Étude théorique : stations de base du réseau téléphonique français

Progression du stage

Paul MÉHAUD, Brendan SÉVELLEC

České Vysoké Učení Technické v Praze

mai - août 2024

Déroulé de la présentation

Introduction

Contexte général

Objectifs

- Déterminer si les stations de base sont en zone urbaine ou rurale ;
- Chercher les stations de bases voisines les unes des autres pour aider à déterminer si les utilisateurs sont en mouvement.

Méthodes

- Approche par la théorie des graphes ;
- Approche par le machine learning.

Données

Données

Arcep

Autorité de régulation des communications électroniques, des postes et de la distribution de la presse.

Jeu de données

Le jeu de données `2023_T4_sites_Metropole.csv` [a] représente les stations de bases au trimestre 4 de 2023 avec leur position géographique (taille : $16,7\,\mathrm{Mo}$).

a. `https://data.arcep.fr/mobile/sites/`

A retenir :

- $108\,838$ sites ;
- 29 attributs.

A quoi ressemble notre base ?

code_op	nom_op	num_site	id_site_partage	id_station_anfr	x	y	latitude	longitude	nom_reg
20801	Orange	00000001A1	nan	0802290015	687035	6985761	49,97028	2,81944	Hauts-de-France
20801	Orange	00000001B1	nan	0642290151	422853	6249263	43,28861	-0,41389	Nouvelle-Aquitaine
20801	Orange	00000001B2	nan	0332290026	416932	6422196	44,84112	-0,58333	Nouvelle-Aquitaine
20801	Orange	00000001B3	nan	0472290005	511106	6349234	44,21666	0,63556	Nouvelle-Aquitaine
20801	Orange	00000001C1	nan	0512290147	836824	6889450	49,09028	4,87333	Grand Est

nom_dep	insee_dep	nom_com	insee_com	site_2g	site_3g	site_4g	site_5g	mes_4g_trim	site_ZB
Somme	80	Curlu	80231	1	1	1	0	0	0
Pyrénées-Atlantiques	64	Jurançon	64284	1	1	1	1	0	0
Gironde	33	Bordeaux	33063	1	1	1	1	0	0
Lot-et-Garonne	47	Agen	47001	1	1	1	0	0	0
Marne	51	Sainte-Menehould	51507	1	1	1	0	0	0

site_DCC	site_strategique	site_capa_240mbps	date_ouverturecommerciale_5g	site_5g_700_m_hz	site_5g_800_m_hz
0	0	0	nan	0	0
0	0	1	2020-12-14	0	0
0	0	1	2021-02-22	0	0
0	0	1	nan	0	0
0	0	1	nan	0	0

site_5g_1800_m_hz	site_5g_2100_m_hz	site_5g_3500_m_hz
0	0	0
0	1	0
0	0	1
0	0	0
0	0	0

Table 1 – Premières valeurs de la base

Description (1/2)

Ce qui nous intéresse

1. *longitude*, *latitude* : coordonnées de chaque site ;
2. *nom_op* : nom commercial de l'opérateur ;
3. *nom_reg*, *nom_dep* et *nom_com* : nom de la région, du département et de la commune d'implantation du site ;
4. *site_xg* : équipement du site en technologie xG ($x \in \{2, \ldots, 5\}$) ;
5. *num_site* : identifiant du site issu du SI de l'opérateur.

Description (2/2)

Ce qu'il faut retenir

1. Répartition équitable du nombre de sites en fonction de l'opérateur ($\simeq 27\,000$) ;
2. $99,6\%$ des sites équipés en 4G ;
3. 6 stations en moyenne par commune.

La construction de cette base ne nous permet pas de faire de statistiques descriptives intéressantes.

From 08/07/24 to 12/07/24

ČVUT
ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

From 08/07/24 to 12/07/24

further advancement on road detection

Cities and big cities connection graph

Figure 1 – cities and big cities connection graph

Little cities linking method

Steps

- For each edge of the connection graph ;
- Detect all little cities close enough to this edge ;
- Create multiple edges to link all these cities together by a path with the same start and ending city than the big edge ;

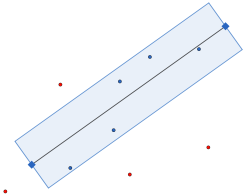

Figure 2 – Little cities selection

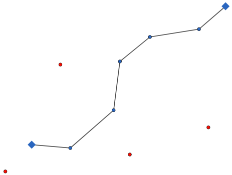

Figure 3 – Cities linkage

New graph after little city linkage

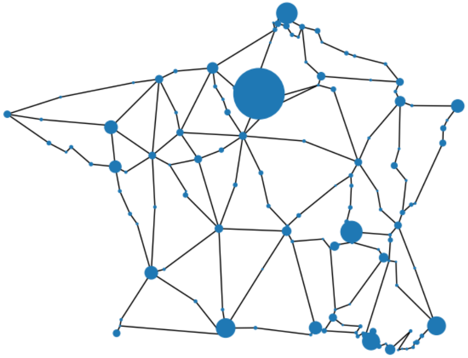

Figure 4 – Final graph (width parameter = 0.2)

New graph on a map

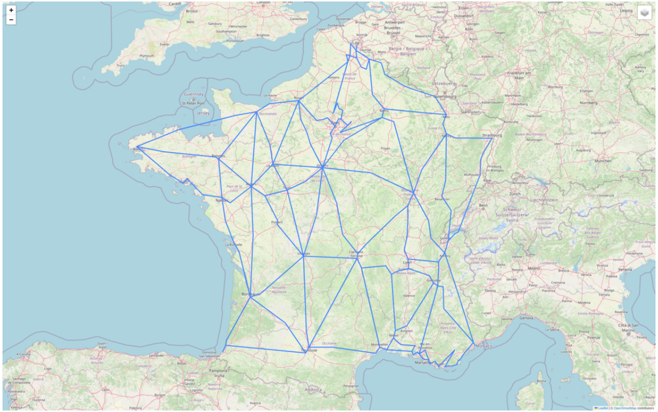

Figure 5 – Final graph on a map (width parameter $= 0.2$)

Reminder : pre-road detection thanks to the clustering methods

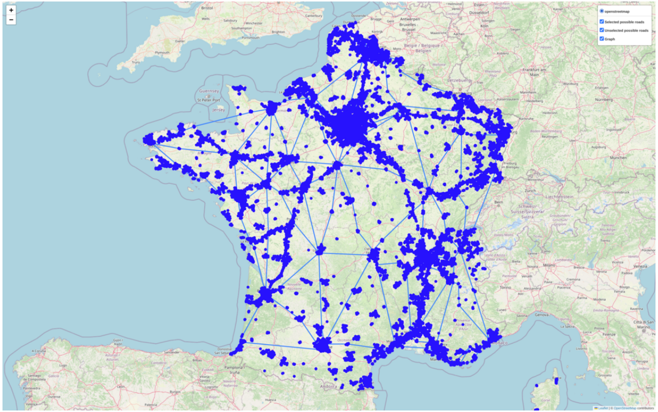

Figure 6 – Road pre-detection and graph

Final selected road base stations

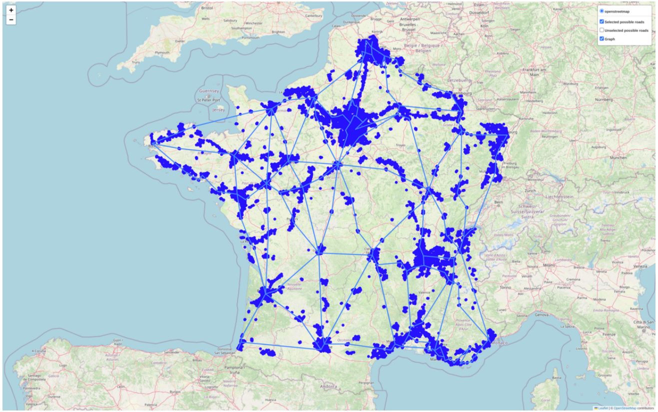

Figure 7 – Nodes detected as roads after the method

From 08/07/24 to 12/07/24

Further advancement on Coverage Calculation

Creation of New Database

- We have created a new database containing detailed information about base stations and their antennas in Normandy, France.

Database Fields

- Station ID (`id_station_anfr`)
- Latitude and Longitude
- Department and Commune Names
- Presence of 2G, 3G, 4G, and 5G technologies
- Antenna Dimensions and Azimuths

Program for Visualization

- We implemented a Python program to visualize the base stations on a map and analyze their coverage using Voronoi diagrams.

Key Steps in the Program

- Load and clean the data.
- Add markers and pop-ups for each base station.
- Calculate and visualize Voronoi cells and antenna sectors.

Visualization Results

- The map is centered on Normandy and shows base stations as markers.
- Each marker provides detailed information about the station and its antennas.
- Voronoi tessellation is used to visualize the coverage area of each base station.
- Antenna azimuths are displayed to indicate coverage directions.

Example Visualization

- The green lines represent Voronoi boundaries.
- The markers indicate base station locations with detailed popup information.
- Direction lines show the azimuths of the antennas.

From 15/07/24 to 18/07/24

From 15/07/24 to 18/07/24

further advancement on road detection

Little advancement

City name detection

For each city detected, we find the closest base station to the center of the said city. We then declare that the name of the city is the value contained in the field "nom_com" of the base station.

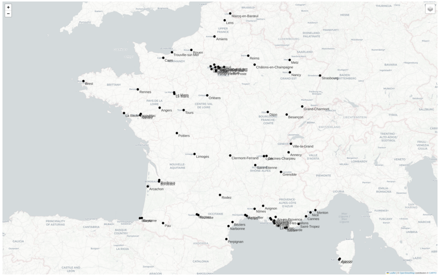

Figure 8 – Cities centers with name

New method (1/2)

Edges weight

For each couple of cities, we compute a weight using this formula :

$$w_{\text{city1, city2}} = \frac{\min(\text{size(city1)}, \text{size(city2)})}{\text{dist(city1, city2)}}$$

With the size of a city refering to the number of base stations detected inside that city.
We the keep all the edges whose weight is superior to a certain value.

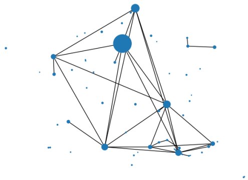

Figure 9 – weight filtration, cap value $= 0.1$

New method (2/2)

Edges weight

We then apply the angle criterion to the resulting graph

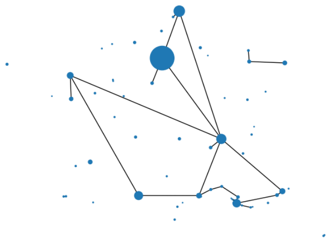

Figure 10 – weight and angle filtration, cap value $= 0.1$, angle value $= 15$ degrees

From 15/07/24 to 18/07/24

Method for Finding Neighboring Base Stations Using Voronoi Diagrams

ČVUT
ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Voronoi Diagrams for Neighboring Base Stations

- The method uses Voronoi diagrams to determine the neighborhood of base stations.
- If the cells share a common boundary, the base stations are considered neighbors.

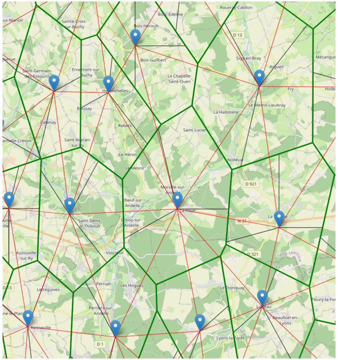

Figure 11 – Voronoi Diagram Showing Neighboring Base Stations

Analysis of Results

Explanation of inaccuracy

- The method considers only geometric proximity without accounting for actual network topology or physical barriers.
- It's analogous to using Delaunay triangulation and assuming all neighboring stations in the triangulation are true neighbors.

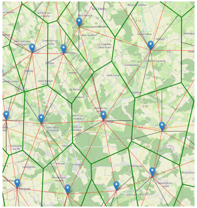

Figure 12 – Voronoi Diagram Showing Neighboring Base Stations

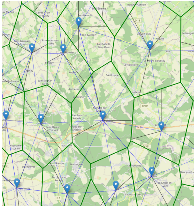

Figure 13 – Delaunay Triangulation of Base Stations

From 15/07/24 to 18/07/24

Introduction to the New Method for Determining Neighboring Base Stations

New Method for Determining Neighboring Base Stations

- The method uses Delaunay triangulation to identify potential neighboring base stations.
- Verification of neighboring status is done by checking antenna coverage angles.
- Stations are considered neighbors if their antenna directions align within their coverage angles.

Determining Antenna Coverage Angles

- Calculate the coverage angle for each antenna by dividing 360 degrees by the number of unique azimuths.
- This gives the angular sector covered by each antenna on the station.

Validating Neighboring Status

- For each pair of stations connected by an edge, check the direction of the connecting line.
- Compare this direction with the antenna coverage angles of both stations.
- If the direction falls within the coverage angles, the stations are real neighbors.

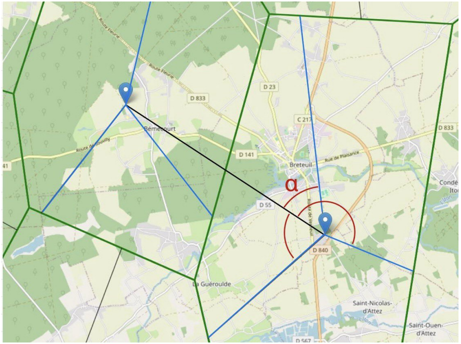

Figure 14 – Validation of Neighboring Status

Detailed Steps with Formulas

- Calculate Direction of the Connecting Line:

$$\theta = \arctan 2(y_2 - y_1, x_2 - x_1)$$

  where (x_1, y_1) and (x_2, y_2) are the coordinates of the two base stations.
- Calculate the Antenna Coverage Angles:

$$\text{Coverage Start Angle} = A - \frac{B}{2}$$

$$\text{Coverage End Angle} = A + \frac{B}{2}$$

  For an antenna with azimuth A and beamwidth B.
- Validate if the Connecting Line Falls Within Coverage Angles: Normalize angles to be within $[0^\circ, 360^\circ]$. Check if the direction θ falls within $[A - \frac{B}{2}, A + \frac{B}{2}]$.
- Bi-Directional Check: Repeat the validation from both base stations' perspectives. If $\Delta\theta$ for both stations is within the beamwidth, they are considered true neighbors.

Updating the Dataset

- Add validated neighboring stations to the dataset.
- Update the data frame with pairs of stations identified as real neighbors.

Result

- Improved accuracy in identifying neighboring base stations.
- More reliable data for network and coverage analysis.

Summary and Conclusion

- The new method combines geometric triangulation with antenna direction validation.
- This approach enhances the accuracy of identifying neighboring base stations.
- This method can be further refined and applied to past methods as an additional criterion.

From 22/07/24 to 26/07/24

From 22/07/24 to 26/07/24

Further advancement on road detection

Adjustements in the method

Stop using the angle criterion

After analysing last week's results, I decided to stop using the angle criterion because it doesn't delete the right edges. Therefore, I fine-tuned the weight calculation method, so that it doesn't need it anymore

New edge weight calculation

For each couple of cities, we compute a weight using this formula :

$$w_{\text{city1, city2}} = \frac{(\min(\text{size}(\text{city1}), \text{size}(\text{city2})))^{1.3}}{(\text{dist}(\text{city1, city2}))^{1.2}}$$

With the size of a city refering to the number of base stations detected inside that city.

Results

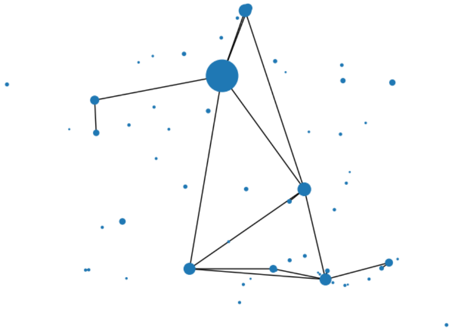

Figure 15 – weight filtration with new calculation, cap value $= 0.2$

Figure 16 – TGV lines covered in 4G by Orange

From 22/07/24 to 26/07/24

State of the art of our different neighbouring methods

Delaunay and simple criteria

Principle of the method

At first we will apply a Delaunay triangulation, to have a potential neighbours graph, and, afterwards, we apply the simple criteria : distance, angle and quadrant, to filter the bad neighbouring connexions.

Criteria

- Distance: we filter every edge longer than $15\,\mathrm{km}$;
- Angle : we filter every edge separated from the nearest one by an angle more narrow than $20°$;
- Quadrant : we filter according to the simple quadrant criterion.

Delaunay and simple criteria - pros and cons

Pros

- Every edge is treated equally, wether it in countryside or in a city;
- This method is really simple to put in place.

Cons

- Every edge is treated equally, wether it in countryside or in a city. This leads to non precise results.

Delaunay and enhanced criteria

Principle of the method

At first we will apply a Delaunay triangulation, to have a potential neighbours graph, and, afterwards, we apply the enhanced criteria : distance, angle and quadrant, to filter the bad neighbouring connexions. Same principle as before but we will take in account if a city is in or not in a city.

3NN cityness classification method

To classify each base station we need a method. Here, we will take the mean distance to the 3 nearest neighbours. Let's call this value γ, in km. So, we have 4 different categories. For each category, we will apply a different distance and angle criterion :

- $\gamma \in]0, 1]$: city center (max_distance $= 2\,\mathrm{km}$, min_angle $= 5°$);
- $\gamma \in]1, 2]$: urban area (max_distance $= 5\,\mathrm{km}$, min_angle $= 15°$);
- $\gamma \in]2, 4]$: extra-urban area (max_distance $= 10\,\mathrm{km}$, min_angle $= 25°$);
- $\gamma \in]4, \infty[$: courntryside (max_distance $= 15\,\mathrm{km}$, min_angle $= 30°$).

And then, of course, we apply the enhanced quadrant criterion.

Delaunay and enhanced criteria - pros and cons

Pros

- We take in account the "cityness" of each base station to compute its neighbours.

Cons

- We have no method to be sure that we choose the correct distances and angles;
- The filtering does not seems good sometimes. On one hand we filter too much and on the other hand, we filter too less.

Those problems could come from the Delaunay triangulation, which is maybe not the base for this use case.

Gabriel graph and enhanced criteria

Principle of the method

At first we will create a potential neighbours graph : the Gabriel graph and, afterwards, we apply the enhanced criteria : distance, angle and quadrant, to filter the bad neighbouring connexions. Same principle as before but we apply a pre-filtering with the Gabriel graph.

Gabriel graph [1]

A Gabriel graph is a subgraph of a Delaunay triangulation. Formally, it is the graph G with vertex set S in which any two distinct points $p \in S$ and $q \in S$ are adjacent precisely when the closed disc having pq as a diameter contains no other points.

And then, of course, we apply the enhanced criteria, as listed before.

1. https://en.wikipedia.org/wiki/Gabriel_graph

Gabriel graph and enhanced criteria - pros and cons

Pros

- We add another level of filtering.

Cons

- We add another level of complexity.

This is not perfect, maybe we can get rid of the Delaunay triangulation.

k-NN and enhanced criteria

Principle of the method

At first we will create a potential neighbours graph with the k-NN method and, afterwards, we apply the enhanced criteria : distance, angle and quadrant, to filter the bad neighbouring connexions.

k-NN [2]

To create this graph we will, for each base station, connect it to its k nearest neighbours.

And then, of course, we apply the enhanced criteria, as listed before.

2. `https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.NearestNeighbors.html`

Gabriel graph and enhanced criteria - pros and cons

Pros

- We get rid of the Delaunay traingulation, so we could connect more base stations.

Cons

- A lot of edges to filter;
- The filtering methods is maybe not adapted.

This is not perfect, but got rid of the Delaunay triangulation.

From 29/07/24 to 02/08/24

ČVUT
ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

From 29/07/24 to 02/08/24

Methodology for Identifying True Neighboring Base Stations Based on Antenna Azimuths

Improved Methodology

The goal is to identify true neighboring base stations by analyzing antenna azimuths.

Azimuth approach

True neighbors are stations within each others line of sight and within the coverage range of their directed antennas.

Defining Coverage Angles

- **Antenna beamwidth**
  - Each antenna has a specified azimuth and coverage angle (beamwidth).
  - Coverage angles are calculated based on azimuth direction and beamwidth, giving start and end angles.
- **Steps to Define Coverage Angles:**
  - For an antenna with azimuth A and beamwidth B:
    - Coverage Start Angle $= A - \frac{B}{2}$
    - Coverage End Angle $= A + \frac{B}{2}$

Verification Process

Step-by-Step Validation

Step 1: Calculate the Direction of the Connecting Line:

- Determine the azimuth angle of the line connecting the two base stations.

Step 2: Validate Antenna Direction for Both Base Stations:

- Check if the connecting line's direction falls within the coverage angles of its antennas.

Step 3: Determine the Shorter Angle:

- Calculate the angle between the azimuth of the antenna and the direction of the connecting line.
- Measure the smaller angle between the azimuth and the connecting line direction.

Step 4: Check Bi-Directional Coverage:

- Perform validation from both base stations' perspectives.
- If conditions are met for both, consider them true neighbors.

Step 5: Applying the Distance Criterion:

Analysis of Results

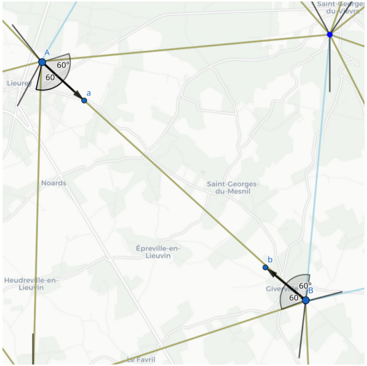

Figure 17 – Visualization of the antenna coverage angle based methodology

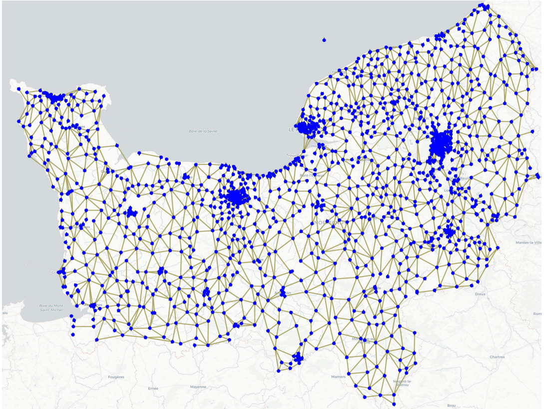

Figure 18 – Results on the map

Issue with Directional Antenna Coverage Methodology

Problem Explanation

- The current methodology focuses on identifying neighboring base stations based on the directional coverage of their antennas.
- However, this approach can miss real neighboring stations that have overlapping coverage areas but whose antennas are not directly oriented towards each other.
- As a result, some base stations that are effectively neighbors in terms of coverage might be excluded from the neighbor graph.
- This limitation highlights the need for a more nuanced approach that considers not just the directional angles but also the actual coverage areas.

Figure 19 – Coverage Overlap Issue

From 29/07/24 to 02/08/24

State of the art of different methods

Modification of criteria

New angle values

For each category, we will apply a different angle criterion :

- $\gamma \in]0,1]$: min_angle $= 5°$;
- $\gamma \in]1,2]$: min_angle $= 10°$;
- $\gamma \in]2,4]$: min_angle $= 15°$;
- $\gamma \in]4,\infty[$: min_angle $= 20°$.

New quadrant criterion

The idea is to use azimuths of each base station to make quadrants but as you will see results are really bad.

How to compare different methods (1/2)

For the comparison between the different methods we will need to put in place a way to compute dissimilarity between two methods.

Dissimiliarity computation

Let Δ be the value we are looking for. Let α the number of edges in method 1 not in method 2 and β the number of edges in method 2 not in method 1. So, we have:

$$\Delta = \alpha + \beta$$

And here is what we have when we compare to azimuth method :

Delaunay:	$\alpha = 0, \beta = 1008, \Delta = 1008$
Gabriel graph:	$\alpha = 684, \beta = 589, \Delta = 1273$
7-NN:	$\alpha = 302, \beta = 3095, \Delta = 3397$
15-NN:	$\alpha = 19, \beta = 9699, \Delta = 9718$

Table 2 – Dissimiliarity with unfiltered methods

How to compare different methods (2/2)

Delaunay and simple-criteria [adq]:	$\alpha = 919, \beta = 477, \Delta = 1396$	Delaunay and simple-criteria [dqa]:	$\alpha = 930, \beta = 489, \Delta = 1419$
Delaunay and enhanced-criteria [adq]:	$\alpha = 873, \beta = 464, \Delta = 1337$	Delaunay and enhanced-criteria [dqa]:	$\alpha = 885, \beta = 470, \Delta = 1355$
Delaunay and enhanced-criteria_v2 [adq]:	$\alpha = 1751, \beta = 362, \Delta = 2113$	Delaunay and enhanced-criteria_v2 [dqa]:	$\alpha = 1755, \beta = 393, \Delta = 2148$
Gabriel graph and enhanced-criteria [adq]:	$\alpha = 1098, \beta = 433, \Delta = 1531$	Gabriel graph and enhanced-criteria [dqa]:	$\alpha = 1098, \beta = 431, \Delta = 1529$
7-NN and enhanced-criteria [adq]:	$\alpha = 1023, \beta = 524, \Delta = 1547$	7-NN and enhanced-criteria [dqa]:	$\alpha = 1035, \beta = 540, \Delta = 1575$
15-NN and enhanced-criteria [adq]:	$\alpha = 899, \beta = 620, \Delta = 1519$	15-NN and enhanced-criteria [dqa]:	$\alpha = 938, \beta = 645, \Delta = 1583$
Delaunay and simple-criteria [aqd]:	$\alpha = 919, \beta = 477, \Delta = 1396$	Delaunay and simple-criteria [qad]:	$\alpha = 930, \beta = 489, \Delta = 1419$
Delaunay and enhanced-criteria [aqd]:	$\alpha = 873, \beta = 466, \Delta = 1339$	Delaunay and enhanced-criteria [qad]:	$\alpha = 883, \beta = 470, \Delta = 1353$
Delaunay and enhanced-criteria_v2 [aqd]:	$\alpha = 1751, \beta = 362, \Delta = 2113$	Delaunay and enhanced-criteria_v2 [qad]:	$\alpha = 1755, \beta = 393, \Delta = 2148$
Gabriel graph and enhanced-criteria [aqd]:	$\alpha = 1098, \beta = 435, \Delta = 1533$	Gabriel graph and enhanced-criteria [qad]:	$\alpha = 1098, \beta = 433, \Delta = 1531$
7-NN and enhanced-criteria [aqd]:	$\alpha = 1021, \beta = 525, \Delta = 1546$	7-NN and enhanced-criteria [qad]:	$\alpha = 1032, \beta = 540, \Delta = 1572$
15-NN and enhanced-criteria [aqd]:	$\alpha = 893, \beta = 612, \Delta = 1505$	15-NN and enhanced-criteria [qad]:	$\alpha = 935, \beta = 638, \Delta = 1573$
Delaunay and simple-criteria [daq]:	$\alpha = 919, \beta = 477, \Delta = 1396$	Delaunay and simple-criteria [qda]:	$\alpha = 930, \beta = 489, \Delta = 1419$
Delaunay and enhanced-criteria [daq]:	$\alpha = 873, \beta = 464, \Delta = 1337$	Delaunay and enhanced-criteria [qda]:	$\alpha = 883, \beta = 470, \Delta = 1353$
Delaunay and enhanced-criteria_v2 [daq]:	$\alpha = 1751, \beta = 361, \Delta = 2112$	Delaunay and enhanced-criteria_v2 [qda]:	$\alpha = 1755, \beta = 393, \Delta = 2148$
Gabriel graph and enhanced-criteria [daq]:	$\alpha = 1098, \beta = 433, \Delta = 1531$	Gabriel graph and enhanced-criteria [qda]:	$\alpha = 1098, \beta = 433, \Delta = 1531$
7-NN and enhanced-criteria [daq]:	$\alpha = 1023, \beta = 524, \Delta = 1547$	7-NN and enhanced-criteria [qda]:	$\alpha = 1032, \beta = 540, \Delta = 1572$
15-NN and enhanced-criteria [daq]:	$\alpha = 898, \beta = 620, \Delta = 1518$	15-NN and enhanced-criteria [qda]:	$\alpha = 935, \beta = 638, \Delta = 1573$

Table 3 – Dissimiliarity with filtered methods

From 29/07/24 to 02/08/24

Mean distances in departments

How all is constructed

The `Population`, `Superficie` and `Densite` data is from here : `https://france.ousuisje.com/departements/classement/superficie.php`.

Mean distance calculation

Let M be the value we are looking for. Let S be the set of one department's base stations. Let N_j be the set of distances to j's nearest neighbours, we will keep 10 of them.

$$M = \frac{\sum_{j \in S} \frac{\sum_{i \in N_j, i \leqslant 10} (N_j)_i}{\mathrm{size}(N_j)}}{\mathrm{size}(S)}$$

So M is the value called `total`.
For `countryside`, it's the same calculation but only on counstryside's base stations. For `city`, we have an additional criterion to get rid of bad values: we only keep, in the 10 nearest neighbours, the one that are less than $6\,\mathrm{km}$ distant.
By the way, all those values are in km.

We can also normalize these values with this formula: $M_{norm} = \frac{M - \min(Ms)}{\max(Ms) - \min(Ms)}$, where Ms is the set of M values for all departments.

Results

nom_dep	Population	Superficie	Densite	city	countryside	total	normalized_city	normalized_countryside	normalized_total
Paris	2166200	105	20433	0.21439	-1	0.21439	0	0	0
Hauts-de-Seine	1517000	176	8619	0.4315	-1	0.4315	0.07834	0	0.05889
Rhône	1667500	3249	513	1.04954	8.52619	1.08092	0.30134	0.79741	0.23503
Hérault	992500	6224	159	1.66912	7.83726	1.81089	0.52491	0.73974	0.43303
Loire-Atlantique	1209000	6815	177	1.75516	8.15056	1.99532	0.55595	0.76597	0.48305
Bas-Rhin	1063000	4755	224	1.78152	7.51702	1.7444	0.56546	0.71294	0.41499
Haute-Savoie	631679	4388	144	1.81713	8.8195	1.78003	0.57831	0.82196	0.42466
Gironde	1376000	10000	138	1.89034	8.44384	2.09651	0.60473	0.79052	0.5105
Seine-Maritime	1245000	6278	198	1.89128	7.1899	1.95555	0.60507	0.68555	0.47226
Pas-de-Calais	1456000	6671	218	1.92476	7.49829	1.96731	0.61715	0.71137	0.47545
Ille-et-Vilaine	930000	6775	133	2.08199	7.58191	2.2183	0.67388	0.71837	0.54353
Calvados	664000	5548	120	2.14504	8.43249	2.44803	0.69663	0.78957	0.60584
Haute-Corse	148000	4666	30	2.15674	9.06187	2.75204	0.70085	0.84225	0.6883
Indre-et-Loire	571500	6127	93	2.23539	8.35991	2.43487	0.72923	0.78349	0.60227
Hautes-Pyrénées	222368	4464	50	2.3107	9.45288	2.80779	0.75641	0.87498	0.70342
Finistère	852418	6733	127	2.32578	8.18865	2.52294	0.76185	0.76916	0.62616
Orne	292337	6103	48	2.41672	10.6345	3.74098	0.79466	0.97389	0.95653
Drôme	457651	6530	67	2.42289	7.99711	2.51054	0.79689	0.75312	0.62279
Ariège	137205	4890	28	2.44526	9.0807	2.83295	0.80496	0.84383	0.71024
Cantal	150772	5726	26	2.55952	9.05477	2.95966	0.84619	0.84166	0.74461
Jura	250857	4999	50	2.56287	8.79631	2.74338	0.8474	0.82002	0.68595
Manche	489500	5938	82	2.74607	8.70393	3.14209	0.9135	0.81229	0.79409
Creuse	124470	5565	22	2.76393	9.53859	3.53196	0.91994	0.88216	0.89984
Eure	541054	6040	90	2.8698	7.90577	2.93429	0.95814	0.74548	0.73773
Haute-Loire	209113	4977	42	2.88781	7.69017	2.85588	0.96464	0.72743	0.71646
Aveyron	271200	8735	31	2.91637	8.62305	3.11151	0.97495	0.80552	0.7858
Dordogne	401500	9060	43	2.9858	8.06246	3.04884	1	0.75859	0.7688

Table 4 – Mean distances (extract), *sorted by city* [in roman Normandy]