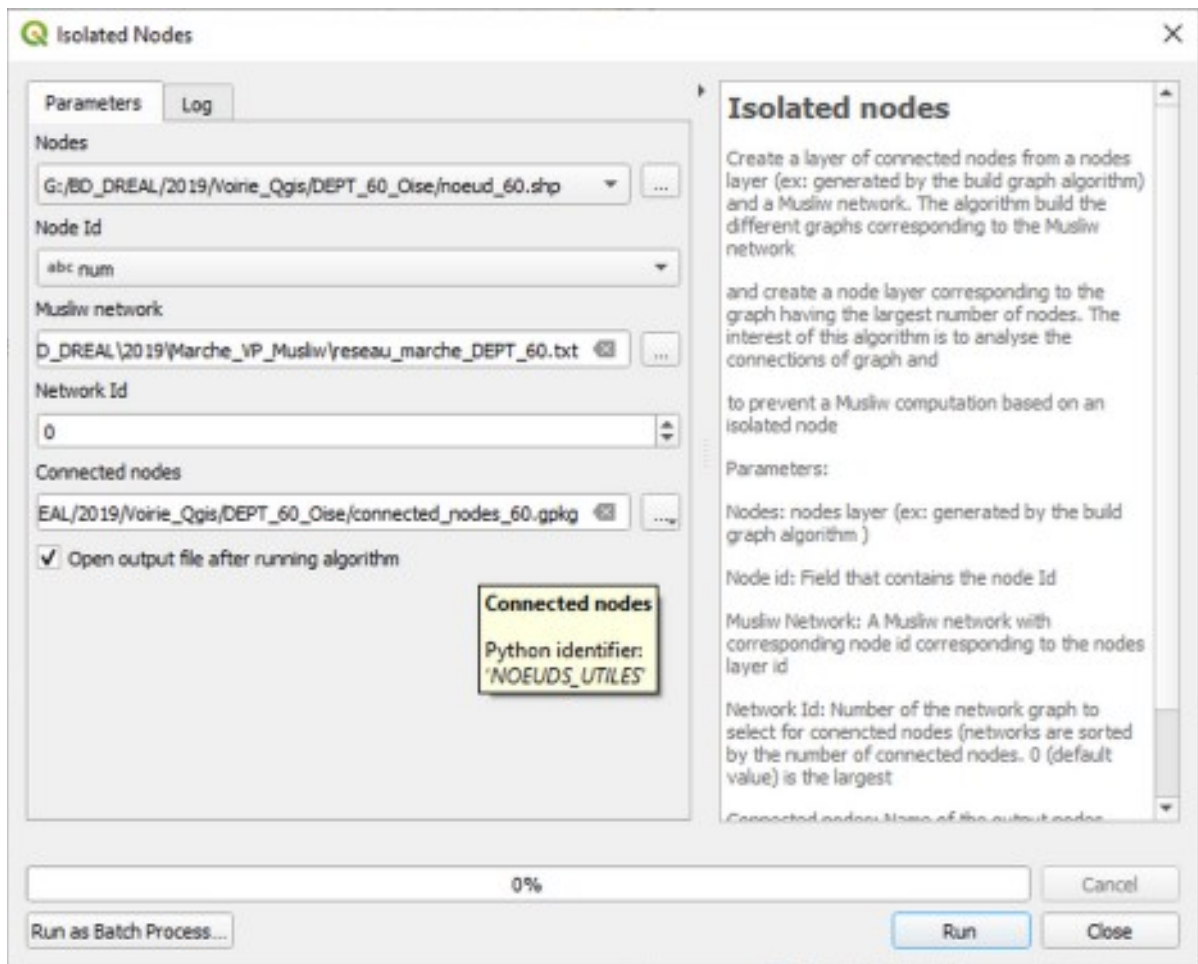
Node Id : **num**

Musliw network : **open walking network .txt**

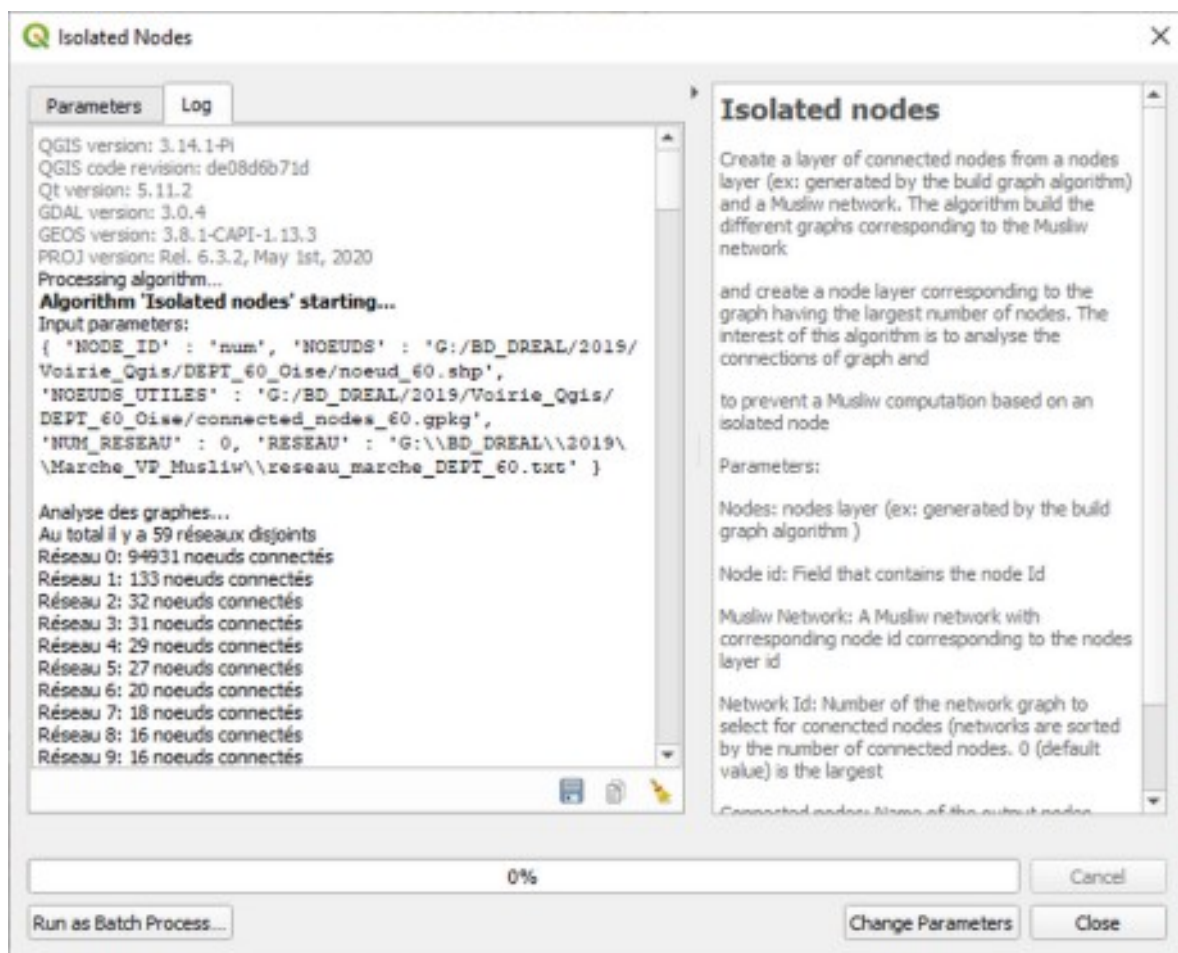
Netowrk Id : 0 in a first step (networks are sorted depending on the greater number of nodes, then 1, or even 2, etc. depending on the network analysis and the homogeneous chosen network

Connected nodes : **Save file as...**





In this example, there are for the network (ID 0), 94 931 connected nodes and 59 non-intersecting networks.



## 10. Prepare GTFS – From a raw GTFS (not prepared for Musliw)

### SCRIPT **Prepare GTFS**

Right click on the script/ **Execute as batch process**

GTFS source folder: Select the **source directory** (where are located the GTFS files)

Network Id : **AE** (AE example for Lille Airport bus line)

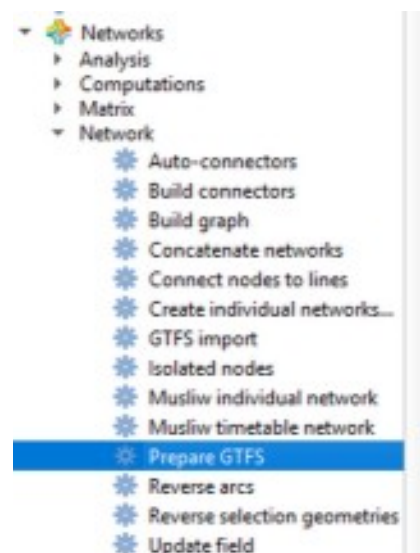
Extract stop\_id ? : **Non** (select Yes if you want to modify the stop\_id string with a formula)

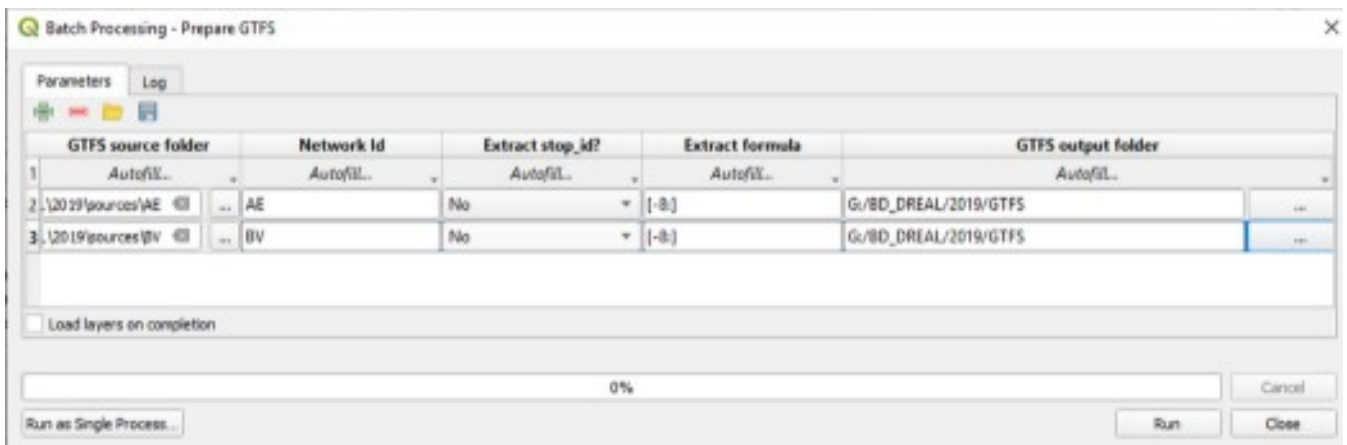
Extract formula : **write the formula: Ignored** when “Extract stop\_id?” is not checked

GTFS output folder : Save result in the **output folder**. Example « **GTFS** » folder

**Run**

**Result:**

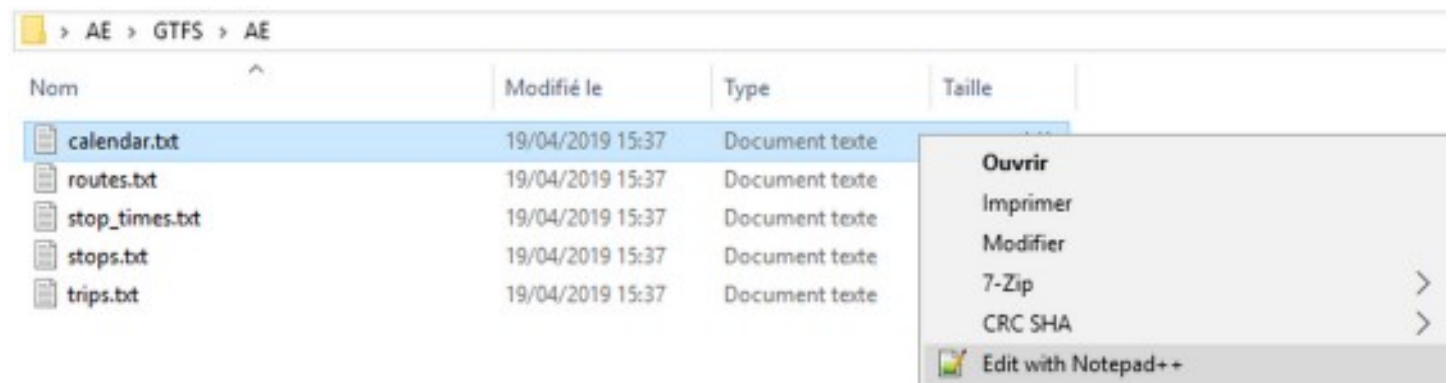




## 11. Generation of a PT network

As the GTFS extractions of the different PT networks have not been done at the same time, it is very important to select for each GTFS a period where timetables are available.

**In the result folder (saved GTFS files), edit (for example with Notepad++) the `calendar.txt` or `calendar_dates` file from each network** in order to choose a typical week (not during school holidays for example). Example : from 10/06/2019 to 16/06/2019



### SCRIPT Musliw timetable network

GTFS Folder : open the directory where the GTFS files are located

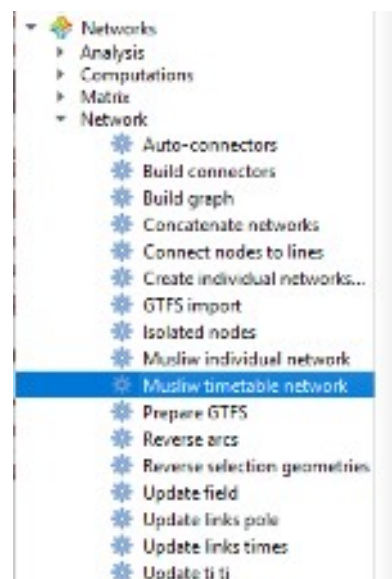
Calendar start: enter the first day of the period (**often a monday for a typical week period, but you can choose any day and any period duration**)

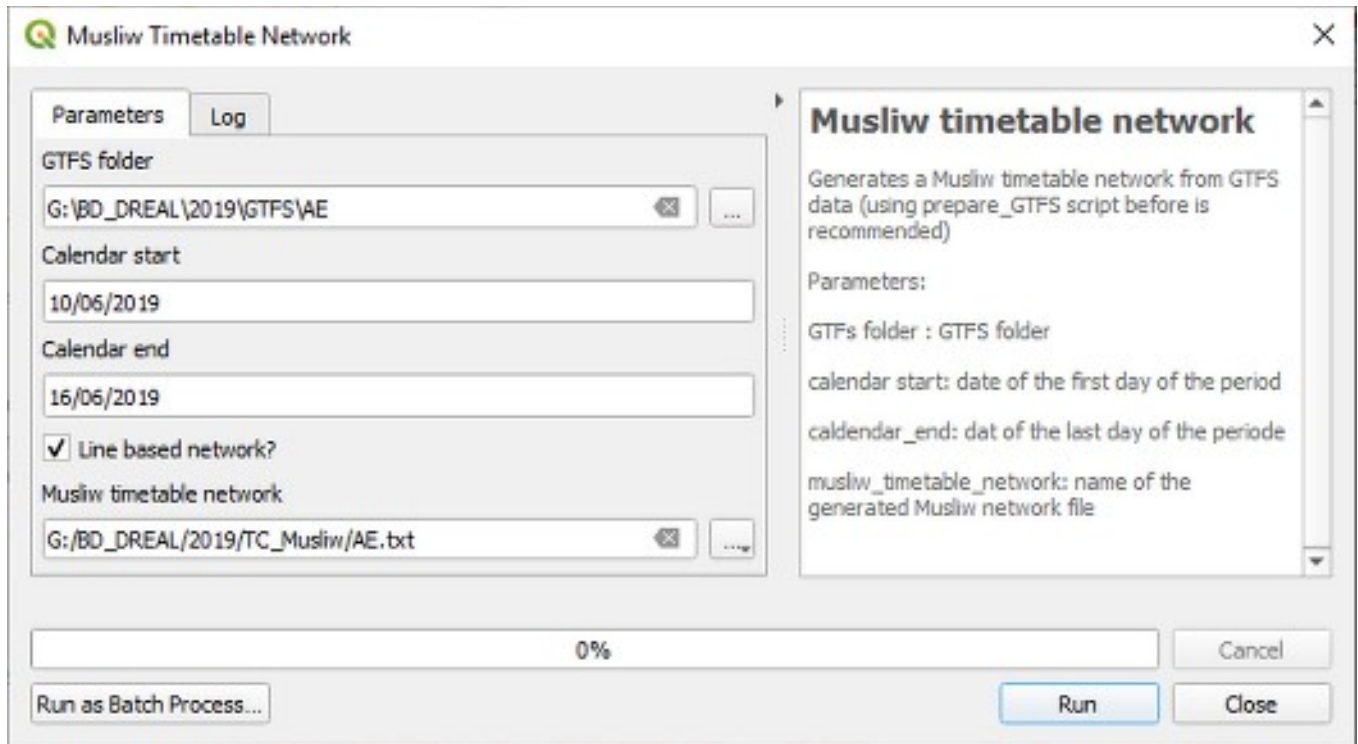
Calendar end: enter the last day of the period (**sunday for a typical week period**)

Lines based network: checked by default for most uses

Musliw timetable network : Save Musliwnetwork name as (**with .txt extension**) – Example AE.txt

Run





## 12. Import PT stops in QGIS

### SCRIPT **GTFS Import**

Right click / Execute as batch process (To save time in case of errors or in order to modify the batch process, don't forget to save the batch process script in Json format)

GTFS Folder : **Select the GTFS directory**

Calendar start : ex: **10/06/2019**

Calendar end : **ex: 16/06/2019** (corresponding to the week from Monday, June 10th 2019 to Sunday, June 16th 2019)

Start time : **keep default values (default values corresponds to the whole day. You can change them if you want to reduce the analyses period ex: peak hours)**

End time : **keep default values**

Table names : **AE** (put the network name, for example AE pour Lille Airport shuttle)

CRS : **let default values (EPSG 2154 for France)**

Encoding : « **utf8\_8\_sig** » **Keep the default value. If this doesn't work, try another encoding« utf8 » or « cp1252 » for example**

Output folder: **Select an output directory** (example "shp" (as output files are shapefiles))

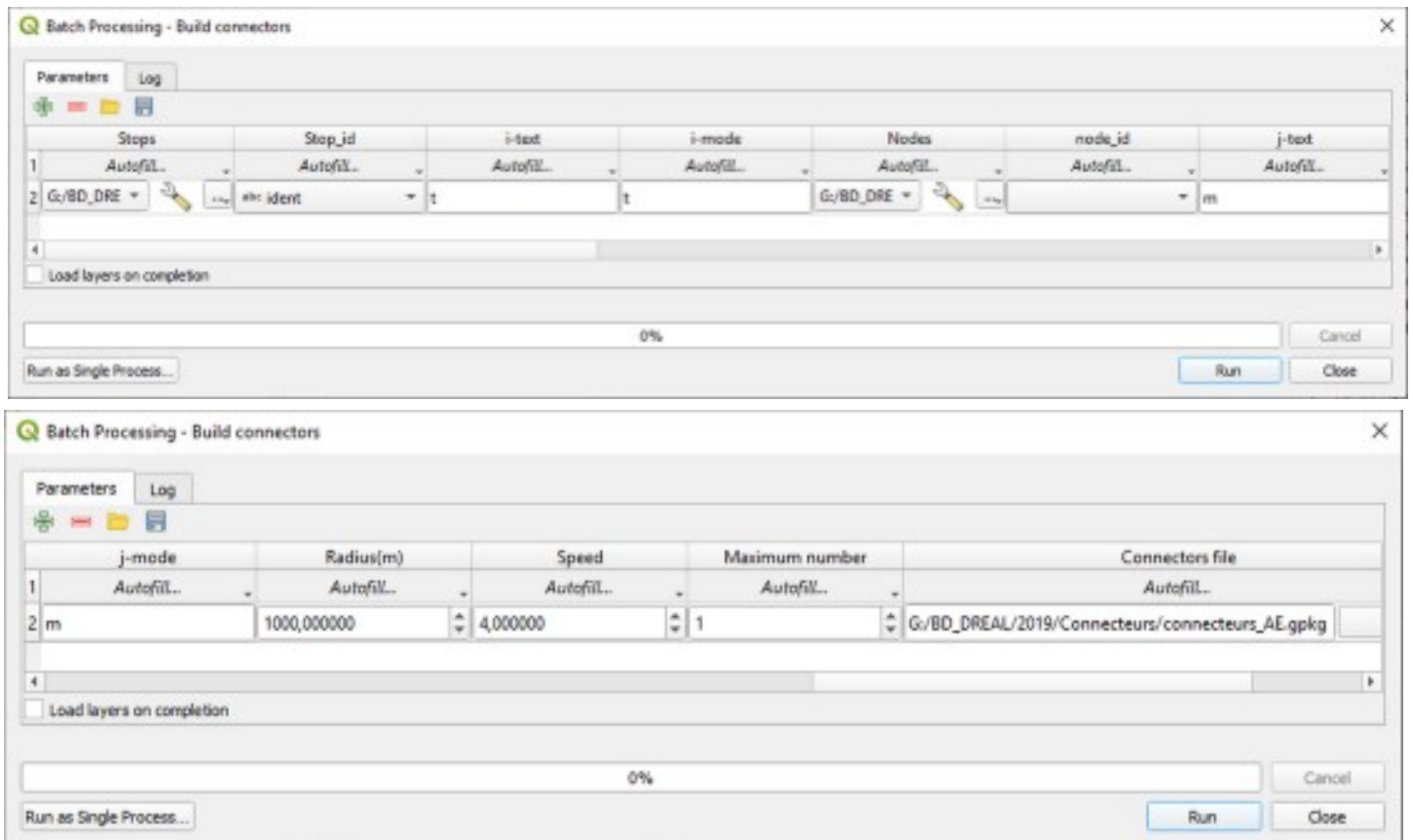
**Run**

**Results :**





stop id : **ident**  
 i-text : **t** (transit)  
 i-mode: **t** (TC) - mode id at i node  
 Nodes : **pedestrian nodes layer**  
 node id : **num**  
 j-texte j : **m** (walking)  
 j-mode : **m** (walking) – mode id at j node  
 search radius : **1000**  
 speed : **0** – **if 0 connector will be with no travel time**  
 maximum number : **1** – **only 1 connector per PT stop**  
 Connectors file : **connecteur\_reseau\_tc\_map** (name of the connectors layer)



## 14. Generation of the multi-modal network

### SCRIPT Concatenate networks

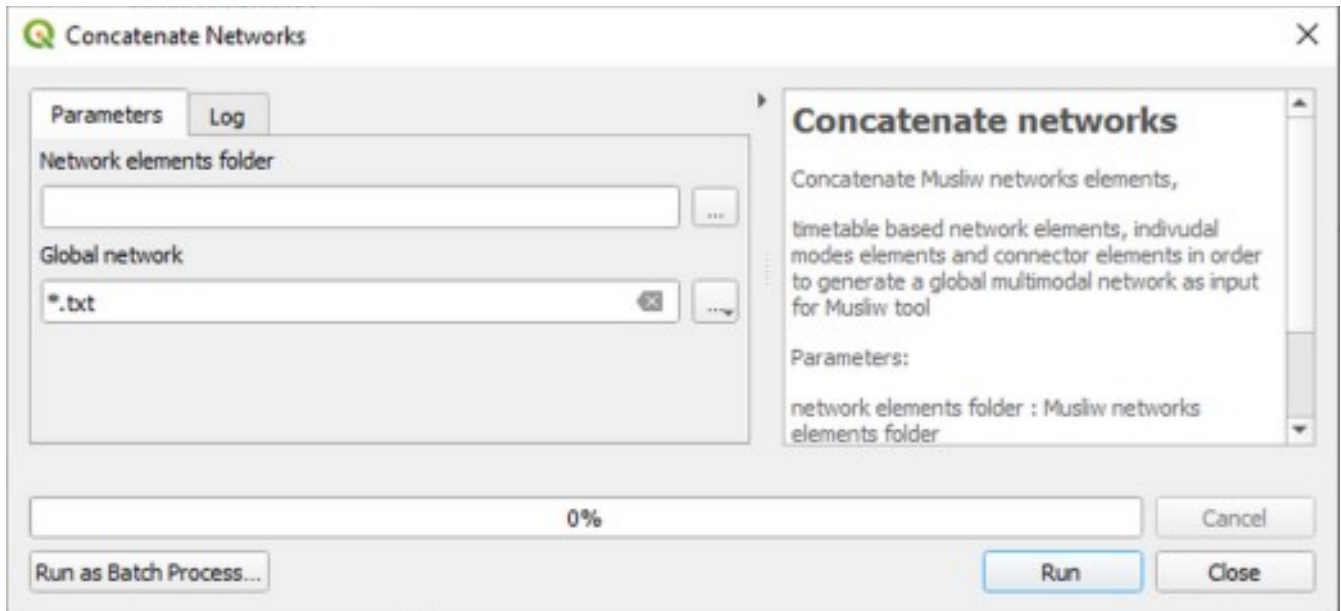
Concatenate different networks files in Musliw format (.txt extension)

Walking network  
 Connectors network  
 PT timetables networks  
 ...

Put each file that need to be used in a specific directory (example : folder « concatenate »  
 network elements foldere : **select the folder where .txt files to be concatenated are located**  
 global network : **save multimodal network as (with txt extension)**  
 Run

**Résultat :**





## Part 2 : generation of the territorial accessibility map

### 15. Matrix generation for Musliw

#### SCRIPT **Musliw simple matrix**

Nodes : **nodes layer**

Node Id : **num**

Trip start : **click directly in Qgis the start point**

Trip end : **select directly in Qgis the end point**

Demand : **number of passengers**

Day : **1 (0 is the first day from the period, 1 for a typical week is corresponding to a Tuesday))**

Time : time **format 00:00:00**

Departure/Arrival : **Choose**

Write mode : **write or append**

Musliw matrix : **save matrix as...**

Musliw Simple Matrix

Parameters Log

Nodes: G:\BD\_REAL\2020\nodes\HCP\_nodes.shp

☐ Selected features only

Node ID: num

Trip start: 707041.585501, 7061563.229501 [EPSG:2154]

Trip end [optional]: 705415.737243, 7059936.381243 [EPSG:2154]

Demand: 1,000000

Day: 1

Time: 00:00:00

Departure/Arrival: Arrival

Write Mode: Write

Musliw matrix: [Save to temporary file]

0%

Run as Batch Process...

Run Close

**Musliw simple matrix**

Generates a Musliw matrix from clicking on 2 nodes to define an origin(1st point)-destination(2nd point)

Parameters:

Nodes: nodes layer (corresponding to the routing arc layer)

Node id: Field that contains the node id

Trip start: Start point (origin)

Trip end: End point (destination)

Demand: number of passengers for assignment

Day: number of the day in the calendar (1 first day of the calendar)

Time: Time of earliest departure or latest arrival

Departure/Arrival: Departure (from Start point to end point forward) - Arrival (from end point to start point backward)

Write mode: Write erase filename if exists/ Append add trip definition at the end of the file

Musliw matrix: Musliw matrix name (text file with ";" separator)

### 16. Musliw parameters file



#### SCRIPT **Musliw parameters**

Individual mode speed factor : **0,25 for cycling**

Output links times : **Without timetables links(travel times file)**

[Enregistrer dans un fichier temporaire]

Musliw Parameters

Parameters Log

In vehicle weight: 1

Waiting weight: 1

Individual mode weight: 1

Boarding weight: 1

Individual mode speed factor: 1

Minimum transfer delay: 2

Maximum transfer delay: 60

Extra day duration: 0

Max. individual time budget: 60,000000

Maximum generalized time: 1500,000000

Toll weight: 0

Output filter [optional]:

Output links times? Without timetable links

☒ Prohibited U-turns?

☐ Output services?

☐ Output transfers?

☐ Output node times?

Algorithm scale: 30

Algorithm exponent: 2

0%

Run as Batch Process...

Run Close

**Musliw parameters**

Create a set of parameters useful for multimodal routing or accessibility computation with Musliw

Produce a parameter file (to be selected in musliw computation)

Parameters:

In vehicle weight: weight factor for travel time in vehicle (time based mode e.g bus, train, metro, airplane, ferry, ...)

Waiting weight: weight factor for waiting time

Individual mode weight: weight factor for individual modes travel times (e.g car, walking, cycling, ...)

Boarding weight: weight factor for boarding time

Individual mode speed factor: homothetic factor that apply to the individual travel times

Minimum transfer delay: minimum safety time for transfer (ex: 5 means that you should wait at least 5 minutes at a stop before the bus leaves)

Maximum transfer delay: maximum waiting time for transfer (ex: 60 means that you will not be able to take a train that leaves more than 60 minutes after you arrived)

Extra day duration: By default Musliw takes only into account timetables of the day selected in the matrix. If you can extend with the timetable of the day after (if departure) or the day before (arrival) you must enter 1

Max. individual time budget: The algorithm path will not explore shortest paths that have a individual time budget greater than this value

Maximum generalized time: The algorithm path will not explore shortest paths that have a total generalized cost greater than this value

Toll weight: weight factor for toll attribute

Output filter: specify the type of links you want in the reports (empty will output all links ex: metro,bus/train will export only metro, bus and train links)

Output links times: select what you want detailed output at link level (no, without or with timetable based links)

Prohibited U-turns: choose if you want to allow u-turns in shortest paths computation

Output paths: Check if you want to output detailed paths

Output services: Check if you want to have detailed outputs for each service

Output transfers: Check if you want to have detailed outputs on turns and transfers

Output nodes times: Check if you want to have detailed outputs at node level

Algorithm scale: Parameter of the shortest path algorithm

Algorithm exponent: the algorithm parameter exponent

Nb classes: The number of buckets of the shortest path algorithm

Parameter file: The output parameter file to be used in musliw computation

## 17. Musliw computation

Computations

Musliw computation

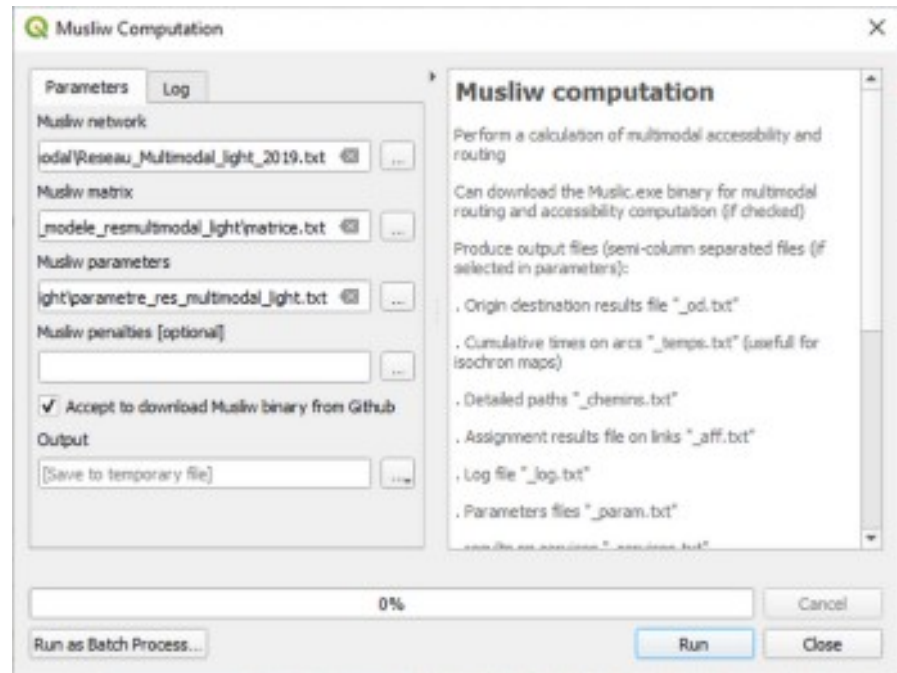
Musliw network : **reseau\_multimodal.txt**

Matrice Musliw : **matrix.txt**

Penalties Musliw (optional) : **penalties and transfers file**

Accept to download Musliw binary from

Github : **Must be if checked if Musliw has not ever being loaded.**



## 18. Update ti tj

SCRIPT **Update ti tj**

Network: **reseau\_voirie.shp**

Travel times file : **Choose the travel times output file generated by Musliw (\_temp.txt)**

Musliw time : **temps**

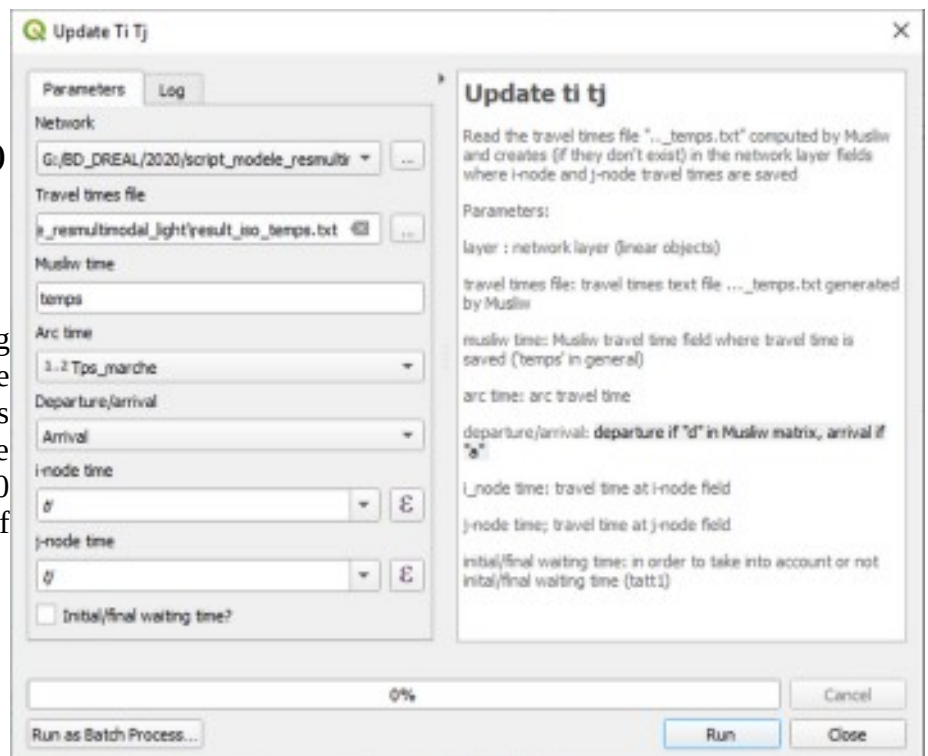
Arc time : **Tps\_marche**

Departure/arrival : **departure if "d" in Musliw matrix, arrival if "a"**

i-node time : **ti if this field exists (update) or possibility to create a new field, in this case write « ti\_bike » for example.**

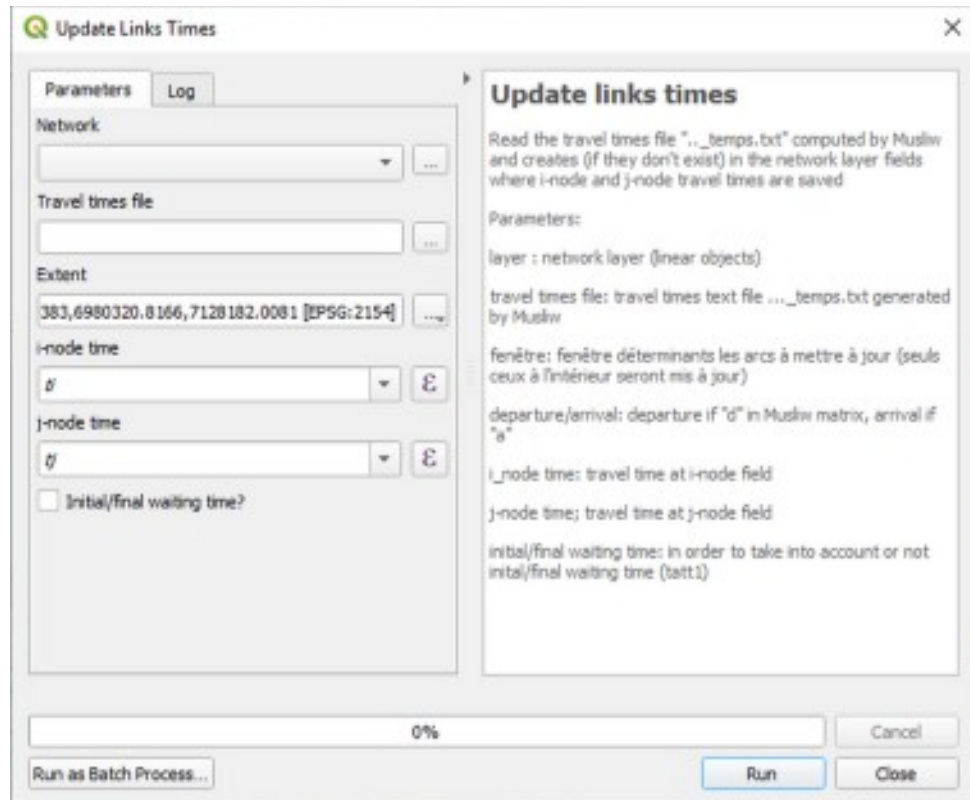
J-node time : **tj (as i-node time but in j)**

Check Initial/final waiting time for taking into account the remaining time from the stop (for example if the arrival time is 8h20 and you asked in the matrix to arrive at 8h30 at latest, then the remaining 10 minutes are not taken into account if checked)



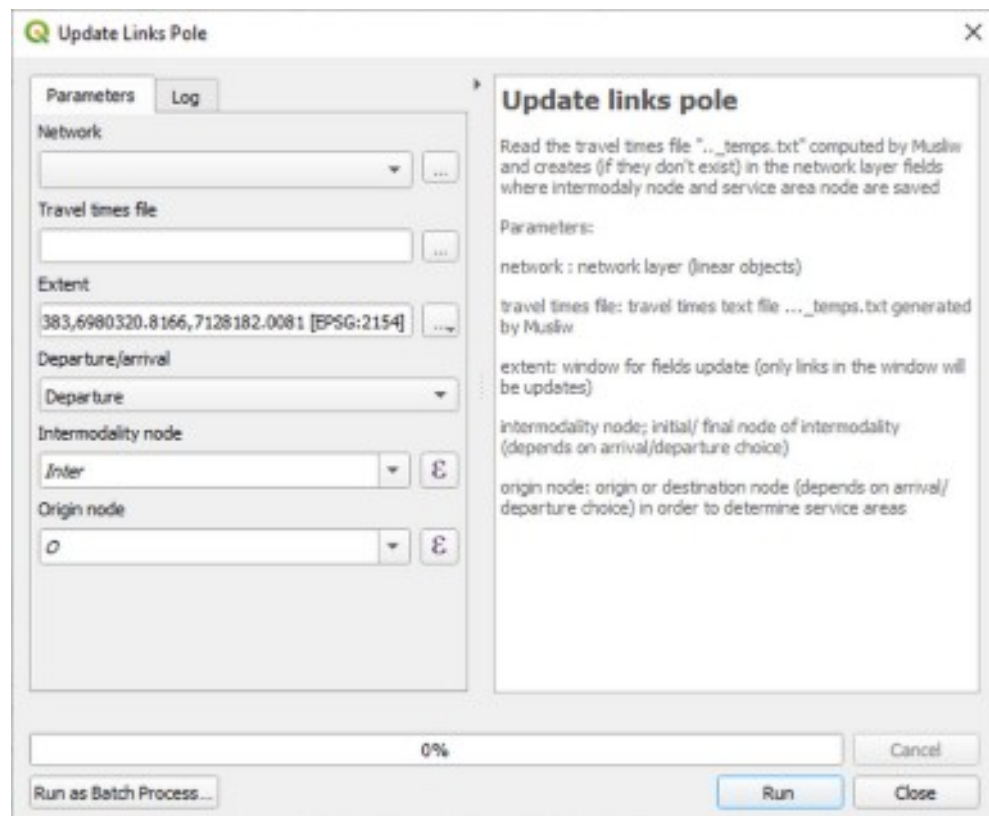
### Option 1: SCRIPT **update links times**

If you just need to make an territorial accessibility map, you can use the script “update links time” instead of “update ti tj”. It uses the Musliw time “temps” by default and get time values directly into travel output times file.



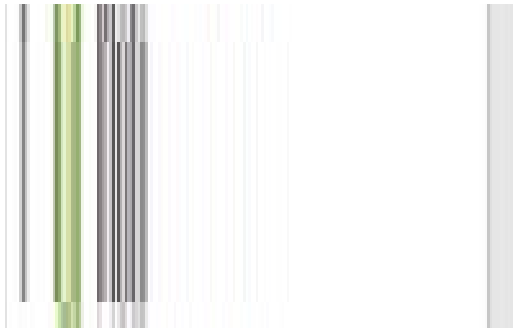
### Option 2: SCRIPT **update links pole**

Use this option if you want to make an discrete territorial analyze map (for example, if you compute a territorial accessibility from several points, this option allow you to generate the catchment area of each point.



## 19. Linear Interpolation

### SCRIPT Linear interpolation



Network : **réseau\_voirie\_shp**

Window : **Allow to define the analysis area**

i-cost: **ti**

j-cost : **tj**

Direction : **Sens**

Spread : **Diffusion**

Impassability : **Impasse**

Pixels nb x : **200**

Pixels nb y : **200**

Pixel size x : **let default values**

Pixel size y : **let default values**

Decimals : **5**

Radius(m) : **1000**

Spread speed : **4** (4 km/h for walking) speed to be adapted depending on mode (15 km/h cycling)

or for a distance map : **60**

Impassable? : **do not check**

individual values: **choose a field for in case of an individual value analysis (ex: pole) or keep '.' as default**

Raster file : save raster as

**Run**



## 20. Isovalue polygons

### SCRIPT Isovalue polygons

Raster : **raster layer**

Band : **1**

Min : **0**

Max : **60** (depends on the duration)

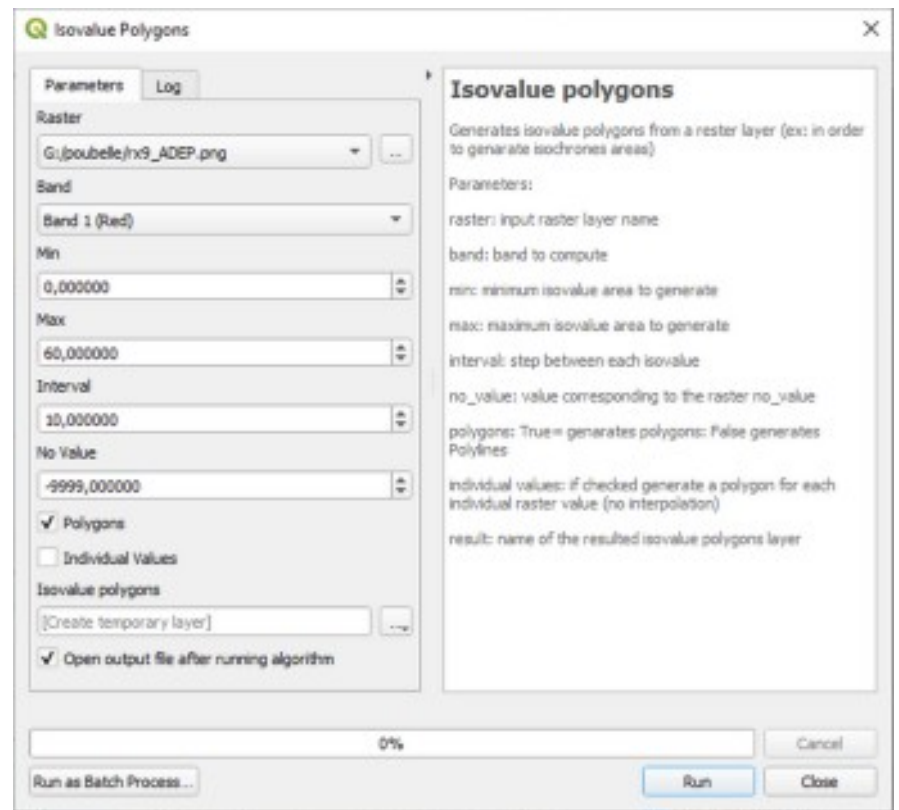
Interval : **10** (iso-value interval)

No value : **let default value**

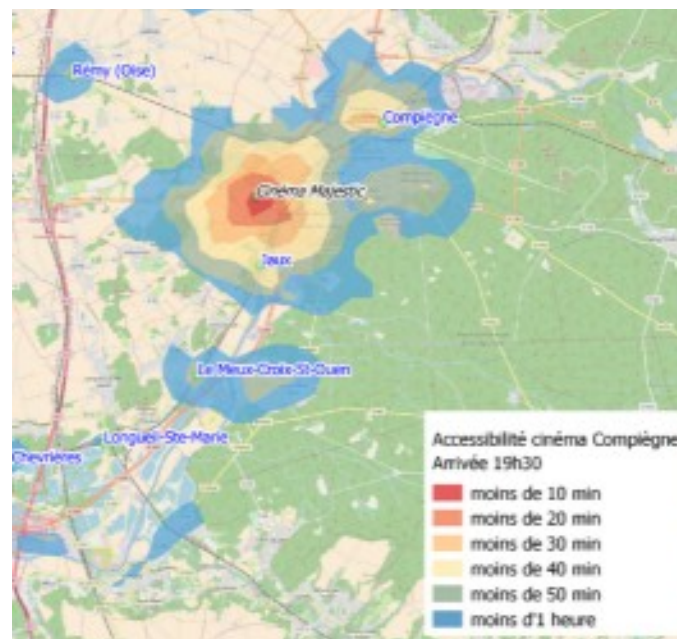
Polygons : **check it**

Individual values: **check only for individual value analysis**

Contours iso-valeurs : **save layer as..**



### Isochrone example

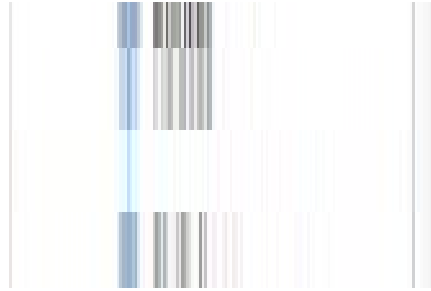


## Partie 3 : To go further...

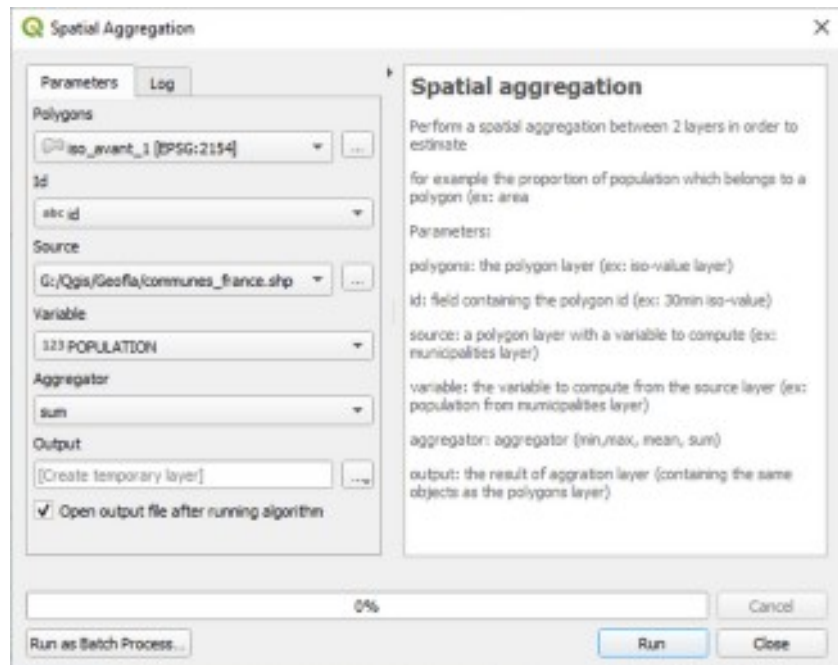
### 21. Compute population inside iso-value polygons

#### SCRIPT **Spatial aggregation**

Open the iso-value polygons layer  
Open an administrative layer with a population field



Polygons : **select the iso-value polygons layer**  
Id : **polygon id** (corresponding to the threshold defined in the polygons layer)  
Source : **administrative layer**  
Variable : **field name containing the population data**  
Aggregator : **sum** (to sum). Choose between sum, mean, minimum, maximum  
Output : **save result as (polygons layer)**



Results

	id	POPULATION
1	0,00000	621,01899
2	5,00000	2381,86852
3	10,00000	5078,68296
4	15,00000	5542,47301



# B- Production of a PT services offer (example : french regional railways offer)

## 1. Get the theoretical offer on the SNCF open data website

(<https://data.sncf.com/explore/?sort=modified> )

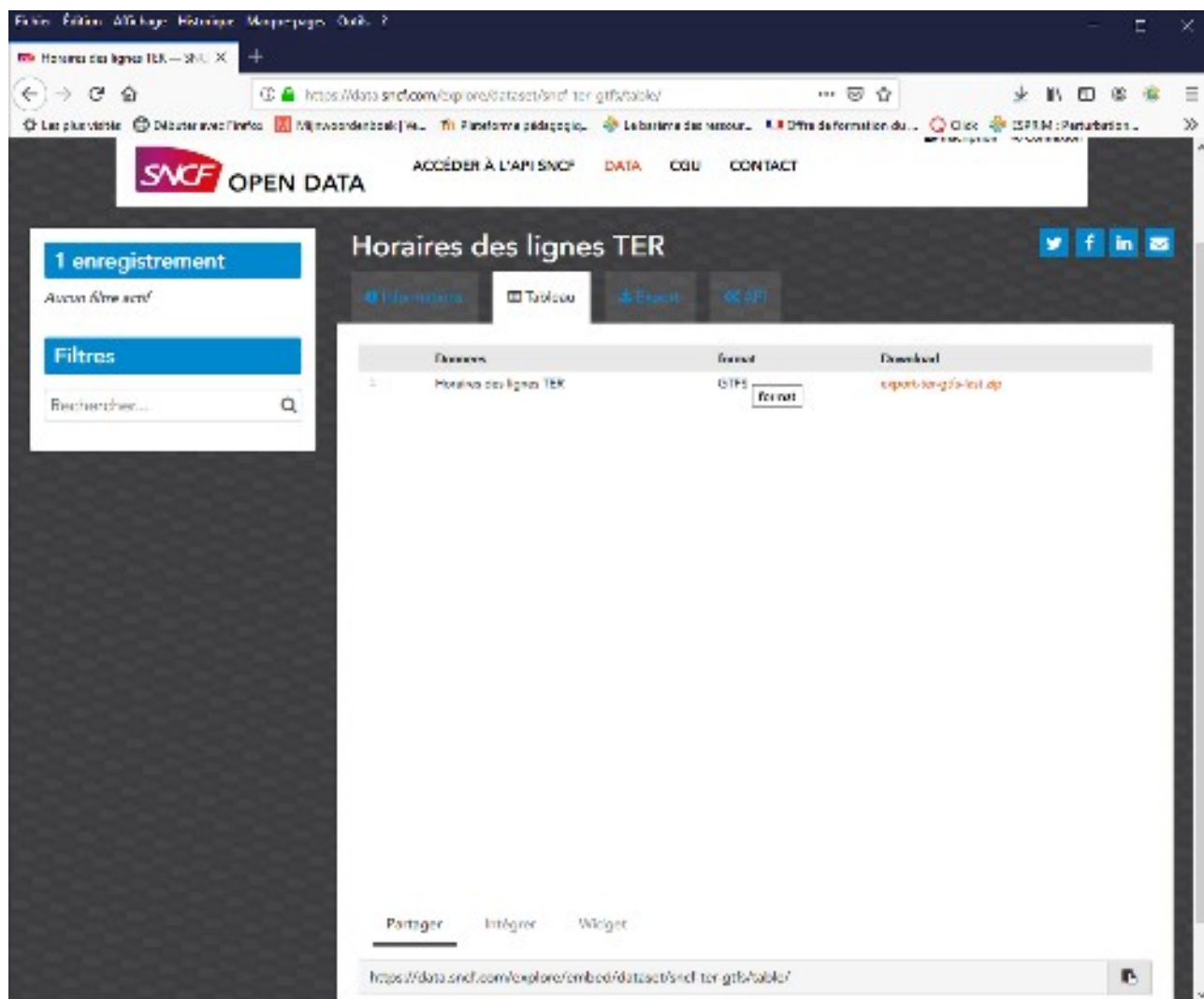
The screenshot shows the SNCF Open Data website interface. The header includes the SNCF logo, 'OPEN DATA', and navigation links: 'ACCÉDER À L'API SNCF', 'DATA', 'CGU', and 'CONTACT'. There are also links for 'Inscription' and 'Déconnexion'.

The main content area displays a list of datasets. The first dataset is 'Horaires des lignes TER' (TER line schedules). It shows the producer as 'TER', the license as 'Open Database License (ODBL)', and the number of items as '1 Alimént'. Below this, there are filters for 'Nom', 'Train', 'Date de voyageur', 'GTF5', and 'France'. To the right of the dataset name, there are icons for 'Tableau', 'Export', and 'API'.

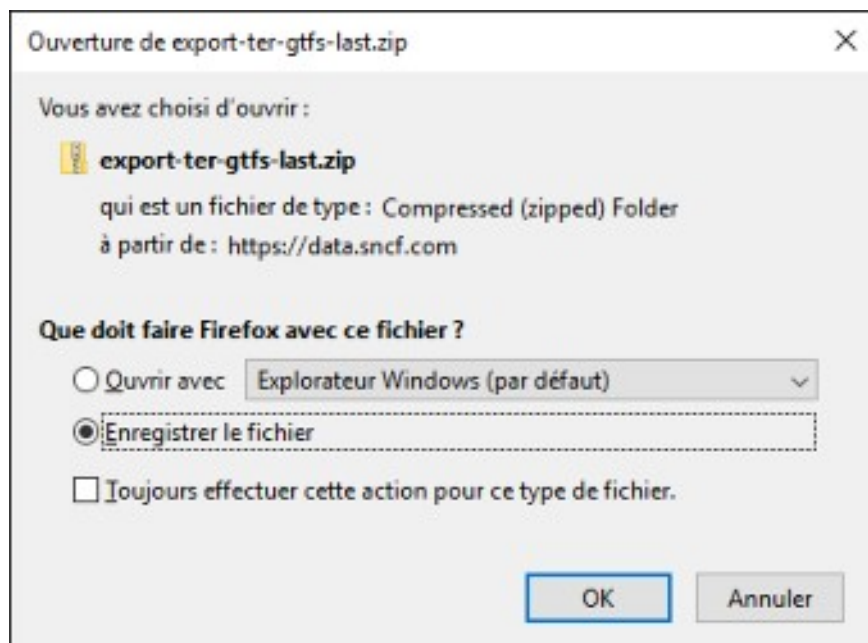
Other datasets visible include 'Horaires des Tram-Train TER Pays de la Loire', 'Horaires des lignes Intercités', and 'Horaires des lignes Transilien'.

On the left side, there is a sidebar with '216 jeux de données' (216 datasets). It includes a search bar 'Trouver un jeu de données' and a 'Filtres' section. The 'Vue' (View) section shows options for 'Analyse', 'Carte', 'Image', and 'Vue personnalisée'. The 'Modifié' (Modified) section shows a list of years from 2014 to 2019 with corresponding counts. The 'Producteur' (Producer) section shows 'SNCF Réseau, DIRECTION FINANCE ACHATS'.

**Choose the services that we want to represent** (In our example “Horaire des lignes TER”).  
Click on « **export-ter-gtfs-line.zip** » to download and save the file



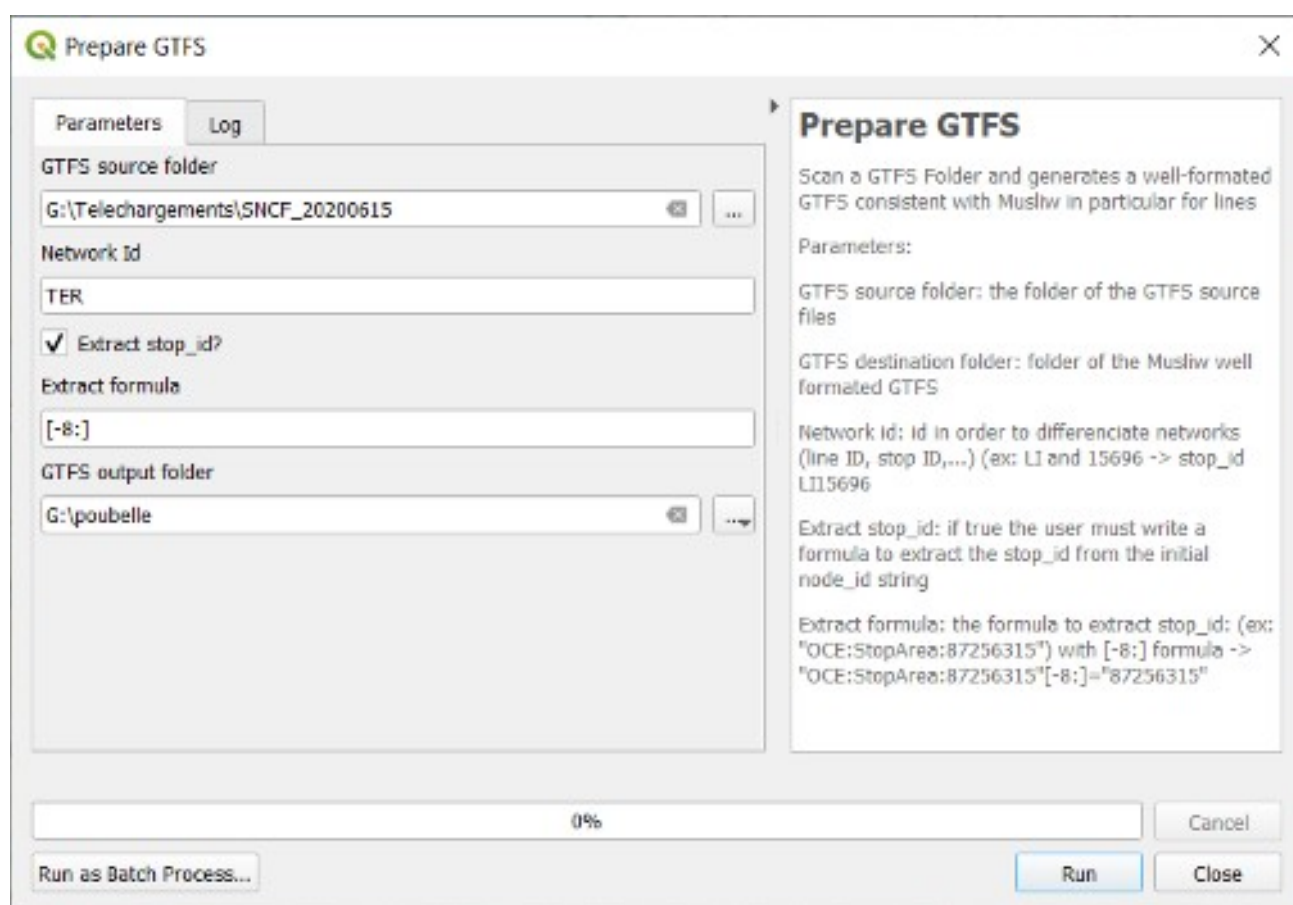
## Accept website policy



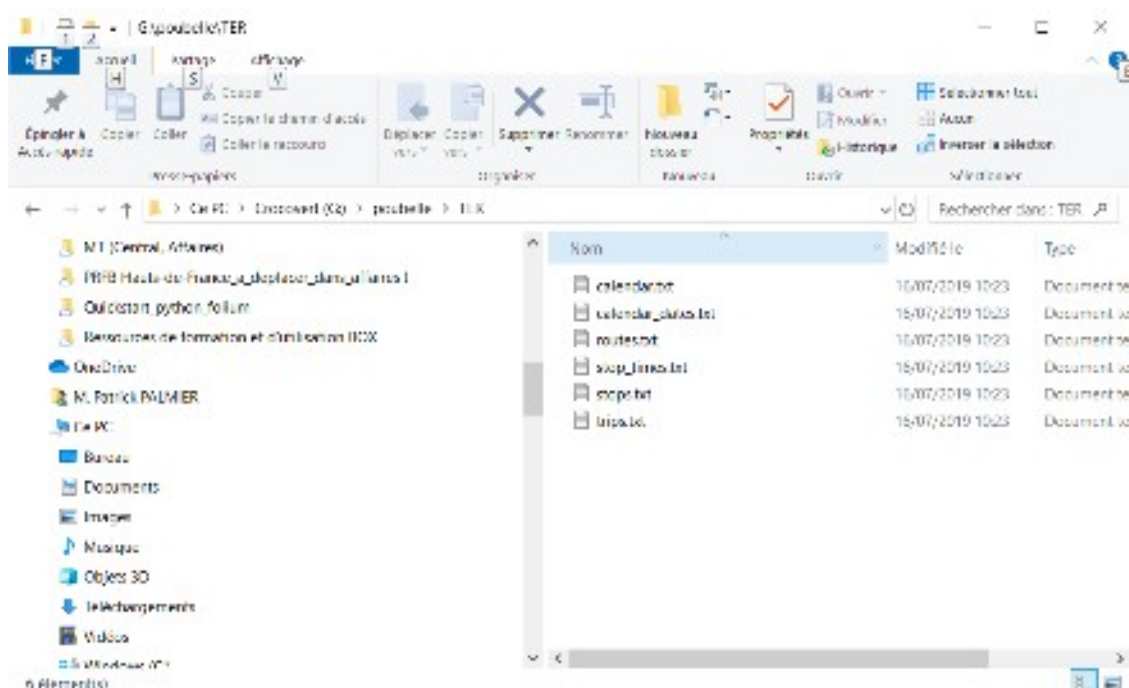
## Save and unpack the archive in a directory

## 2. GTFS pre-processing

A pre-processing has first to be done on the raw GTFS with the « **prepare GTFS** » algorithm in order to be correctly formatted for “networks” plugins scripts and in particular with the mutli-modal calculator Musliw.

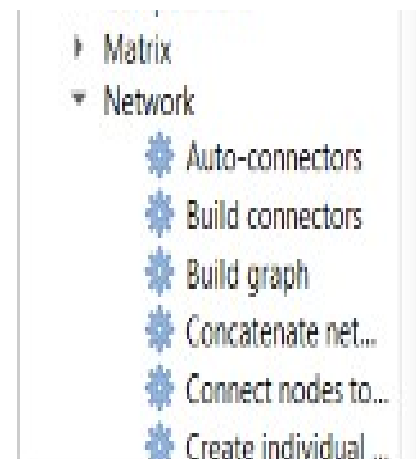


The script generates a new GTFS dataset compliant with networks algorithms



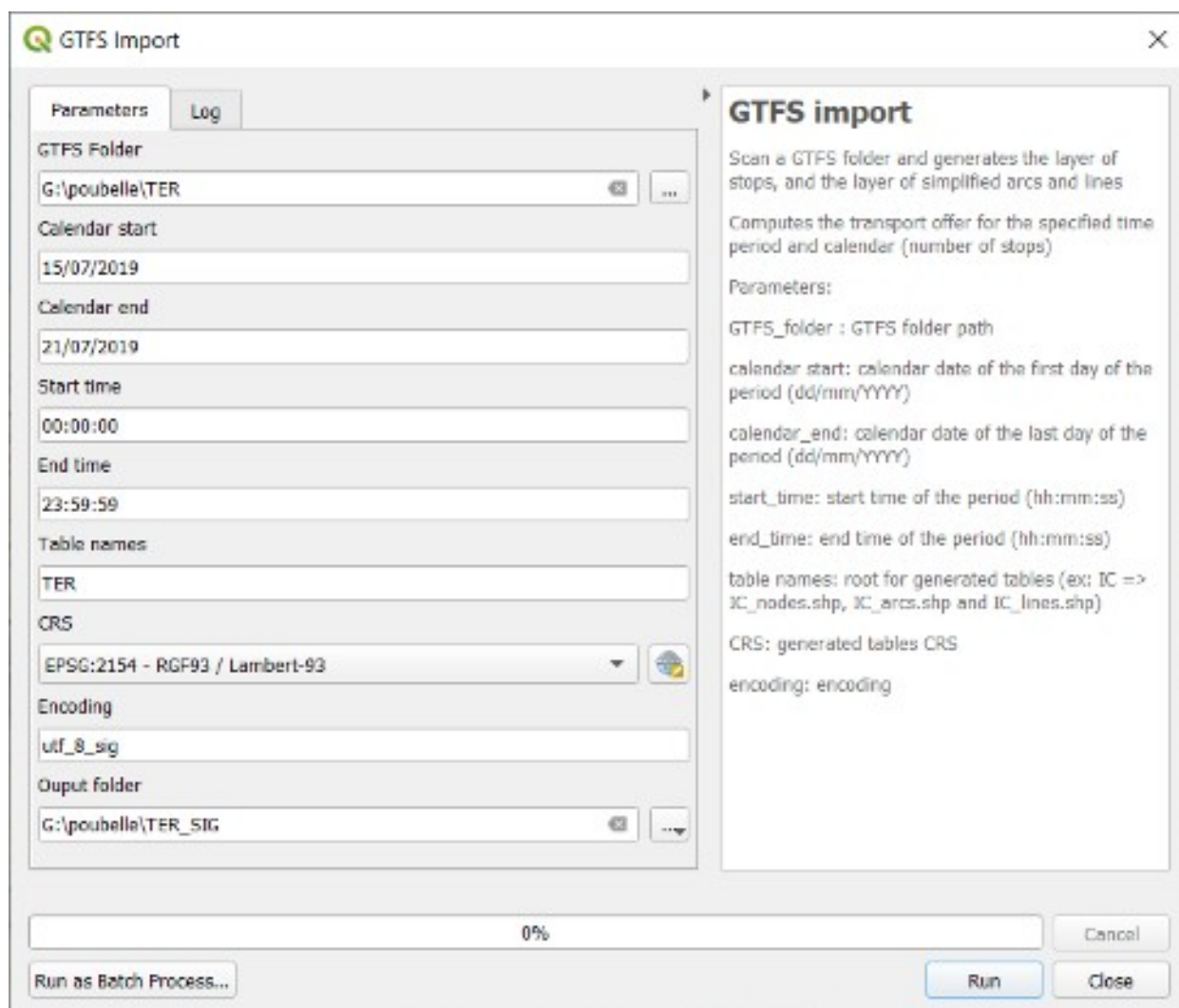
### 3. GTFS import in Qgis

Run the « **import GTFS** » algortihm to be able to view the corresponding offer in Qgis



the start and end dates of the calendar must be dates where corresponding data (timetables) are available in the GTFS file.

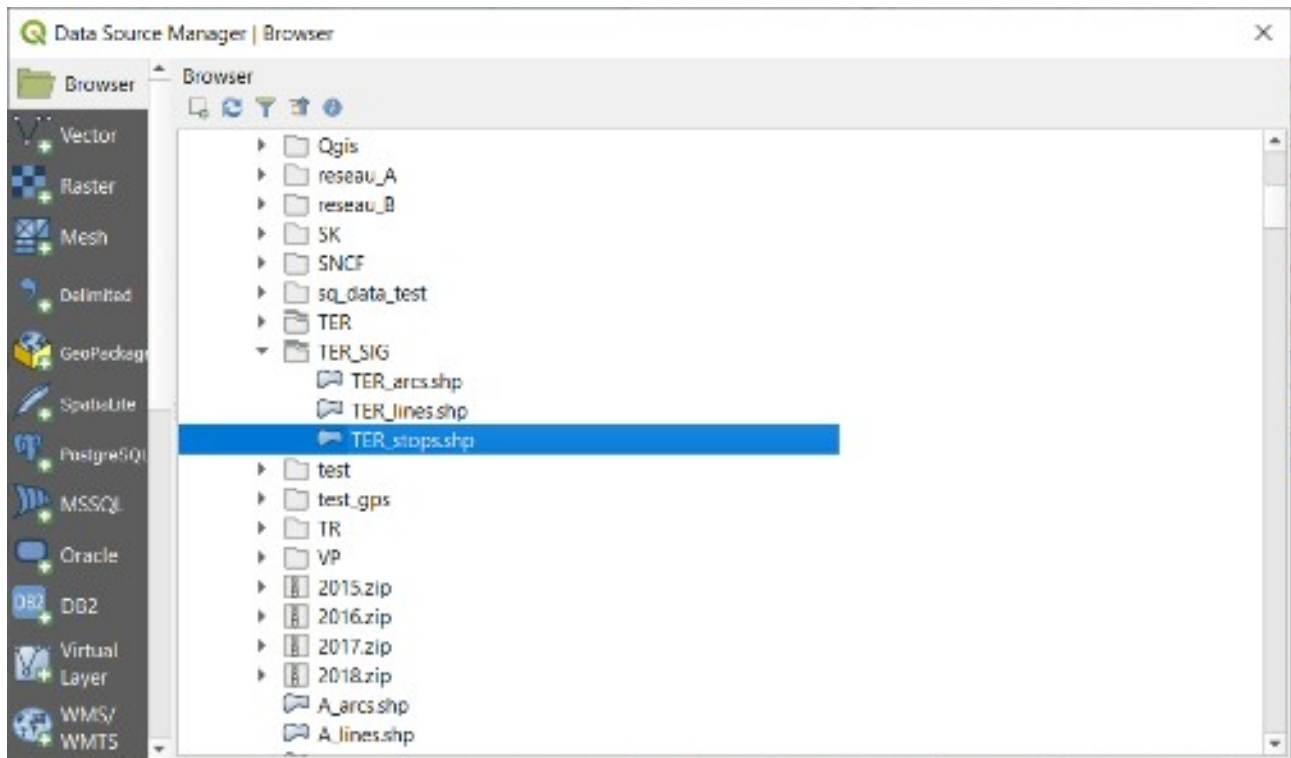
For that point, it is recommended to edit **calendar.txt** and/or **calendar\_dates.txt** to choose a week or a period where timetables are available



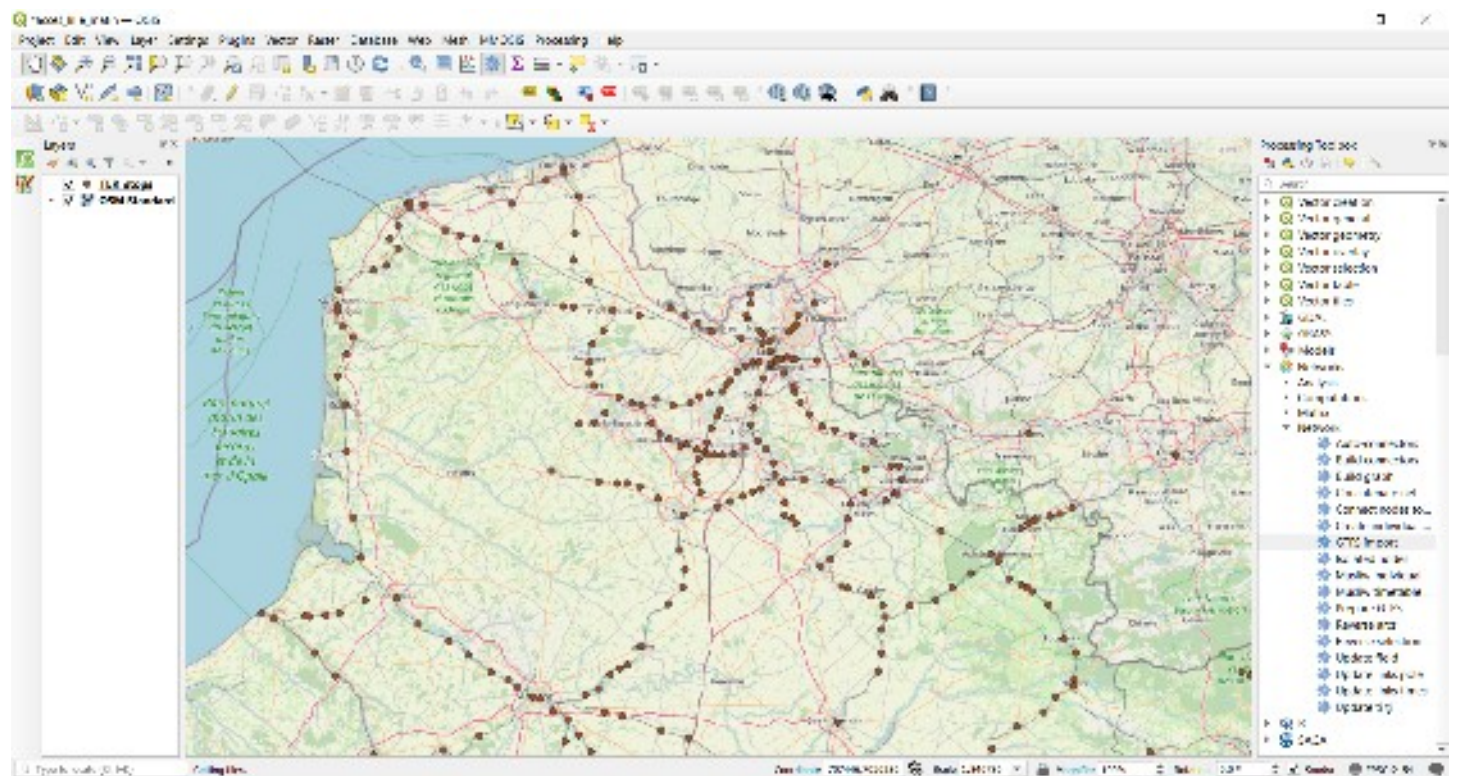


## 4. View stops

Load in Qgis the « **\_stops.shp** » layer which contains regional railways services stops



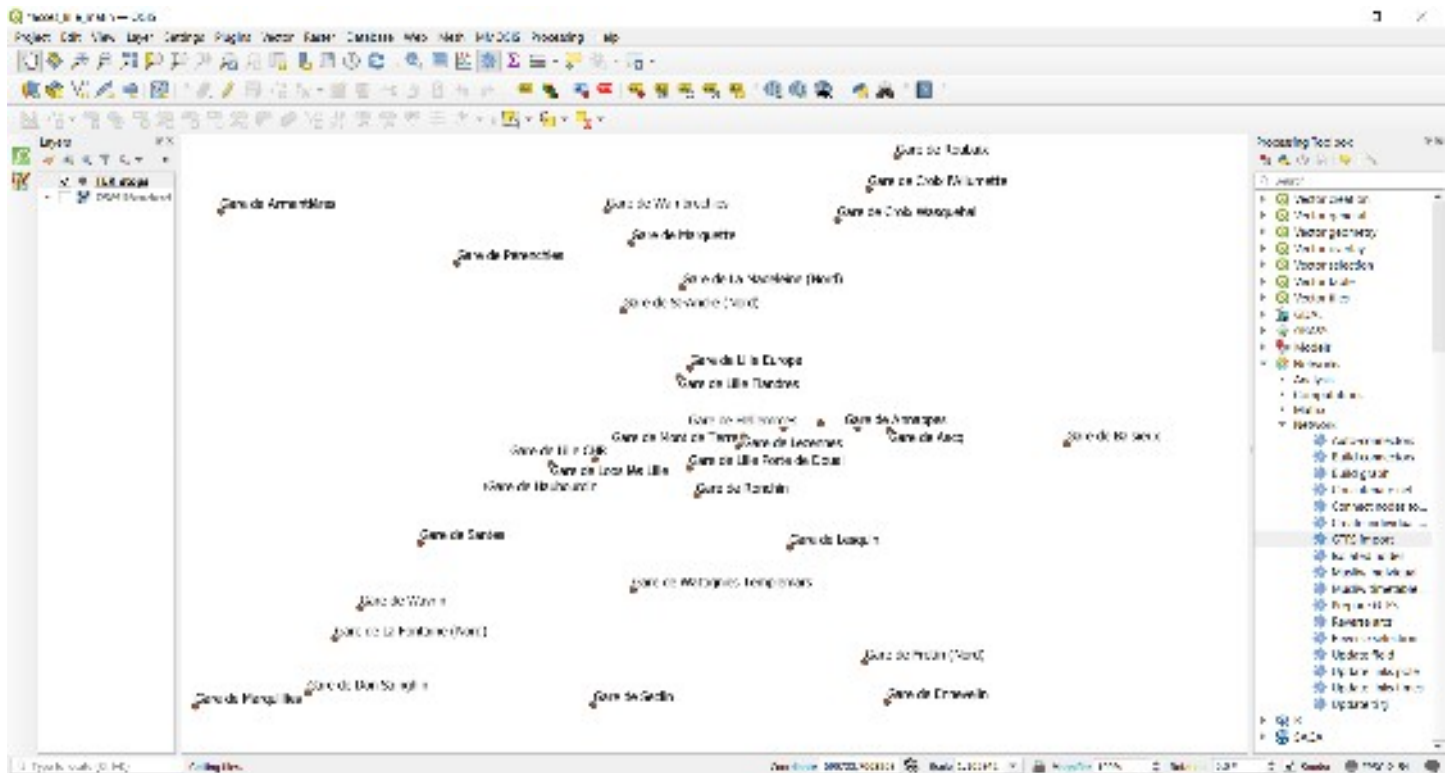
Results (zoom in Lille regional area)



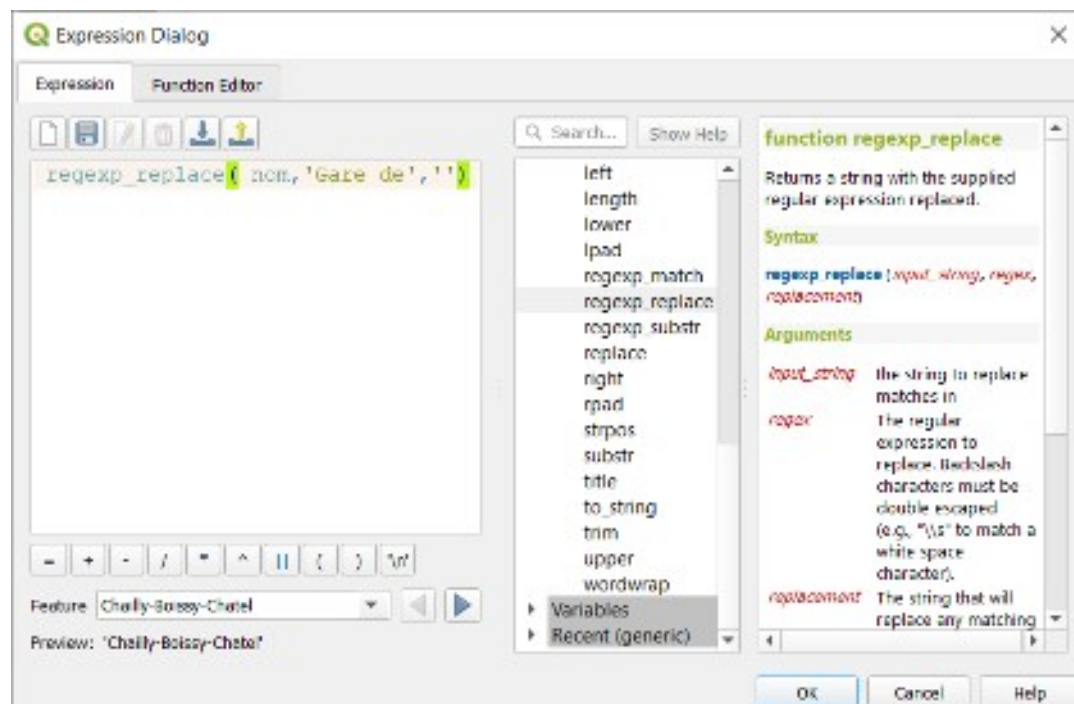
## 5. Show stations names

To show the stations names, you have to show labels with the field “**nom**”

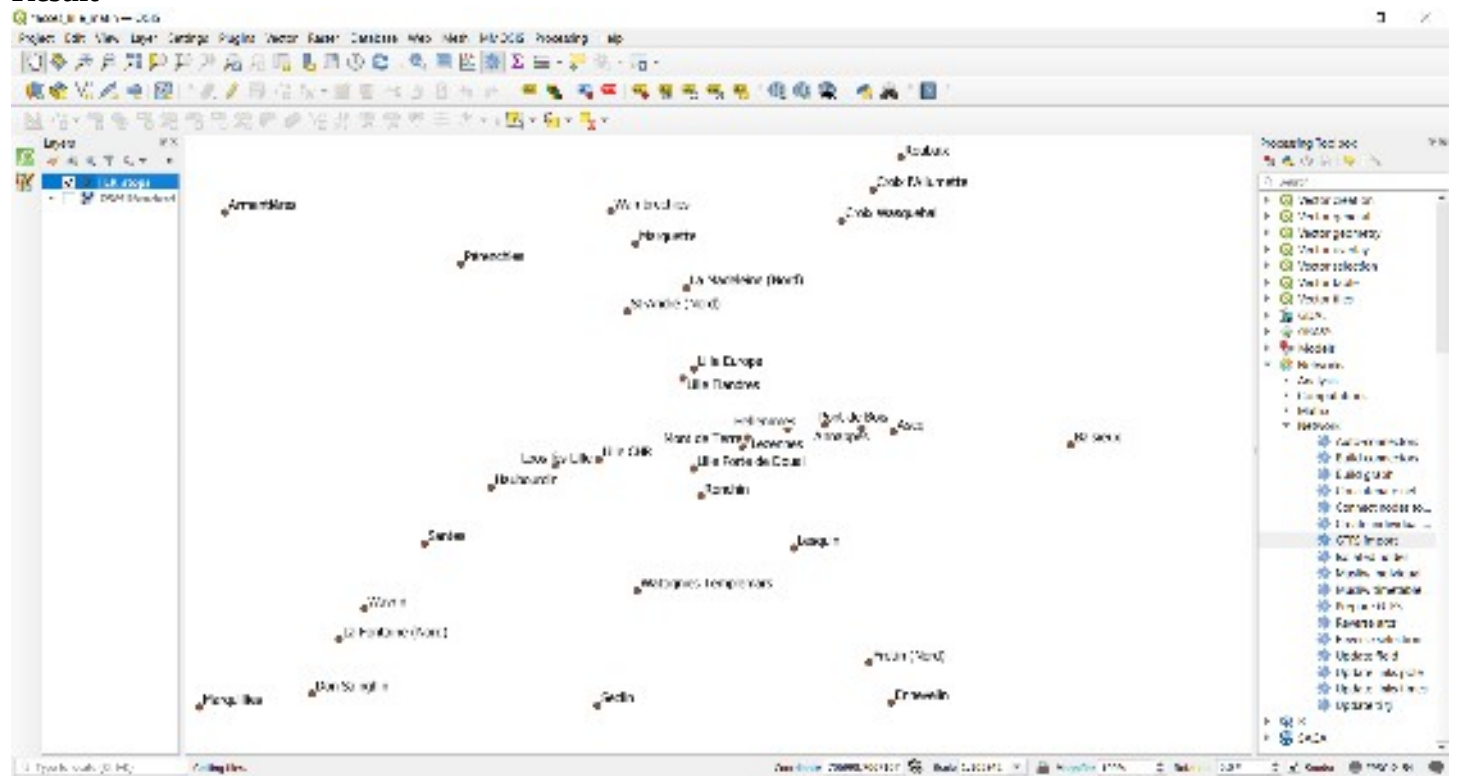
On the other hand, we notice that the station name is defined with “**Gare de**” before the names of the station



To not show “Gare de”, it is possible to use a **regular expression** which will replace “Gare de” by an empty string.



## Result

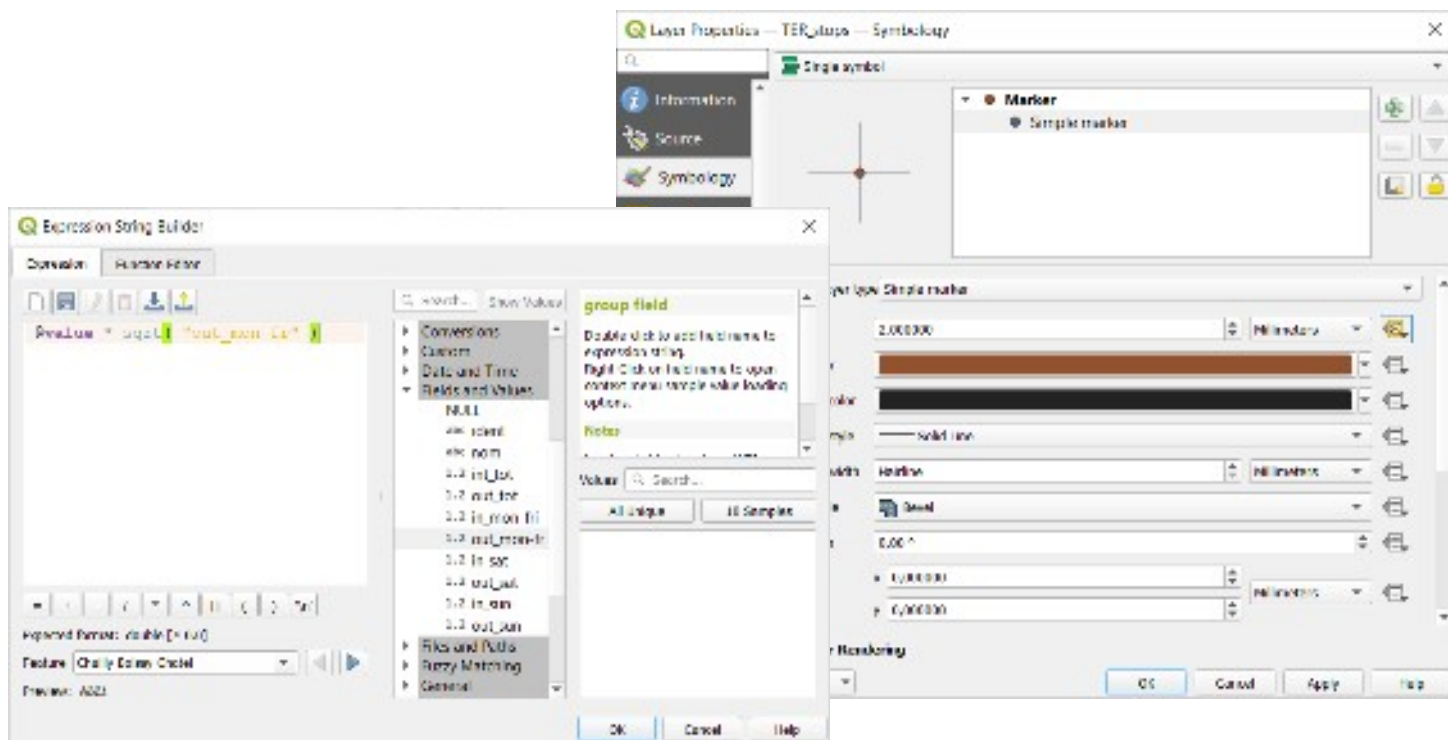




## 6. Produce a proportional symbol size map

To perform a **map with proportional symbol size**, you have to **customize the symbol size** with an expression containing the variable to represent

“out\_mon-fri” field represents the average daily number of departures during the time interval you’ve chosen in script “GTFS import”. The keyword **@value** represents the width attribute value. To keep it in the expression allow to adjust the size directly by modifying the value in the size width, without having to modify the expression.



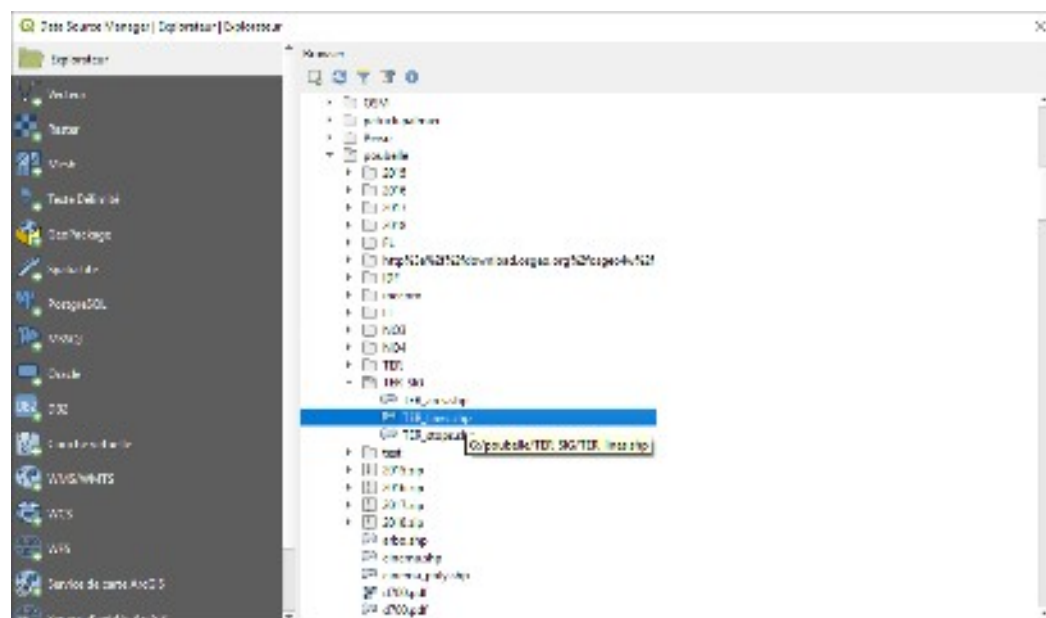
### Result

## 7. View the number of services by link

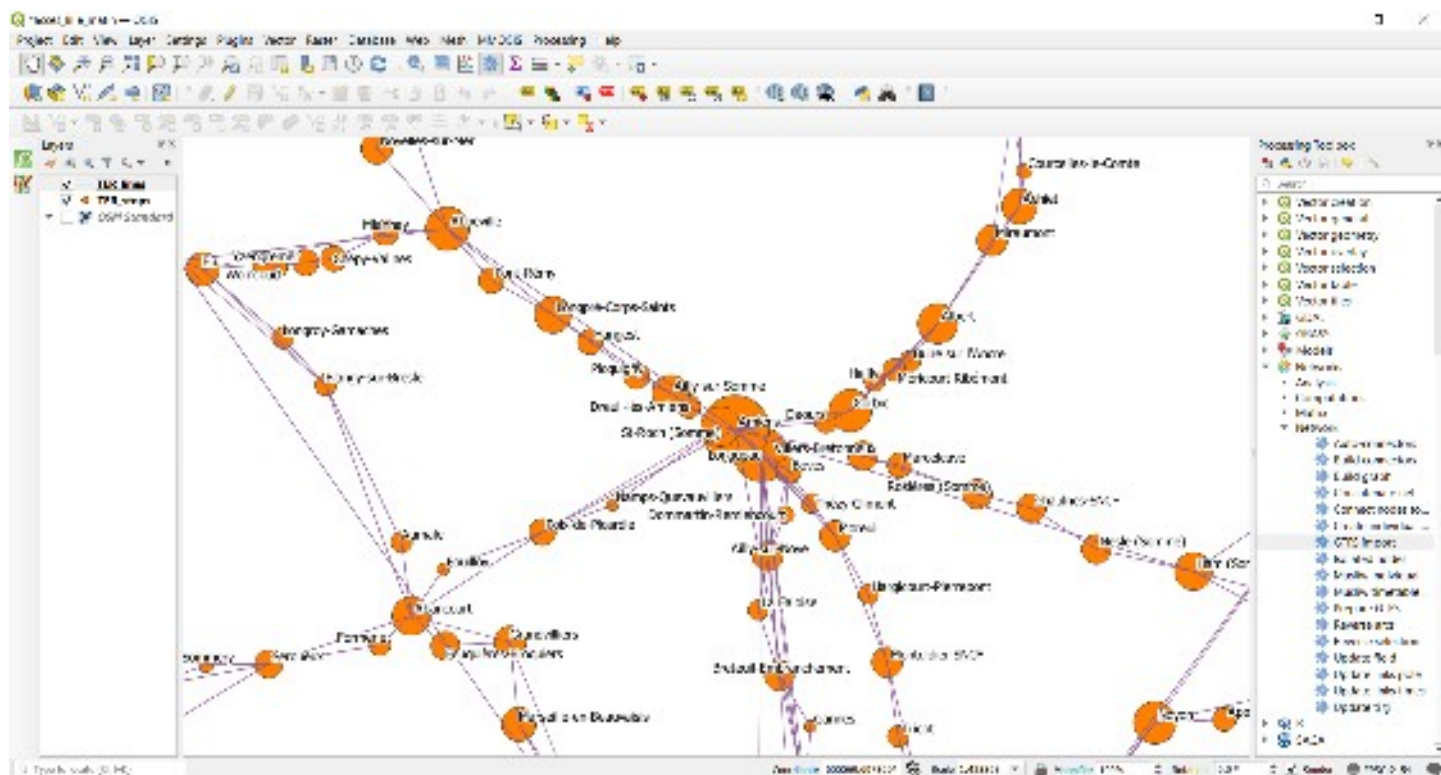
To view the **number of services by link**, you have to load the “\_lines” layer.

On the other hand, the line shape is simplified. it consists of a straight line between two successive stops or stations. That could generate an unclear representation due to the overlay of direct and omnibus trains

**Load the « lines » layer**



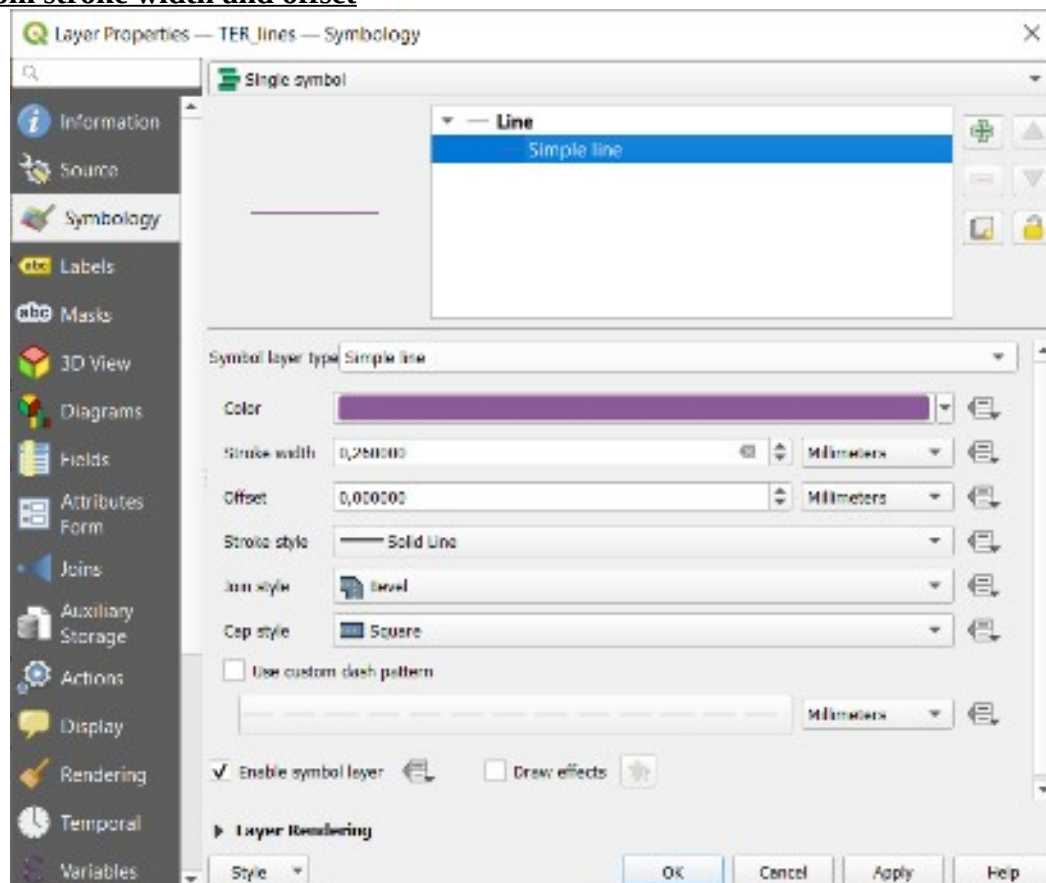
## Result



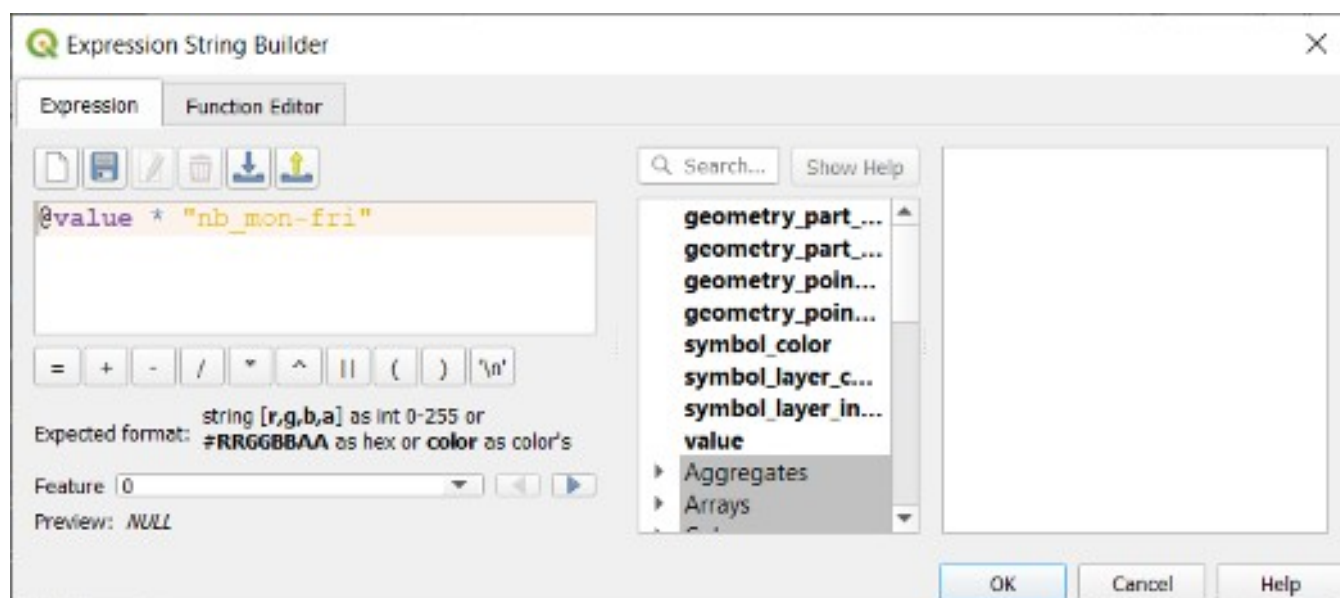
## 8. Widths proportional to the number of circulations

**Warning:** If a network contains a common core where several lines pass in the same road (two successive stops), the settings below is no more adapted and need more complex expression not presented in this document that allow to view the different lines juxtaposed instead of overlapped

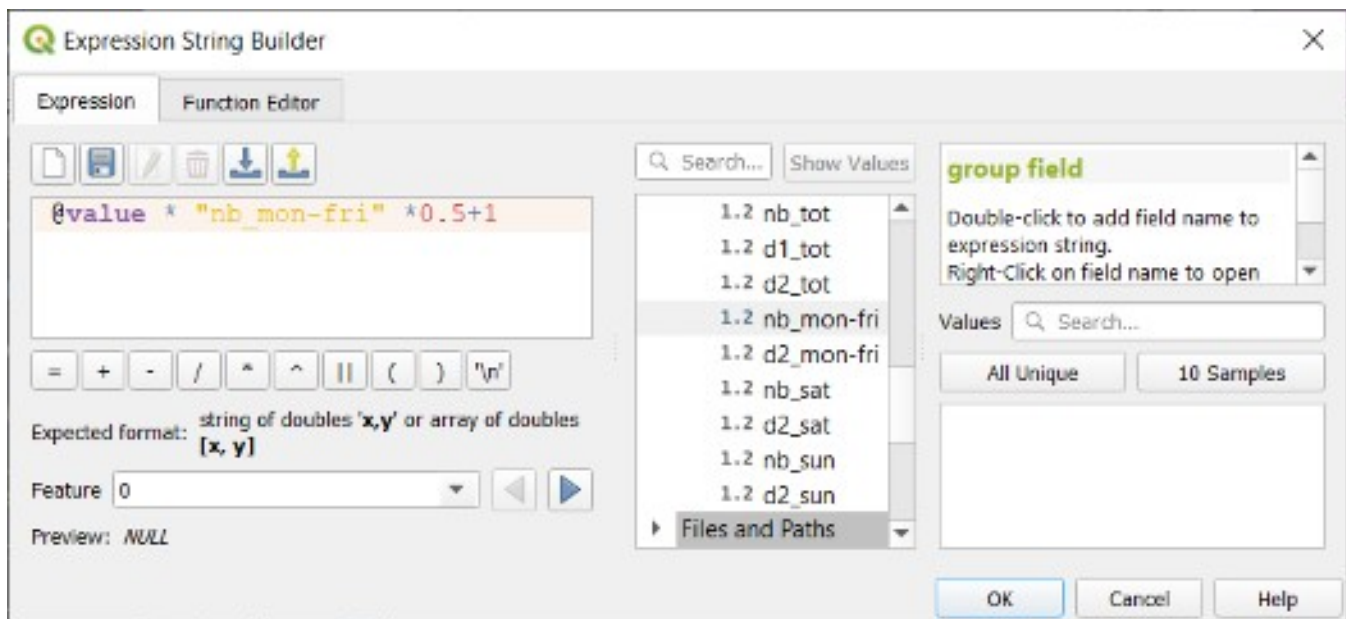
### Define a custom stroke width and offset



### Stroke width

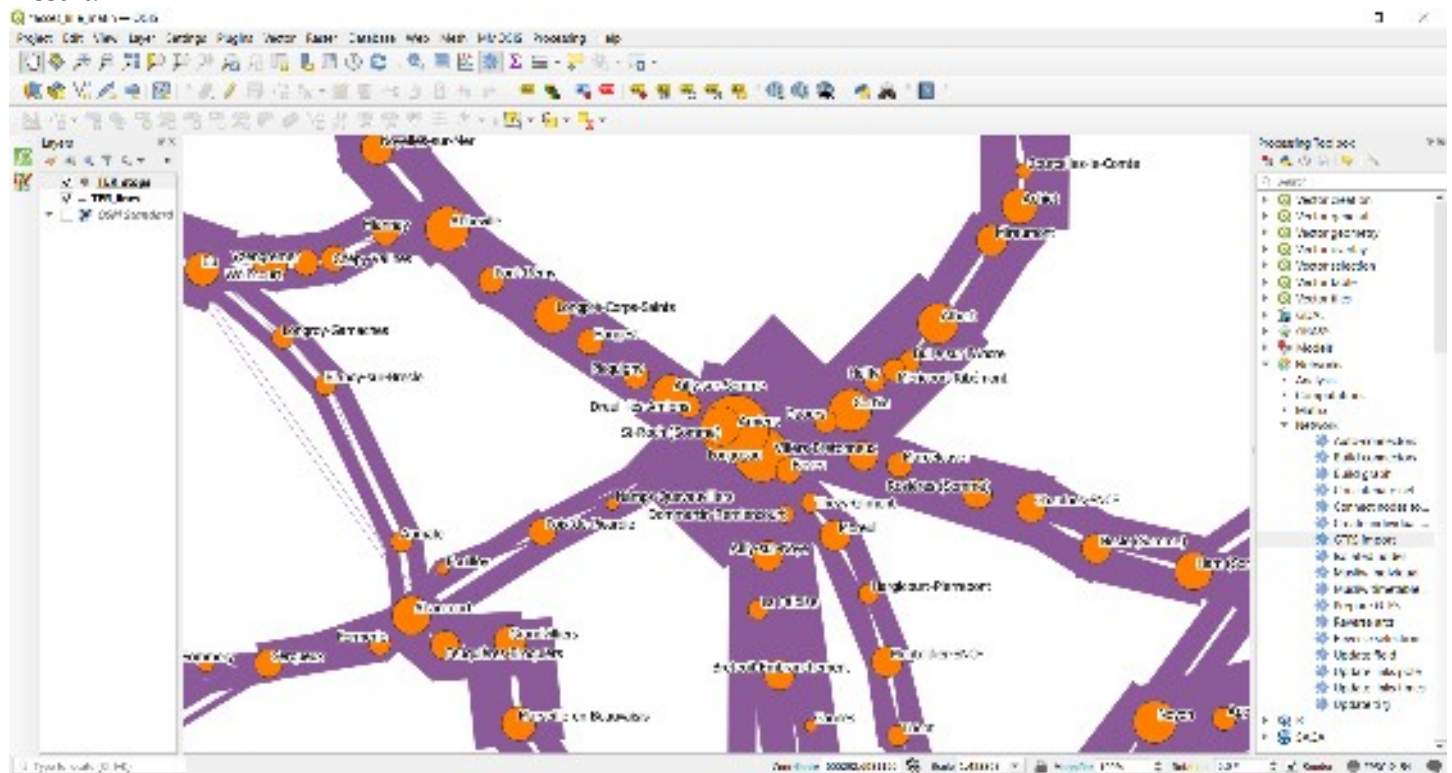


## Offset



NB : Numeric values in stroke width form and offset from have to be identical with the expressions defined above.

Result:

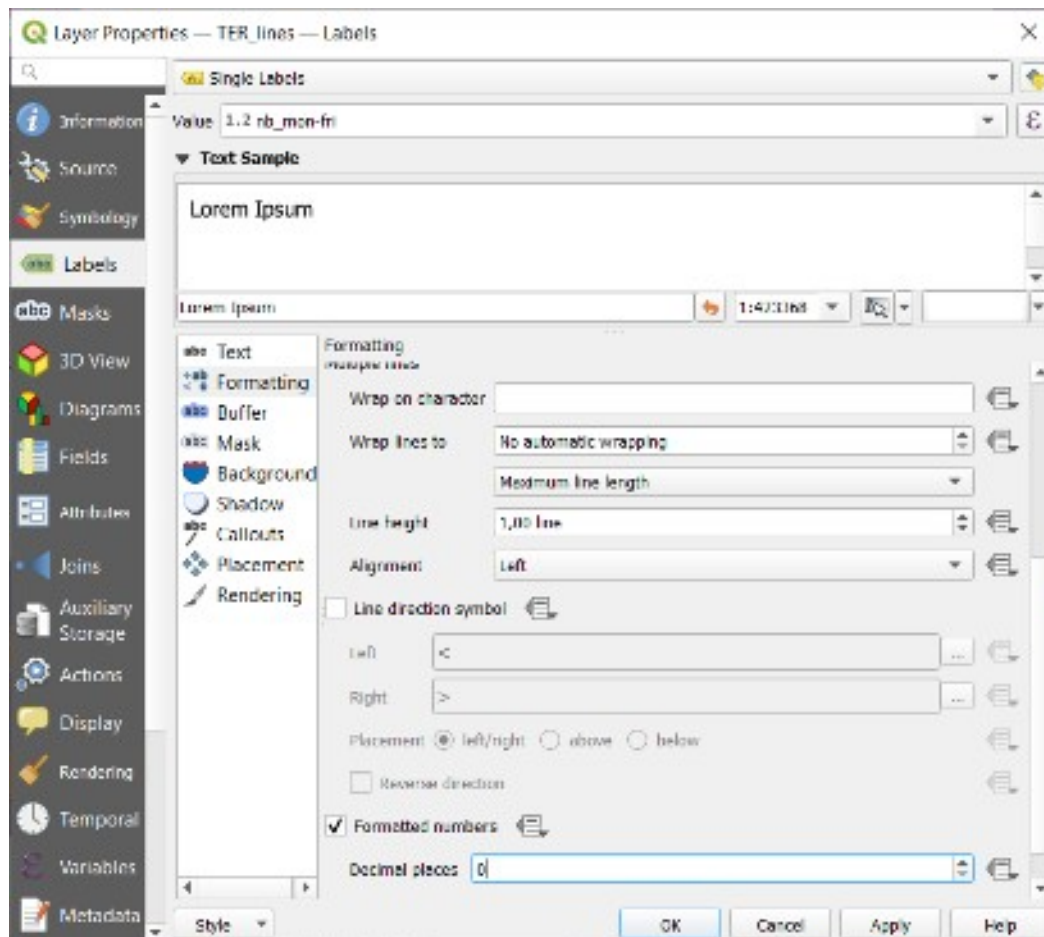




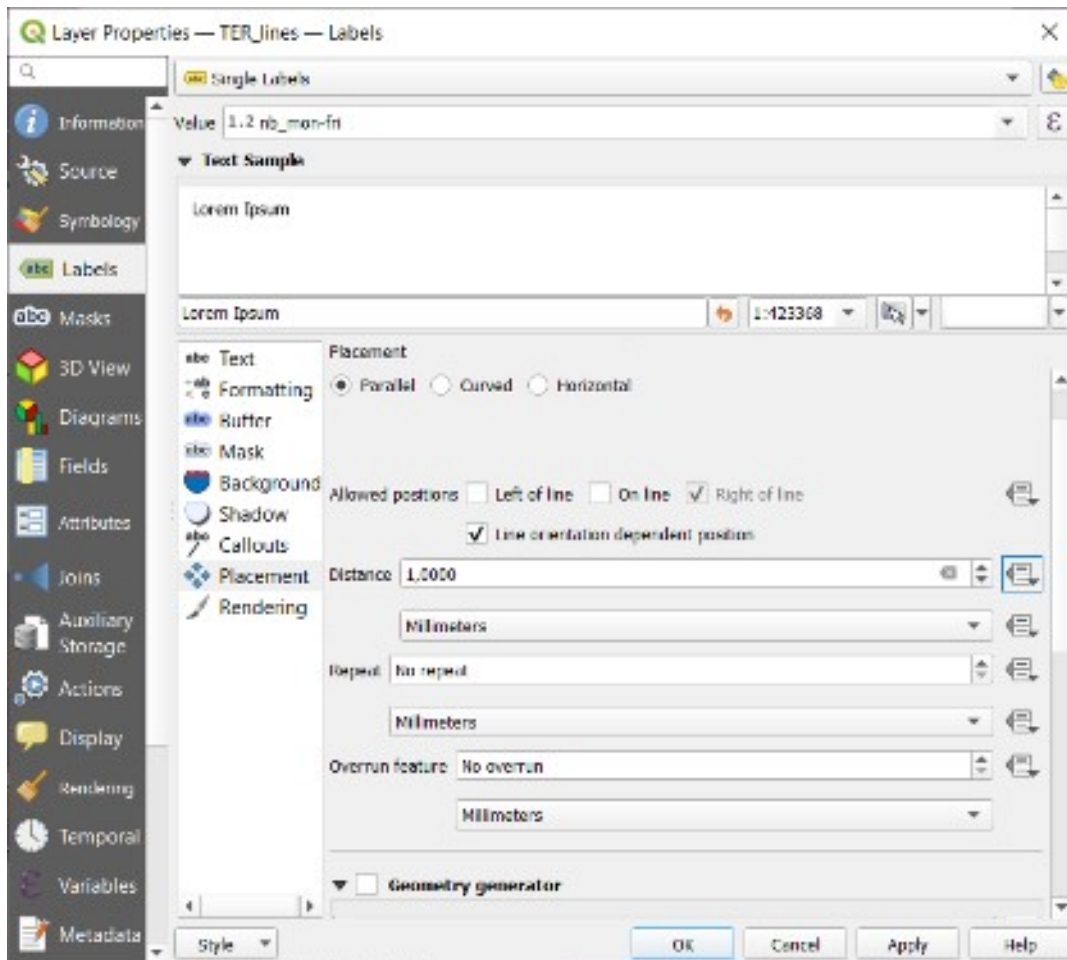
## 9. Label the number of services

Choose **nb\_mon\_fri** (average number of train courses the selected week from Monday to Friday included). As it is an average, specify formatted numbers with “0” in place of decimals

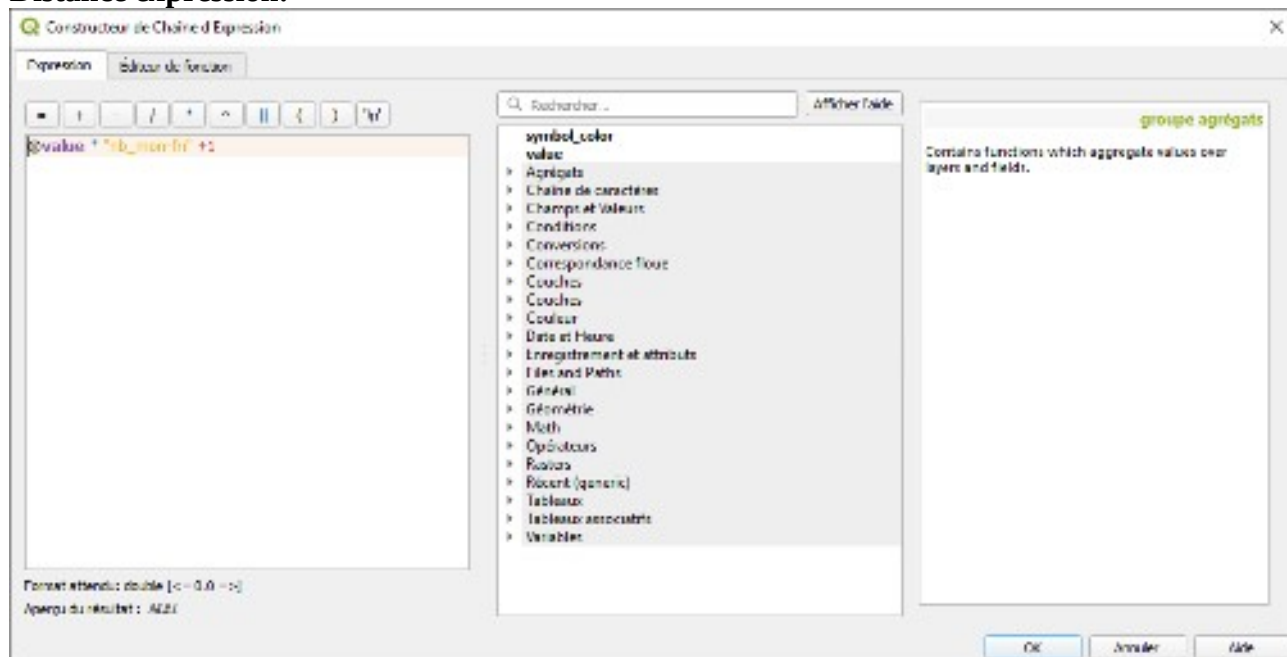
PS: In general, rather than simple labels, choose "rule-based labels". This will allow you to filter the labels more easily, because if you have set up your simple labels in a complex way, changing the type of labels will make you lose this settings which will have to be renewed with the rule-based labels.



Set the same distance as the strike width and offset value on the lines layer. Check **“line orientation dependent position”** and **“right of line”** a traffic map rendering.



## Distance expression:





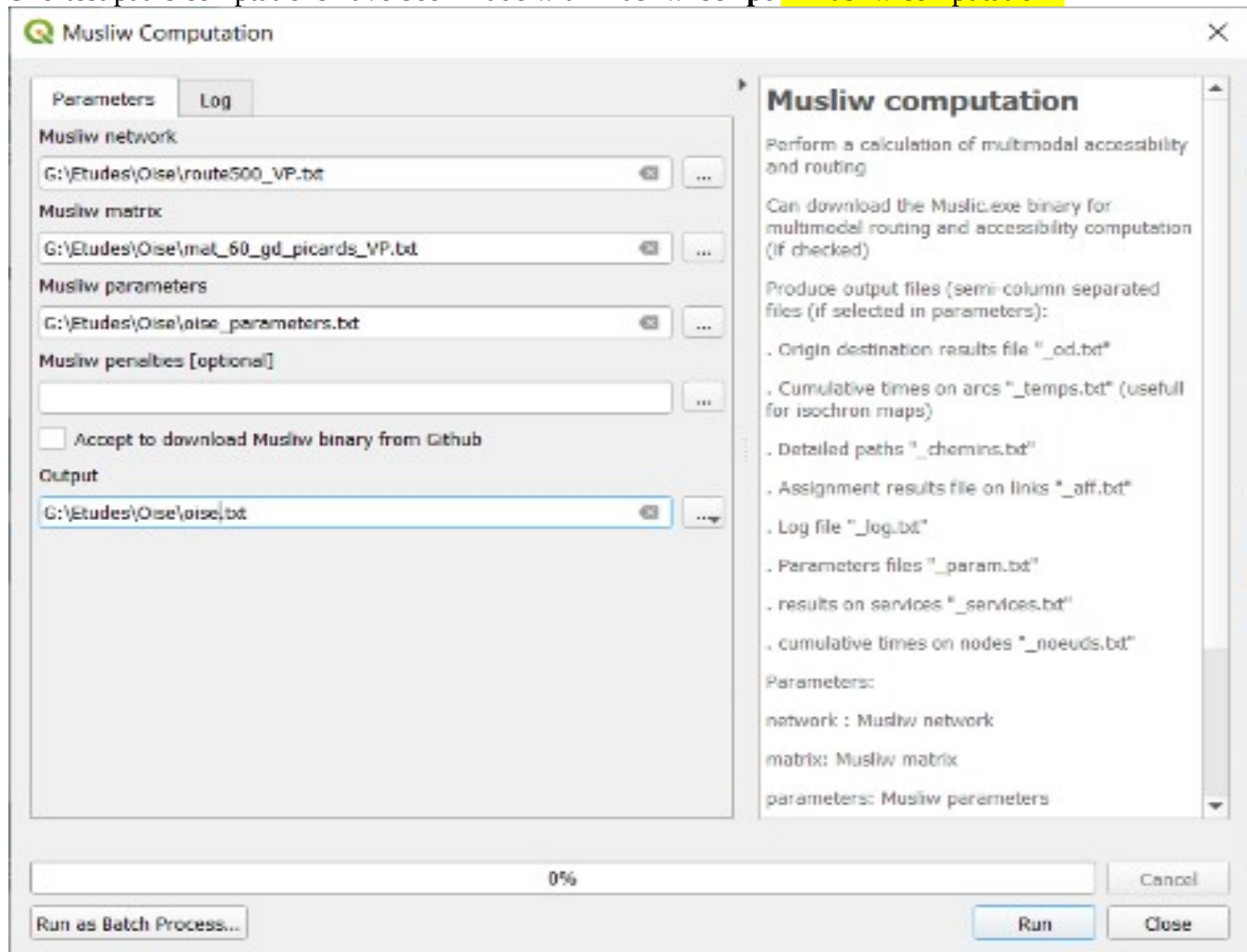


# C- Production of a traffic map

The aim of this tutorial is to produce a traffic map

In a first step a Musliw matrix has been generated from the trip file of the “Grands Mobiles Picards” household travel survey. The matrix was built with car as driver trips from people living in the Oise Territory. The objective is to view desire lines of drivers who live in Oise territory.

Shortest paths computations have been made with Musliw script « Musliw computation»



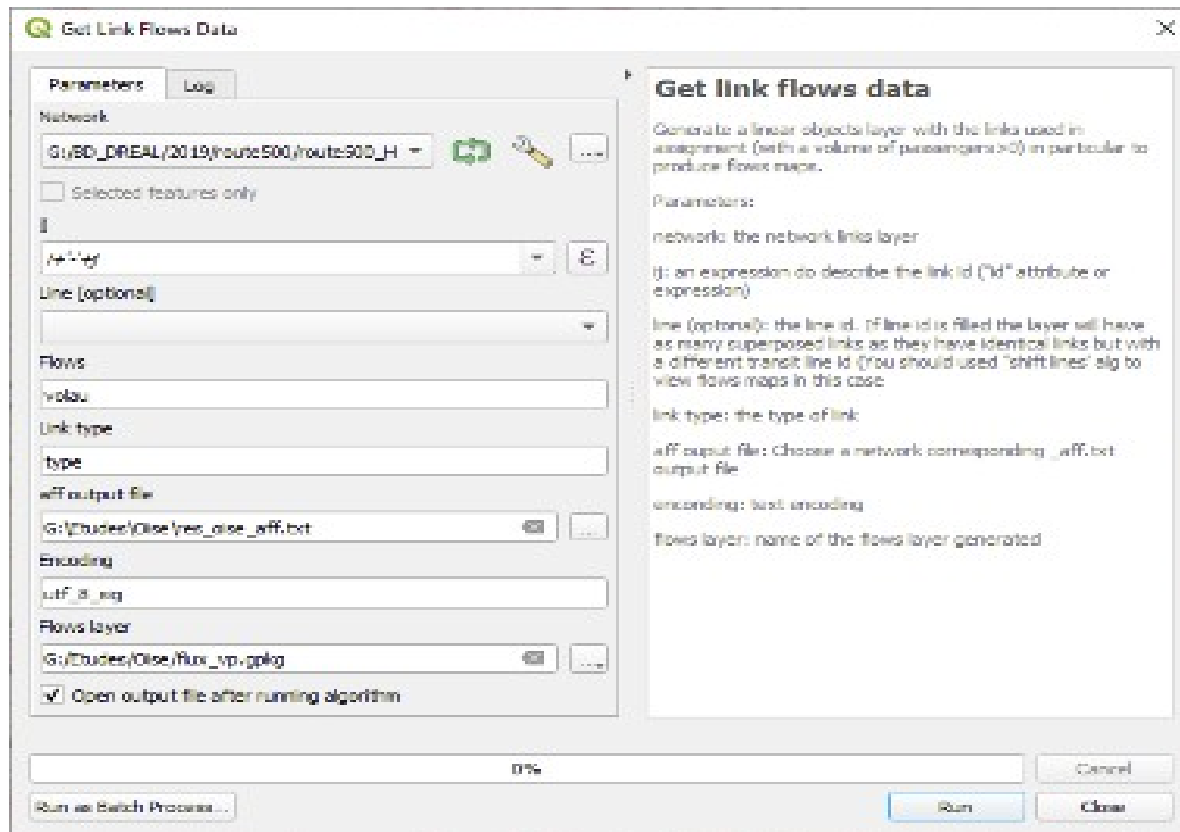
The “\_aff” file contains the list of arcs (here route500) with auto-volumes on shortest paths .

## 1. Get traffic volumes

The road network used here is the Route500 network

As route500 is the used network, it has to be the reference in order to get Musliw results calculations data

The traffic volumes fields is often called **volau** (auto volumes), and you have to get data from the **“.aff”** file which contains informations about the number of cars on each “route500” arc.



Le **script** génère une **table flux\_vp.shp** qui contient les arcs route500 ayant été empruntés par au moins une OD de la matrice.

flux\_vp — Features Total: 20958, Filtered: 20958, Selected: 0

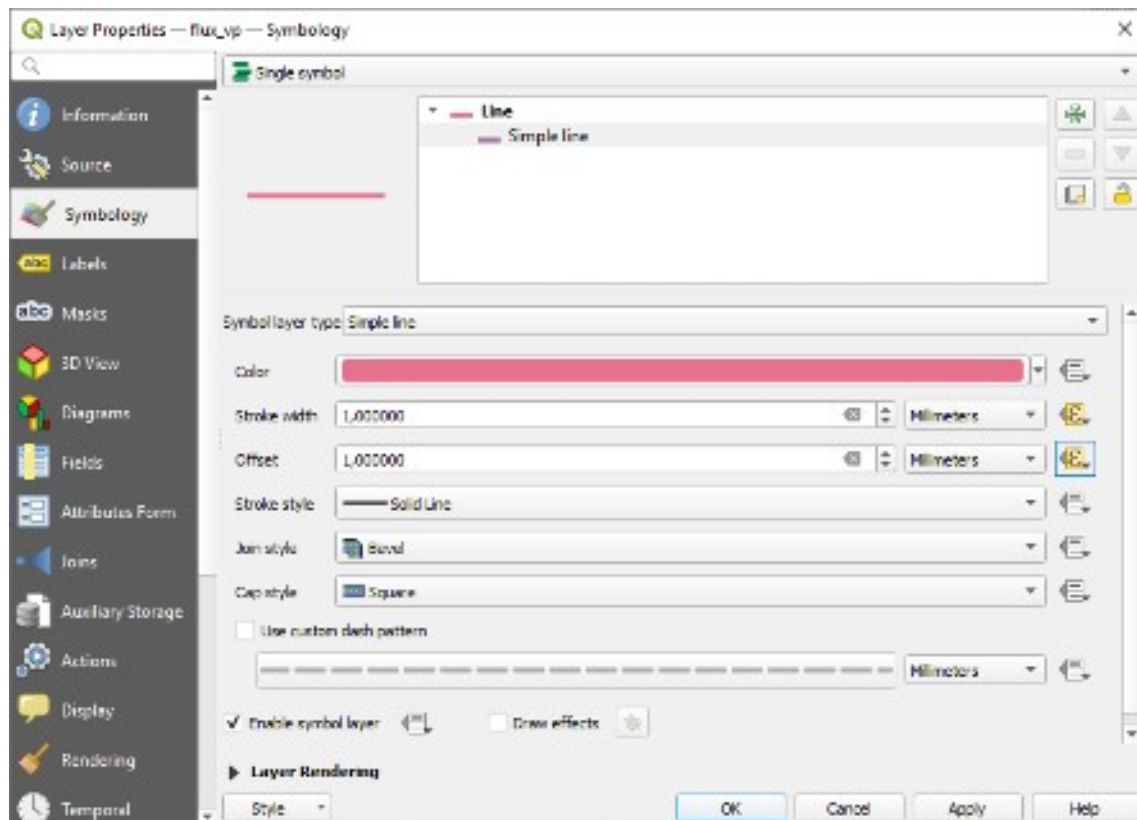
	i	j	ij	volu
1	v182533320228...	v182534205228...	v182533320228...	271,0000
2	v182319291228...	v182326723228...	v182319291228...	56,00000
3	v182074827229...	v182084313229...	v182074827229...	5644,000
4	v182475884228...	v182480714228...	v182475884228...	4444,000
5	v181562335229...	v181573797229...	v181562335229...	66,00000
6	v182360066229...	v182396484229...	v182360066229...	670,0000
7	v181663081228...	v181653220228...	v181663081228...	34,00000
8	v181814382229...	v181819873229...	v181814382229...	1572,000
9	v181431278229...	v181414171229...	v181431278229...	83,00000
10	v182685000229...	v182693442229...	v182685000229...	113,0000
11	v180499802226...	v180506992226...	v180499802226...	23,00000
12	v181873015230...	v181866836230...	v181873015230...	128,0000
13	v182428572229...	v182430210229...	v182428572229...	587,0000
14	v182481685229...	v182462523229...	v182481685229...	102,0000
15	v184065303229...	v184067513229...	v184065303229...	36,00000
16	v182620965228...	v182631450228...	v182620965228...	168,0000

Show All Features

flux-vp data table

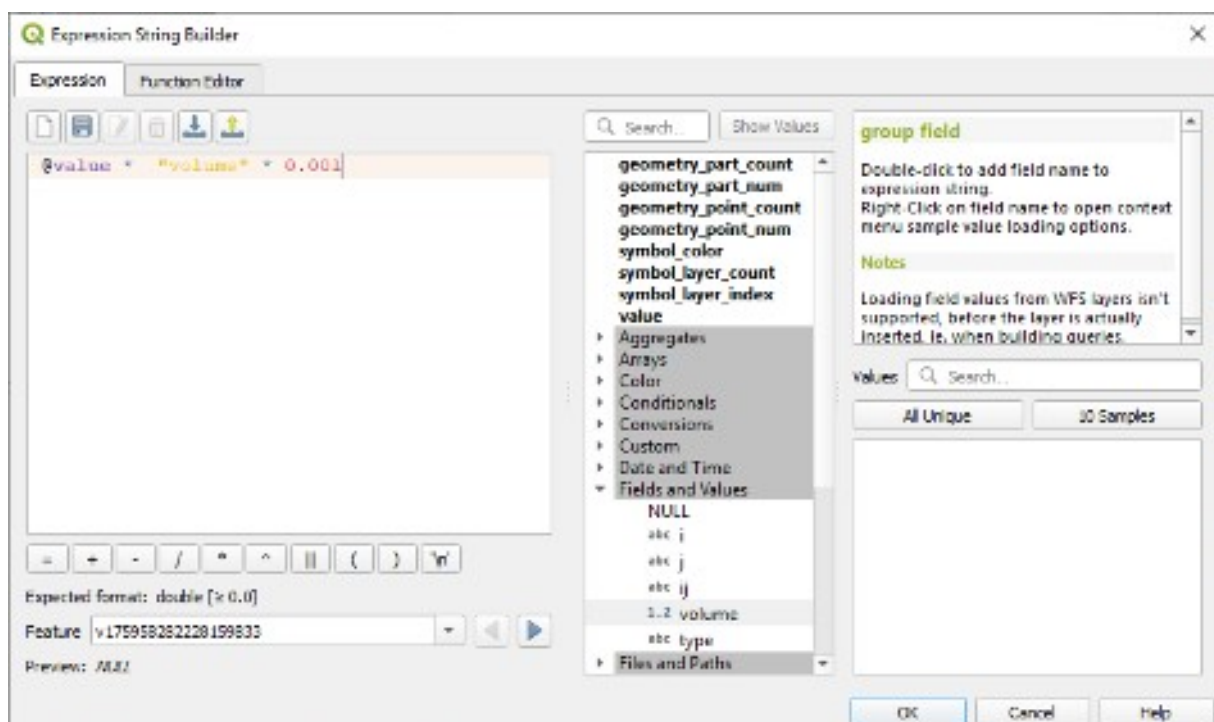
## 2. Define proportional widths

In layer properties choose symbology to **define the line style**. Numeric values of « **stroke width** » and « **Offset** » must be identical



## 3. Set stroke width with an expression

Stroke width




Then set offset as volumes are defined by traffic directions. Then set **offset** expression as traffic is depending of directions. The **+0.2 value** is useful in order to visually differentiate both directions.

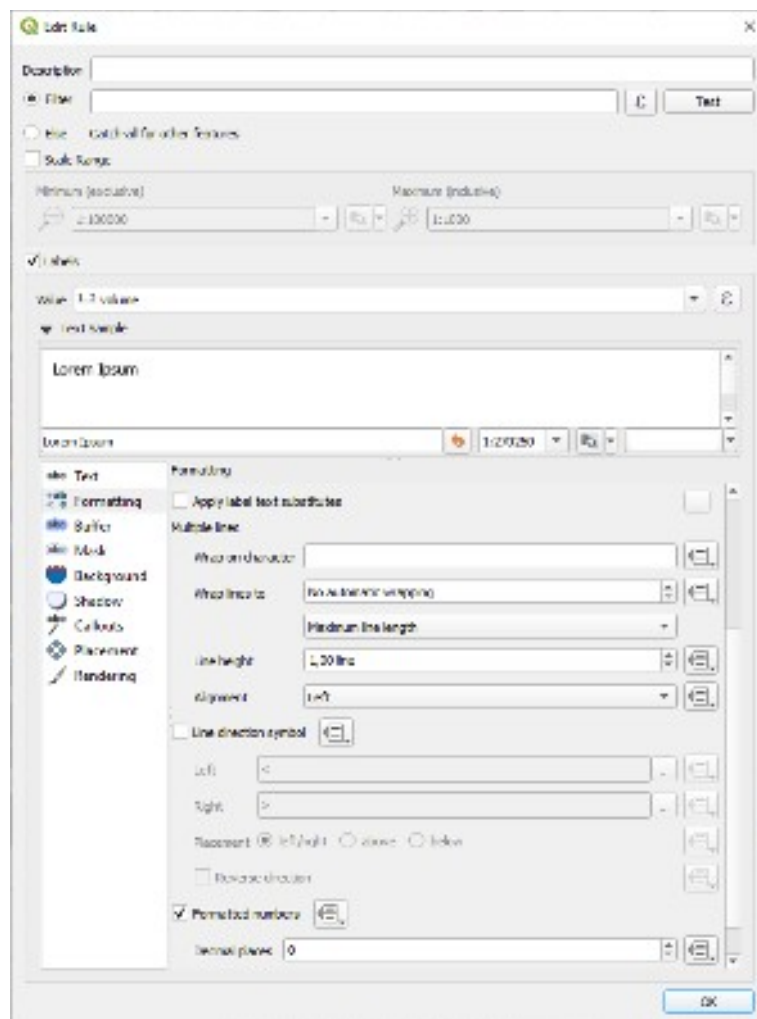






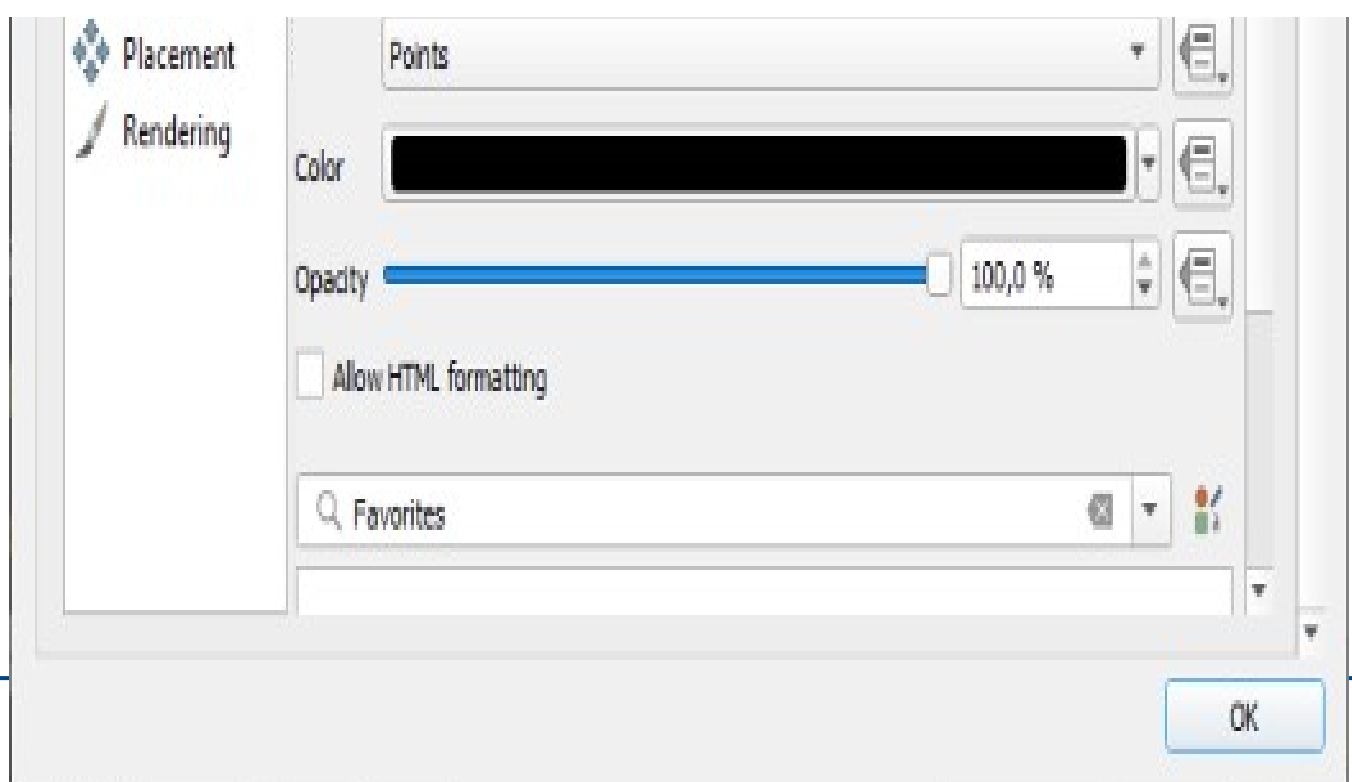
## 4. Show traffic volumes flows

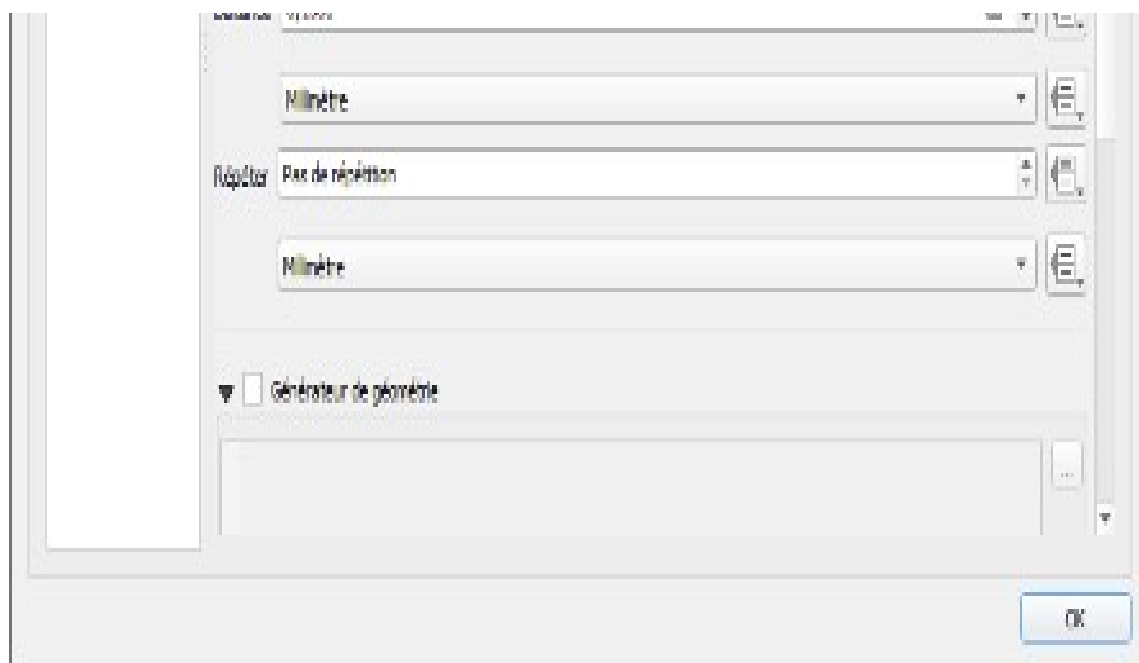
Create **rule based labels** because you will filter small figures later. It is important to click on the  in order to add a rule, even if there is no filter in a first step.



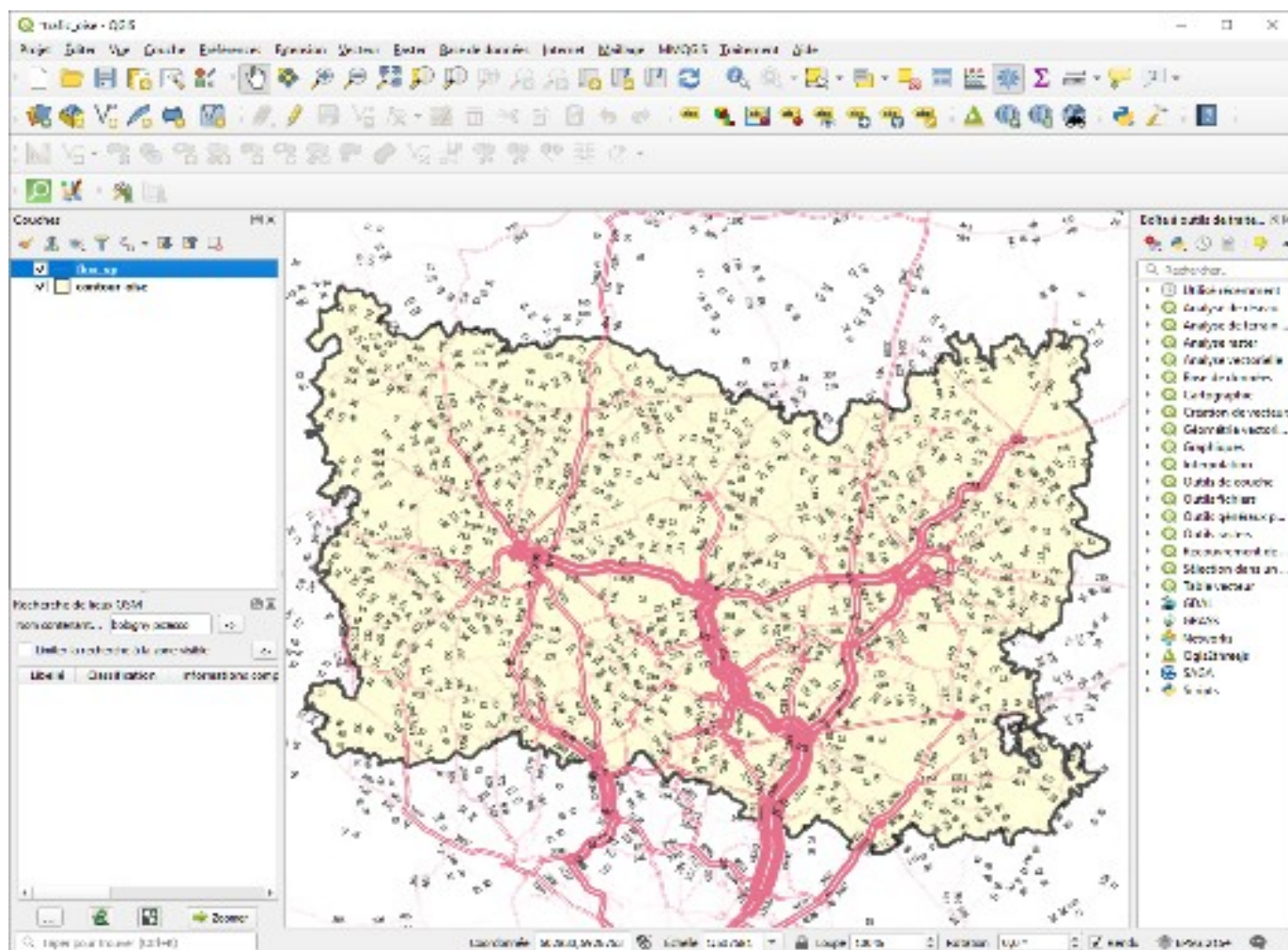
Formatting text

Possibilité de définir la position de l'étiquette





## Result



The map has too many labels and is becoming unreadable. Moreover, the precision to the nearest unit is illusory considering the survey sample rate.

We will filter labels for volumes < 2000, **merge labels** for adjacent links sharing the same value and round volumes to the nearest hundred.

**Edit Rule**

Description:

☒ **Filter** Value >= 2000

☐ **Else** Catch-all for other features

☐ **Scale Range**

Minimum (exclusive):

Maximum (inclusive):

☒ **Labels**

Value:

**Text Sample**

Text:

Text:

**Formatting**

**Multiple lines**

Wrap on character:

Wrap lines to:

Maximum line length:

Line height:

Alignment:

☐ **Line direction symbol**

Left:

Right:

Placement: ☒ left/right ☐ above ☐ below

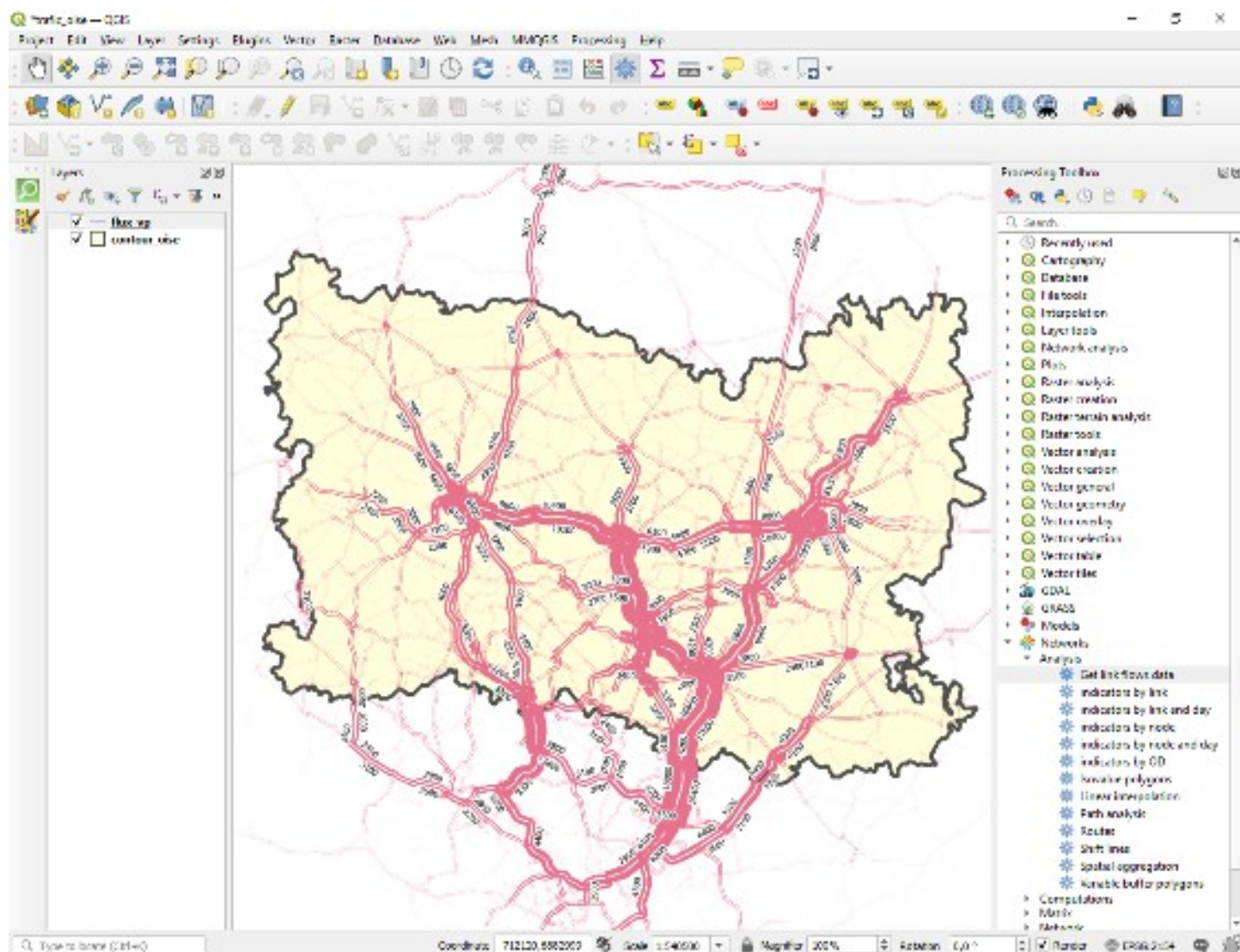
☐ **Reverse direction**

☒ **Formatted numbers**

Decimal places:

Define the **same distance as the stroke width and offset values**. Check **“Line orientation dependent position”** and **“Right of line”** for a traffic map-like rendering

## Result



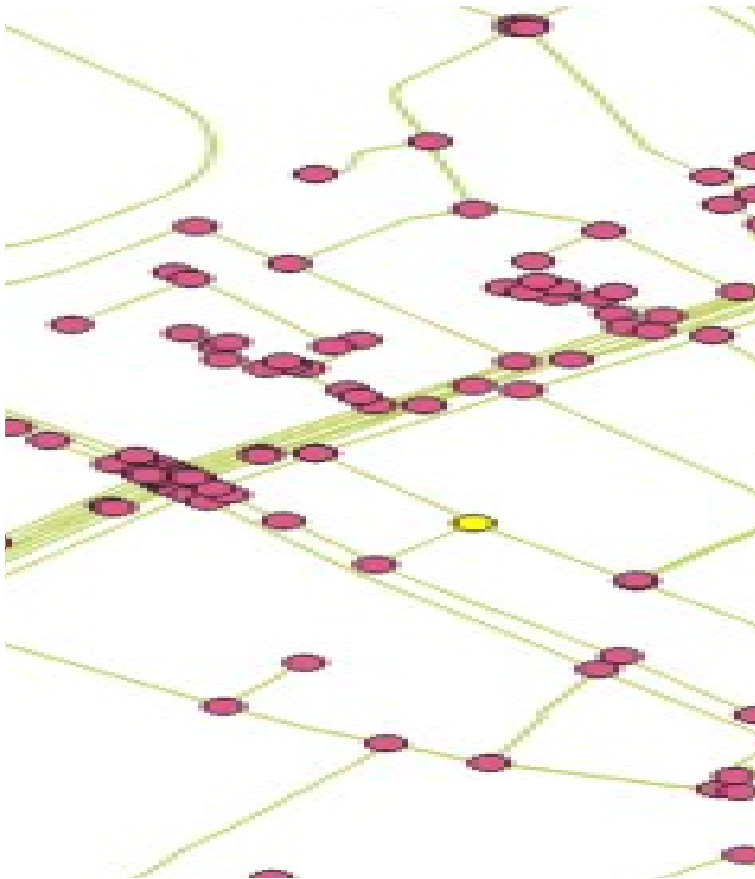
# **D- Create a territorial accessibility map on a time period and/or from/to several points**

## **1. Create the accessibility map and the area of influence map**

### **1.1. Matrix creation**

Example with a point during a period (several arrival or departure times)

**Select a point** (in yellow on the map)



**SCRIPT** **Musliw matrix simple list**

nodes : **select the nodes layer** (the layer where the node is selected)

Check **Selected features only**

Fill **start\_time** and **End time fields** of the period : example 16:00:00 – 17:00:00



Fill **time interval in minutes** : example 15,0000

Check **diagonal matrix** (same origin, same destination.

Save matrix as.

**Musliw Matrix Simple List**

Parameters Log

Nodes  
G:/BD\_DREAL/2019/Voie\_Qgis/DEPT\_59\_ [Refresh] [Help] [Browse]

☐ Selected features only

Node ID  
abc num

Demand  
1,000000

Day  
1

Start time  
16:00:00

End time  
17:00:00

Step  
15,000000

Departure/Arrival  
Departure

☒ Diagonal matrix?

☐ OD label?

Musliw matrix  
G:/poubelle/matrice\_point\_16h\_17h.txt [Refresh] [Browse]

0%

Run as Batch Process... Run Cancel Close

**Musliw matrix simple list**

Generates a Musliw matrix from a point layer and a period of time (from start time to end time with a step in minutes)

the script generates a full square matrix (N od) or a diagonal matrix (N od with same origin and same destination)

Parameters:

Nodes: nodes layer (corresponding to nodes layer or the graph)

Node Id: Field that contains the node Id

Demand: number of passengers for assignment

Day: number of the day in the calendar (1 first day of the calendar)

Start time: Beginning of the time period

Step: Step time in minutes

Departure/Arrival: Departure (from Start point to end point forward) - Arrival (from end point to start point backward)

Diagonal matrix: Check if you want only a diagonal matrix instead of a full square matrix

OD label: If True an origin-destination ID will be written combining o and d IDs separated by a '-'

Musliw matrix: Musliw matrix name (text file with ";" separator)

**Result: The generated matrix**

matrice\_point\_16h\_17h.txt - Bloc-notes

Fichier Edition Format Affichage ?

m183098922230661500;m183098922230661500;1.0;1;960.0;d

m183098922230661500;m183098922230661500;1.0;1;975.0;d

m183098922230661500;m183098922230661500;1.0;1;990.0;d

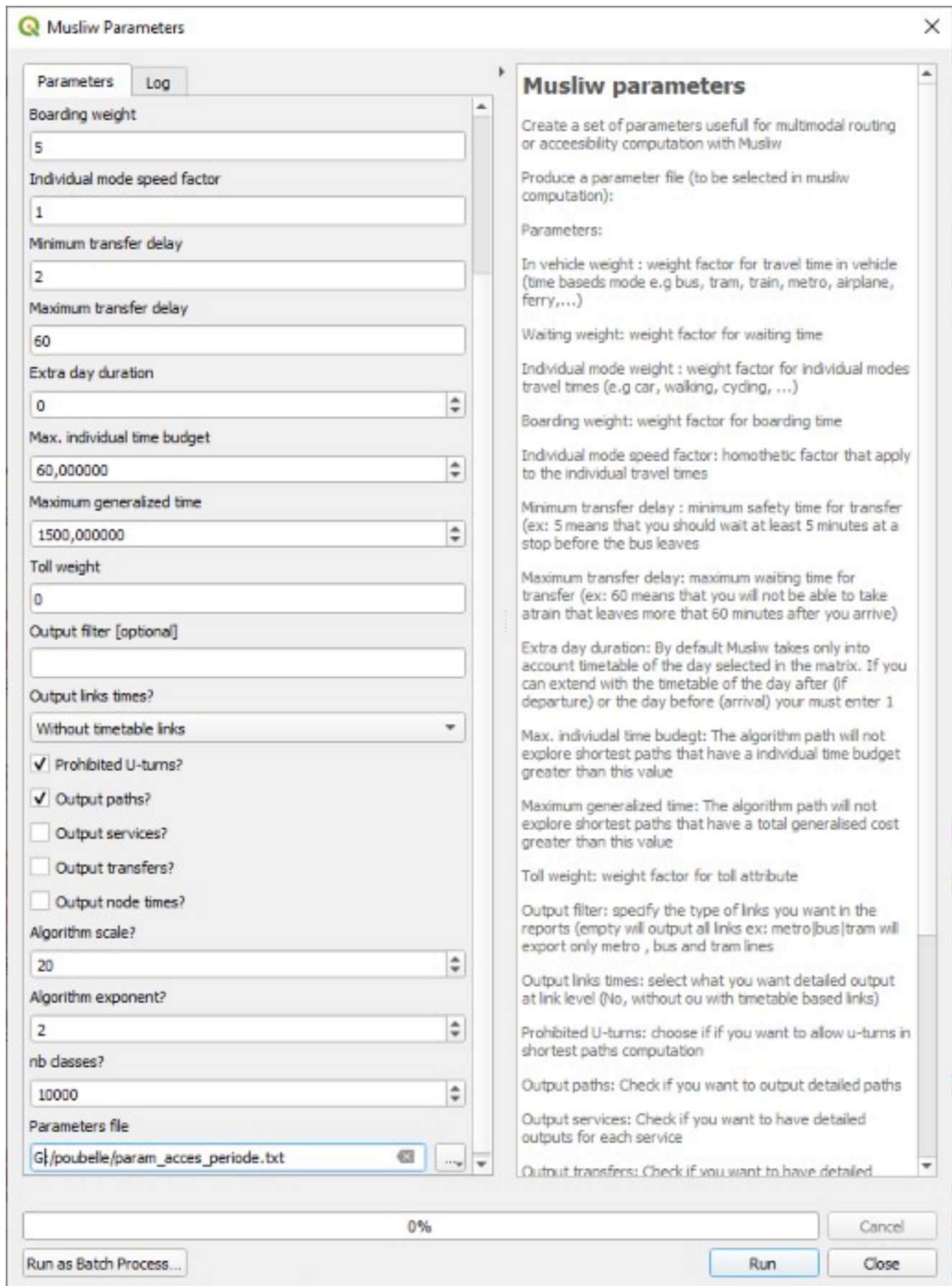
m183098922230661500;m183098922230661500;1.0;1;1005.0;d

Ln 5, Col 1



## 1.2. Customize Musliw computation parameters

In the field “output link times” choose « **Without timetable links** »



**Musliw Parameters**

Parameters Log

Boarding weight  
5

Individual mode speed factor  
1

Minimum transfer delay  
2

Maximum transfer delay  
60

Extra day duration  
0

Max. individual time budget  
60,000000

Maximum generalized time  
1500,000000

Toll weight  
0

Output filter [optional]

Output links times?  
Without timetable links

☒ Prohibited U-turns?

☒ Output paths?

☐ Output services?

☐ Output transfers?

☐ Output node times?

Algorithm scale?  
20

Algorithm exponent?  
2

nb classes?  
10000

Parameters file  
G:\poubelle\param\_acces\_periode.txt

0%

Run as Batch Process...

Run Cancel Close

**Musliw parameters**

Create a set of parameters usefull for multimodal routing or accesibility computation with Musliw

Produce a parameter file (to be selected in musliw computation):

Parameters:

In vehicle weight : weight factor for travel time in vehicle (time baseds mode e.g bus, tram, train, metro, airplane, ferry,...)

Waiting weight: weight factor for waiting time

Individual mode weight : weight factor for individual modes travel times (e.g car, walking, cycling, ...)

Boarding weight: weight factor for boarding time

Individual mode speed factor: homothetic factor that apply to the individual travel times

Minimum transfer delay : minimum safety time for transfer (ex: 5 means that you should wait at least 5 minutes at a stop before the bus leaves

Maximum transfer delay: maximum waiting time for transfer (ex: 60 means that you will not be able to take atrain that leaves more that 60 minutes after you arrive)

Extra day duration: By default Musliw takes only into account timetable of the day selected in the matrix. If you can extend with the timetable of the day after (if departure) or the day before (arrival) your must enter 1

Max. indiidual time budegt: The algorithm path will not explore shortest paths that have a individual time budget greater than this value

Maximum generalized time: The algorithm path will not explore shortest paths that have a total generalised cost greater than this value

Toll weight: weight factor for toll attribute

Output filter: specify the type of links you want in the reports (empty will output all links ex: metro|bus|tram will export only metro , bus and tram lines

Output links times: select what you want detailed output at link level (No, without ou with timetable based links)

Prohibited U-turns: choose if if you want to allow u-turns in shortest paths computation

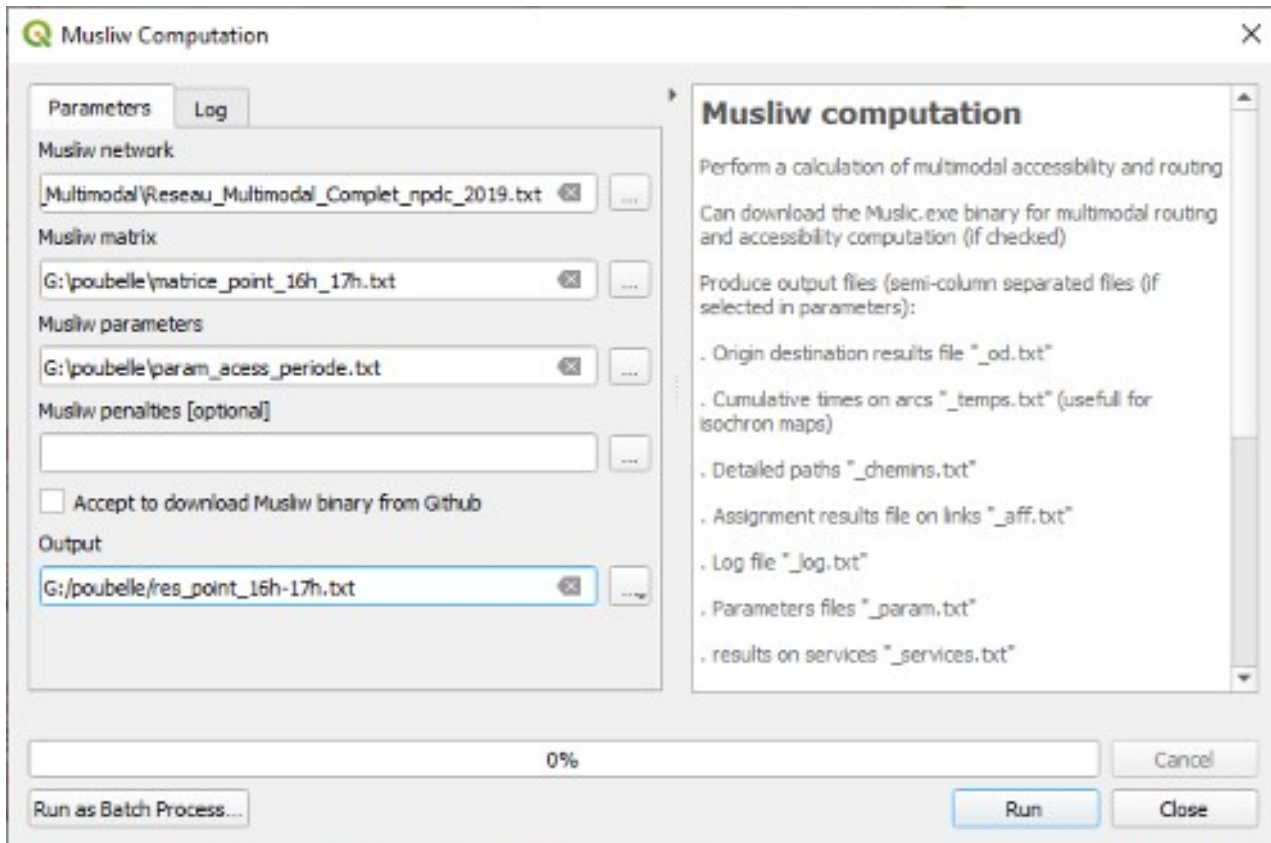
Output paths: Check if you want to output detailed paths

Output services: Check if you want to have detailed outputs for each service

Output transfers: Check if you want to have detailed

### 1.3. Musliw computation

SCRIPT **Musliw computation**



### 1.4. Compute travel times aggregation

From Musliw computation out files, we are going to use the **travel times output file**.

In this file, there are several travel times for each link, because there are 4 simulations during the period.

Here, we have at most 4 travel times per link, as there are 4 simulation during the period (one simulation every 15 minutes) (cf matrix)


Then, we can't represent directly the isochrones. It is necessary to perform an statistic analysis (min, mean, max, ...)

To do this we will use the SCRIPT **indicators by link** (You can also use if you want "indicator by node" if you want a representation based on nodes)

Link times output files : **fichier temps** (computation result)

Variable : choose **temps**

**Check Remove initial/final waiting time**

 Indicators by Link

ParametersLog

Link times output file

G:\poubelle\res\_point\_16h-17h\_temps.txt

Variable

temps

Filter

1

☒ Remove initial/final waiting time?
   
☐ Time based links only?

Link indicators file

G:/poubelle/synthes\_arc\_point\_16h\_17h.txt

indicators by link

Produce indicators aggregated by link. This analysis is usefull when you want to study the variation of travel times during a time period

Parameters:

link times ouput file: the Musliw link times output file (\_temps.txt)

variable: The name of the variable for indicator computation (temps (time) by default)

filter: expression to filter nodes times outfile (ex: ncorr

remove initial/final boarding time: If checked the initial or final waiting time (between the excepted arrival or departure time

and the real one is subtracted from the total travel time

time based links only: If checked only time based links are analysed

0%

Run as Batch Process...

Run

Cancel

Close

## 1.5. Update aggregated ti tj

The following process is similar as a standard creation of a territorial accessibility map, except that for the **“update of ti tj”** script, you must choose which time we want to represent in the variable **«Musliw time »** (the minimum travel time : min, the average travel time : avg, the maximum travel time : max).

**Update Ti Tj**

Parameters Log

Network  
G:/BD\_DREAL/2019/Voie\_Qgis/DEPT\_59\_Nord/DEPT\_...

Travel times file  
G:\poubelle\synthes\_arc\_point\_16h\_17h.txt

Musliw time  
avg

Arc time  
1.2 Tps\_marche

Departure/arrival  
Departure

i-node time  
tj

j-node time  
tj

☐ Initial/final waiting time?

0%

Run as Batch Process... Run Cancel Close

**Update ti tj**

Read the travel times file "...\_temps.txt" computed by Musliw and creates (if they don't exist) in the network layer fields where i-node and j-node travel times are saved

Parameters:

layer : network layer (linear objects)

travel times file: travel times text file ...\_temps.txt generated by Musliw

musliw time: Musliw travel time field where travel time is saved ('temps' in general)

arc time: arc travel time

departure/arrival: departure if "d" in Musliw matrix, arrival if "a"

i\_node time: travel time at i-node field

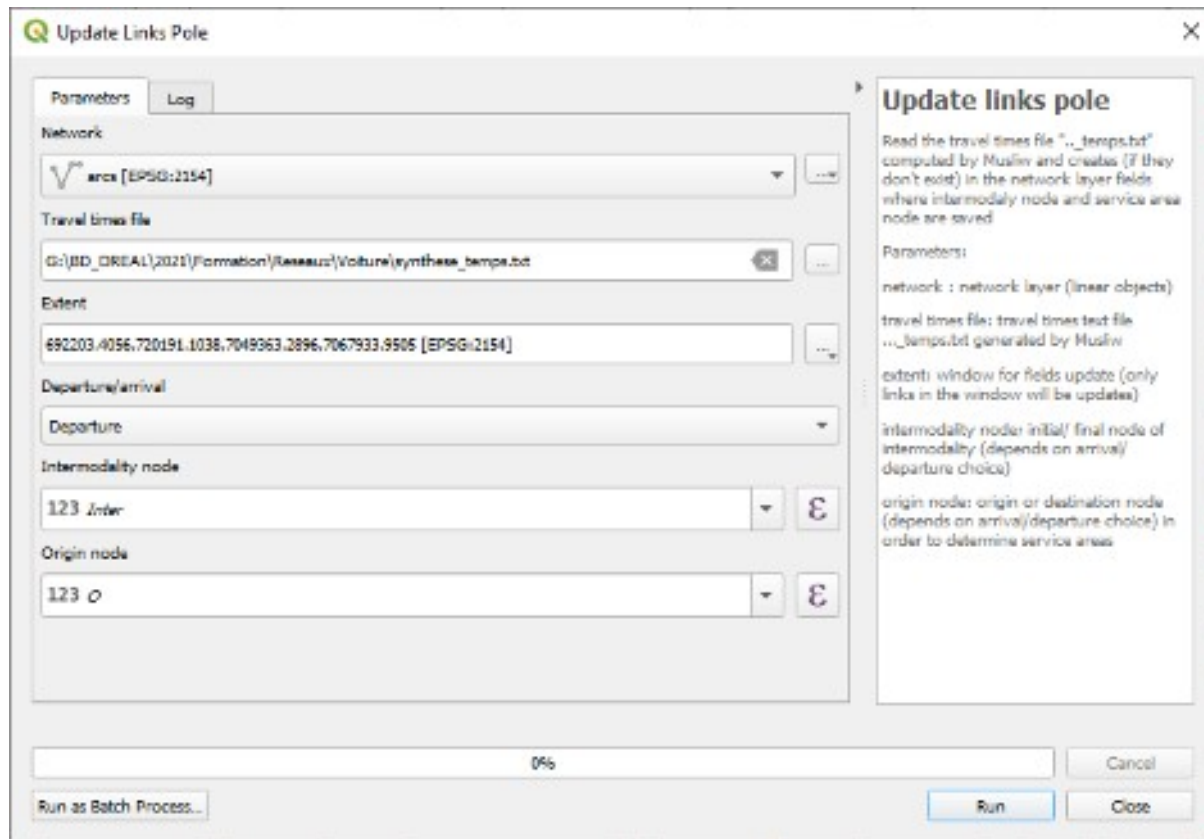
j-node time; travel time at j-node field

inital/final waiting time: in order to take into account or not inital/final waiting time (tatt1)

## 1.6. Update of the links pole

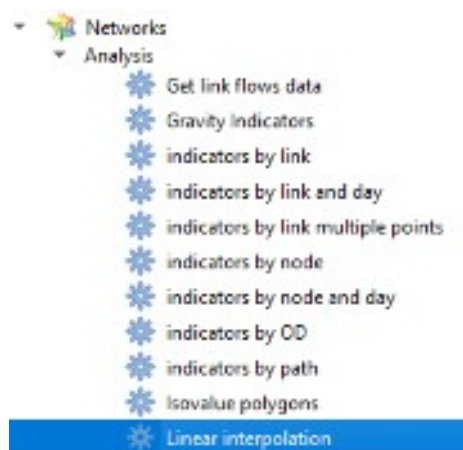
- Network
  - Auto-connectors
  - Build connectors
  - Build graph
  - Concatenate network files
  - Concatenate networks folder
  - Connect nodes to lines
  - Create individual networks arcs
  - Create updated links
  - GTFS import
  - Isolated nodes
  - Musliw individual network
  - Musliw timetable network
  - Prepare GTFS
  - Reverse arcs
  - Reverse selection geometries
  - Update field
  - Update links pole**

If you also want to create a map of the areas of influence or catchment areas (if there are several points), you must then update for each arc the pole to which it is attached, i.e. the node identifier from which the travel time is the smallest



To make the map, start again from A- Making an accessibility map, part 2 (making the accessibility map), level 5 (starting the linear interpolation).

## 1.7. Linear interpolation



To be able to create the map of areas of influence or catchment areas, you must, in addition to the identical settings made for the single-point map, fill in the individual values field.

You must indicate the identifier that defines your areas of influence.

It is a text variable:

For example:

O, O\_min,... For the closest starting or ending point



Pole, pole\_min, ... : For the intermodality point

**Linear Interpolation**

Parameters Log

Network  
Vues [EPSG:2154]

Window  
692283.4056,720091.1038,7049363.2896,7067933.9505 [EPSG:2154]

Icost  
1,2 s

Jcost  
1,2 s

Direction  
2

Spread  
123 3

Impossibility  
123 3

Pixels nb x  
100

Pixels nb y  
100

Pixel size x  
-1,000000

Pixel size y  
-1,000000

Decimals  
5

Radius(m)  
500,000000

Spread speed  
123 4,0

☐ Impossible?

Individual values  
Q\_min

Raster file  
G:\BD\_OREAL\rdjongo\reseaux\vp\raster.tif

☒ Open output file after running algorithm

Linear interpolation

Implements a linear based interpolation in order to build a raster representing isovalue from a linear objects file and travel times at i-node and j-node

Parameters:

network : network layer  
window : working area  
i-node cost : cost at node i  
j-node : cost at node j  
direction : flow direction ('0','1','2','3'): '0' prohibited, '1' flow in the object direction, '2' flow in the reverse object direction, '3' flow in both directions  
spread : side of spread inside blocks ('0','1','2','3') ('0' spread prohibited, '1' right-side spread only, '2' left-side spread only, '3' both sides spread  
impossibility : impossibility ('0','1') ('0' impassable road, '1' traversable road)  
number of pixel x : number of pixels in x of the output raster  
number of pixel y : number of pixels in y of the output raster  
pixel size in x : pixel size in x(m) (optional)  
pixel size in y : pixel size in y (m) (optional)  
decimals : number of decimals for approximation (e.g. 6 correspond to 1e-6)  
radius : search radius m inside blocks  
spread speed : speed of spread inside blocks in km/h (60 for isodistance maps)  
impossible : when selected impassable elements are taken into account for isovalue computations  
individual values: (Optional) field for individual values polygons (ex: stations access area)  
result : output raster layer

0%

Run as Batch Process...

Run Cancel Close

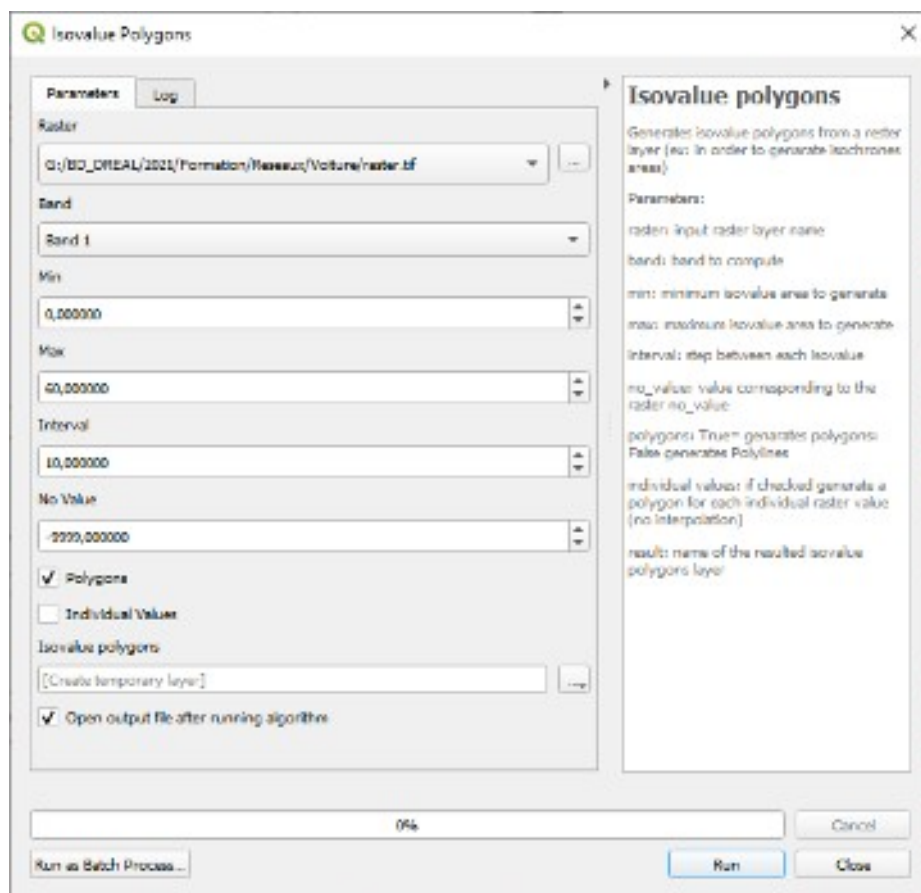
## 1.8. Drawing of isovalue polygons and areas of influence.

For the drawing of isovalue polygons, the settings are identical to those of the single point map. For the drawing of the areas of influence or catchment areas, the "individual values" option must also be checked.

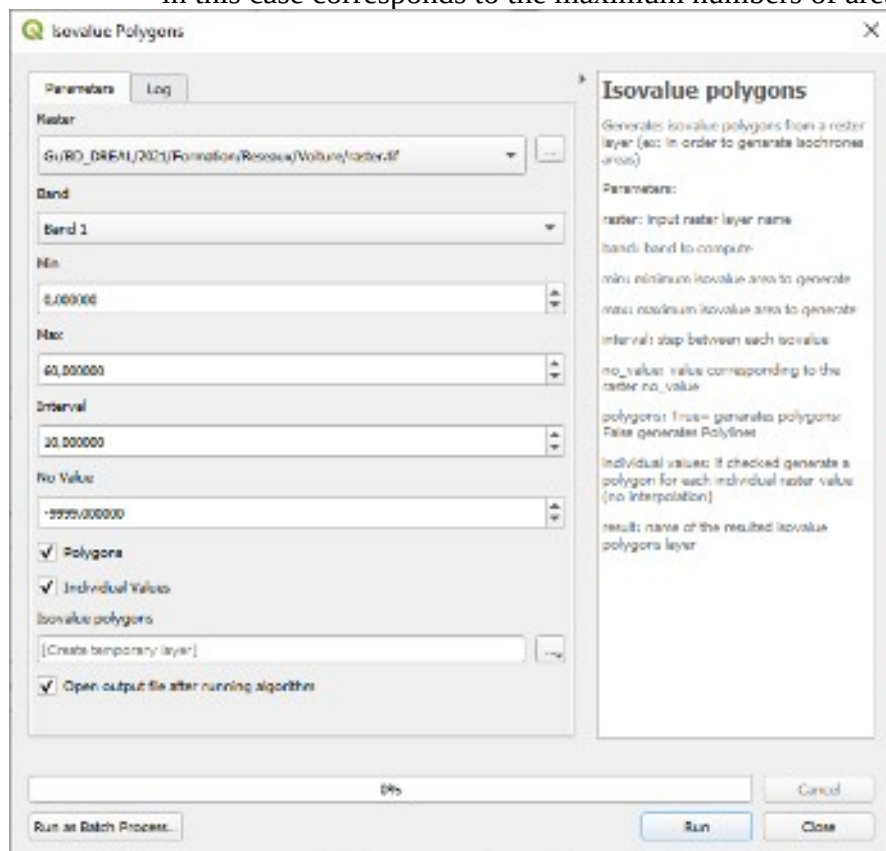
So, to have both maps, you have to launch the "iso value polygons" algorithm twice

- Once with the box "individual values" unchecked for the drawing of iso-value polygons





- Once with the box "individual values" checked for drawing areas of influence
  - For areas of influence if their number is large, it is necessary to increase the max parameter, which in this case corresponds to the maximum numbers of areas of influence to be represented



# **E- Simplified accessibility map**

The interest of developing plugins in the form of processing algorithm managers is to :

- Perform batch processing: Run the algorithm in batch several times
- Build models by chaining the algorithms together (can also be run in batch)

The realization of accessibility maps requires to execute successively several scripts or algorithms.

If the understanding of the process of realization of these maps is important for the consultant so that he can understand the process, the parameters that he can modify, it seems useful to have a tool that allows to realize the maps in an automatic way with predetermined default parameters.

For this purpose, Cerema has developed, thanks to the model tool of the processing box, 3 models (1 per network) to generate accessibility maps very simply.

- isochrone\_multimodal (multimodal network with detailed road network, large scale)
- isochrone\_multimodal\_light (multimodal network with simplified walking road network, small scale)
- isochrone\_route500 (car road network, medium scale)

## **1. Installation**

Each model is delivered in the form of a directory containing :

- Musliw network
- GIS network (arcs, nodes)
- Musliw parameters file
- Qgis style file
- model file (.model3)

Copy each directory to your disk:

Copy the model files (.model3) to your processingmodels directory: C:\Users\<username>\AppData\Roaming\QGIS\QGIS3\profiles\default\processing\models

## **2. Settings**

The principle of models in Qgis is to be able to chain several algorithm scripts.

A model is made of :

- algorithms (scripts)
- inputs (parameters to be entered by the user)
- links between algorithms



Each parameter of each script can be defined by either :

- a value: constant defined by the user for the parameterization (e.g. "num" for the name of the node number in the node table)
- an input (here origin for the start and end points)
- an expression
- the result of another algorithm

The user must initially define all the parameters of each algorithm to adapt them to his needs and context (file paths, day, time, number of pixels, etc.)

He can also add inputs if he wants the user to enter a parameter at each use. For example, here the time has been set to 9am. It could be useful to add the time input so that the user has to specify the desired arrival or departure time each time.

Once the template has been modified and customized, the changes must be saved. However, the models are only loaded when Qgis is started. In order to use the modified version, it must also be loaded into the project (green icon to the right of "save as"), otherwise the old version is used.

### 3. Making the map

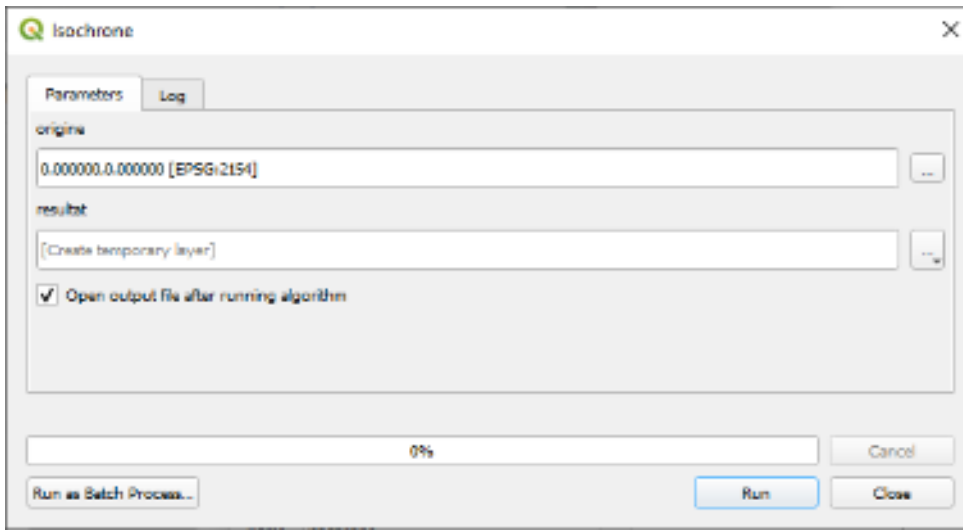
To make the map, you need to run the template in question as a script. They are located in "Models".

First of all, you need to define in the map window the area of study of the map. The map will be defined on the visible part of the map only



We can see that now only two parameters are required for the accessibility card multimodal accessibility map :

- the origin: point to click on the map
- the result : vector file in which the isochrones will be generated



All other parameters are defined by the user beforehand by default. The user can always change them if he wishes but he will have to save the model again, or add an input if this parameter is to be changed often.

## 4. Tips for setting parameters

When building models composed of a succession of several scripts, it may be interesting to set the paths of frequently used tables or files dynamically.

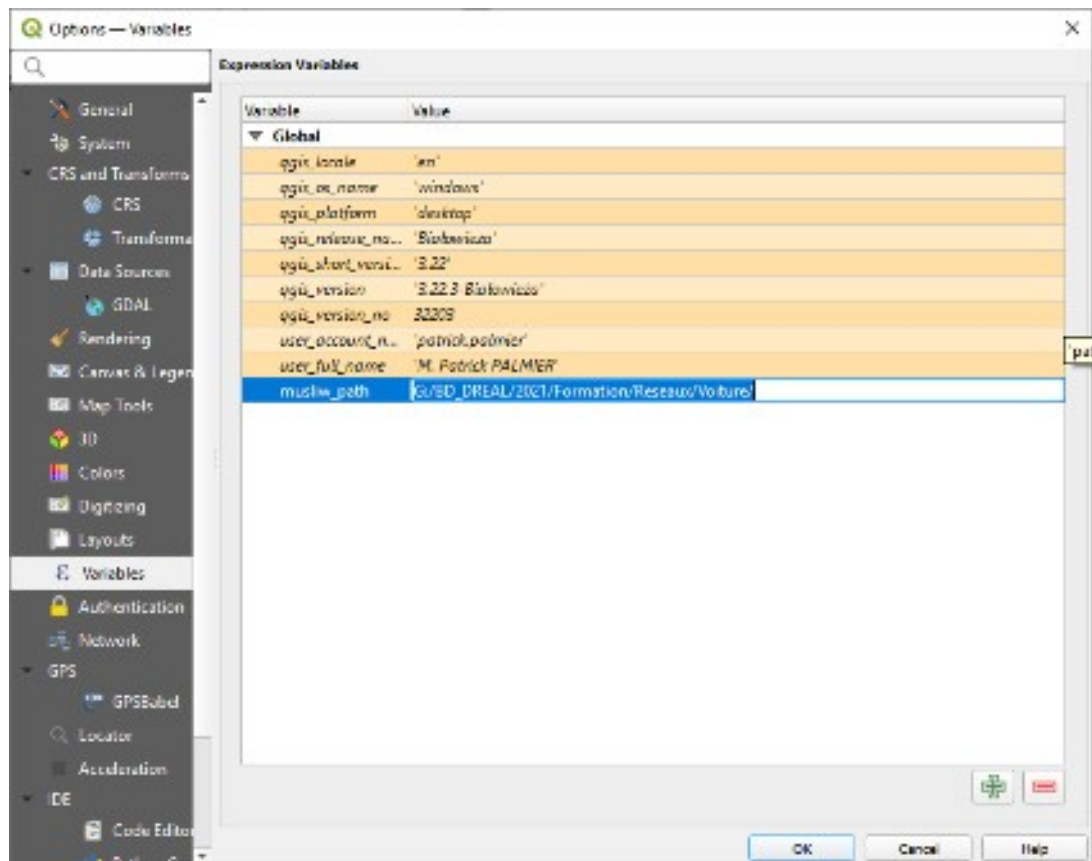
In the creation of accessibility maps in particular, the user may need to create maps for different modes (walking, cycling, public transport, car, ... and on several study areas).

For this purpose, an interesting method consists in naming the files used for each mode in the same way and placing them in a different directory with the name of the mode, for example

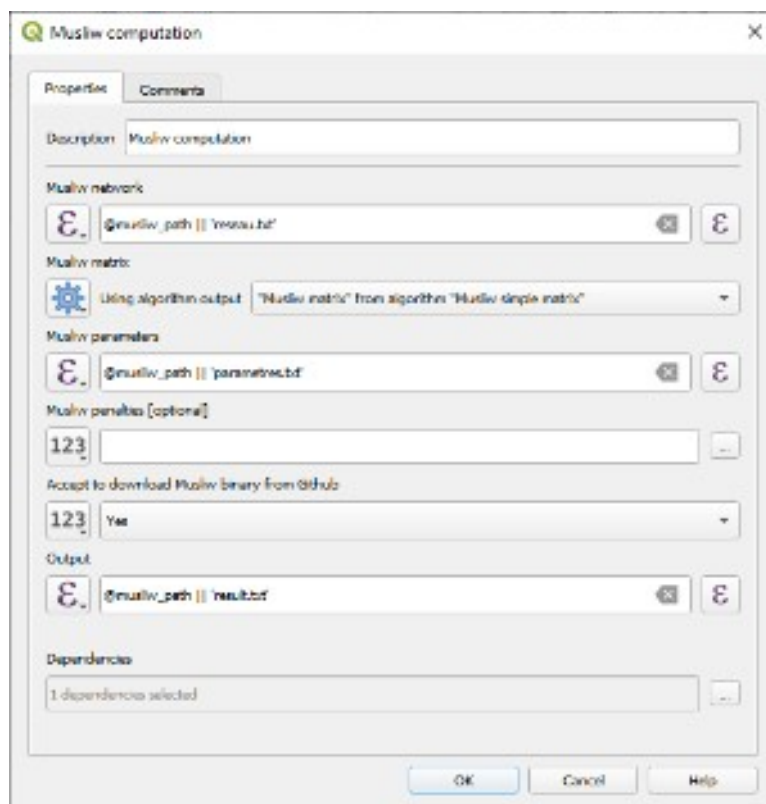
- reseau.txt : Multimodal network in Musliw format (txt)
- arcs.gpkg : The table of the road network used by the mode we want to study (walking, car, bicycle, ...)
- nodes.gpkg : The table of nodes associated with the arcs.gpkg table
- parametres.txt : The calculation parameters of the mode in question
- result.txt : The name of the result file

Then the trick is to create a variable in Qgis that will point to the directory of the mode you want to study. For example, here, we must add the variable "musliw\_path".





In the following two images you have an example of how to parameterize directory names using the @musliw\_path variable. You have to define the parameter as an expression and concatenate the name of the variable with the name of the file (same as above for each directory).



Linear interpolation

Properties Comments

Decline  
123 5

Radius(m)  
123 testvalue

Spread speed  
123 4.0

Impossible?  
123 No

Individual values  
123 1'

Raster file  
E: \msubv\_path \ raster.tif

Dependencies  
1 dependencies selected

OK Cancel Help

# F- Create an accessibility map based on a grid

## 1. Advantages and drawbacks

The main advantage of this method is that the realization of the map is much faster since it only requires the steps of creating the matrix and the calculation and then a thematic analysis.

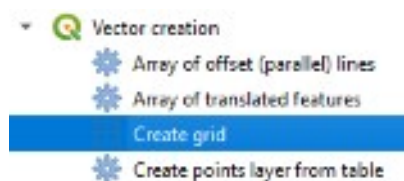
Moreover, this method is particularly interesting when there is a need to cross-reference accessibility results with other data (e.g. gravity accessibility) since all the necessary data can be disaggregated to the level of each grid cell.

However, as the grid is fixed, it is scale dependent. If we zoom in too much, the meshes become too big and this gives a pixelated effect and a loss of precision. Finally, as the grid is fixed, the initial step of creating the grid and preparing the data is a bit time consuming.

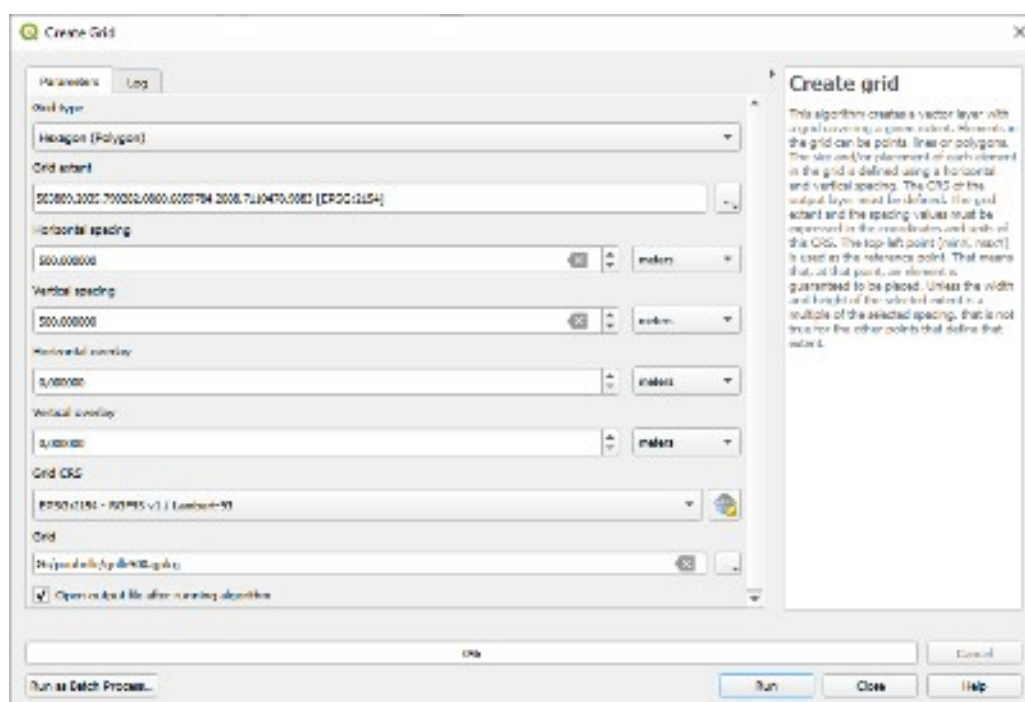
## 2. Creating the grid

The grid can be created in several ways.

To stay with the providers of processing algorithms, we can use the Qgis script "Create grid" located in the "Vector creation" group

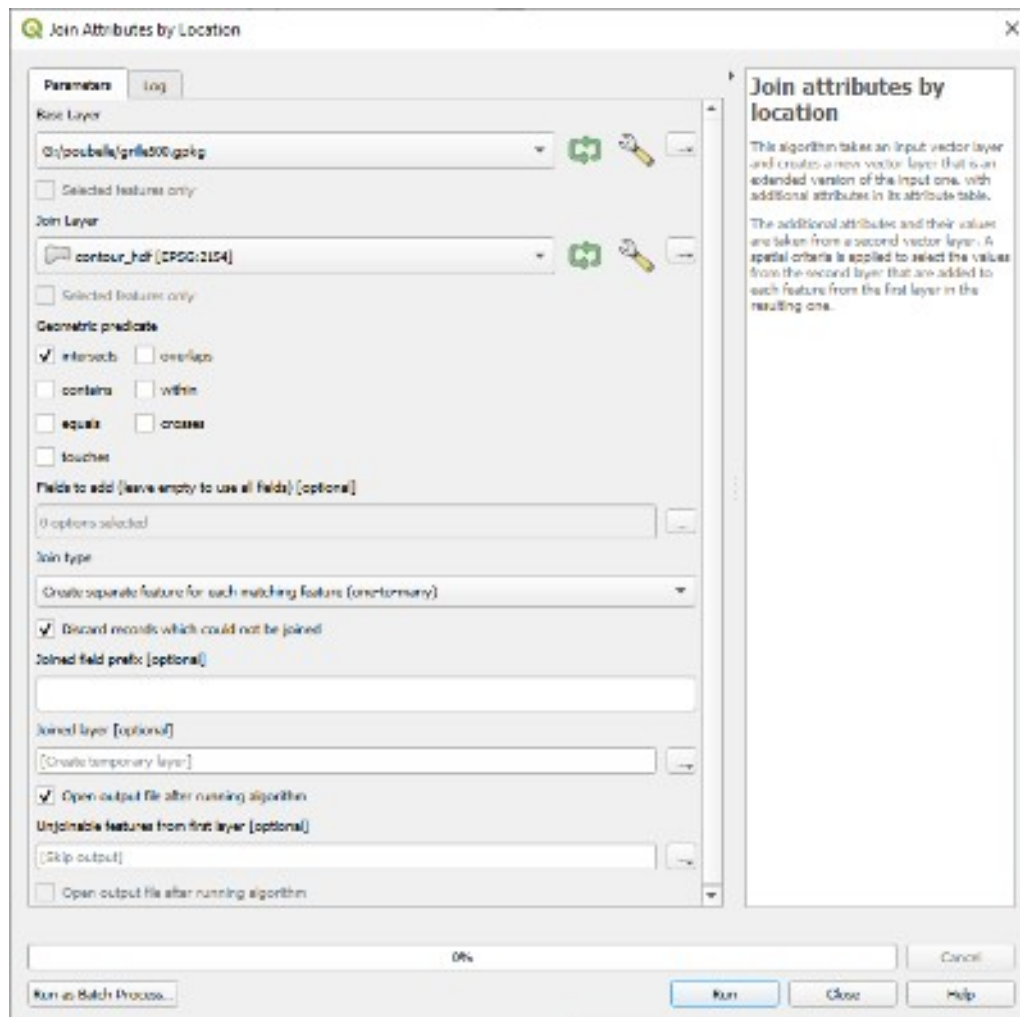
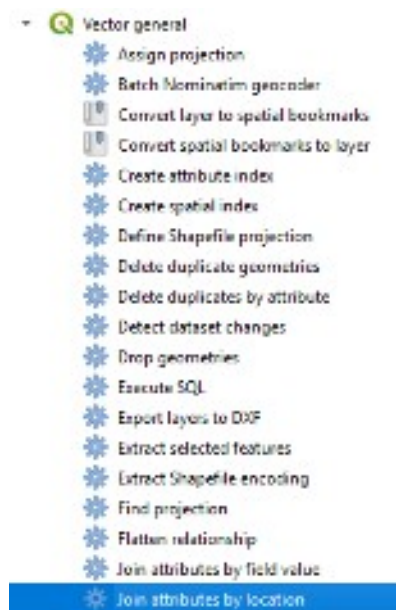


The following script generates a hexagonal grid on a defined area from a loaded layer (Hauts-de-France Region) with a spacing of 500m.



### 3. Cut the grid according to the outline of the study area

The grid can then be lightened by deleting the cells located outside the study area (here the Hauts-de-France Region) with the script “join attributes by location”

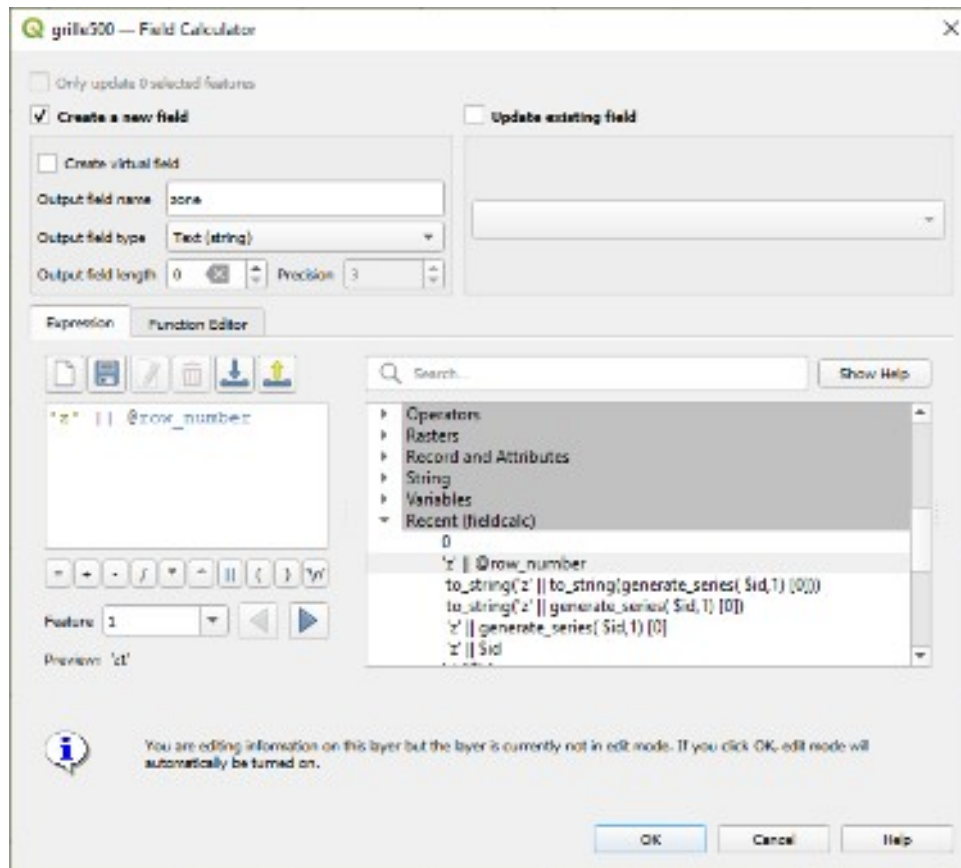


Make sure to check "delete records that cannot be joined".

## 4. Create a zone identifier

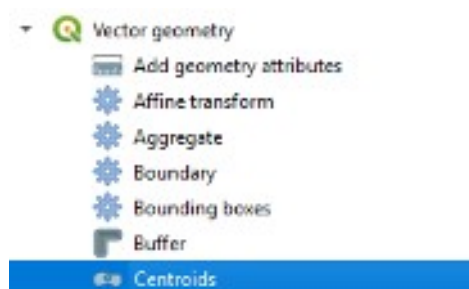
Then you have to add a field in the grid corresponding to a cell identifier (or zone identifier)

To do this, you must create a text field "zone" for example defined by the expression `'z'+@row_number`

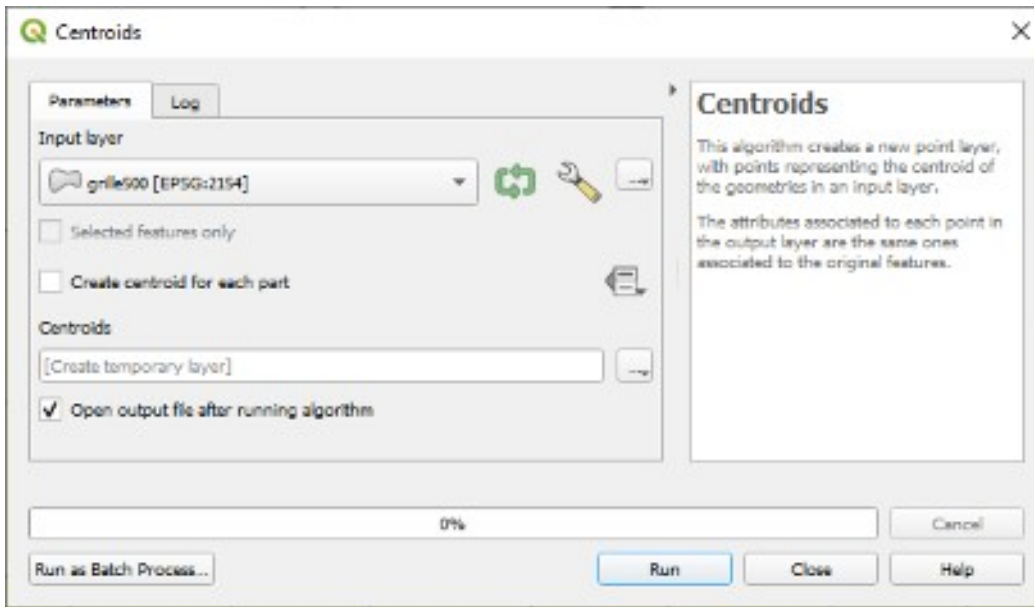


## 5. Make a copy of the grid in centroids

The objective is to have each grid cell connected to the road network. This means treating each cell as a point of interest (POI). To do this, it is necessary to create a layer of points corresponding to the centroids of each cell of the grid. To do this, we must use the "centroids" algorithm.







## 6. Create the grid - road network connectors

In order to integrate the centroids of the grid cells into the accessibility calculation, it is necessary to generate the connectors between each of the centroids and the nodes of the road network.



Special attention must be paid to the values entered for "node mode i" and "node mode j".

In the example below, 'z' is entered for 'node mode i' and 'm' for 'node mode j'. This means that the connectors :

- from zone 'z' to road node 'm' will have the type 'zm'
- from road node 'm' to zone 'z' will have the type 'mz'.

These types will be the ones used in the calculation parameters to filter the results only on the zones and not on the whole network.

Build Connectors
×

Parameters

Log

Stops

G:\BD\_DREAL\2021\Formation\Grille\grille\_hdf\_centroids\_4326.gpkg

☐ Selected features only

Stop\_id

abc num

i-text

z

i-mode

z

Nodes

G:\BD\_DREAL\2021\Formation\Rezeaux\TC\_light\noeuds.gpkg

☐ Selected features only

node\_id

abc num

j-text

m

j-mode

m

Radius(m)

1000.000000

Speed

4.000000

Maximum number

1

ID field size

40

Connectors file

G:\poubelle\connecteurs\_zones.gpkg

☒ Open output file after running algorithm

Build connectors

Generate a layer of linear objects corresponding to connections between two points layers (e.g. PT stops and roads nodes)

Parameters:

stops : stops layer

stop\_id : stop id

nodes : nodes layer

node\_id : node id

radius : search radius(m)

i-text : text at node i (start node)

i-mode : mode at node i (start node)

j-text : text at node j (end node)

j-mode : mode at node j (end node)

speed : travel speed on connectors (0= infinite speed)

connectors file : name of the resulting connectors layer

NB: the tool generates in the same time a txt file in Muxiv format (same name as connectors file with txt extension)

0%

Run as Batch Process...

Run

Close

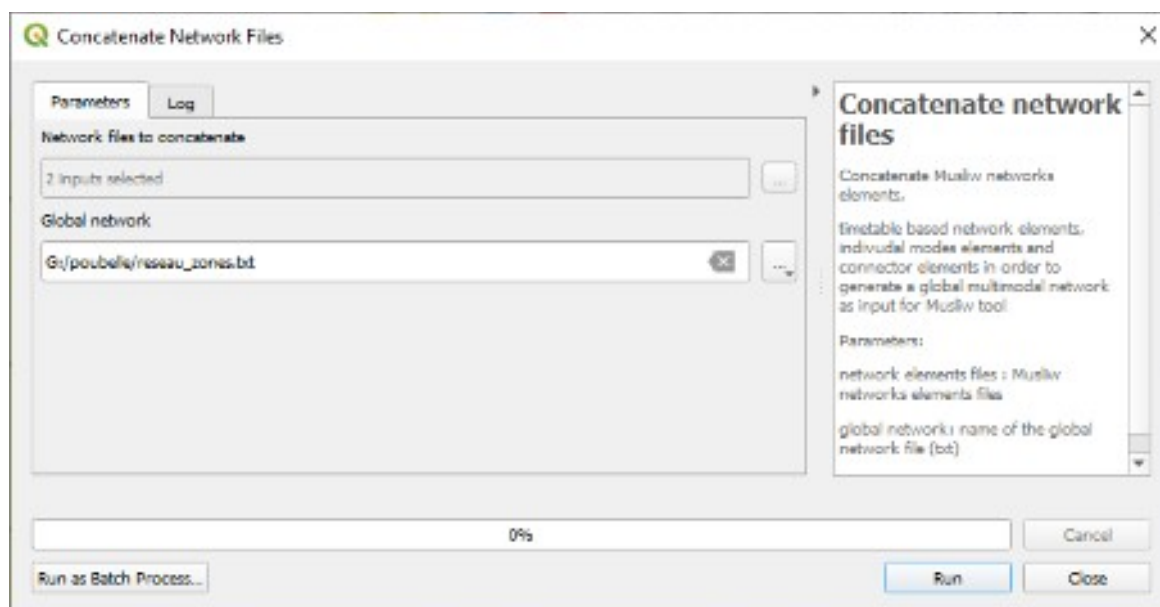
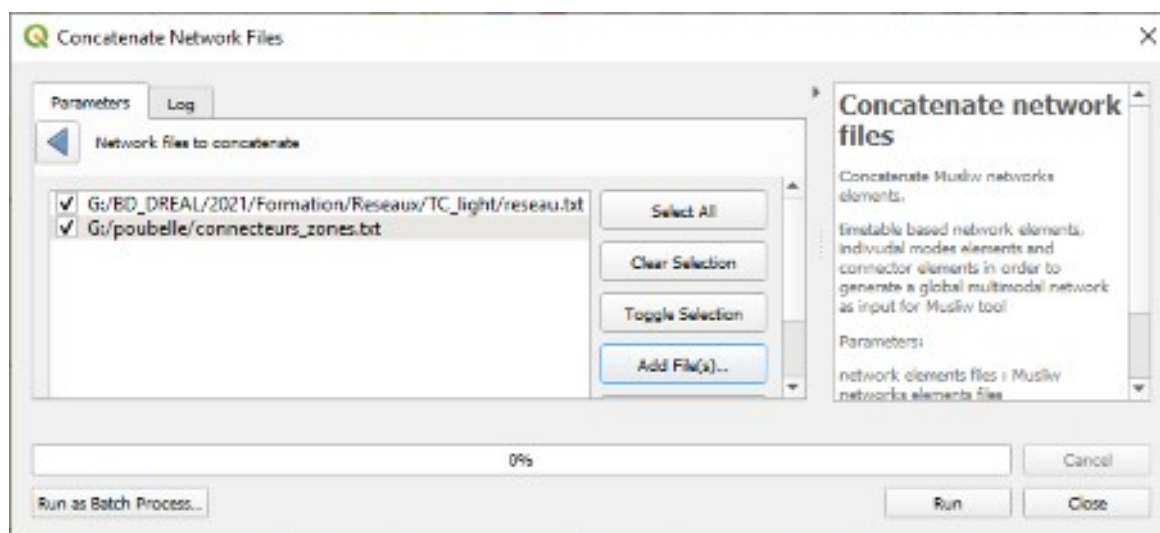
Cancel

72

## 7. Insert connectors into the multi-modal network

The "Build connectors" algorithm generates a table of linear objects corresponding to the connectors (here connectors\_zones.gpkg) and the "txt" file of the connectors in Musliw format ready to be integrated into the multimodal network.

To do this, the file connectors\_zones.txt (connectors in Musliw format) must be concatenated with the file "reseau.txt" of the multimodal network in Musliw format.



## 8. Create an adapted parameter set and launch the calculation

To be able to visualize the results, it is necessary to generate a specific parameter set a little bit different from the usual parameter set for the realization of classical isochrones maps.



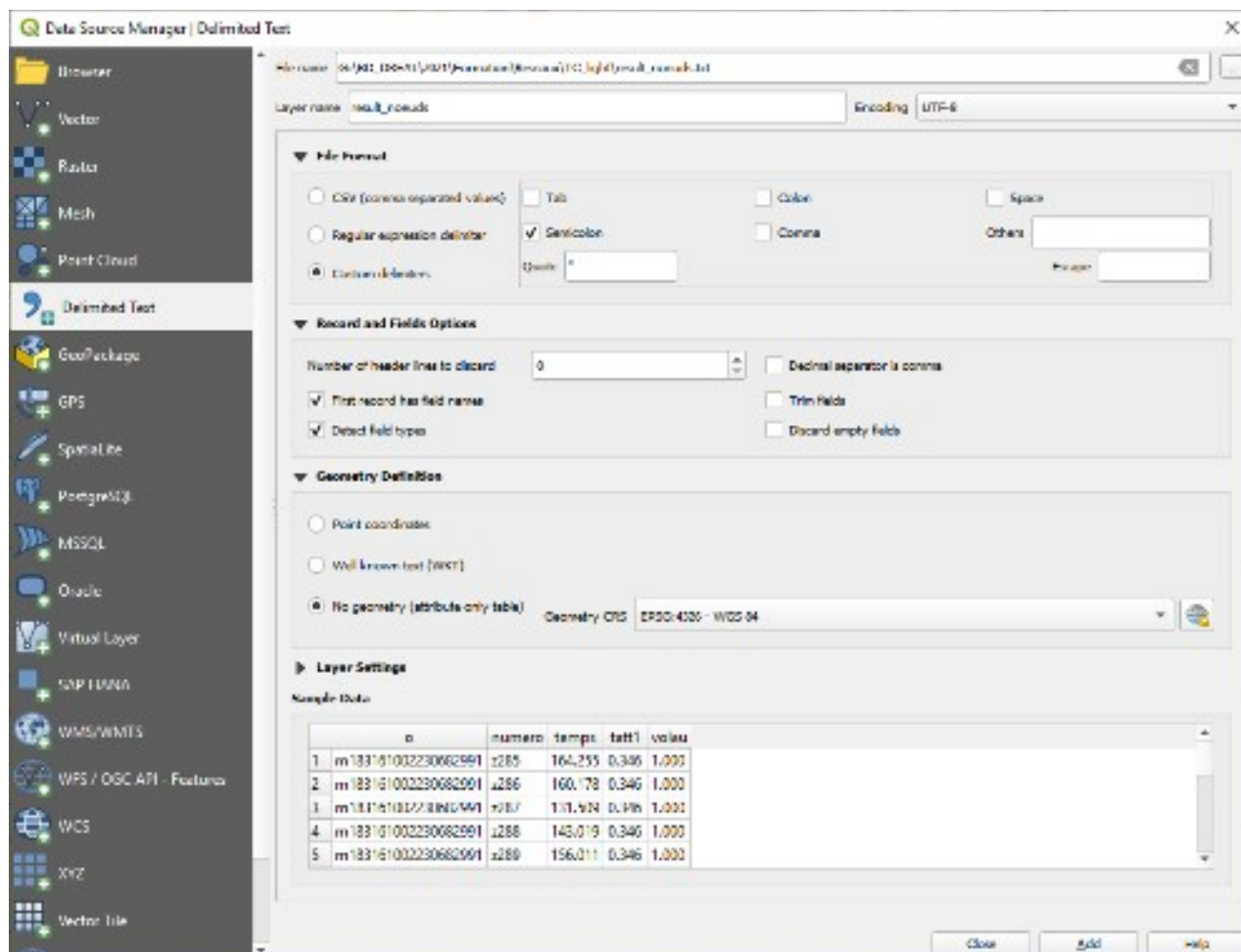
In particular, you should :

- Enable "output nodes?"
- Enable "detailed travel times" "compact mode" if you want to reduce the size of the node file
- Set up the "output filter".
  - zm (if the mode identifier is z for zones, and m for the network walk) with an "endpoint" calculation)
  - mz (if the mode identifier is z for the zones, and m for the network on) with a "starting point" calculation)

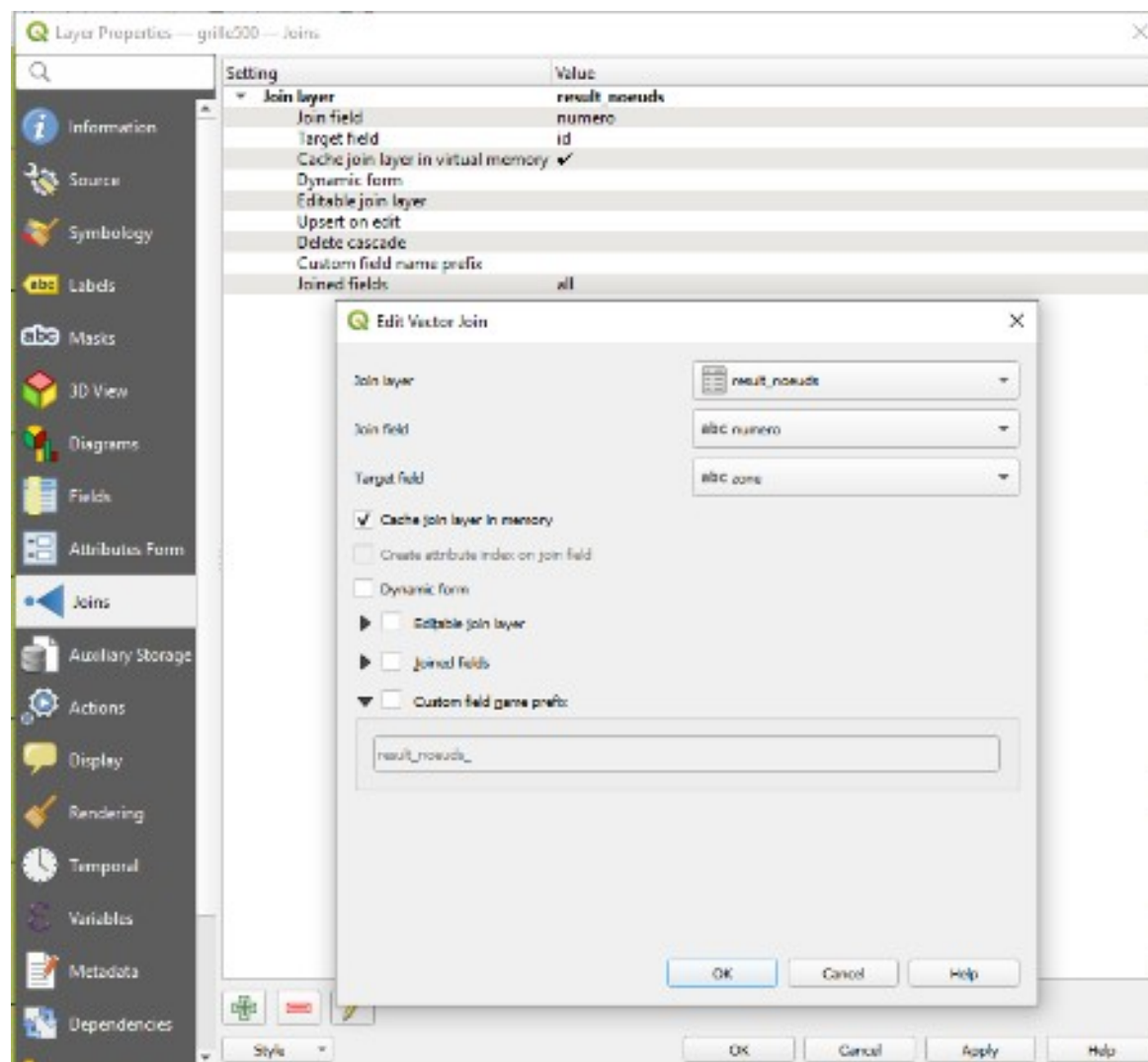
Then, it is necessary to launch the Musliw calculation

## 9. Visualize the results

Import the Delimited Text file ...\_noeuds.txt

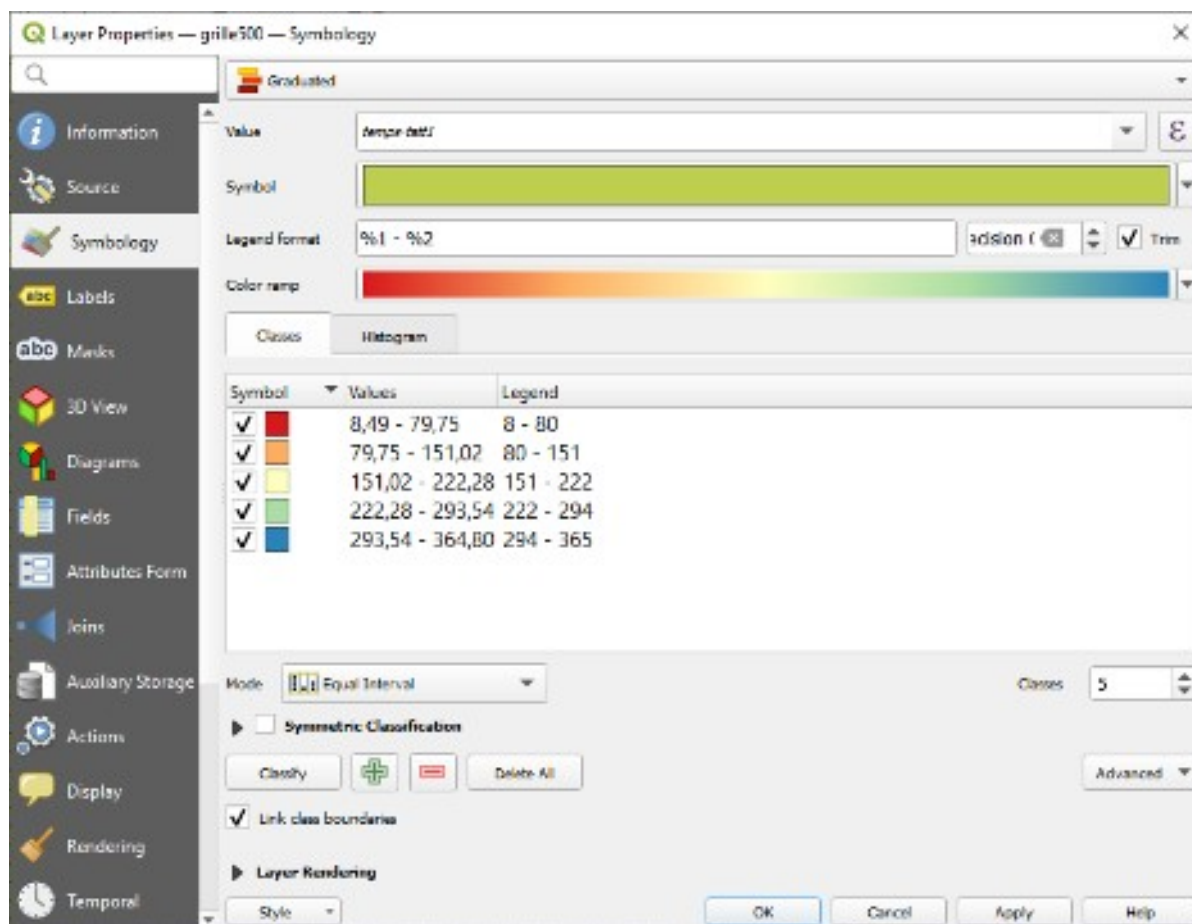


Perform the join with the grid layer

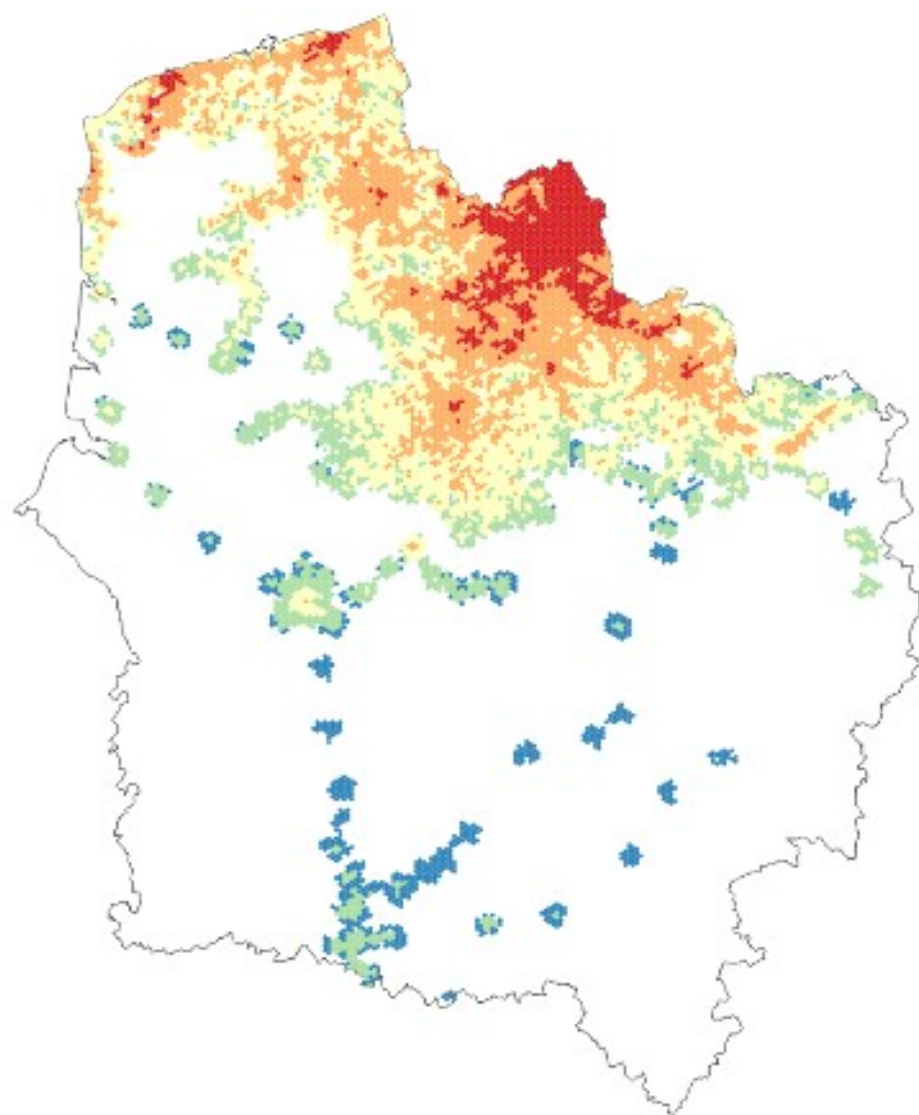


Make a graduated representation on " temps " or " temps -tatt1 ".





We obtain the following map



# **G- Creation of gravity accessibility maps with or without spatial and/or modal competition**

## **1. Presentation**

### **1.1. Introduction**

Unlike accessibility to the nearest service, which provides information on the fastest travel time to access the nearest facility, gravity accessibility makes it possible to construct indicators that take into account the accessibility not of the nearest facility but of all facilities.

Indeed, it is not the same thing to be located 10 minutes from the nearest general practitioner who happens to be the only one within an hour, for example, as it is to be located 10 minutes from the nearest one, but with the presence of one or more others within 15 minutes. The gravity method also makes it possible to determine whether it is more interesting to be located 10 minutes from a single doctor, or 15 minutes from two doctors

### **1.2. Principle**

The principle of gravity accessibility is to give a value or a weight to each equipment (e.g. to each general practitioner) which depends on its accessibility. Gravity accessibility then consists in summing up the weights of each equipment. The value of the weight is defined by a decreasing function of the distance called the conductance or resistance function (distance, access time, generalized access time, cost, etc.).

### **1.3. Function used for accessibility by gravity**

For accessibility to services, the principle chosen is as follows:

- We want a service on the spot ( $t=0$ ) to count as 1
- A service not accessible or at an infinite time counts as 0
- A service located at  $t=t_0$  counts as 0.5

The choice of  $t_0$  is a parameter defined by the user. In the example we have taken it equal to the average time observed in the Household Travel Surveys according to the mode of transport:

- Car:  $t_0=15$  minutes
- Public transport:  $t_0=30$  minutes

Ideally, these parameters should be calibrated on the basis of survey results that make it possible to assess the attractiveness of one facility in relation to another, from the standpoint of accessibility alone.

In this study, the resistance function used is of the form  $2^{-\left(\frac{t}{t_0}\right)^n}$

## 1.4. Required data

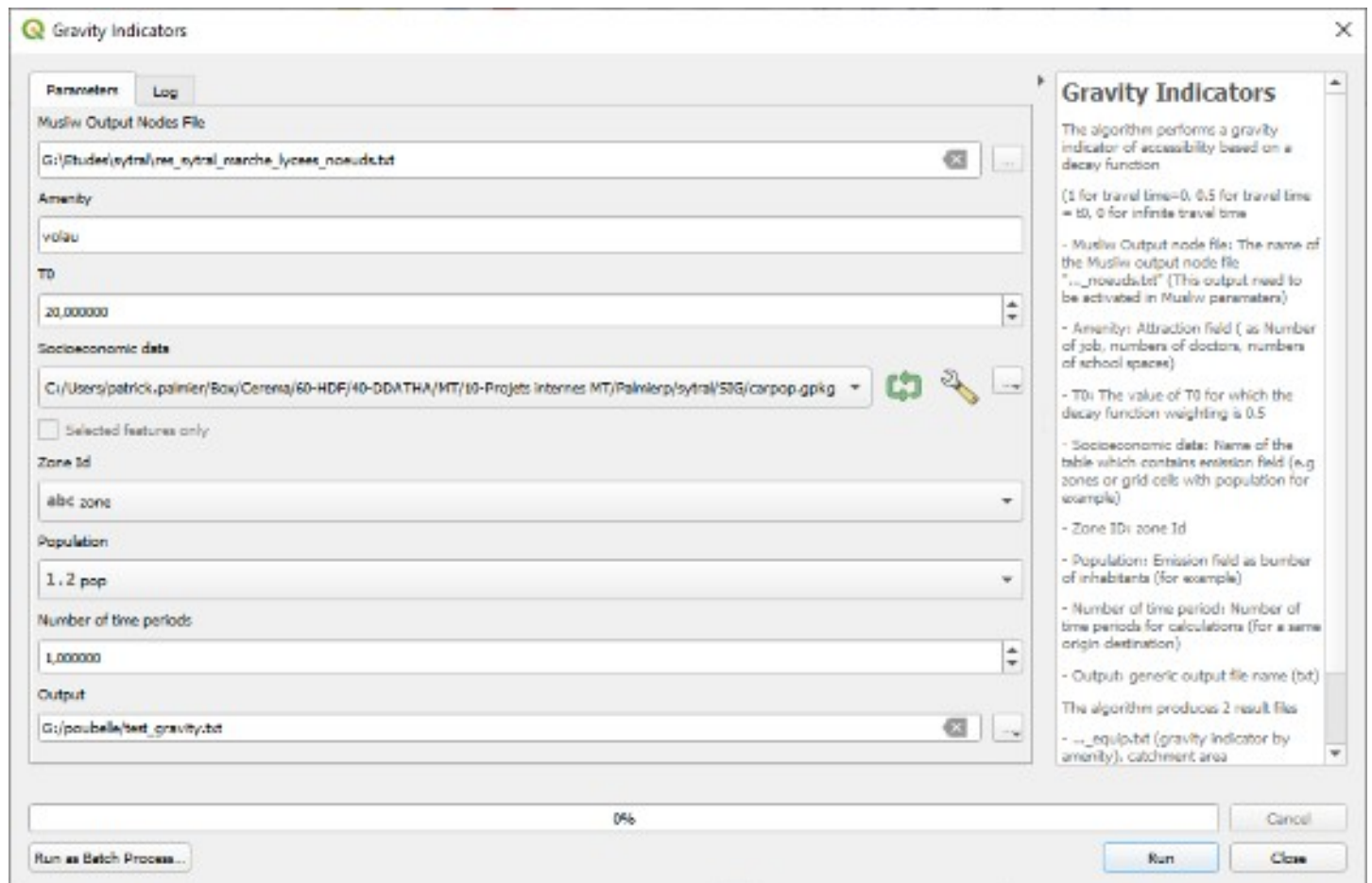
To calculate the gravity accessibility for a service in a territory, it requires :

- A table of areas of the territory (polygons or grid) that contains data on the population(s) for which we are looking for territorial accessibility to the service
- A table of facilities (amenities representing the attractiveness of the service, e.g. table of doctors, table of schools, jobs). This table contains a column characterizing the attractiveness of the service (e.g.: number of pupils for schools, number of jobs, '1' for doctors if we do not know their average number of consultations)
- An accessibility calculation based on a matrix between each generator and each population zone.

## 2. Gravity indicators with or without spatial competition

For the gravity calculation, it is necessary to check that in the parameterization file, the output 'Nodes file' is activated and that in the 'output filter' only the sections connecting the zones to the road network (outbound or inbound) are activated, depending on whether the calculation is done with a starting point or an end point.

Depending on the size of the matrix (at least nb zones\*nb equipments \* nb hours per od ) the script will, based on the result node file, calculate gravity indicators taking into account the socio-economic data of the concerned population.



The script produces two result files of delimited text type with ";" as separator.

- A file ...\_zones.txt: This file contains the gravity accessibility indicators related to each zone with the following fields:
  - zone: zone identifier
  - nb: number of DOs that contributed to the calculation of the indicator
  - n\_tot: number of accessible amenities without taking into account spatial competition
  - w\_n: gravity accessibility indicator with spatial competition without taking population into account
  - w\_pop: gravitational accessibility indicator with spatial competition with population taken into account (nb of amenities per capita). In the example below, this is the number of college seats per capita.

zone	nb	w_n	n_tot	w_pop
Z2	117	0.444	1'909	0.00810
Z3	117	0.303	1'263	0.00562
Z4	117	0.334	1'405	0.00618
Z5	117	0.295	1'228	0.00548
Z6	117	0.318	1'322	0.00594
Z7	117	0.370	1'563	0.00690
Z8	117	0.400	1'701	0.00743
Z9	104	0.035	126	0.00086
Z10	107	0.038	137	0.00092
Z11	117	0.165	677	0.00337
Z12	117	0.183	755	0.00372
Z13	117	0.160	653	0.00326
Z14	117	0.332	1'421	0.00654
Z15	117	0.331	1'417	0.00652
Z16	117	0.323	1'339	0.00590
Z17	117	0.332	1'389	0.00610
Z18	117	0.396	1'688	0.00727
Z19	117	0.342	1'438	0.00631
Z20	117	0.386	1'641	0.00709
Z21	117	0.376	1'595	0.00691
Z22	117	0.392	1'671	0.00718
Z23	117	0.280	1'153	0.00526
Z24	117	0.435	1'866	0.00806

- A file ...\_equip.txt : This file contains the gravity accessibility indicators relative to each of the equipments
  - equip: equipment identifier
  - nb: number of DOs that contributed to the calculation of the indicator
  - pop: total population of areas that contributed to the calculation of the indicator
  - w\_n: gravity accessibility indicator with spatial competition without taking population into account.
  - w\_pop: gravity accessibility indicator with spatial competition with population taken into account (catchment area)
  - w\_pop2 : population by amenity (by doctor, by job, by college place, by equipment, ...). . In the example below, this is the number of inhabitants per college place



- vol : volume of amenities of the equipment (here number of college places)

equip	nb	pop	w_n	w_pop	w_pop2	vol
0690636S	45942	1'351'589	1'072	39'900	4.0	81
0690514J	46271	1'353'322	6'753	484'244	25.3	646
0690653K	46839	1'355'298	7'334	487'667	28.0	245
0690032K	46680	1'354'660	8'195	573'483	29.4	858
0691626T	46179	1'352'939	3'072	215'994	12.2	294
0690281F	45780	1'352'187	4'550	300'998	17.7	405
0690698J	46654	1'354'617	6'876	486'064	25.0	562
0690554C	46764	1'354'966	5'695	467'076	23.6	530
0011276G	1673	96'284	114	387	0.8	1261
0690125L	46866	1'355'225	5'137	412'216	19.5	351
0691724Z	46836	1'355'284	7'502	551'326	27.6	166
0690651H	46619	1'354'702	6'728	508'744	25.0	492
0693330V	43923	1'345'727	985	18'615	14.2	1141
0690035N	45895	1'352'216	5'598	419'154	21.4	943
0690563M	46391	1'353'739	4'370	346'552	16.3	608
0694197M	45690	1'350'843	2'411	124'537	9.4	137
0693619J	46733	1'354'408	3'281	255'085	13.6	816
0692968B	46722	1'354'785	4'827	303'898	19.6	303
0690027E	46622	1'354'728	7'496	564'146	27.9	1083
0693045K	46441	1'353'837	3'192	244'540	13.2	361
0693368L	46802	1'354'954	7'971	556'225	29.6	34
0693373S	45846	1'351'953	5'870	490'006	23.1	375
0690712Z	46391	1'353'739	4'370	346'552	16.3	156
0690642Y	46339	1'353'635	5'751	439'282	21.9	118
0693478F	46621	1'354'391	4'413	170'467	15.3	1254
0693200D	43699	1'344'959	927	17'758	13.4	270
0690109U	46379	1'353'997	4'859	418'424	20.6	284
0690103M	46252	1'353'124	5'500	430'451	22.3	1043
0690522T	46165	1'353'006	6'948	491'457	24.6	1405
0690074F	45591	1'351'483	4'387	297'859	16.8	925
0693974V	46572	1'354'004	6'081	462'877	24.2	94
0693446W	45651	1'351'909	4'215	318'111	16.2	835
0691680B	45690	1'350'843	2'411	124'537	9.4	380

### 3. Gravity indicators with spatial and modal competition

The gravity indicators with spatial and modal competition are a generalization of the above indicators. In addition to spatial competition for access to facilities, the calculation also incorporates modal competition between several subpopulations that differ in their use of transport modes.

In order to calculate this indicator, the area layer must contain columns corresponding to the subpopulations to be studied.

Here in the example, the population is divided into 3 subpopulations in the following columns

pvp0: the population of households without cars

pvp1 : the population of households with one car

pvp2: the population of households with 2 or more cars

For each of the sub-populations, we need to know the modal split of the trips (here for the home-college pattern) if we are interested in the accessibility to the colleges.

	Vélo	TC+marche	Voiture
pvp0	0%	100%	0%
pvp1	1%	88%	11%
pvp2	2%	74%	24%

**Multimodal Gravity Indicators**

The algorithm performs a gravity indicator of accessibility with spatial and modal competition based on a decay function

(1 for travel time=0, 0.5 for travel time = t0, 0 for infinite travel time)

- Modes: a table with one row per mode with:
  - name: name of the mode (e.g car)
  - t0: The value of t0 for which the decay function weighting is 0.5 for this specific mode
  - modal share: a python dict where keys must correspond to socioeconomic data columns, and value to the modal share of this mode for this population (e.g. {'pvp0': 1.0, 'pvp1': 0.88, 'pvp2': 0.74})
  - musliw nodes file: The name of the Musliw output mode file "...\_nodes.txt" (This output need to be activated in Musliw parameter)
  - Amenity: Attraction field (e.g. Number of job, numbers of doctors, numbers of school spaces)

The mode table must be filled in as follows:

- mode : The mode label
- t0 : The value of the time in minutes for which the resistance function is 0.5
- modal share: The modal share of the mode for each of the subpopulations (in the form of a Python dictionary where the keys are the column names of the subpopulations)
- Musliw node file: Musliw node file of the gravity indicator of the mode in question (see G 2)

It is therefore necessary to have previously calculated the gravity indicators for each of the modes (see G 2)

**Multimodal Gravity Indicators**

The algorithm performs a gravity indicator of accessibility with spatial and modal competition based on a decay function

(1 for travel time=0, 0.5 for travel time = t0, 0 for infinite travel time)

- Modes: a table with one row per mode with:
  - name: name of the mode (e.g car)
  - t0: The value of t0 for which the decay function weighting is 0.5 for this specific mode

mode	t0	modal share	Musliw nodes file
1 marche	22	{ 'pvp0':1.0, 'pvp1':0.88, 'pvp2':0.74 }	G:/Etudes/sytral/res_sytral_marche_lycees_no...
2 Velo	9	{ 'pvp0':0, 'pvp1':0.01, 'pvp2':0.02 }	G:/Etudes/sytral/res_sytral_velo_lycees_noeuds...
3 voiture	14	{ 'pvp0':0.0, 'pvp1':0.11, 'pvp2':0.24 }	G:/Etudes/sytral/res_sytral_vp_lycees_noeuds.txt

The script produces two result files of delimited text type with ";" as separator.

- A file ...multi\_zones.txt: This file contains the gravity accessibility indicators related to each zone with the following fields:
  - zone: zone identifier
  - nb: number of DOs that contributed to the calculation of the indicator
  - pop: total population of the area
  - n\_tot: gravity accessibility indicator with spatial competition without taking population into account
  - w\_pop: gravity accessibility indicator with spatial and modal competition with population taken into account (nb of amenities per capita). In the example below, this is the number of college seats per capita.
  - pvp0,pvp1,pvp2: This is the w\_pop indicator but relative to each subpopulation. The name is that of the corresponding column name in the area table.

zone	nb	n_tot	w_pop	pvp0	pvp1	pvp2
Z2	1072	233	0.0178	0.0281	0.0123	0.0196
Z3	1072	143	0.0105	0.0150	0.0072	0.0125
Z4	1072	158	0.0119	0.0177	0.0082	0.0137
Z5	1072	139	0.0101	0.0144	0.0070	0.0122
Z6	1072	151	0.0114	0.0169	0.0079	0.0133
Z7	1072	272	0.0186	0.0224	0.0127	0.0244
Z8	1069	290	0.0201	0.0255	0.0138	0.0258
Z9	1018	51	0.0024	0.0008	0.0016	0.0043
Z10	1027	51	0.0024	0.0008	0.0016	0.0043
Z11	1057	71	0.0045	0.0049	0.0031	0.0061
Z12	1057	76	0.0050	0.0059	0.0034	0.0066
Z13	1057	69	0.0044	0.0046	0.0030	0.0060
Z14	1057	157	0.0114	0.0164	0.0079	0.0133
Z15	1057	156	0.0114	0.0163	0.0079	0.0132
Z16	1072	148	0.0102	0.0149	0.0070	0.0120
Z17	1072	155	0.0111	0.0161	0.0077	0.0132
Z18	1072	205	0.0154	0.0234	0.0106	0.0175
Z19	1072	175	0.0126	0.0181	0.0087	0.0151
Z20	1072	179	0.0139	0.0221	0.0096	0.0152
Z21	1072	174	0.0134	0.0212	0.0093	0.0148
Z22	1072	192	0.0146	0.0227	0.0101	0.0163
Z23	1072	137	0.0099	0.0137	0.0068	0.0122
Z24	1069	313	0.0223	0.0296	0.0153	0.0277
Z25	1072	325	0.0233	0.0316	0.0160	0.0286
Z26	1057	24	0.0023	0.0031	0.0016	0.0027
Z27	1057	61	0.0037	0.0033	0.0025	0.0054
Z28	997	50	0.0023	0.0006	0.0015	0.0042
Z29	997	50	0.0023	0.0006	0.0015	0.0042
Z30	1057	69	0.0043	0.0046	0.0030	0.0060

- A file ...\_equip.txt : This file contains the gravity accessibility indicators relative to each of the equipments
  - equip: equipment identifier
  - nb: number of DOs that contributed to the calculation of the indicator
  - pop: total population of areas that contributed to the calculation of the indicator
  - nb: number of DOs that contributed to the calculation of the indicator
  - w\_pop: gravity accessibility indicator with spatial competition with population taken into account (catchment area)
  - pvp0,pvp1,pvp2: gravity accessibility indicator with spatial competition with population subpopulations taken into account (catchment area by subpopulation)(column names identical to those of the zones table)
  - w\_pop2 : population by amenity (by doctor, by job, by college place, by equipment, ...) . In the example below, this is the number of inhabitants per college place



- w\_pop\_pvp0, w\_pop\_pvp1, w\_pop\_pvp2: sub-population by amenity (per doctor, per job, per college place, per facility, ...). . In the example below, this is the number of inhabitants per college place for each population category
- vol : volume of amenities of the equipment (here number of college places)
- 

equip	pvp0	pop pvp0	w pop pvp0	pvp1	pop pvp1	w pop pvp1	pvp2	pop pvp2	w pop pvp2	nb	pop	w pop	w pop2	vol
0881614E	41588	257118	0.29	15276	625588	2.80	14934	422024	5.40	408125	1304731	34799	8.48	
0890546J	8721	257009	0.34	18828	625541	2.33	11614	422064	5.23	418554	1304704	37163	7.90	
0891675W	49555	257114	3.03	92009	625671	7.56	32019	422310	3.57	422928	1305095	173583	14.16	
0882337R	28179	257109	1.77	62013	625678	6.33	30949	422281	5.63	417258	1305070	122141	13.73	
0892090A	5411	257117	0.32	15138	625582	2.68	12901	421911	4.61	401713	1304591	33530	7.59	
0011338Z	0	7	0.00	408	73290	0.52	1252	106406	1.61	220266	179702	1659	2.32	
0891484N	81574	257038	0.62	21799	625414	3.58	14541	422222	4.59	423201	1304674	45015	8.79	
0891497C	8391	257112	1.12	23407	625590	8.00	20623	422041	8.09	410004	1304742	50500	14.09	
0892159X	9800	257104	1.80	24766	625582	6.31	16182	422104	6.18	418653	1304770	50749	14.29	
0891690U	1863	257028	0.07	7331	625068	1.56	8140	421349	3.90	386444	1303436	17724	5.53	
0890621A	0	251153	0.00	102	601500	0.25	495	393411	0.02	345693	1248152	605	1.07	
0891479H	33639	257110	2.12	71862	625689	7.79	36502	422303	7.07	417654	1305102	142003	16.98	
0893306H	3492	257030	0.18	9019	625356	1.29	6766	422085	2.43	421092	1304471	18776	3.90	
0011151W	15	189537	0.00	1141	452584	0.71	1064	294905	1.57	323019	907096	3020	2.28	
0891489C	73206	257120	5.32	126643	625708	11.48	46766	422378	6.96	425793	1305206	246615	23.76	
0891481K	9744	256900	0.60	21005	625023	3.01	14286	421182	4.79	418889	1303562	45005	8.40	
0893890D	2259	257109	0.21	7221	625642	1.57	8029	422169	2.50	412605	1304920	15509	4.38	
0890002C	0	3	0.00	512	71738	0.61	1534	105251	2.20	208235	176991	2047	2.80	
0891864J	48302	257118	3.05	92120	625700	8.56	38308	422340	4.82	422841	1305156	175759	16.46	
0890624D	112	258532	0.00	2637	622772	1.46	5429	418455	3.79	380890	1297759	8739	5.28	
0892842Y	21920	257037	1.22	42817	625435	4.71	20785	422278	4.52	423858	1304750	85523	10.45	
0891863H	95294	257119	7.61	165245	625712	14.90	57114	422380	8.47	423735	1305191	317623	28.98	
0891869P	35214	257023	2.16	68568	625392	7.40	32150	422108	6.72	419313	1304601	135900	16.35	
0891852Z	23980	257041	1.73	57228	625294	8.56	36329	421949	10.74	412701	1304284	117537	21.03	
0893286X	7	237542	0.00	401	584247	0.75	1481	382384	3.11	318308	1164772	1885	3.88	
0894317T	656	258600	0.11	3339	624740	0.98	3229	420911	1.67	405398	1302532	7423	2.98	
0892583H	1262	256990	0.65	4136	622266	2.48	3660	416526	3.09	337638	1295782	8058	6.21	
0892390Y	64406	257117	4.46	118605	625680	10.77	42470	422318	5.17	421842	1305114	223574	19.80	
0893479G	34473	257103	1.89	59220	625643	4.99	22760	422276	3.71	422121	1305023	116472	10.58	
0893895T	38395	257113	2.13	72799	625657	7.26	34918	422292	7.12	424773	1305062	146112	16.51	
0892520P	5208	258978	0.28	11805	624972	1.53	7743	421443	2.77	407508	1303393	24758	4.08	
0890599B	74658	257111	5.18	130499	625654	10.72	44108	422278	4.64	421827	1305044	249465	20.54	
0892343X	17632	257108	2.18	35872	625638	5.95	17622	422223	4.75	417636	1304968	71126	12.88	
0892340U	41461	257113	2.64	78008	625689	7.04	29301	422298	4.87	421575	1305078	148182	14.55	
0890551Z	39562	257116	2.90	72098	625705	7.50	28355	422350	4.78	424515	1305171	140035	15.18	
0892703N	65439	257108	4.59	114618	625665	10.54	47080	422294	7.49	422685	1305057	227138	22.62	
0893267Y	4260	257115	0.46	18067	625638	4.08	18042	422148	8.63	414093	1304999	38389	11.17	
0892419E	1743	258967	0.85	6674	624207	3.85	7546	418534	5.39	388418	1300708	15964	10.08	
0894312M	3701	257118	0.17	10888	625540	2.33	10197	421785	4.54	388108	1304443	24788	7.04	
0892335N	2750	258905	0.32	12613	625141	3.40	13024	421608	8.21	408519	1303821	29187	9.93	
0892620Z	87991	257120	6.92	152592	625726	14.14	58983	422391	8.25	425958	1305238	299566	29.31	

# H- Using processing models

## 1. Introduction

The advantage of developing the "networks" extension in the form of an algorithm provider is the ability to create complex processing models by combining different algorithms and also to carry out batch processing. To make things easier for users, processing models have been developed and made available to perform the most frequent complex tasks.

These models are as follows (sorry in french):

- Isochrone: Drawing a monosite isochrone
- Isochrone multiples: Drawing of a multi-site isochrone
- Réseau individuel choix référence comple: Drawing of catchment areas
- Individual network full reference choice: Generation of a road network from OSM over the study area (exhaustive network)
  - Réseau individuel choix référence light: Generation of a road network from OSM on study area (simplified network)
  - Réseau individuel fenêtre complet: Generation of a road network from OSM on the study area defined by the visible canvas (exhaustive network)
  - Réseau individuel fenêtre light: Generation of a road network from OSM on the study area defined by the visible canvas (simplified network).
- Réseau Transport en commun: Generate a public transport network from GTFS data and connectors to the road network.

## 2. Installation

- Go to Preferences/User profile/Open active profile folder to open the corresponding folder
- Then go to python/plugins/networks and copy the .model3 files into the processing/models directory from the active profile folder.
- Deactivate and reactivate the processing extension so that models are taken into account



### 3. Running the model



Go to the edit box in the Templates/Cerema menu and double-click on the template you want to run.

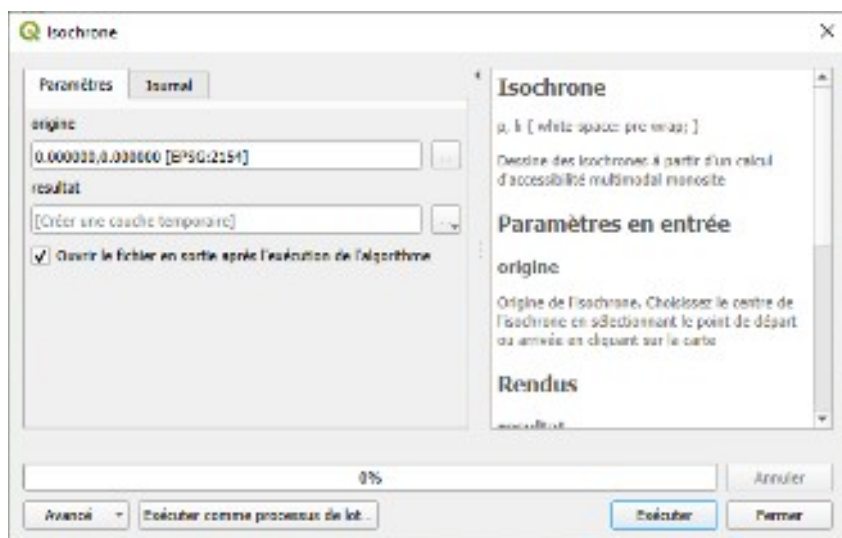
Enter the required fields and run the procedure.

Only the essential parameters are requested.

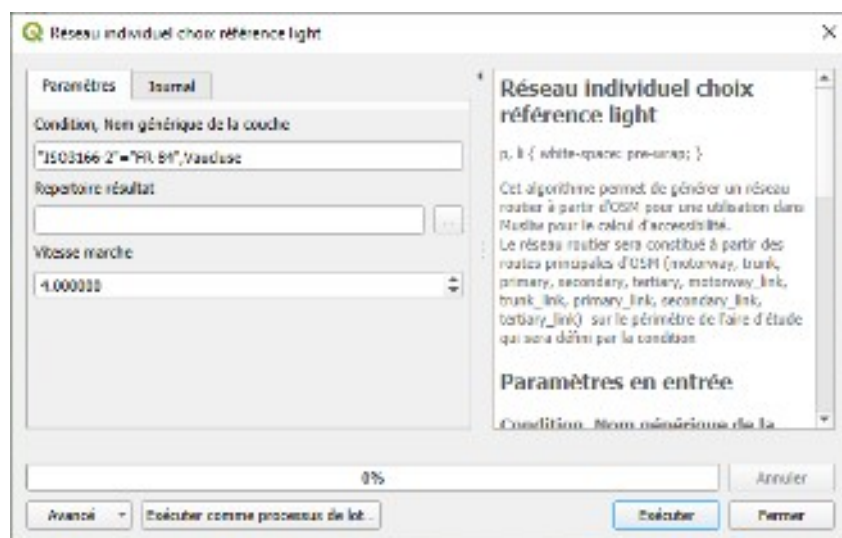
However, you can customize the template by editing it. This allows you to change the value of the various algorithm parameters.

It is also possible to add entries to the dialog box to ask the user to specify one or more other parameters.

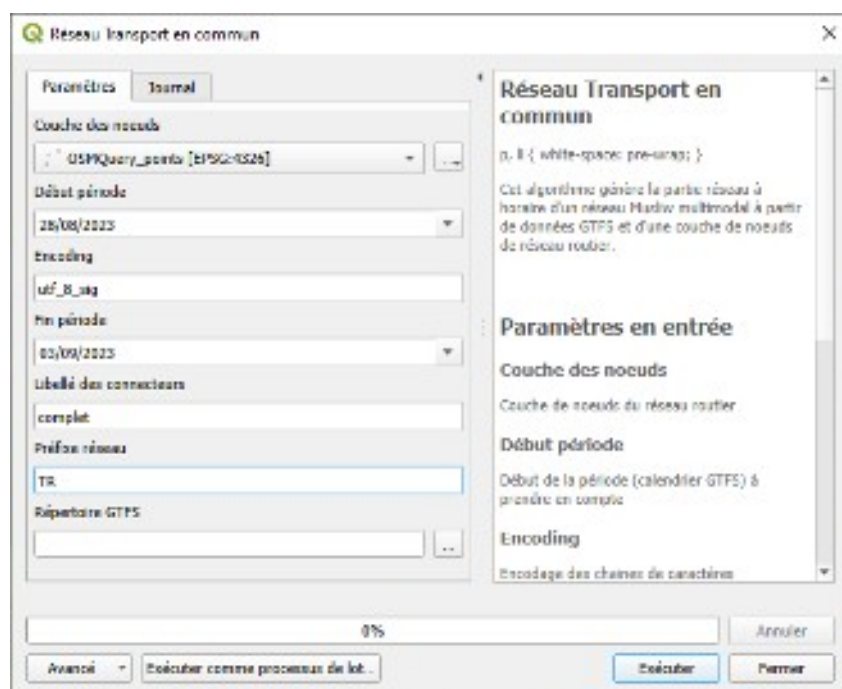
Ex: Isochrone dialog box



Ex: individual network by reference light dialog box



Ex: Public transport network dialog box



## 4. Description of model outputs

- Isochrons and multiple isochrons:  
The model produces a layer of iso-value polygons representing the isochrons by time step (default 0 to 120 minutes in 10-minute steps). These values can be customized by modifying the parameters of the "iso-value contours" algorithm.
- Catchment area :  
The model produces a layer of polygons representing the catchment area of the nearest point in terms of multimodal accessibility (network indicated in the Musliw calculation) of each hub indicated in the matrix (departure or arrival point).
- Individual networks :  
These models generate the following files:

- A table of road arcs with the addition of the following columns (direction, diffusion, dead end, length, time\_m, i, j , ij)
- The corresponding road network file in Musliw format (.txt)
- Public transport network :
  - a GIS file of nodes: a layer of point objects made up of stop points with information on the name of the stop and the frequency of use
  - an GIS file of arcs: a linear object file representing a line linking two consecutive stops, with indications of the initial node number and final node number.
  - a GIS file of lines: a file of linear objects made up of lines linking two consecutive stops on the same public transport line, with the initial node number, the final node number and the line number indicated. The layer also contains frequency information.
  - a public transport network file in Musliw format (.txt)
  - a GIS file of connectors between public transport stops and road network nodes
  - a file of connectors in Musliw format (.txt)

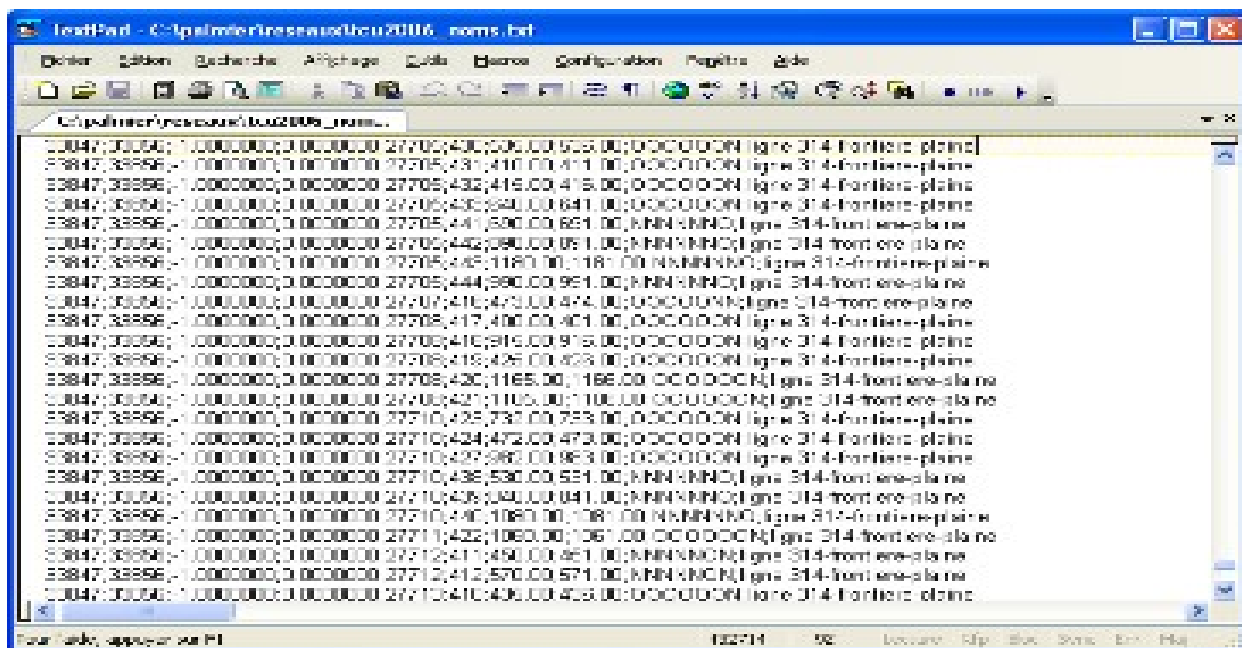
# I- Musliw documentation

For the computation of multimodal accessibility based on timetables and time on links, it is necessary to import a network and a demand matrix. It is also necessary to set up parameters for calculating the TC generalized time. The following paragraphs describe the process.

This calculation also works if there are no links based on timetables (ex : walking, cycling or car network only)

## 1. The « network » file

To define a time based network in MUSLIW, you need a file of the following type .



The network file is a "Delimited Text File" with ";" » As separator.

NB: MUSLIW automatically handles the problems of decimal separator '.' or ','. It converts them automatically according to the settings defined by your operating system.

- The different fields in order are:
  - origin node number of the link; maybe also an alphanumeric string;
  - destination node number of the link; maybe also an alphanumeric string;
  - travel time of the section:
    - number of minutes, if the mode of transport on the link is individual: car, walking, cycling;
    - -1 or , if the mode of transport on the section is a public transport with schedules.
  - length of the section :
    - length ;
    - «0» if length is unavailable.
- PT line number : (a number )
  - if positive (> 0) :

–line number (integer), for links of public transport with schedules ;

•si negative ( $< 0$ ) :

- ◆◦ period id during which the travel times will be identical The link is then considered to be of individual transport type variable in time. You need to describe the time periods during which time is applied. The identifier must be different for each period with a different travel time. An example of coding can be found below:
  - ◆ • -1 for the morning peak period;
  - ◆ • -2 for the evening peak period;
  - ◆ • -3 for the off-peak period;
  - ◆ • -4 for the night;
  - ◆ • -5 for Sunday and holidays.

•Service id :

- service id ( $> 0$ ) id (integer), for public transport sections at times;
- service id ( $> 0$ ) the range number (integer) for the individual transport links taking into account schedules and the calendar ;
- -1, for individual transport links (VP, walk, bike) without taking into account schedules or calendar. The section will be accessible 24/7, and will invariably present the same travel time.

•Departure time at start node :

- departure time, in minutes (past midnight), for public transport sections at times (eg 6h00 =  $6 * 60 = 360$ minutes);
- start time of the usable period, in minutes, for individual transport sections taking into account timetables and schedules;
- -1, if the mode of transport is individual without taking into account schedules and calendar: VP, walking, cycling

•arrival time at end node:

- arrival time, in minutes (past midnight), for public transport sections at times (eg 6:00 =  $6 * 60 = 360$ minutes);
- end time of the usable period, in minutes, for the individual transport sections taking into account schedules and calendar;
- -1, if the mode of transport is individual: car, walking, cycling.

•service circulation schedule :

- string of "n" characters: n = length in days of the period. The circulation of the service for each day of the period is determined by "O" for "circulates" or "N" for "does not circulate". For example, for a service that only runs on the 10th day of the period, the corresponding string of characters will consist of 9 "N" and then an "O" in 10th position. The chain of a service that runs every day will be composed of n "O". This chain determines in the same way, the days of circulation the links of the type individual transport and those defined with schedules and calendar;
  - -1, if the mode of transport is individual: car, walking, cycling.
- -1, if the mode of transport is individual: car, walking, cycling.

•label :

- the label is useful in the result files for identifying sections and routes by name rather than just node, line and service numbers. The part of the label before the first vertical separator «|» Generally determines the line identifier. The user is not limited in number of characters.

•Link type :

- the type of link makes it possible to define time and cost weighting parameters differently according to its considered type. Thus, a default connection time of 5 minutes for the network type 0 and 35 minutes for the Eurostar or aerial sections can be set to take account of the registration. Link type

could be a text string (ex : Tram)

- Toll:
  - The value to enter is the monetary cost on the section. Musliw will then calculate the optimal path taking into account the toll that is introduced in the generalized time thanks to a new weighting parameter that can depend on the type of link. In output Musliw will also provide a "toll" element that indicates the cumulative toll on the origin-destination.

The limit of the number of links and services is not defined. It is the memory of the computer that will set the limit of the allowable size..

For 32-bit operating systems, the maximum size of an object is 2 GB. Thus, MUSLIW will not be able to handle larger networks, even if the RAM is higher (4 GB is the maximum manageable size).

Conversely, this limit is much higher for 64-bit systems such as Windows 7, Linux 64, etc.

The major advantage of having a network integrating the service circulation calendar is to be able to vary the periods of the accessibility study without having to recode the network accordingly (especially with regard to Saturdays, Sundays , holidays, etc.).

The example below shows the coding of node numbers by strings and the type of network coding (0 for all sections except "station; bruxelles\_E \*" which is type 1).

```
depart;gare;5;-1;-1;-1;-1;-1;-1;MARCHE;0
gare;bruxelles_tgv;-1;-1;1;1;480;530;0000000;TGV|LILLE-BRUX_TGV;0
gare;bruxelles_E*;-1;-1;2;2;500;532;0000000;E*|LILLE-BRUX_E*;1
bruxelles_tgv;bruxelles_centre;5;-1;-1;-1;-1;-1;-1;MARCHE;0
bruxelles_E*;bruxelles_centre;5;-1;-1;-1;-1;-1;-1;MARCHE;0
```

### **1.1.1 Possible use of the type of link**

- The type of link has two essential uses:
  - differentiate time and cost weighting parameters by type;
  - perform calculations of shorter paths on a part of the network;
  - to filter the individual sections of a particular type (must enter a corresponding negative "cmap" negative;)
  - To filter the sections of a particular type of time, you have to enter a negative "cveh".
  - allow statistical operations by type of network

### **1.1.2 Example of coding of individual LINKS taking into account time periods and calendar**

The following excerpt provides an example of a stretch definition whose travel time depends on the time period and calendar :

```
11;151;5.3;0;-1; 1;420;540;000000N
```



```

11;151;4.1;0;-3; 1;540;960;OOOOOON
11;151;4.1;0;-3; 2;1140;1200;OOOOOON
11;151;4.6;0;-2; 1;960;1140;OOOOOON
11;151;3.5;0;-4; 1;1200;1440;OOOOOON
11;151;3.5;0;-4; 2;0;420;OOOOOON
11;151;3.6;0;-5; 1;360;1140;NNNNNNO
11;151;3.1;0;-6; 2;0;360;NNNNNNO
11;151;3.1;0;-6; 3;1140;1440;NNNNNNO

```

The travel times are:

- 3.5 minutes at night (20h-24h, 0h-6h);
- 4.1 minutes a day (9am to 4pm, 7pm to 8pm);
- 4.6 minutes in the evening (16h-19h);
- 5.3 minutes in the morning (7am to 9am) from Monday to Saturday;
- 3.1 minutes Sunday at night (0h-6h, 20h-24h);
- 3.6 minutes on Sunday during the day (6h-20h).

**NB: Do not forget that if you enter the time periods and a calendar for individual sections and that there are periods of the day or days in the calendar that are not defined, the link will be inaccessible for these periods (which would occur in the case of a closed road or street). It is therefore important to cover the time and calendar spectrum in the description**

## 2. The penalties and transfers definition file

This file contains all the necessary information for the introduction of penalties and prohibitions of turning movements and transfers.

The penalties and transfers definition file is a "Delimited Text" file with ";" » as delimiter

```
35046;35482;3802;35047;302;1.5  
28325;28264;-1;28347;-1;-1  
28347;28325;-1;28442;-1;0.25
```

•The different fields are in order:

- nj: node number of the intersection or stop considered;
- ni: origin node number;
- line number of the incoming link (ni-> nj);
- nk: final node number;
- line number of the outgoing link (nj-> nk);
- penalty time:
- 0: no penalty;
- -1: movement or transfer prohibited;
- otherwise: value of the penalty time in minutes.

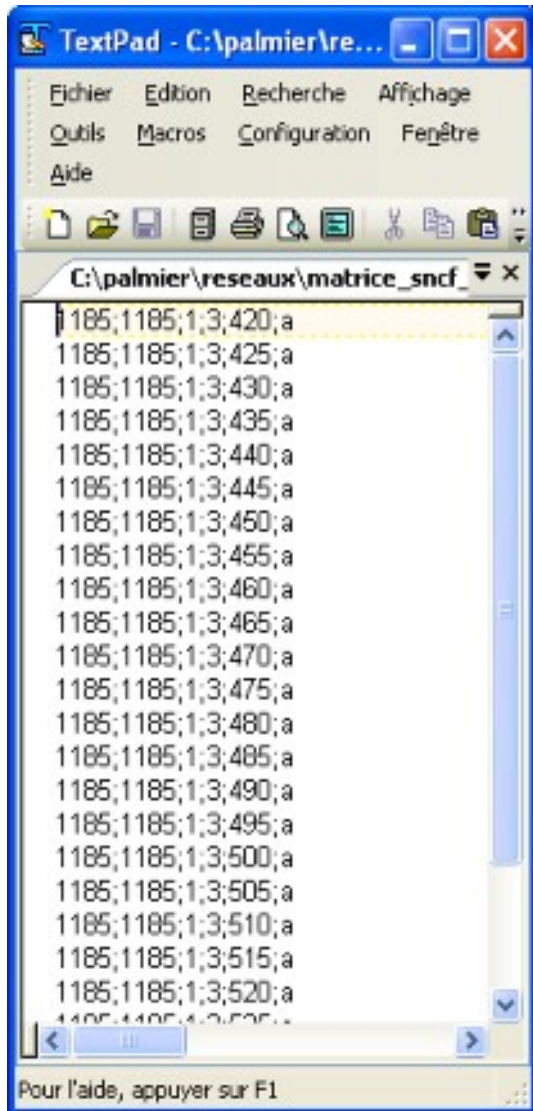
If the penalty time is relative to a transfer, ie one of the two line numbers is not "-1" the penalty time will be used as the transfer time instead of the default transfer time set in the parameters that will be used by default for all other network transfers

The fact of having distinct line numbers for different periods describing the travel time links sections of the individual transport type makes it possible to describe penalties for turning movements or for different transfers depending on the periods.

### 3. The «Matrix» file

#### 3.1.1 Standard specifications

The matrix file is a "Delimited Text" file with ";" » as separator.



The different fields in order are:

- origin node: number of the origin node: this can be any node of the network;
- destination node: destination node number: this can be any network node;
- demand: the volume of the demand that you want to assign between the origin node and the destination node;
- day: indicates the desired day of departure or arrival. This number refers to the period defined for each service in the "Network" file. The first day of the period is day 1. Thus, day 3 is the 3th day of the period. Be careful the number of the day must always be strictly smaller than the number of days of the period;
- time: indicates the desired departure or arrival time in minutes. The time in a day varies between 0 and 1439 minutes. The software tolerates negative hours or more than 1439. It will increment or decrement for the calculation of the number of days \* 1440 for the time to be between 0 and 1439;

- type of time desired:
  - "d" for departure: the calculation will be done from the origin node starting at the indicated day and time by iteratively searching for the different successors to the destination node;
  - "a" for arrival: the calculation will be done from the destination node at the indicated day and time by iteratively searching the different predecessors to the origin node.

There is no size limit for the number of rows in the matrix file.

When looking for the shortest path, a link is taken into account in the route if it allows to reach a successor section with a cost strictly lower than the path chosen as the shortest in the current iteration . Thus, if several itineraries are equivalent, the first found will be the one selected; the entire demand of the row of the corresponding matrix file will be assigned to it.

### 3.1.2 Advanced specifications

MUSLIW offers the possibility to detail for each line of the matrix file the parameters of the calculation (in blue) and even possibly the parameters of the algorithm and outputs (in green).

However, if the calculation parameters (blue part) can be indicated without being those of the algorithm and outputs (green part), the reciprocal is not true. In order to be able to indicate the algorithm and output parameters in batch mode, it is also necessary to indicate the calculation parameters.

As shown in the example below, the fineness of the setting can vary depending on the lines of the matrix file. On the other hand, to be active, each color block must be complete.

```
1546;1368;1;1;420;a;od1_420 ;1;1.5;3;5;1;2;120,0;30,2,120
1546;1368;1;1;435;a;od1_435 ; 1;1.5;3;5;1;2;120,0;30,2,120, true;0;0;50;10000;2
1546;1368;1;1;450;a;od1_450 ; 1;1.5;3;5;1;2;120,0;30,2,120, true;0;0;50;10000;2
1546;1368;1;1;465;a
1546;1368;1;1;480;a ;od1_480
1546;1368;1;1;495;a;od1_495;1;1.5;3;5;1;2;120,0;30,2,120, true;1;0;50;10000;2
1546;1368;1;1;510;a;od1_510 ;1;1.5;3;5;1;2;120,0;30,2,120
1546;1368;1;1;525;a;od1_525 ; 1;1.5;3;5;1;2;120,0;30,2,120, true;0;0;50;10000;2
1546;1368;1;1;540;a;od1_540 ; 1;1.5;3;5;1;2;120,0;30,2,120, true;0;0;50;10000;2
```

The field between the red and blue blocks is a text field to identify the row of the matrix. If this field is empty, Musliw will take by default the line number of the file.

The usefulness of this field lies especially in the case of important matrix files for which the calculation of one or more lines did not give the expected results. This identifier makes it possible to restart the calculation only on Ods you need to recalculate (by the constitution of a specific file matrix) and to easily replace the results of the first calculation by those of the new one.

Blue block in order :	Green block in order :
•In vehicle weight ;	•Detailed paths (true/false) ;
•Waiting weight;	•Detailed travel times(0 None, 1 without tum based links, 2 with time based links) ;
•Individual mode time weight ;	•Algorithm (0 GGA with buckets, 1 Dijkstra with

	buckets) ;
•Transfer weight ;	•Algorithm scale parameter ;
•Individual mode time scale;	•Max of buckets ;
•Minimum transfer time;	•Algorithm power parameter.
•Maximum transfer time ;	•Output filter
•Number of days.	
•Maximum individual mode cumulative time	
•Toll weight	
•Maximum generalized cumulative time	

Since the calculation weighting parameters can be differentiated according to the type of links, they can be entered separated by a "|". Ex 1,1,5; 3; 2; 1; 2 | 5 | 35; 0 specifies a different connection time per link type (2 minutes for type "0", 5 minutes for type "1", and 35 minutes for type "2")..



## 4. Computation time optimization

To minimize the calculation time, it is necessary to sort the matrix file according to the rules below:

For calculations from a starting point, the file must be sorted by:

- Origin;
- Day;
- Hour.

For calculations from an end point, the file must be sorted by:

- Destination ;
- Day;
- Hour.

Indeed, MUSLIW calculates the shortest paths of one point to all the other points, so if the origin, the day and the hour of departure or arrival wished are identical between two consecutive lines of the matrix file, it is not necessary to recalculate all the shortest paths, resulting in significant computing time savings for large networks and large matrix files.

## 5. Computation procedure

### 5.1. Basic case

The procedure is launched via the menu bar "Procedures> Time based assignment". You must then fill in the dialog box that appears, as shown below.

**Musliw Computation**

Parameters Log

Musliw network

Musliw matrix

Musliw parameters

Musliw penalties [optional]

☐ Accept to download Musliw binary from Github

Output

[Save to temporary file]

0%

Run as Batch Process...

Run Close

**Musliw computation**

Perform a calculation of multimodal accessibility and routing

Can download the Musliw.exe binary for multimodal routing and accessibility computation (if checked)

Produce output files (semi-column separated files (if selected in parameters):

- . Origin destination results file "\_od.txt"
- . Cumulative times on arcs "\_temps.txt" (usefull for isochron maps)
- . Detailed paths "\_chemins.txt"
- . Assignment results file on links "\_off.txt"
- . Log file "\_log.txt"
- . Parameters files "\_param.txt"
- . results on services "\_services.txt"

The user must enter :

- The network on which he wants to calculate the accessibility or the multimodal routes
- The matrix (origin-destination matrix, also defining the days and times of departure or arrival and the number of passengers)
- The penalties for turning movements or connections
- Calculation parameters and result data (from a parameter file)

The parameter file can be generated in two ways:

- From the "Musliw parameters" script
- By reusing an existing parameter file or one produced by a previous calculation.

Each Musliw calculation generates a "\_param.txt" file that corresponds to the parameter file used for the calculation. You have to fill the following settings

- In vehicle weight : weight for time based travel times (on board the vehicle);
- Waiting weight : weight for waiting times;
- Individual mode weight: weighting for individual transport mode travel times (Walking, MAP, Bike);
- Maximum generalized time : Maximum cumulative generalized cost
- Maximum individual mode time: Maximum cumulative individual time
- Boarding weight: weight for transfer times;
- Individual mode factor speed: uniform multiplier coefficient of all individual transport travel times.

Performed once before launching all calculations. In the case where the individual transport times have been defined from a constant speed, this coefficient makes it possible to test a different walking speed without

having to modify the travel times in the description of the network. For example, for a network whose walking times have been defined with a speed of 4km / h, if we want to perform calculations for people walking at 5km / h, we will take a TMAP coefficient equal to 1, 25 without having to change the travel times in the description of the network;

- Minimum transfer delay : minimum connection time. This is the minimum time possible between arrival at a stop and the possibility of getting into a vehicle. This time will be weighted by "COR weight" and taken into account in the generalized time;

- Maximum transfer delay : maximum match time. This is the maximum permissible time between arrival at a stop and the possibility of getting into the next vehicle. Beyond this value the correspondence is considered unattractive and will not be taken into account in the search for a shorter path;

- Extra day duration: determines the number of days during which the algorithm will search for the shortest paths. '0' indicates that only services on the day indicated in the origin or destination concerned will be examined. Without this possibility, the search for itineraries lasting several days or very infrequent, such as international regular bus lines or even questions of the "accessibility by staying N days at destination" type, can not be carried out

- Toll weight: toll weighting. This weighting may be different depending on the type of links.
- Algorithm: Graph Growth Algorithm at intervals; Dijkstra at intervals;
- Algorithm scale: 200 (see algorithm parameters);
- nb classes: 10000 (see algorithm parameters);
- Algorithm exponent: 2 (see algorithm parameters);
- Output paths: by activating this box, MUSLIW will write, for each origin-destination, the route, that is to say the succession of links and services with the different components of intermediate time reconstituting the route. In case of many lines in the matrix file, this file can quickly be very huge;
- Output links times?: in case of many lines in the matrix file, the output file with the detailed time can quickly be very large. It is therefore possible to choose the following options:
  - no output: no output;
  - without timetable links : write for each origin-destination, the travel times of all accessible individual transport-type links, but no time based links.
  - with timetable links: writing for each origin-destination the travel times of all accessible links of the individual or time based transport type.
  - compact mode: writing in the file \_nœuds.txt only the fields strictly necessary to generate the isochrones, ( o, numero, temps, tatt1, volau), useful in particular for the gravitational calculation which generates very voluminous files
- Output filter : This field allows you to specify the types of links that you want to include in the detailed output file. If nothing is specified, all accessible links will be printed in the result file. This filter is cumulative with the detailed time option selected. To filter several types of sections it is necessary to enter a list separated by a "| ". Ex: entering "1 | 2 | 4" as a filter will lead to include only accessible sections of type 1,2 or 4 in the detailed time output file
- Prohibited U-turns: If checked, it prohibits by default the ability to perform U-turns (same origin node and same end node in a rotating movement or a transfer). These prohibitions will also be effective for links of transport schedules;
- Output services : this option allows you to generate a results file giving details of the flow volumes per service, for non-zero ones;
- Output transfers: this option makes it possible to generate a file result specifying for each turning movement or transfer node, line by line.

- Output nodes: This option allows to generate a results file for nodes (from the links connected to the node with the minimum cost)

## 5.2. Advanced settings

The advanced settings allow you to apply weights and values depending on the type of network.

The parameters that can be set according to the type of network are:

- On-board weighting;
- Waiting time weighting ;
- Individual time weighting;
- Correspondence weighting;
- Individual time scaling factor;
- Minimum transfer time;
- Toll weighting

The parameterization is done by a key/value system per type of section, separating the different types by a "|" and the key and the value by a ":". Thus, in the example above, "5|2:35" means that the connection time is 5 minutes by default and 35 minutes for network type "2". When there is no particular value defined for a type in question, MUSLIW defaults to the value of the modality for which the key is absent or is equal to "0" (to be expected).

If no type is entered, the type of all the sections is set to "0" by default.

## 5.3. The “results” files

MUSLIW outputs four result files that provide different information:

- time and volume by origin-destination, for those requested in the input matrix;
- all accessible links within a time limit fixed by origin or destination, for all origins and destinations of the matrix;
- volumes by segment and line segment when these volumes are non-zero;
- the detailed paths for origin-destination requested in the input matrix.

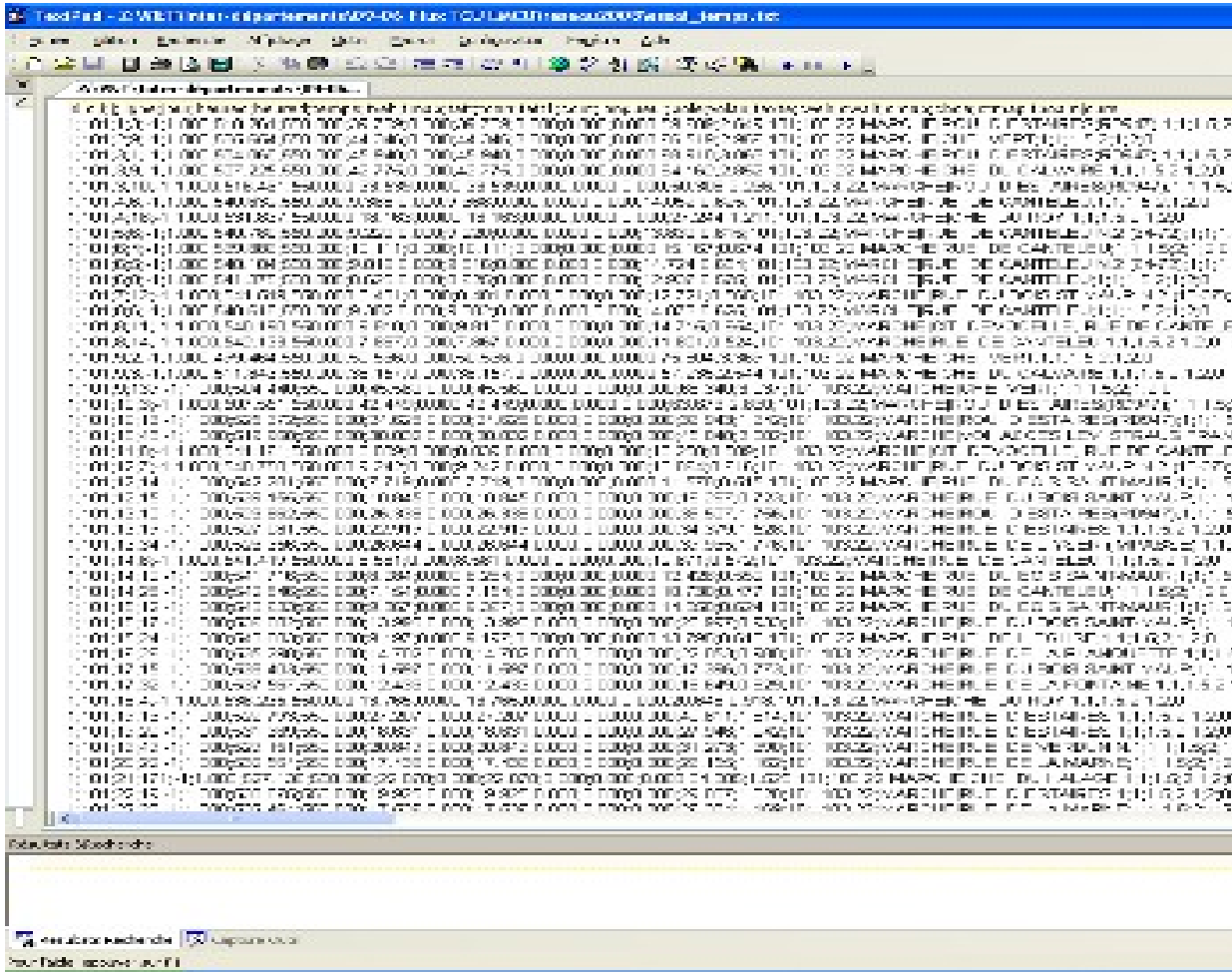




- tcorr: time spent in transfers, this time is equal to the number of trips made multiplied by the time of transfers;
  - ncorr: number of boardings. This number represents the number of boardings in a transit vehicle whose service is described by schedules;
  - tatt1: waiting time before the first boarding (direction d) or waiting time between the actual arrival time and the desired arrival time (direction a);
  - cout: generalized time minimized by the search algorithm for shortest paths.
- [generalized time] = [TC weight] \* [vehicle time] + [ATT weight] \* [wait time] + [MAP weight] \* [on time] + [number of TC trips] \* [COR time];
- longueur: cumulative length;
  - pole: node number of the first intermodal point (transition from the individual transport network to time based network). If the whole route is done on the individual network, the pole has the value of the node origin or destination according to the direction of calculation;
  - volau: volume of demand on the last link of the route;
  - texte: list of public transport lines used on the route. For this feature to provide suitable or actionable results, the names of the time based timetable links must start with an identifier (which may be of type characters) of the line followed by a "-". It is this identifier which will be taken up in the succession of the borrowed lines, otherwise, the identifier used will be the character string defined from the beginning of the name of the section until the first «-».
  - nbpop: number of iterations when calculating the shortest paths. This indicator makes it possible to optimize the parameters of the algorithm for a minimum calculation time. For an equivalent destination-origin, a smaller nbpop value indicates a more optimal computation time;
  - ttoll: cumulative toll along the route.



## 7. <FILENAME> TEMPS.TXT



This result file will only be filled in if the “Sortie temps” box has been activated. The structure of this file is identical to that of the file origins-destinations except for the last field on borrowed lines which is absent.

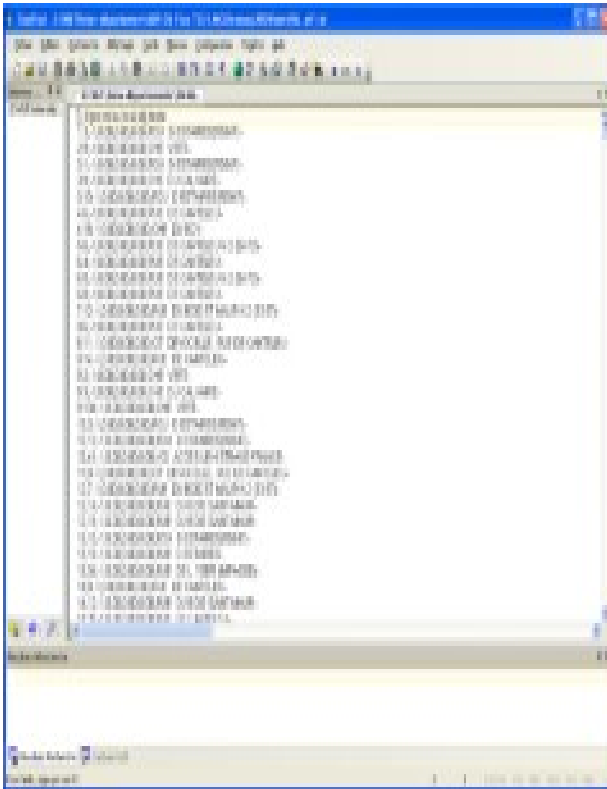
The first line indicates the name of the fields.

Then, for each line of the matrix file, the file gives the set of sections of the network that are accessible with the generalized time calculation parameters defined in the dialog box described in 4.3. This file can therefore quickly become very large for large networks:

- id: sequential number corresponding to the row number of the matrix file;
- o: origin node number (direction d) or destination (direction a) of movement;
- i: origin node number of the accessible section;
- j: end node number of the accessible section;
- numtrc: number identifying the section
- jour: day of travel;
- heureo: departure time of the trip:
  - if time type = "d", this is the desired start time indicated in the matrix file;
  - if type of time = "a", it is the latest departure time to reach the destination before the arrival time;

- heured: arrival time of the trip:
    - if type of time = "d", it is the earliest arrival time to reach the destination starting at the departure time;
    - if type of time = "a", this is the desired arrival time indicated in the matrix file;
  - temps: travel time between the origin node and the destination time (minutes). It is given by the formula  $\text{time} = [\text{heured}] - [\text{heureo}]$ ;
  - tveh: time spent on public transport (minutes);
  - tmap: walking or individual transport time (minutes);
  - tatt: waiting time at a stop before boarding a public transport (minutes);
  - tcorr: time spent in transfers, this time is equal to the number of trips made in TC multiplied by the time of matches;
  - ncorr: number of transfers. This number corresponds to the number of rides in a transit vehicle whose service is described by schedules;
  - tatt1: waiting time before getting on the first TC (direction d) or waiting time between the actual arrival time and the desired arrival time (direction a);
  - cost: generalized time minimized by the search algorithm for shorter paths.
- $[\text{generalized time}] = [\text{TC weight}] * [\text{vehicle time}] + [\text{ATT weight}] * [\text{wait time}] + [\text{MAP weight}] * [\text{on time}] + [\text{number of TC trips}] * [\text{COR time}]$ ;
- length: cumulative length;
  - pole: node number of the first intermodal point (transition from the individual transport network to the hourly network). If the whole route is done on the individual network, the pole has the value of the node origin or destination according to the direction of calculation;
  - volau: volume of demand on the last link of the route;
  - previous: Number of the previous section. This field makes it possible to reconstitute the routes by going back successively the preceding sections
  - type: link type indicator
  - toll: Cumulative toll along the route
  - ti: travel time between the origin node and the destination time (minutes) at the other end of the link. For example ti temps is relating to the time at the end node of the link, ti is relating yo the time a the start node of the link, and reciproqually.

## 8. <FILENAME> AFF.txt



The assignment results file is a "Delimited text" file with ";" as separator. It contains a record by link and line segment and includes as fields:

- i: start node number of the section;
- j: end node number of the section;
- line :
  - TC line identifier if the mode is at times;
  - -1 if the mode is of the individual transport type;
- volau: volume assigned on the link;
- boai: number of boardings to the initial node;
- alij: number of alighting sat the end node;
- text: name of the section.
- type: type of section
- toll: toll of the section



- temps: travel time between the origin node and the destination time (minutes);

time = [heured] - [heureo];

- heuro: time at the origin of the link if starting point or at arrival if point of arrival;
- tveh: time spent on public transport (minutes);
- tmap: walking or individual transport time (minutes);
- tatt: waiting time at a stop before boarding a public transport (minutes);
- tcorr: time spent in transfers, this time is equal to the number of trips made in public transport multiplied by the time of matches;
- ncorr: number of transfers. This number corresponds to the number of boarding in a transit vehicle whose service is described by schedules;
- tatt1: waiting time before getting on the first time based vehicle (direction d) or waiting time between the actual arrival time and the desired arrival time (direction a);
- cout: generalized time minimized by the search algorithm for shortest paths.

[generalized time] = [in vehicle weight] \* [in vehicle time] + [wait weight] \* [wait time] + [individual modes weight] \* [individual modes time] + [number of boardings] \* [transfer time];

- length: cumulative length;
- pole: node number of the first intermodal point (transition from the individual transport network to the time based network). If the whole route is done on the individual network, the pole has the value of the node origin or destination according to the direction of calculation;
- volau: volume of demand on the link for this specific trip;
- boai: number of boardings to the node origin of the link for the considered trip;
- alij: number of alightings at the destination node of the link for the considered trip;
- text: name of the link;
- type: type of link
- toll: detail of the cumulative toll along the way

By consulting this file and for all the lines having the same origin and destination nodes, it is possible to know the route taken on the chosen origin-destination, to know the volumes, boarding and alighting by service and to make detailed analyzes ( trees, paths taken, ...)

## 10. <FILENAME> NOEUDS.TXT

The output file by nodes is in the directory that the user specified by clicking "OK". It is a "Delimited text" file with ";" as separator. It contains one line per rline of the "matrix" file with fields as:

- id: sequential number corresponding to the line number of the matrix file or identifier of origin-destination;
  - o: origin node number of the trip;
  - d: destination node number of the trip;
  - jour: day of travel;
  - i: node identifier
  - heureo: departure time of the trip:
    - if time type = "d", this is the desired start time indicated in the matrix file;
    - if type of time = "a", it is the latest departure time to reach the destination before the arrival time;
  - heured: arrival time: arrival time of the trip:
    - if type of time = "d", it is the earliest arrival time to reach the destination starting at the departure time;
    - if type of time = "a", this is the desired arrival time indicated in the matrix file;
  - temps: travel time between the origin node and the destination time (minutes):
    - time = [arrival time] - [departure time];
  - tveh: time spent on public transport (minutes);
  - tmap: walking or individual transport time (minutes);
  - tatt: waiting time at a stop before boarding a public transport (minutes);
  - tcorr: time spent in transfers, this time is equal to the number of boardings multiplied by the time of transfers;
  - ncorr: number of transfers. This number represents the number of boardings in a transit vehicle whose service is described by schedules;
  - tatt1: waiting time before getting on the first vehicle (direction d) or waiting time between the actual arrival time and the desired arrival time (direction a);
  - cout: generalized time minimized by the search algorithm for shortest paths.
- [generalized time] = [in vehicle weight] \* [in vehicle time] + [wait weight] \* [wait time] + [individual modes weight] \* [individual modes time] + [number of boardings] \* [transfer time];
- length: cumulative length;
  - pole: node number of the first intermodal point (transition from the individual transport network to the time based network). If the whole route is done on the individual network, the pole has the value of the node origin or destination according to the direction of calculation;
  - ttoll: cumulative toll along the route.

## 11. <NOM FICHIER> LOG.TXT

MUSLIW provides a small log file that summarizes information that can be useful when you want to find the files and parameters used or evaluate the differences in computation time corresponding to algorithm parameters. Here is an example of log file (in french) :

Version: 1.5.1.211

Début de la procédure: vendredi 04 mars 2011 11:39:16.968

Paramètres par défaut:



Temps correspondance minimum par défaut:2  
Temps correspondance maximum par défaut:120  
Pondération correspondance:5  
Pondération attente:1  
Pondération temps à horaires:1  
Pondération temps individuel:1.5  
Coefficient temps individuel:1  
Temps individuel maximal:30  
Pondération du péage:0  
Nombre de jours:1  
Interdiction demi-tours:False  
Algorithme:0  
Nombre d'intervalles:10000  
Paramètre d'échelle de l'algorithme:15  
Paramètre exposant de l'algorithme:2  
Sortie chemins:True  
Sortie temps:1  
Noms fichiers sortie:C:\palmier\reseaux\tutu  
Réseau:C:\palmier\reseaux\transpole2006\_musliw.txt  
Noeuds:32084  
Liens:102580  
Services horaires:359508  
Matrice:C:\palmier\reseaux\matrice\_dt\_2.txt  
Début Calcul: vendredi 04 mars 2011 11:39:23.906  
Fin Calcul: vendredi 04 mars 2011 11:39:35.203  
Temps Calcul:11.296875 sec

The log file summarizes:

- the version of Musliw
- the default calculation parameters (the specific parameters by origin-destination are in the result files);
- the paths of the input and output files;
- size elements of the network (nodes, links, turning movements and transfers, services);
- elements to evaluate the calculation time;
- non accessible origins and / or destinations.

## 12. <FILENAME> SERVICES.TXT

The detailed services output file is a "Delimited text" file with ";" as separator. It describes in detail all the volumes of flows and boardings-alightings per service.

```
i;j;ligne;service;hd;hf;regime;volau;boia;alij;texte;type
908272;688583;1026;1026;1065;1145;OOOOOOO;1.00;0.00;0.00;ZRH DUSSwiss/Crossair;2
8000082;8000085;9510136;9510136;1291;1297;OOOOOOO;1.00;0.00;0.00;Düsseldorf Flughafen Düsseldorf Hbf;1
8000084;8003553;9510136;9510136;1359;1365;OOOOOOO;1.00;0.00;0.00;Düren Langerwehe;1
8000085;8001584;9510136;9510136;1300;1305;OOOOOOO;1.00;0.00;0.00;Düsseldorf Hbf Düsseldorf-Benrath;1
8000178;8000084;9510136;9510136;1348;1358;OOOOOOO;1.00;0.00;0.00;Horrem Düren;1
8000207;8000208;9510136;9510136;1335;1339;OOOOOOO;1.00;0.00;0.00;Köln Hbf Köln-Ehrenfeld;1
8000208;8000178;9510136;9510136;1340;1347;OOOOOOO;1.00;0.00;0.00;Köln-Ehrenfeld Horrem;1
8000209;8003368;9510136;9510136;1322;1327;OOOOOOO;1.00;0.00;0.00;Köln-Mülheim Köln Messe/Deutz;1
8000348;8000406;9510136;9510136;1377;1382;OOOOOOO;1.00;0.00;0.00;Stolberg(Rhein)Hbf Aachen-Rothe Erde;1
8000406;8000001;9510136;9510136;1383;1387;OOOOOOO;1.00;0.00;0.00;Aachen-Rothe Erde Aachen Hbf;1
8001584;8006713;9510136;9510136;1306;1314;OOOOOOO;1.00;0.00;0.00;Düsseldorf-Benrath Leverkusen Mitte;1
8001886;8000348;9510136;9510136;1373;1376;OOOOOOO;1.00;0.00;0.00;Eschweiler Hbf Stolberg(Rhein)Hbf;1
8003368;8000207;9510136;9510136;1329;1332;OOOOOOO;1.00;0.00;0.00;Köln Messe/Deutz Köln Hbf;1
8003553;8001886;9510136;9510136;1366;1372;OOOOOOO;1.00;0.00;0.00;Langerwehe Eschweiler Hbf;1
8006713;8000209;9510136;9510136;1315;1321;OOOOOOO;1.00;0.00;0.00;Leverkusen Mitte Köln-Mülheim;1
```

- i: origin node;
- j: destination node;
- ligne: line number;
- service: service number;
- hd: start time at node i;
- hf: arrival time at node j;
- regime: timetable for the circulation of the service;
- volau: volume of service flow;
- boai: number of boardings at node i;
- alij: number of alightings at node j;
- texte: description of the link;
- type: type of links
- 
- Only services with non-zero flow are reported into the file.

### 13. <FILENAME> TURNS.TXT

The turns output file is a "Delimited text" file with ";" as separator. It describes in detail all of the turning movements and transfers at a junction or stop.

```
j;i;lignei;textei;k;lignek;textek;volau;
8000082;8000085;9510115; Dusseldorf Hbf Dusseldorf Flughafen;688583;-1; DUS - Duesseldorf Flughafen ;1;a
688583;8000082;-1; Duesseldorf Flughafen - DUS ;908272;1026;ZRH DUSSwiss/Crossair;1;a
8000084;8003553;9510141; Langerwehe Dren;8000178;9510136; Horrem Dren;1;a
8000207;8000208;9511313; Kln-Ehrenfeld Kln Hbf;8003368;9510136; Kln Messe/Deutz Kln Hbf;1;a
8000406;8000001;9510139; Aachen Hbf Aachen-Rothe Erde;8000348;9510136; Stolberg(Rheinl)Hbf Aachen-Rothe Erde;1;a
8000085;8001584;9511014; Dusseldorf-Benrath Dusseldorf Hbf;8000082;9510136; Dusseldorf Flughafen Dusseldorf Hbf;1;a
8000178;8000084;9510139; Dren Horrem;8000208;9510136; Kln-Ehrenfeld Horrem;1;a
8003553;8001886;9510137; Eschweiler Hbf Langerwehe;8000084;9510136; Dren Langerwehe;1;a
8001584;8006713;9510139; Leverkusen Mitte Dusseldorf-Benrath;8000085;9510136; Dusseldorf Hbf Dusseldorf-Benrath;1;a
8003368;8000207;9212850; Kln Hbf Kln Messe/Deutz;8000209;9510136; Kln-Mlheim Kln Messe/Deutz;1;a
8000208;8000178;9510139; Horrem Kln-Ehrenfeld;8000207;9510136; Kln Hbf Kln-Ehrenfeld;1;a
8000209;8003368;9510137; Kln Messe/Deutz Kln-Mlheim;8006713;9510136; Leverkusen Mitte Kln-Mlheim;1;a
8006713;8000209;9510135; Kln-Mlheim Leverkusen Mitte;8001584;9510136; Dusseldorf-Benrath Leverkusen Mitte;1;a
8000348;8000406;9510139; Aachen-Rothe Erde Stolberg(Rheinl)Hbf;8001886;9510136; Eschweiler Hbf Stolberg(Rheinl)Hbf;1;a
8001886;8000348;9510139; Stolberg(Rheinl)Hbf Eschweiler Hbf;8003553;9510136; Langerwehe Eschweiler Hbf;1;a
```

- j: node corresponding to a junction or a stop;
- i: origin node;
- ligne i: line number of the incoming;
- textei: label of the origin link;
- k: destination node number;
- lignek: line number of the destination link;
- textek: label of the destination link;
- volau: volume of the flow of turning movements or line-to-line transfers.
- 
- Only the turning movements or line-by-line transfers volumes, whose flow volume are greater than-zero, appear in the file.