SANET:

Spatial Analysis along Networks

User Guide/Manual for SANET Standalone

June 14, 2020

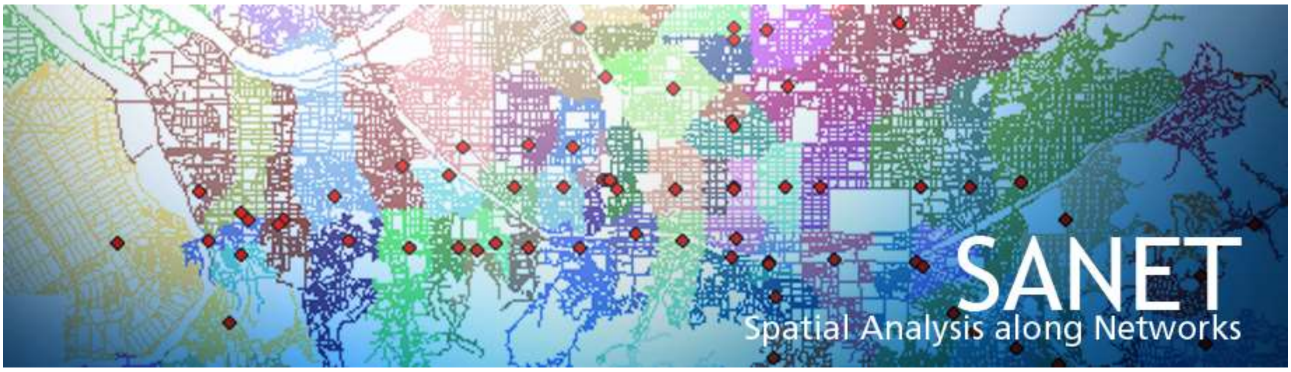


Table of Contents page

1. About SANET

1.1 Functions and Versions

SANET Standalone statistically analyzes spatial patterns of events that occur on/alongside networks.

Current version of SANET Standalone Version 1.0 is a licensed program.

The effective license period for SANET Standalone Version 1.0 is one year.

SANET is developed by the SANET Team (Leader: Atsu Okabe).

SANET Version 3.1 is for ESRI ArcGIS Ver.9.1 & 9.2 with Windows XP.

SANET Version 4.0 is for ESRI ArcGIS Ver.9.3 with Windows Vista and Windows XP.

SANET Version 4.1 is for ESRI ArcGIS Ver.10 with Windows 7, Vista and Windows XP.

1.2 Copyright

The program is copyrighted by PASCO and is intended for the use of students, academic researchers, non-profit researchers and educators.

SANET can be distributed freely on educational and research purposes, but cannot be re-sold.

1.3 Use conditions

SANET Team distributes the program only to those who agree on the following conditions

- The user will use SANET for nonprofit purposes only.
- The authors will not bear responsibility for any trouble that the user may meet in the use of SANET.
- When the user uses SANET, he/she will report to the authors his/her name, affiliation, address and e-mail address.
- When the user publishes any results obtained by using SANET, he/she will explicitly state in the paper that he/she used SANET. Also, he/she will send a reprint of the paper to the authors.
- The authors appreciate the reports of users which help us discover and isolate bugs within SANET.

1.4 Citation in publication

SANET must be cited correctly in any papers or publication that use results obtained from SANET. Also it should be acknowledged the use of SANET, Spatial Analysis along Networks developed by the SANET Team (leader: Atsu Okabe), Tokyo, Japan. In addition, in case any correspondence exists between specific members, his/her name is most preferably being cited.

1.5 Contact

SANET team contact information is as follows.
Request for distribution of SANET Program and sending papers should contact to
Atsu Okabe [atsu@csis.u-tokyo.ac.jp]

Technical questions relating to the SANET software and manual should contact to
Atsu Okabe [atsu@csis.u-tokyo.ac.jp]

1.6 I/O file types

The current Version is SANET Standalone 1.0.
The SANET program inputs spatial data (e.g. accident incidence spots data, retail store location data) with ESRI Shapefile.

The SANET program computes various spatial factors and gives results in forms of Shapefile, CSV file or R file for chart the result.

2. How to resister and run SANET Standalone

2.1 Hardware and Software requirements

Concerning the interface either ESRI ArcMap or QGIS Desktop is required to visualize the results obtained from running the SANET Standalone.

GNU R is also preferably installed for the better performance of the results.

2.2 Unzipping SANET Standalone executable file

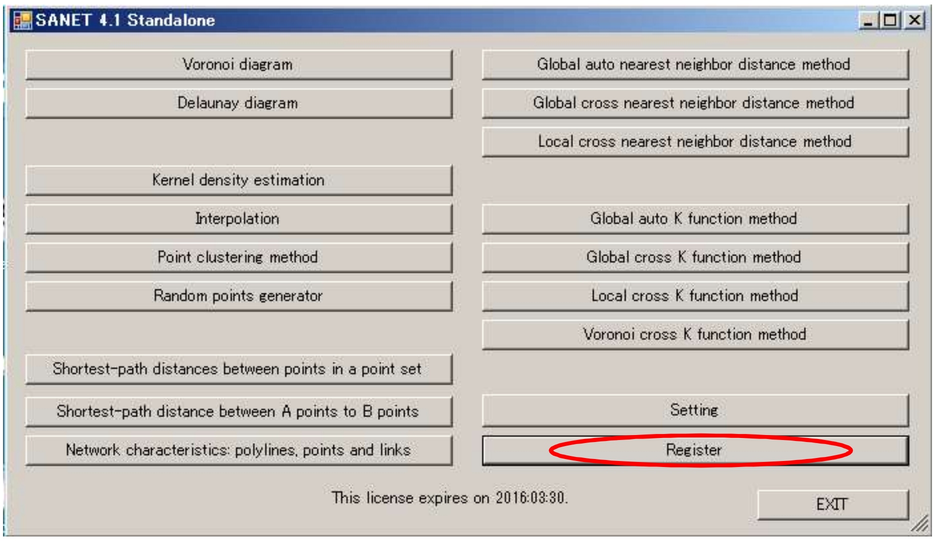
Firstly, download the SANET Standalone executable file (zip).

Secondly, unzip the file and put it in your preferable place.

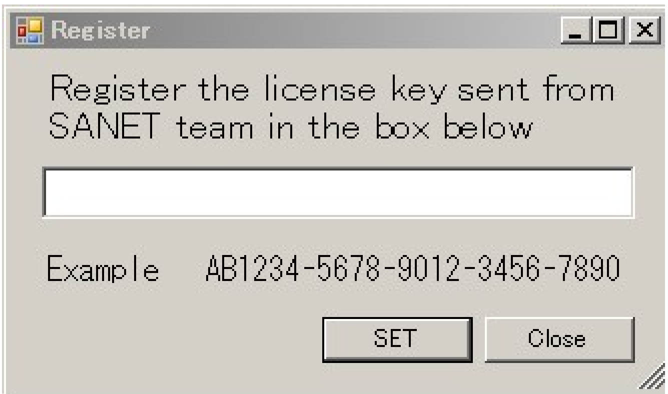
Thirdly, double click setup.exe. Since the setup wizard appears in Japanese, please refer to pp.77-78 for an illustration.

2.3 Placing license key to run SANET Standalone

Once you launched the SANET Standalone, below window will appear. On the bottom you will find the resister button.

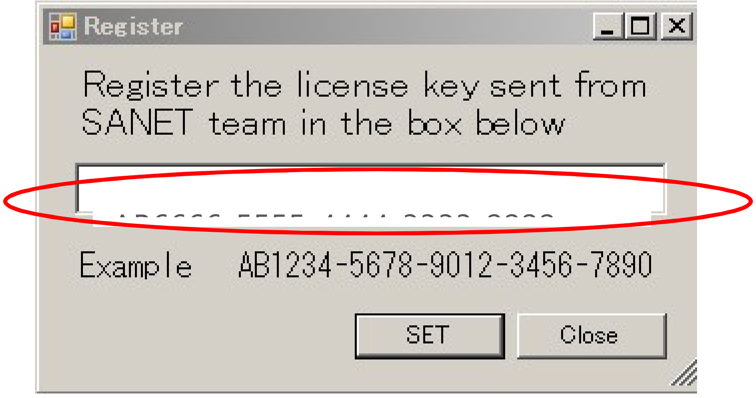


Click the button and the license register window appears.



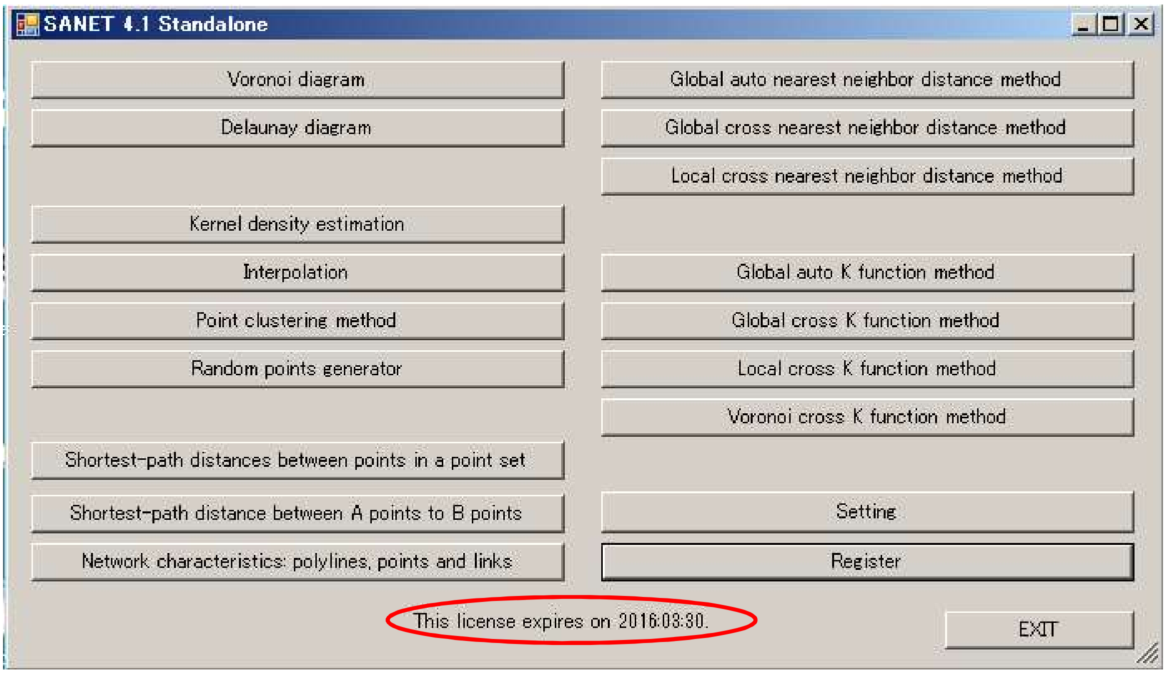
Place the license number which the SANET team have sent to you after receiving your registration form.

To obtain the license number, download a registration form, fill up the form and email it to the SANET contact person (see the section 1.5 above). Having qualified as an eligible user, the license key will be sent to you via email.



Input your license key to your SANET License Register.

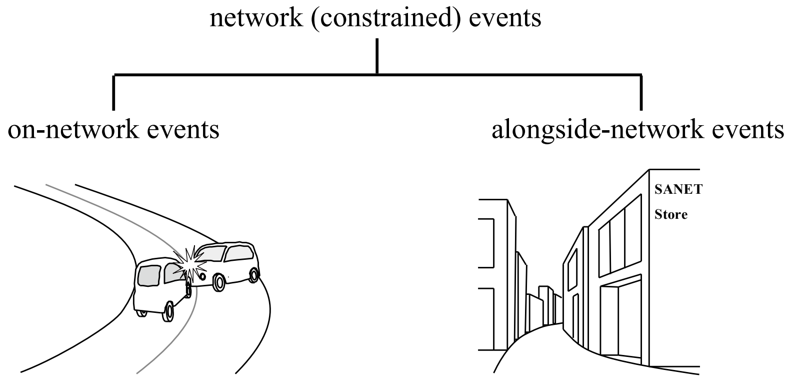
Click the SET button and you will find all SANET tools in your window.



Enjoy SANET until your license expires.

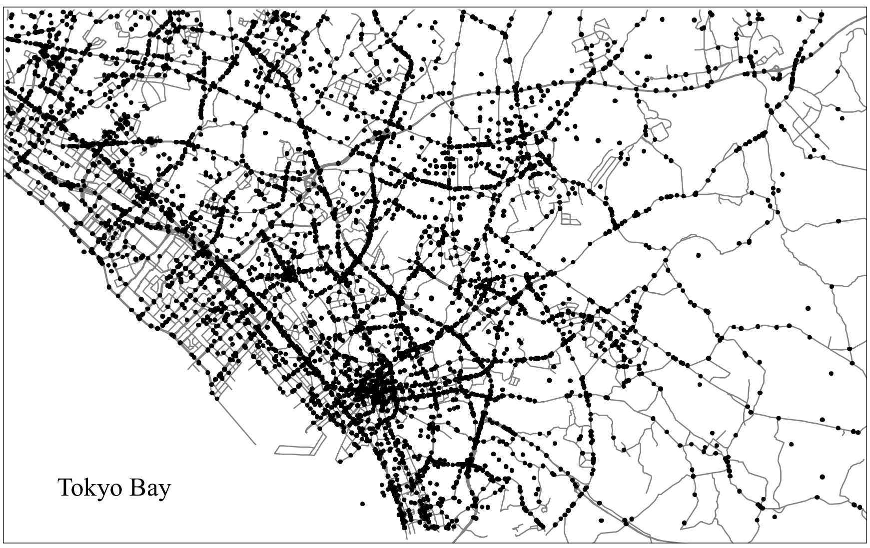
3. What is SANET?

In the real world, there are many and various kinds of network events. Those events may be classified into two types: the events occurring exactly on networks, termed *on-network events* and those occurring alongside networks, termed *alongside-networks* (Figure 1.1).



Source: Figure 1.1 in Okabe and Sugihara (2012). Network (constrained) events consisting of on-network events and alongside-network events.

Typical examples of on-network events are: traffic accidents (shown in Figure 1.2), road-kills of animals, street crimes, beaver lodges in watercourses, leakages in gas pipe lines and river contamination. Alongside-network events include advertisement agencies (Figure 1.3), fast-food shops, convenience stores, fashionable boutiques and other kinds of facilities locating alongside streets in urbanized areas. Almost all facilities in urbanized areas are regarded as alongside-network events because their entrances are adjacent to streets.



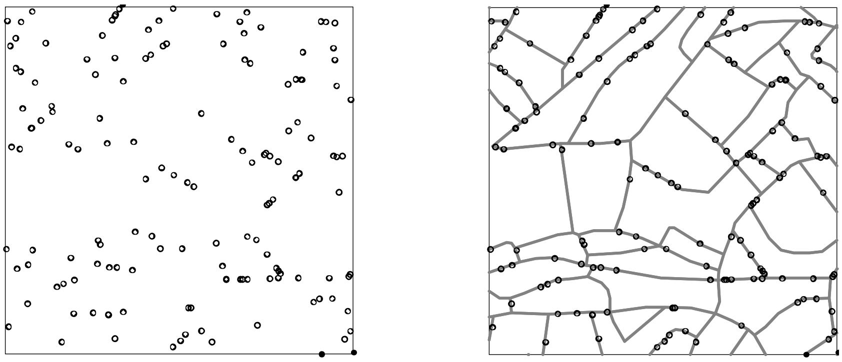
Source: Figure 1.2 in Okabe and Sugihara (2012). Sites of traffic accidents around Chiba station, Japan (private roads are not shown).

Source: Figure 1.3 in Okabe and Sugihara (2012). The distribution of advertisement agency sites (the black points) alongside streets (the gray line segments) in Shibuya ward, one of the sub-central districts in Tokyo.

Traditionally, network events are analyzed with spatial methods assuming Euclidean distance on a plane, referred to as *planar spatial analysis*. However, this assumption is difficult to accept in practice when analyzing network events, in particular, in urbanized areas, because Euclidean distances and their corresponding shortest-path distances are significantly different. As a matter of fact, an empirical examination shows that the difference is more than 20% when Euclidean distances are less than 400 meters.

Alternatively, network spatial analysis assumes the shortest-path distance on networks. This analysis potentially enables more practical investigation of network events than planar spatial analysis, but it requires heavy geometrical and topological computations. This difficulty hindered its development. To overcome this difficulty, SANET, a GIS toolbox has been developed. Using this toolbox, application-oriented GIS scientists, who are not always skilled in programming, can now easily perform network spatial analysis with detailed data (not spatially aggregated data, but such as objects in Figures 1.2 and 1.3).

Network spatial analysis is not only practical but also theoretically sound because it can avoid misleading statistical inference when network events are examined. A clear example is provided in Figure 1.4. Having observed the distribution of points in panel (a), nobody would consider that points are randomly distributed. This is true when a plane is assumed but this becomes false when a network is assumed. In fact, the points in panel (b) are randomly generated according to the uniform distribution over the network (the configuration of points in panel (a) and that in panel (b) is the same). This shows that planar spatial analysis is likely to lead to false conclusions when applied to network events.



(a) (b)

Source: Figure 1.4 in Okabe and Sugihara (2012). Point distributions: (a) nonrandomly distributed points on a bounded plane, (b) randomly distributed points on a network (note that the point distributions in (a) and (b) are the same).

As is noticed from the above discussion, tools in SANET are practically as well as theoretically useful for examining network events.

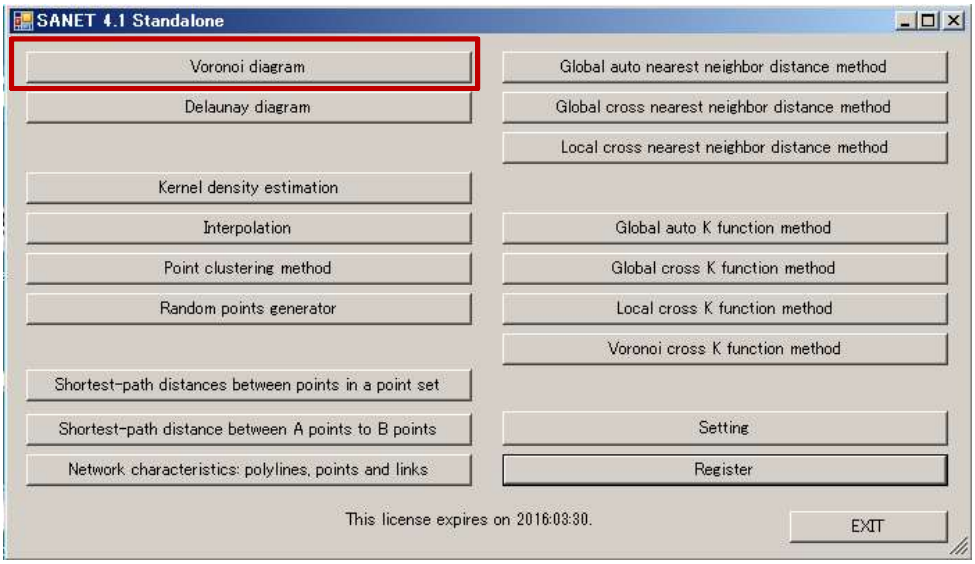
4. Analytical Tools

SANET Toolbox includes the following tools: The following sections show how to operate these tools using the sample data downloadable from the SANET website. Note that each section is self-contained; therefore, the user can directly go to any section you want to read.

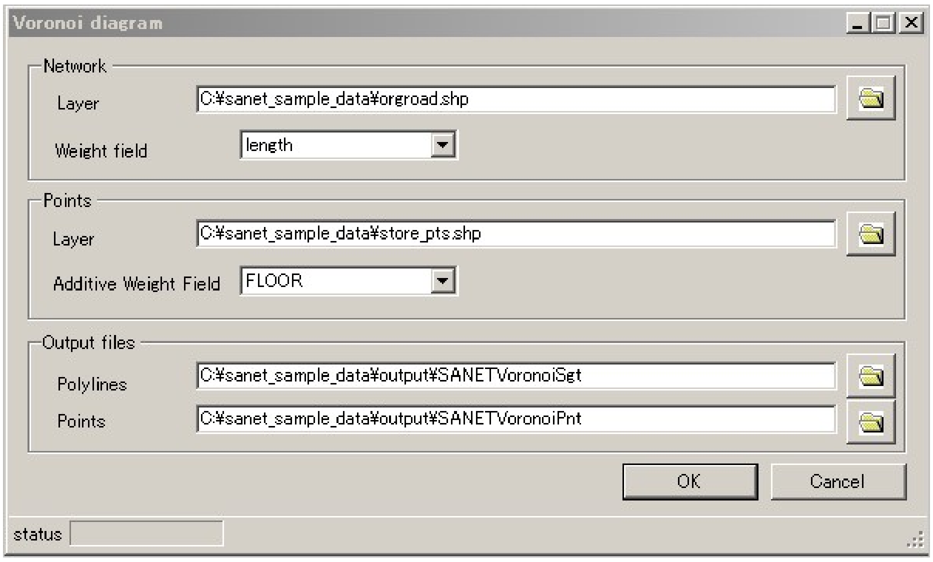
4.1 Tool 01: Voronoi diagrams

This tool generates the ordinary Voronoi diagram and the additively weighted Voronoi diagram for a given generator set of points placed on a given network. Details of this diagram are described in Chapter 4 of Okabe and Sugihara (2012).

Click the "Voronoi diagram" in the SANET menu.



Then the following window appears.



Click to choose a file of network (e.g. orgroad.shp: street network in the sample data set).

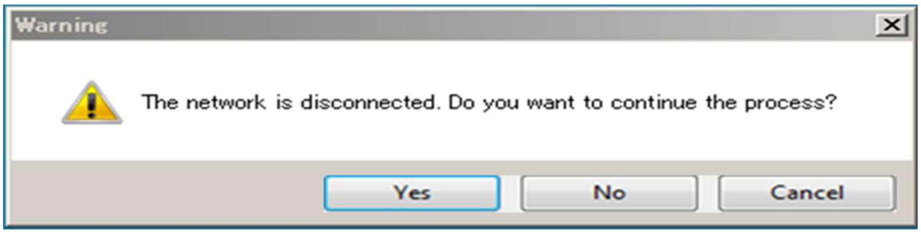
Click to choose a set of points that generates Voronoi diagrams (e.g. store_pts.shp: 3 shop locations in the sample data set).

If you use the ordinary Voronoi diagram, leave the 'Additively Weighted Field" blank. If you use the additively weighted Voronoi diagram, choose ▼ to set the field of the file of the generation point set where weights are given.

Click to store the resulting output files, and

Click "OK".

If the following window does not appear, the network is completely connected. Proceed to the next step marked by *** below.



If this window appears, the given network is not completely connected. In that case, see p.55 to find out your next step to proceed.

***

When the network is completely connected or if you click "Yes" in the warning window, the program begins to run. Having no trouble such as memory overflows, you may obtain the following files as the output.

SANETVoronoiPnt.dbf	2015/09/05 12:00	DBF ファイル	8 KB
SANETVoronoiPnt.shp	2015/09/05 12:00	SHP ファイル	2 KB
SANETVoronoiPnt.shx	2015/09/05 12:00	SHX ファイル	1 KB
SANETVoronoiSgt.dbf	2015/09/05 12:00	DBF ファイル	14 KB
SANETVoronoiSgt.shp	2015/09/05 12:00	SHP ファイル	6 KB
SANETVoronoiSgt.shx	2015/09/05 12:00	SHX ファイル	1 KB

The attribute table of the output is as follows.

SANETVoronoiSgt

FID	Shape	SgtID	FromX	FromY	FromZ	ToX	ToY	ToZ	FromPntID	ToPntID	Length	Weight	AcsID
0	Polyline ZM	0000000021 98AA0	-12294.018089	-94280.195156	0	-12318.714671	-94279.981244	0	00000000219E4D0	00000000219E2F7	24.697508	24.697508	2
1	Polyline ZM	0000000021 98A97	-12346.520667	-94279.740399	0	-12370.802803	-94307.22961	0	00000000219E4C5	00000000219E4AF	36.678043	36.678043	2
2	Polyline ZM	0000000021 98A8E	-12370.802803	-94307.22961	0	-12371.636283	-94346.125346	0	00000000219E4AF	00000000219E4A4	38.904665	38.904665	2
3	Polyline ZM	0000000021 98A85	-12371.636283	-94346.125346	0	-12339.780471	-94346.148412	0	00000000219E4A4	00000000219E302	31.855821	31.855821	2
4	Polyline ZM	0000000021 98A19	-12541.561239	-94260.272725	0	-12513.648485	-94247.687783	0	00000000219E4BA	00000000219E48E	30.618665	30.618665	1
5	Polyline ZM	0000000021 98A10	-12513.648485	-94247.687783	0	-12478.785545	-94231.969232	0	00000000219E48E	00000000219E478	38.242613	38.242613	1
6	Polyline ZM	0000000021 98A07	-12478.785545	-94231.969232	0	-12419.503497	-94228.917951	0	00000000219E478	00000000219E2EC	59.360521	59.360521	1
7	Polyline ZM	0000000021 989FE	-12380.086187	-94226.889118	0	-12364.764656	-94218.407556	0	00000000219E46D	00000000219E462	17.512459	17.512459	2
8	Polyline ZM	0000000021 989F5	-12364.764656	-94218.407556	0	-12327.302503	-94219.331954	0	00000000219E462	00000000219E457	37.473556	37.473556	2
9	Polyline ZM	0000000021 989EC	-12327.302503	-94219.331954	0	-12311.666106	-94244.151963	0	00000000219E457	00000000219E44C	29.334787	29.334787	2
10	Polyline ZM	0000000021 989E3	-12311.666106	-94244.151963	0	-12294.018089	-94280.195156	0	00000000219E44C	00000000219E4D0	40.131836	40.131836	2
11	Polyline ZM	0000000021 989DA	-12294.018089	-94280.195156	0	-12295.229049	-94346.180671	0	00000000219E4D0	00000000219E499	65.996626	65.996626	2
12	Polyline ZM	0000000021 989D1	-12295.229049	-94346.180671	0	-12296.832859	-94433.572699	0	00000000219E499	00000000219E441	87.406743	87.406743	2

Shape: polylines

SgtID: link ID (a polyline ID).

FromX, FromY, From Z: from the node (x, y, z).

ToX,ToY, ToZ: to the node (x, y, z).}.

Length: the length of a link.

AcsID: the Voronoi subnetwork ID to which a link belongs.

To visualize the result, please see section 5.1 / section 5.3.

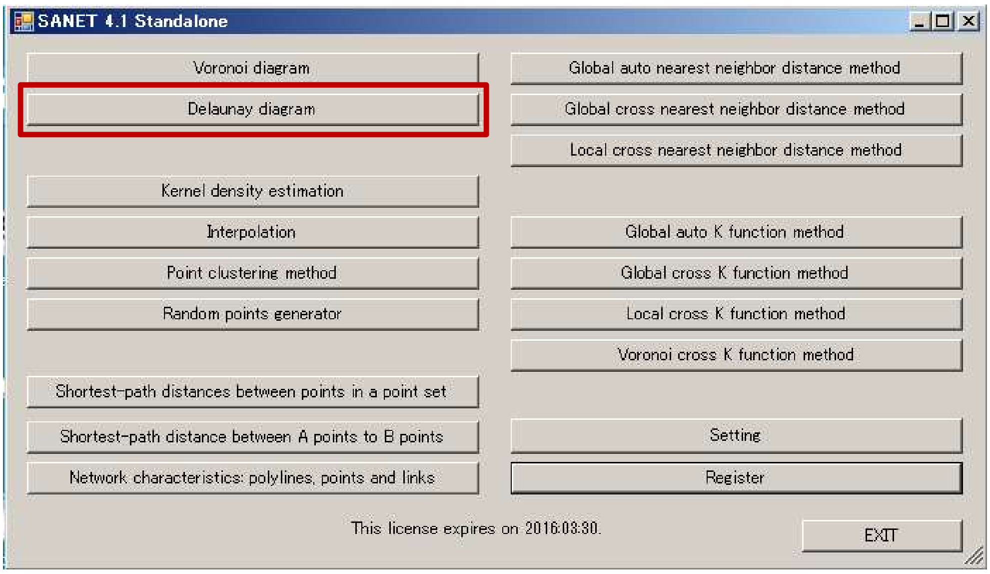
Reference

Okabe, A. and K. Sugihara (2012) *Spatial Analysis along Networks: Statistical and Computational Methods*, Chichester: John Wiley, a volume in the Wiley series of Statistics in Practice.

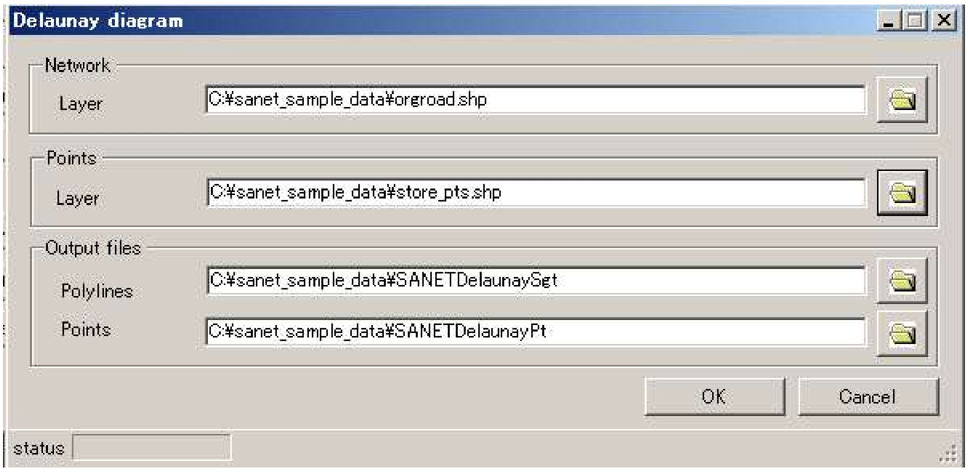
4.2 Tool 02: Delaunay diagram

Originally, a Delaunay triangulation (a diagram on a plane) is defined as the reciprocal figure of the corresponding Voronoi diagram on a plane. Stated explicitly, generate line segment between the generators whose Voronoi polygons share the same Voronoi edge. The resulting line segments form a triangular network, termed the Delaunay triangulation of the given Voronoi diagram. Delaunay diagrams on a network are similar to Delaunay triangulations on a plane in spirit. Given a Voronoi diagram on a network, generate the shortest-path between two generators whose Voronoi sub-networks are adjacent. The resulting shortest-paths form a Delaunay diagram on a network.

Click the "Delaunay diagram" in the SANET menu.



Then the following window appears.

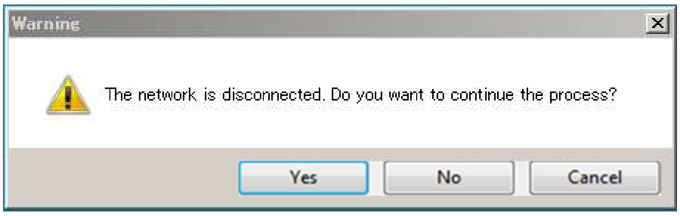


Click to choose a file of network (e.g. orgroad.shp: street network in the sample data set).

Click to choose a set of points that generates Delaunay diagrams (e.g. store_pts.shp: 3 shop locations in the sample data set. Ignore Weight field.).

Click to store the resulting output files, and click “OK”.

If the following window does not appear, the network is completely connected. Proceed to the next step marked by *** below.



If this window appears, the given network is not completely connected. In that case, see p.55 to find out your next step to proceed.

***

When the network is completely connected or if you click “Yes” in the warning window, the program begins to run. Having no trouble such as memory overflows,

SANETDelaunayPt.dbf	2015/09/06 17:30	DBF ファイル	1 KB
SANETDelaunayPt.shp	2015/09/06 17:30	SHP ファイル	1 KB
SANETDelaunayPt.shx	2015/09/06 17:30	SHX ファイル	1 KB
SANETDelaunaySgt.dbf	2015/09/06 17:30	DBF ファイル	8 KB
SANETDelaunaySgt.shp	2015/09/06 17:30	SHP ファイル	3 KB
SANETDelaunaySgt.shx	2015/09/06 17:30	SHX ファイル	1 KB

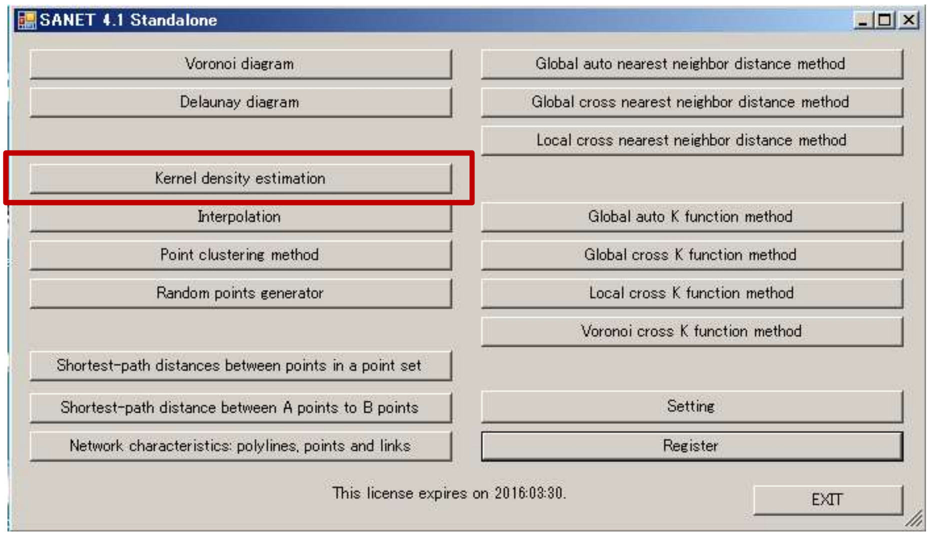
you may obtain the following files as the output.

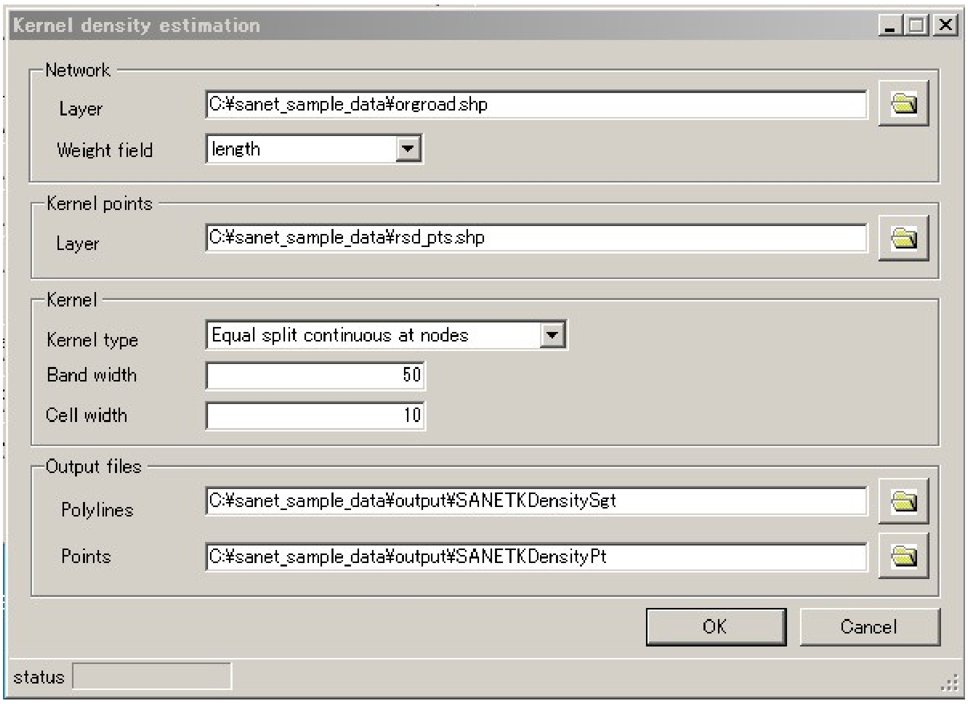
To visualize the result, please refer section 5.1 / section 5.3.

4.3 Tool 03: Kernel density estimation

For a given set of points on a given network, this tool estimates the density of points on the network. For details, see Chapter 9 and Section 12.2.5 in Okabe and Sugihara (2012).

Click “Kernel density estimation”. Then the following window appears.





Choose the file of a network (e.g., orgroad.shp: street network in the sample data set.).

Choose the file of a set of points (e.g., rsd_pts.shp: location of points in the sample data set.).

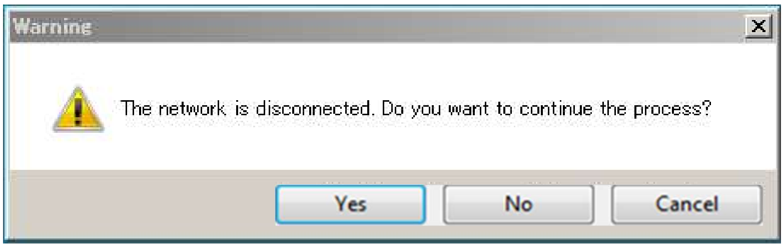
Choose ▼ one of the two estimation methods: “equal split continuous at nodes” or “equal split discontinuous at nodes”.

Fill in a band width and a cell width. 50 and 10 are put respectively in this example. Note that if you use a large band width and a small cell width, computation time becomes long. Try to use a fairly small band width and a large cell size satisfying that the former is larger than the latter. If the computation time is within your time allowance, change those values. Our experience says [band size] = 10*[cell size]. We also note that you are supposed to use the same grid coordinates system as that of the network.

Choose to store the resulting output files, and

Click “OK”.

If the following window does not appear, the network is completely connected. Proceed to the next step marked by *** below.



If this window appears, the given network is not completely connected. In that case, see p.55 to find out your next step to proceed.

***

When the network is completely connected or if you click “Yes” in the warning window, the program begins to run. If you do not have any trouble such as memory overflows, you may obtain the following files as the output.

SANETKDensityPt.dbf	2015/09/06 21:50	DBF ファイル	38 KB
SANETKDensityPt.shp	2015/09/06 21:50	SHP ファイル	9 KB
SANETKDensityPt.shx	2015/09/06 21:50	SHX ファイル	2 KB
SANETKDensitySgt.dbf	2015/09/06 21:50	DBF ファイル	74 KB
SANETKDensitySgt.shp	2015/09/06 21:50	SHP ファイル	28 KB
SANETKDensitySgt.shx	2015/09/06 21:50	SHX ファイル	2 KB

To visualize the result, please see section 5.1 / section 5.3.

Reference

Okabe, A. and K. Sugihara (2012) *Spatial Analysis Along Networks: Statistical and Computational Methods*, Chichester: John Wiley, a volume in the Wiley series of Statistics in Practice.

4.4 Tool 04: Interpolation

For a given set of known attributes values at sample points on a given network, this tool interpolates unknown attribute values at arbitrary points on the network using the inverse distance weighting method. Stated explicitly, the tool predicts the value $\hat{z}_0$ at p_0 as the weighted average of the known attribute z_j values at the points p_j of a neighborhood $P_{\mathrm{N}}(p_0)$ of p_0, i.e.:

$$\hat{z}_0 = \sum_{p_i \in P_{\mathrm{N}}(p_0)} w_i z_i ,$$

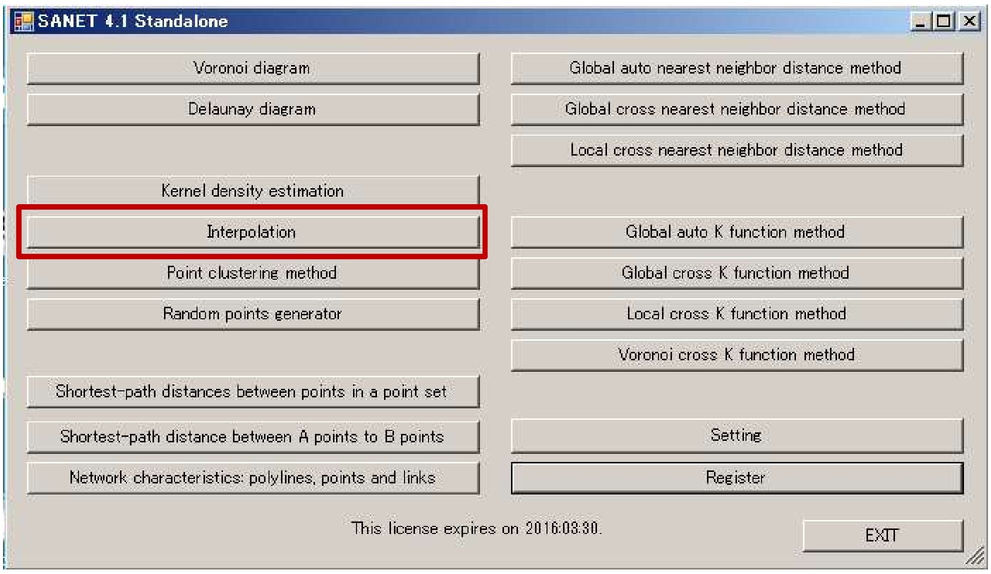
where the weight w_i is given by:

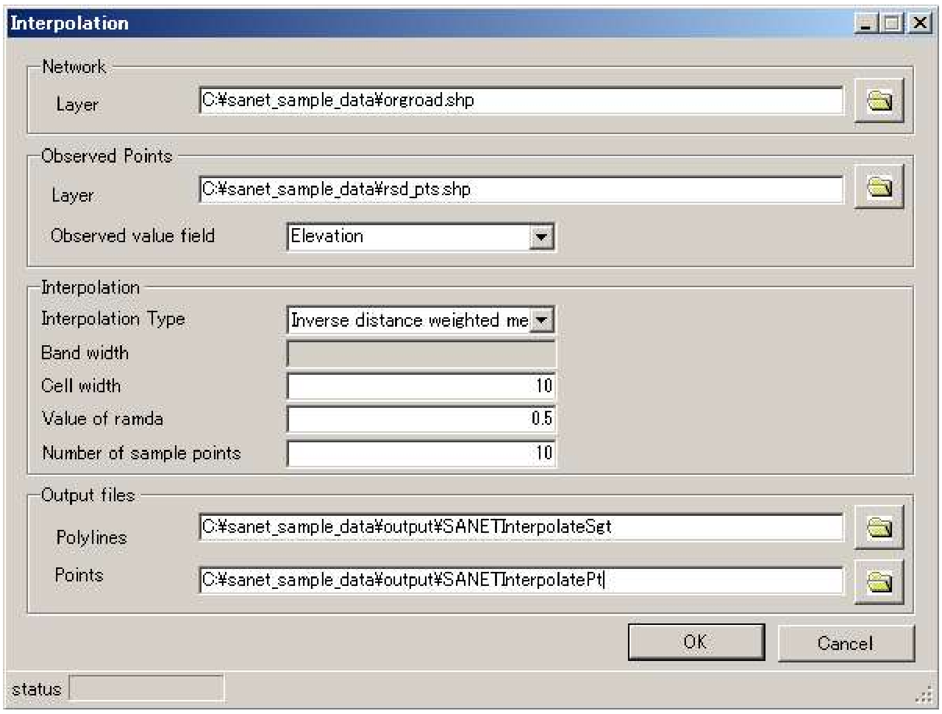
$$w_i = \frac{d_{\mathrm{S}}(p_0, p_i)^{-\alpha}}{\sum_{p_j \in P_{\mathrm{N}}} d_{\mathrm{S}}(p_0, p_j)^{-\alpha}} ,$$

where α is a positive predetermined parameter.

For details, see Chapter 9 and Section 12.2.5 in Okabe and Sugihara (2012).

Click “Interpolation”. Then the following window appears.





Choose  the file of a network (e.g., orgroad.shp: street network in the sample data set.).

Choose  the file a set of points at which their attribute values are known (e.g., location points in the sample data set.).

Choose ▼ the field in which weights are given.

Choose ▼ “Inverse distance weighting method”.

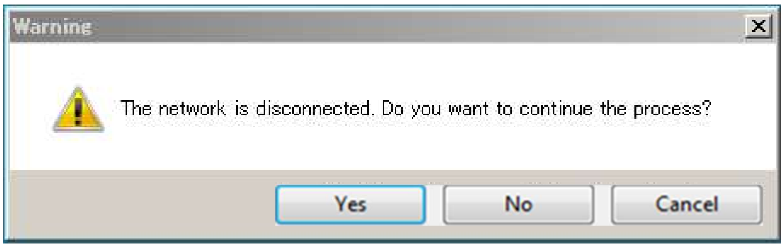
Fill in

- cell width (This determines the resolution of the resulting figure; default is 10, but might require much memory; you may start a larger value, say 10).
- value of α , say 0.5.
- number of points in the neighborhood $P_{\mathrm{N}}(p_0)$, say 10.

Choose  to store the resulting output files, and

Click “OK” .

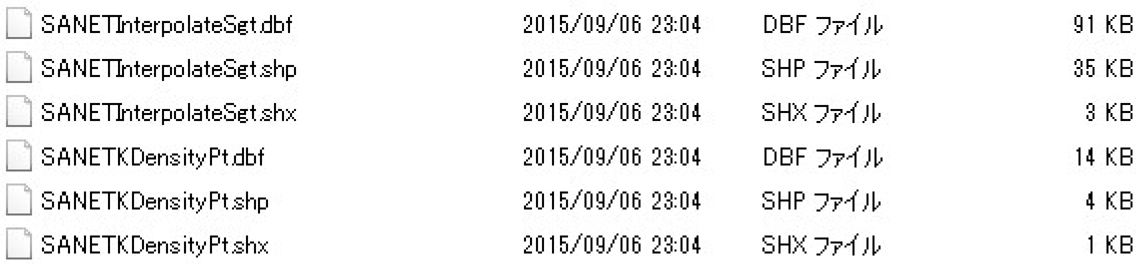
If the following window does not appear, the network is completely connected. Proceed to the next step marked by *** below.



If this window appears, the given network is not completely connected. In that case, see p.55 to find out your next step to proceed.

***

When the network is completely connected or if you click “Yes” in the warning window, the program begins to run. If you do not have any trouble such as memory overflows, you may obtain the following six files as the output.



SANETInterpolateSgt.dbf	2015/09/06 23:04	DBF ファイル	91 KB
SANETInterpolateSgt.shp	2015/09/06 23:04	SHP ファイル	35 KB
SANETInterpolateSgt.shx	2015/09/06 23:04	SHX ファイル	3 KB
SANETKDensityPt.dbf	2015/09/06 23:04	DBF ファイル	14 KB
SANETKDensityPt.shp	2015/09/06 23:04	SHP ファイル	4 KB
SANETKDensityPt.shx	2015/09/06 23:04	SHX ファイル	1 KB

To visualize the result, please see section 5.1 / section 5.3.

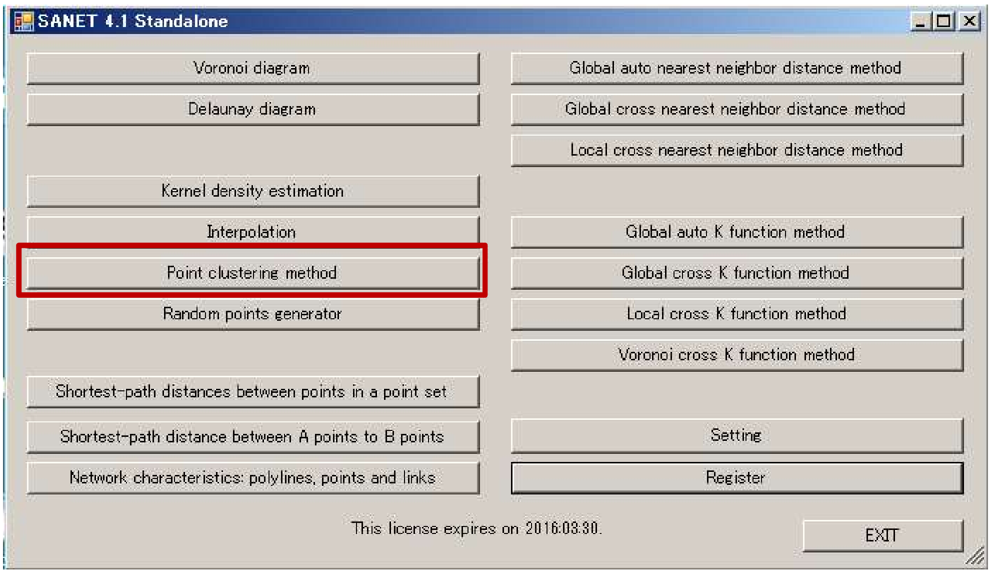
Reference

Okabe, A. and K. Sugihara (2012) *Spatial Analysis Along Networks: Statistical and Computational Methods*, Chichester: John Wiley, a volume in the Wiley series of Statistics in Practice.

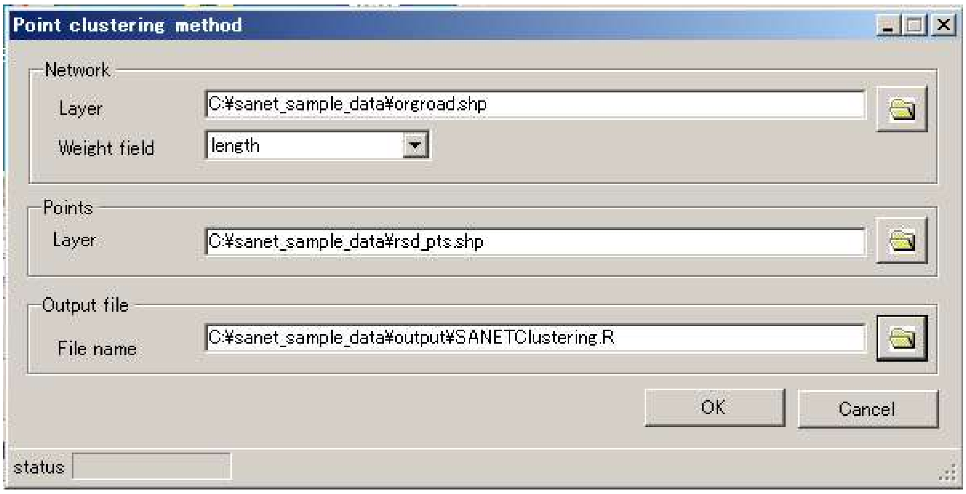
4.5 Tool 05: Point clustering method

This tool clusters points in a given point set on a given network by the closest-pair clustering method. For details, see Sections 8.1.2.1 and 12.2.4 in Okabe and Sugihara (2012).

Click the "Point clustering method" in the SANET menu.



Then the following window appears.



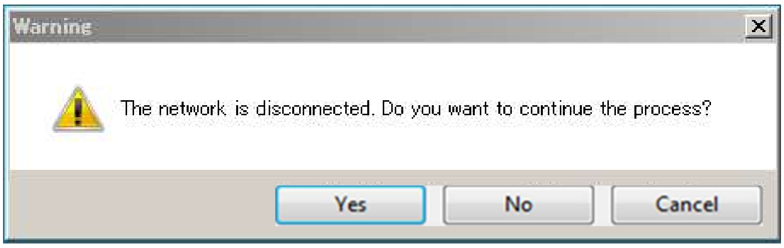
Choose  the file name of a network (e.g., orgroad.shp: street network in the sample data set).

Choose  the file name of a set of points (e.g., location points in the sample data set.).

Choose  to store the resulting output files, and

Click “OK” .

If the following window does not appear, the network is completely connected. Proceed to the next step marked by *** below.



If this window appears, the given network is not completely connected. In that case, see p.55 to find out your next step to proceed.

***

When the network is completely connected or if you click “Yes” in the warning window, the program begins to run. If you do not have any trouble such as memory overflows, you may obtain the following “SANETClustering.R” file as the output.

To visualize the result, please see section 5.2.

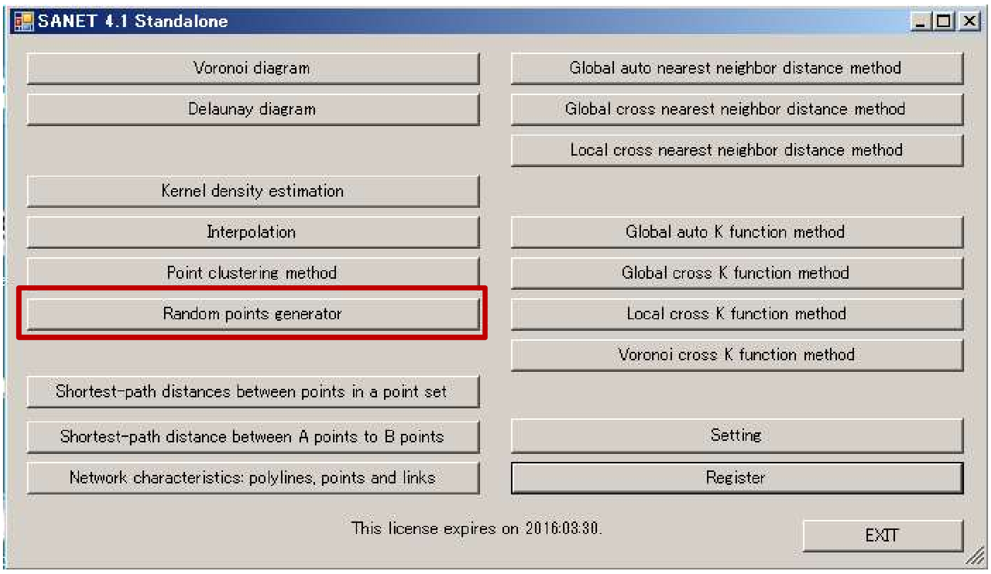
Reference

Okabe, A. and K. Sugihara (2012) *Spatial Analysis Along Networks: Statistical and Computational Methods*, Chichester: John Wiley, a volume in the Wiley series of Statistics in Practice.

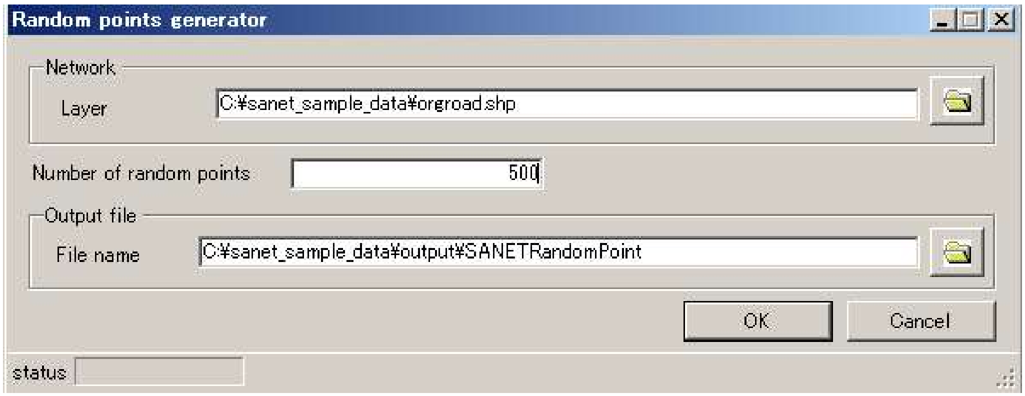
4.6 Tool 06: Random points generator

This tool generates random points on a given network according to the uniform distribution over the network. The resulting points are a realization of the complete spatial randomness (CSR) often used for a null-hypothesis. For details, see Sections 2.4.2 3.4.5 and 12.1.4 in Okabe and Sugihara (2012).

Click the “Random points generator” in the SANET menu.



Then the following window appears.



Choose  the file name of a network (e.g., orgroad.shp: street network in the sample data set).

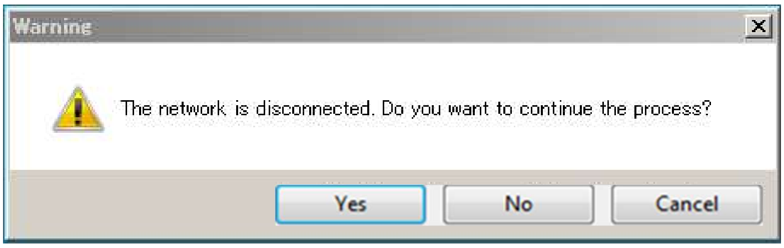
Fill in a number of random points, say 200.

Choose  to store the resulting output files, and

Click “OK”.

If the following window does not appear, the network is completely connected.

Proceed to the next step marked by *** below.



If this window appears, the given network is not completely connected. In that case, see p.55 to find out your next step to proceed.

***

When the network is completely connected or if you click “Yes” in the warning window, the program begins to run. If you do not have any trouble such as memory overflows, you obtain the following files in the output file.

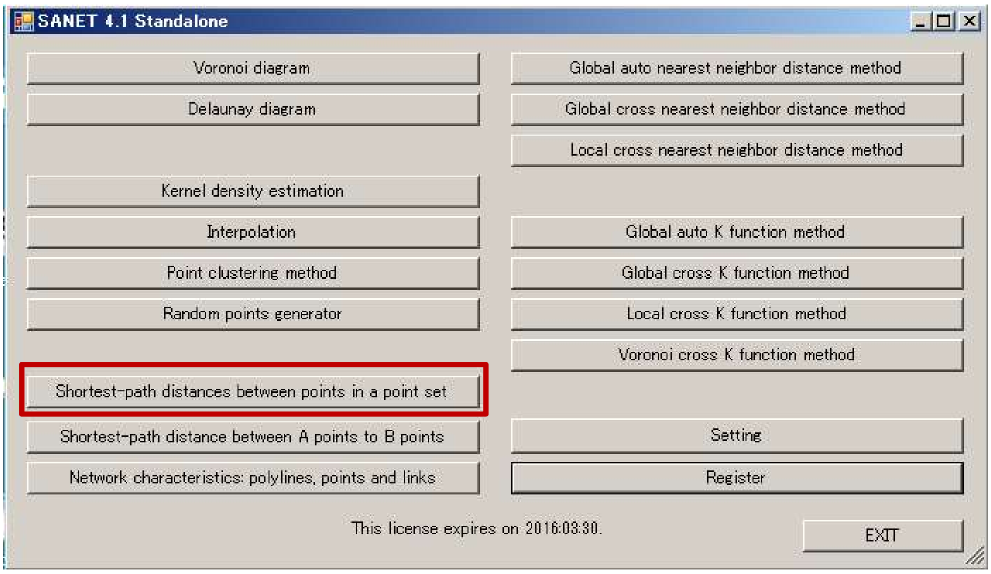
To visualize the result, please see section 5.1 / section 5.3.

Reference

Okabe, A. and K. Sugihara (2012) *Spatial Analysis Along Networks: Statistical and Computational Methods*, Chichester: JonWiley, a volume in the Wiley series of Statistics in Practice.

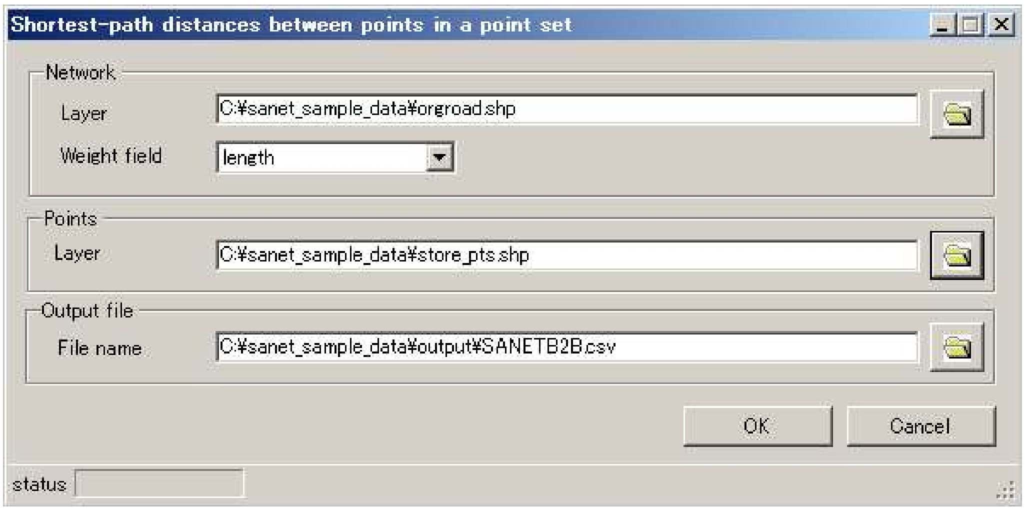
4.7 Tool 07: Shortest-path distances between points in a point set

This tool computes the shortest-path distances between any pair of points in a given set of points placed on a given network. For details, see Section 12.1.3 in Okabe and Sugihara (2012).



Click the “Shortest-path distances between points in a point set” in the SANET menu.

Then the following window appears.



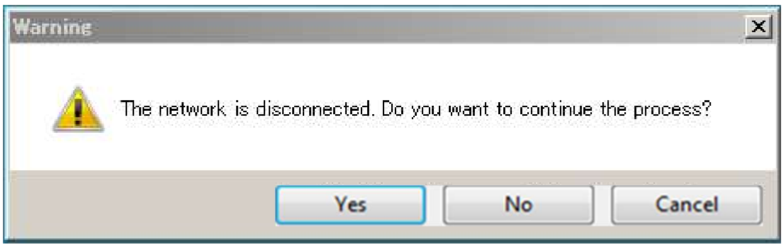
Choose  the file name of a network (e.g., orgroad.shp: street network in the sample data set).

Choose  the file name of a set of points (e.g., store_pts.shp: store location points in the sample data set)

Choose  to store the resulting output file, and

Click “OK”.

If the following window does not appear, the network is completely connected. Proceed to the next step marked by *** below.



If this window appears, the given network is not completely connected. In that case, see p.55 to find out your next step to proceed.

***

When the network is completely connected or if you click “Yes” in the warning window, the program begins to run. If you do not have any trouble such as memory overflows, you obtain the following file.

The contents of the file are as follows.

	1	2	3
1	FromPntID	ToPntID	Distance
2	0	1	313.0112
3	0	2	342.2287
4	1	0	313.0112
5	1	2	342.3649
6	2	0	342.2287
7	2	1	342.3649
8	AVERAGE	332.535	

The first column indicates “from the *i*-th point of the point set”.
The second column indicates “to the *j*-th point of the point set”.
The last column indicates the shortest-path distance between those points. For instance, the shortest-path distance from the 0th point to the first point is 313.0112 and so forth.
AVERAGE of the last row shows the average of all the distances.

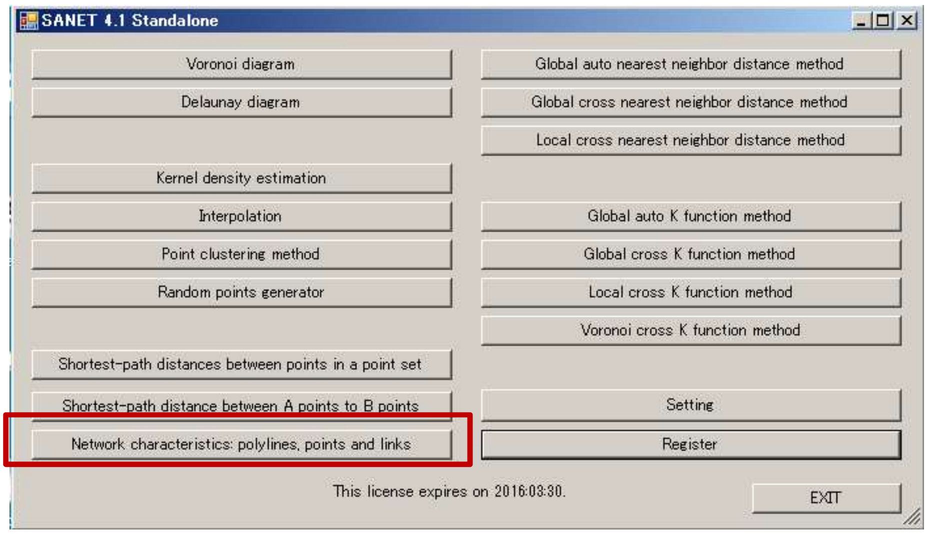
Reference

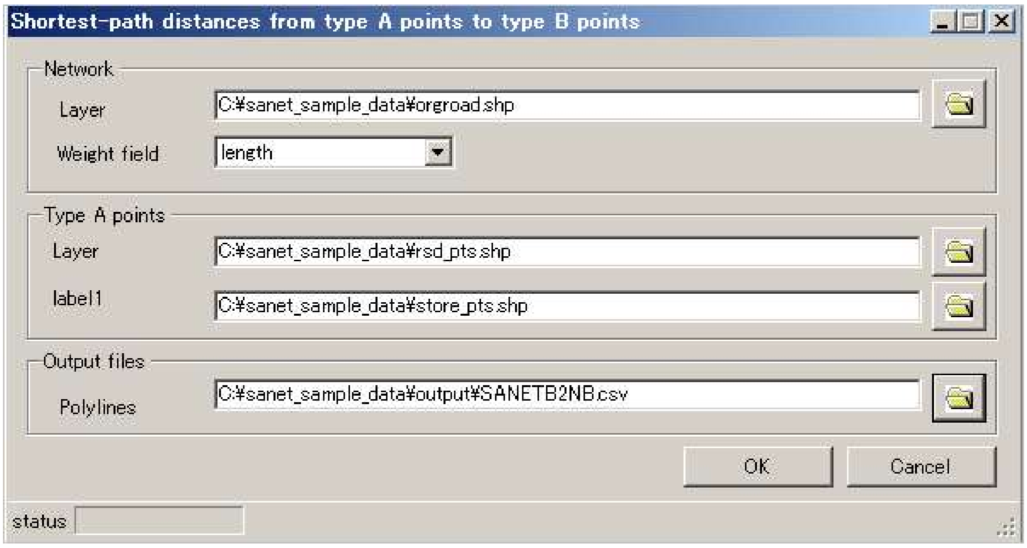
Okabe, A. and K. Sugihara (2012) *Spatial Analysis Along Networks: Statistical and Computational Methods*, Chichester: John Wiley, a volume in the Wiley series of Statistics in Practice.

4.8 Tool 08: Shortest path distances from type A points to type B points

This tool computes the shortest-path distance from each point in a given set of type A points to each point in a given set of type B points, where those points are on a given network. For details, see Section 12.1.3 in Okabe and Sugihara (2012).

Click the “Shortest-path distances between type A points and type B points” in the SANET menu. Then the following window appears.





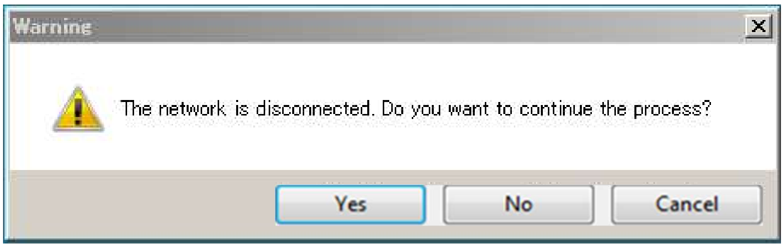
Choose  the file name of a network (e.g., orgroad.shp: street network in the sample data set).

Choose  the file name of a set of type A points (e.g., rsd_pts.shp: location points in the sample data set).

Choose  the file name of a set of type B points (e.g., store_pts.shp: store location points in the sample data set).

Choose  to store the resulting output file, and click “OK”.

If the following window does not appear, the network is completely connected. Proceed to the next step marked by *** below.

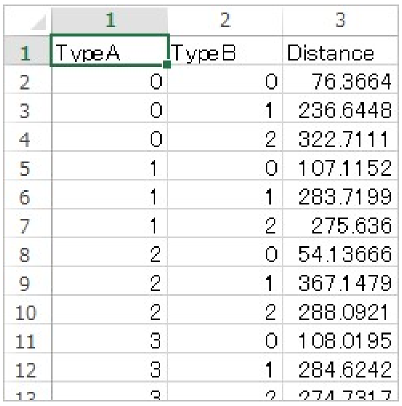


If this window appears, the given network is not completely connected. In that case, see p.55 to find out your next step to proceed.

***

When the network is completely connected or if you click “Yes” in the warning window, the program begins to run. If you do not have any trouble such as memory overflows, you obtain the following file.

The contests are as follows.



	1	2	3
1	TypeA	TypeB	Distance
2	0	0	76.3664
3	0	1	236.6448
4	0	2	322.7111
5	1	0	107.1152
6	1	1	283.7199
7	1	2	275.636
8	2	0	54.13666
9	2	1	367.1479
10	2	2	288.0921
11	3	0	108.0195
12	3	1	284.6242

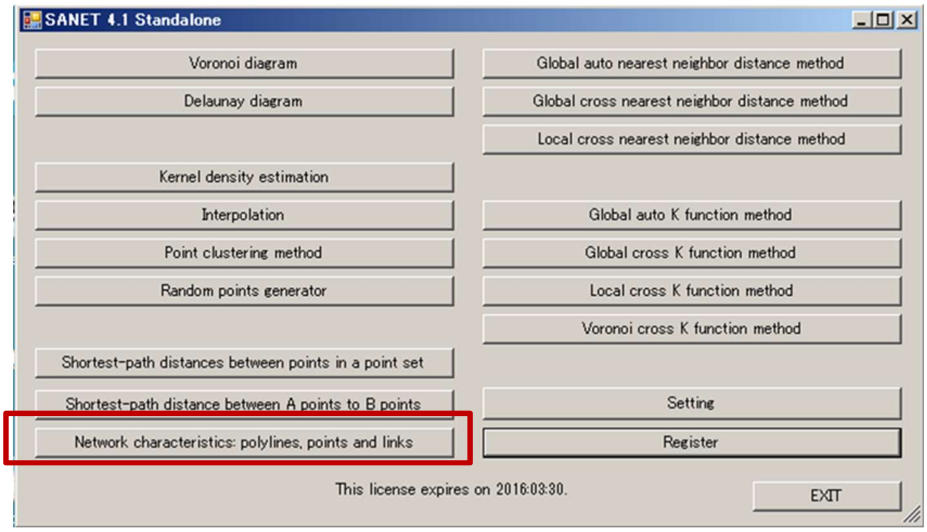
The first column indicates "from the *i*-th point of the type A point set.
The second column indicates "to the *j*-th point of the type B point set.
The last column indicates the shortest-path distance between those points. For instance, the shortest-path distance from the 0th point of the type A to the 0th point of the Type B is 76.3664.

Reference

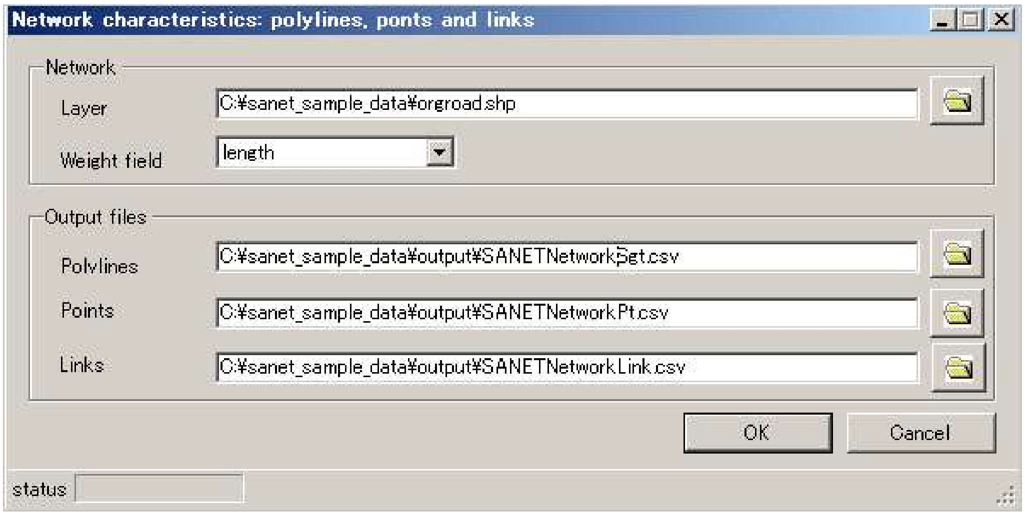
Okabe, A. and K. Sugihara (2012) *Spatial Analysis Along Networks: Statistical and Computational Methods*, Chichester: John Wiley, a volume in the Wiley series of Statistics in Practice.

4.9 Tool 09: Network Characteristics: polylines, points and links

This tool shows the characteristics of polylines, points and links forming a given network.



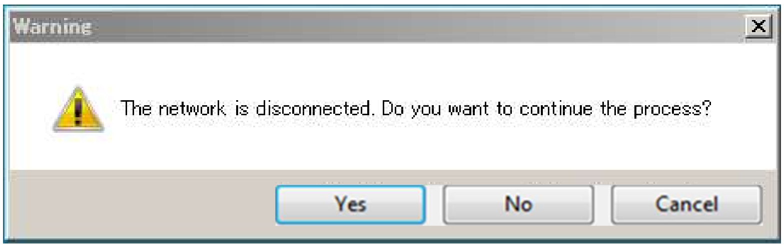
Click the “Network characteristics: polylines, points and links” in the SANET menu. Then the following window appears.



Choose  the file name of a network (e.g., orgroad.shp: street network in the sample data set).

Choose  to store the resulting output files, and click “OK”.

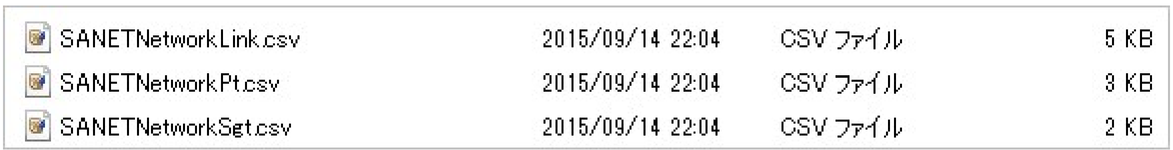
If the following window does not appear, the network is completely connected. Proceed to the next step marked by *** below.



If this window appears, the given network is not completely connected. In that case, see p.55 to find out your next step to proceed.

***

When the network is completely connected or if you click “Yes” in the warning window, the program begins to run. If you do not have any trouble such as memory overflows, you obtain the following three files.



- SANETNetworkLink.csv

	1	2	3	4
1	PntID	SgtID	AdjacentPntID	FromToFlg
2	000000002044FAB0	0000000020DE73D0	000000002044FA00	From
3	000000002044FAB0	0000000020DE6800	000000002044F270	To
4	000000002044FAB0	0000000020DE74F0	000000002044F740	From
5	000000002044FA00	0000000020DE73D0	000000002044FAB0	To
6	000000002044FA00	0000000020DE7340	000000002044F8A0	From
7	000000002044F8A0	0000000020DE7340	000000002044FA00	To
8	000000002044F8A0	0000000020DE72B0	000000002044F7F0	From
9	000000002044F7F0	0000000020DE72B0	000000002044F8A0	To
10	000000002044F7F0	0000000020DE7220	000000002044F740	From
11	000000002044F740	0000000020DE7220	000000002044F7F0	To

This table shows that:

SgtID: line segment ID;

PntID and AdjacentPntID are end nodes of a line segment;

FromToFlg:

From: from PntID to AdjacentPntID;

To: from AdjacentPntID to PntID.

- SANETNetworkPnt.csv

	1	2	3	4
1	PntID	X	Y	Z
2	000000002044FAB0	-12294.01809	-94280.19516	0
3	000000002044FA00	-12346.52067	-94279.7404	0
4	000000002044F8A0	-12370.8028	-94307.22961	0
5	000000002044F7F0	-12371.63628	-94346.12535	0
6	000000002044F740	-12295.22905	-94346.18067	0
7	000000002044F950	-12541.56124	-94260.27273	0
8	000000002044F690	-12513.64849	-94247.68778	0
9	000000002044F530	-12478.78555	-94231.96923	0
10	000000002044F480	-12380.08619	-94226.88912	0
11	000000002044F3D0	-12364.76466	-94218.40756	0

This table shows that the (x, y, z) coordinates of a point specified by PntId.

- SANETNetworkSgt.csv

	1	2	3	4
1	SgtID	FID	Length	Weight
2	0000000020DE73D0	0	52.504547	52.504547
3	0000000020DE7340	0	36.678043	36.678043
4	0000000020DE72B0	0	38.904665	38.904665
5	0000000020DE7220	0	76.407254	76.407254
6	0000000020DE6B60	1	30.618665	30.618665
7	0000000020DE6AD0	1	38.242613	38.242613
8	0000000020DE6A40	1	98.83001	98.83001
9	0000000020DE69B0	1	17.512459	17.512459
10	0000000020DE6920	1	37.473556	37.473556
11	0000000020DE6890	1	29.334787	29.334787

This table shows that the length of a link specified by SgtID.

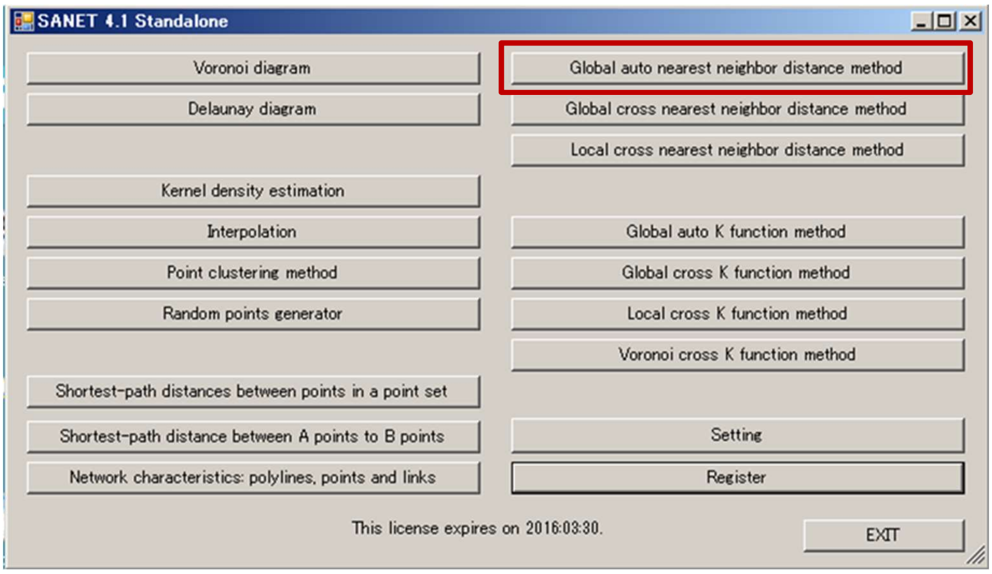
Reference

Okabe, A. and K. Sugihara (2012) *Spatial Analysis along Networks: Statistical and Computational Methods*, Chichester: John Wiley, a volume in the Wiley series of Statistics in Practice.

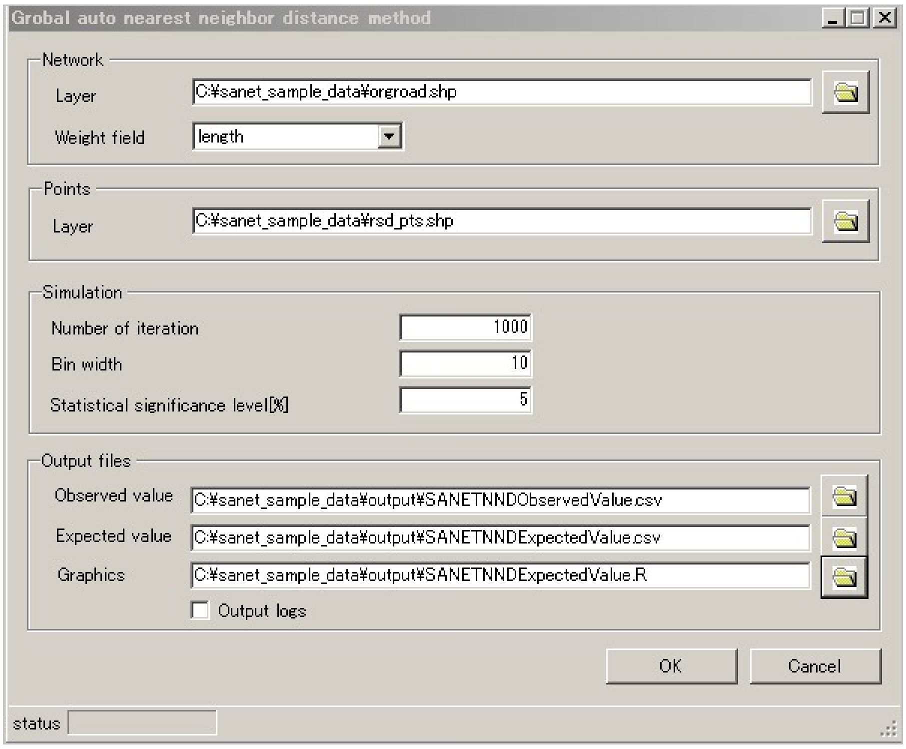
4.10 Tool 10: Global auto nearest neighbor distance method

This tool tests the complete spatial randomness (CSR) hypothesis in terms of the shortest-path distance from every point in a given set of points placed on a given bounded network to its next nearest point in the set. Note that in the literature, the *global auto nearest neighbor distance method* is simply referred to as the *nearest neighborhood distance method*. The CSR hypothesis means that points are independently and identically distributed according to the uniform distribution over the network, or points follow the homogeneous binomial point process on the bounded network. A general description about the nearest neighbor distance method is provided in Chapter 5 in Okabe and Sugihara (2012); specifically, the global auto nearest neighbor distance method is shown in Section 5.1.2 and its application in Section 12.2.2.1.

Click the “Global auto nearest neighbor distance method” in the SANET menu.



Then the following window appears.



Choose  the file name of a network (e.g., orgroad.shp: street network in the sample data set).

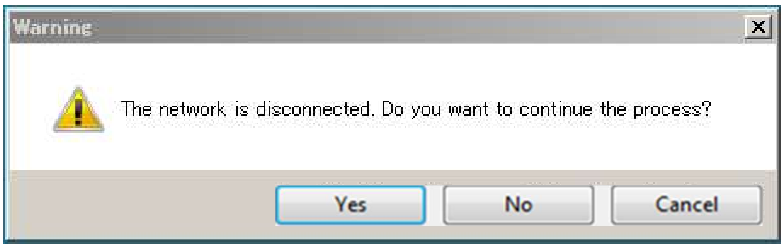
Choose  the file name of a set of points (e.g., rsd_pts.shp: location points in the sample data set).

Fill in:

- a number of iterations for Monte Carlo simulation (a default values is 1000),
- a bin width (a continuous distance is divided by the equal bin width; a default value is 10; in this case, the resulting intervals are 0-10, 10-20, 20-30,…..) ,
- one-sided statistical significance level (the default value is 5%).

Choose  to store the resulting output files. If you want to have intermediate data, check “Output logs”, which may require much memory. Click “OK”.

If the following window does not appear, the network is completely connected. Proceed to the next step marked by *** below.



If this window appears, the given network is not completely connected. In that case, see p.55 to find out your next step to proceed.

***

When the network is completely connected or if you click “Yes” in the warning window, the program begins to run. If you do not have any trouble such as memory overflows, you obtain the following three files.

SANETNNDExpectedValue.csv	2015/09/15 10:50	CSV ファイル	1 KB
SANETNNDExpectedValue.R	2015/09/15 10:50	R ファイル	2 KB
SANETNNDObservedValue.csv	2015/09/15 10:50	CSV ファイル	1 KB

The contents of the resulting files are as follows.

- SANETNNDObservedValue.csv

	1	2	3
1	FromPntID	ToPntID	Distance
2	0	5	15.528911
3	1	3	0.904277
4	2	8	9.905195
5	3	1	0.904277
6	4	7	16.369132
7	5	0	15.528911
8	6	28	15.829072
9	7	1	12.545751
10	8	2	9.905195
11	9	3	34.552302
-	-	-	-
41	39	8	31.504595
42	40	44	21.362772
43	41	26	66.644104
44	42	45	36.088645
45	43	45	39.115285
46	44	40	21.362772
47	45	42	36.088645
48	46	42	44.857939
49	47	34	44.755411
50	AVERAGE	23.701932	

The first column indicates “from the *i*-th point”.

The second column indicates “to its nearest neighbor point”. Note that the same ID points mean different points are placed at the same location.

The last row “AVERAGE” indicates the average nearest neighbor distance. Note that the same FromPtId implies that those points are placed at the same location.

- SANETNNDExpectedValue.csv

	1	2
1		
2	Lower	13.419826
3	Upper	19.66787
4	ALL AVERAGE	16.523159
5	VARIANCE	232.809281

The first column indicates the *i*-th iteration of Monte Carlo simulation.
The second column indicates "from the *j*-th point" in a given point set.
Third column indicates "to its nearest neighbor point".
The last column indicates the shortest-path distance between them.
The last row of the last (1000) iteration (AVERAGE) indicates the average of the nearest neighbor distance for the *j*-th iteration.
At the end of this file, "Lower" indicates the lower critical value for the one-sided significance level is 5 %, while "Upper" indicates the upper critical value for the one-sided significance level is 5 %
"ALL AVERAGE" indicates the average nearest neighbor distances for 1000 iterations.

The Clark-Evans index is given by AVERAGE in the table of SANETObservedValue.csv divided by ALL AVERAGE in the table of SANETExpectedValue.csv. In the above example, the value of the index is 23.70/16.52=1.43.

- SANETNNDExpectedValue.R.

To visualize SANETNNDExpectedValue.R, please see section 5.2.

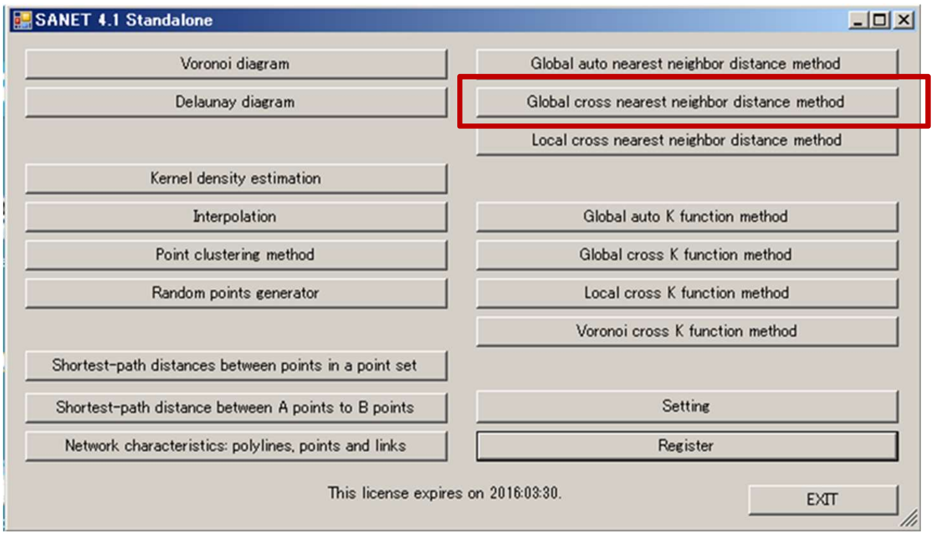
Reference

Okabe, A. and K. Sugihara (2012) *Spatial Analysis Along Networks: Statistical and Computational Methods*, Chichester: John Wiley, a volume in the Wiley series of Statistics in Practice.

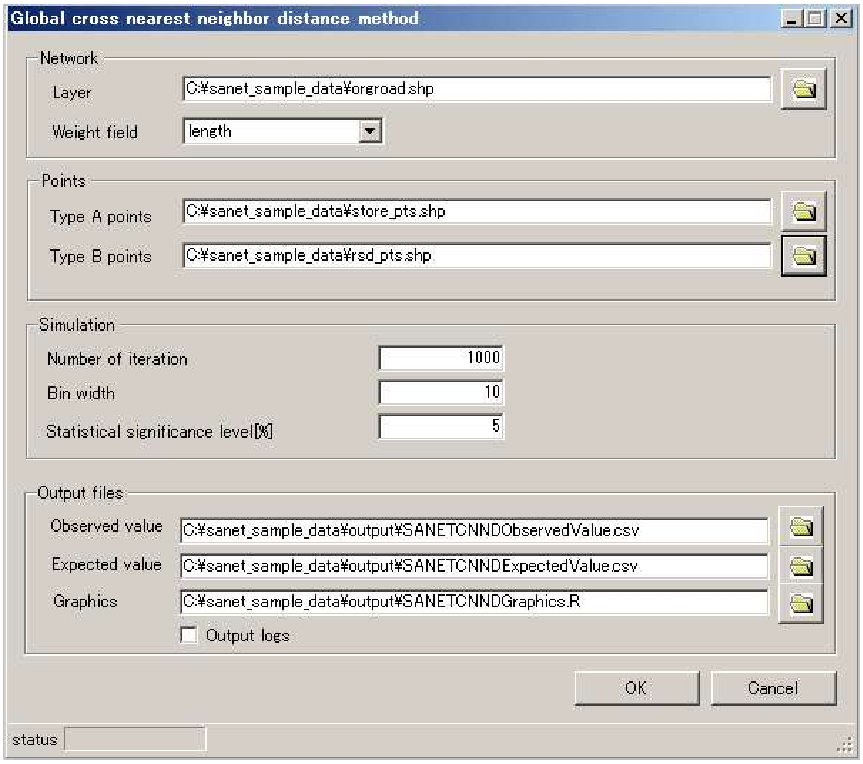
4.11 Tool 11: Global cross nearest neighbor distance method

This tool tests the complete spatial randomness (CSR) hypothesis in terms of the shortest-path distance from each point in a given set of type B points to its nearest point in a given set of type A points. In the literature, the *global cross nearest neighbor distance method* is sometimes referred to as the *conditional nearest neighborhood distance method*. In general, it is assumed that type B points are temporal, while type A points are stable over time; for instance, the former points are restaurants and the latter points are railway stations. This method tests the effect of railway stations on the distribution of restaurants in comparison with the CSR hypothesis. The CSR hypothesis means that points are independently and identically distributed according to the uniform distribution over the network, or points follow the homogeneous binomial point process on the bounded network. A general description about the nearest neighbor distance method is provided in Chapter 5 in Okabe and Sugihara (2012); specifically, the global cross nearest neighbor distance method is shown in Section 5.2.2 and its application in Section 12.2.2.2.

Click the "Global cross nearest neighbor distance method" in the SANET menu.



Then the following window appears.

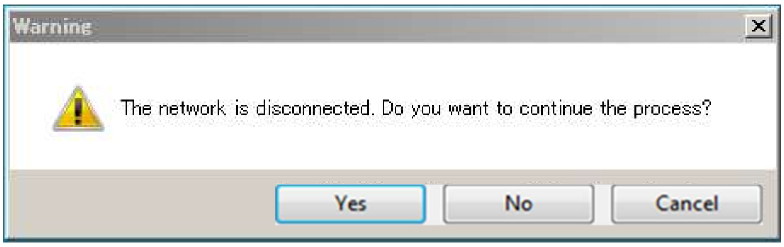


Choose the file name of a network (e.g., orgroad.shp: street network in the sample data set).

Choose the file of type A points (e.g., store_pts.shp: store location points in the sample data set); and the file of type B points (e.g., rsd_pts.shp: location points in the sample data set).
Fill in:

- a number of iterations for Monte Carlo simulation (a default values is 1000),
- a bin width (a continuous distance is divided by the equal bin width; a default value is 10; in this case, the resulting intervals are 0-10, 10-20, 20-30,…..) ,
- a statistical significance level (the default value is 5%; one-sided).

Choose to store the output resulting files, and click OK.
If the following window does not appear, the network is completely connected. Proceed to the next step marked by *** below.



If this window appears, the given network is not completely connected. In that case, see p.55 to find out your next step to proceed.
***
When the network is completely connected or if you click “Yes” in the warning window, the program begins to run. If the memory is enough, the following three files are obtained as the output.

SANETCNNDExpectedValue.csv	2015/09/15 11:50	CSV ファイル	25 KB
SANETCNNDGraphics.R	2015/09/15 11:50	R ファイル	5 KB
SANETCNNDObservedValue.csv	2015/09/15 11:50	CSV ファイル	1 KB

The contents of the resulting files are as follows.

- SANETCNNDObservedValue.csv

	1	2	3
1	TypeB	TypeA	Distance
2	0	0	76.3664
3	1	0	107.1152
4	2	0	54.13666
5	3	0	108.0195
6	4	0	78.20031

- - - - -

46	44	2	187.2046
47	45	2	115.1871
48	46	2	74.3575
49	47	2	17.92685
50	AVERAGE	119.5153	
51	VARIANCE	4801.486	

The first column indicates “from the *i*-th point in the type B point set”.
The second column indicates “to its nearest neighbor point in the type A point set”.
The third column indicates the shortest-path distance between those points.
The last row “AVERAGE” indicates the average nearest neighbor distance from all type B points to their nearest type A points.

- SANETCNNDExpectedValue.csv

	1	2
1	AVERAGE 0	109.2403
2	AVERAGE 1	112.9089
3	AVERAGE 2	187.033
4	AVERAGE 3	123.9661
5	AVERAGE 4	112.8184
6	AVERAGE 5	198.177
7	AVERAGE 6	133.709
8	AVERAGE 7	162.4418

- - - - -

996	AVERAGE 995	121.995
997	AVERAGE 996	171.1544
998	AVERAGE 997	173.2064
999	AVERAGE 998	145.0828
1000	AVERAGE 999	148.1178
1001		
1002	Lower	104.1329
1003	Upper	187.033
1004	ALL AVERAGE	135.6321
1005	ALL VARIANCE	645.8519

The first column indicates the *i*-th iteration of Monte Carlo simulation.
The second column indicates “from the *j*-th point of type B”.
Third column indicates “to its nearest neighbor point of type A”.
The last column indicates the shortest-path distance between those points. The last row of the *i*-th iteration, AVERAGE, indicates the average distance of from each type B point to its nearest type A point.
At the bottom, ALL AVERAGE indicates the average of AVERAGEs, and ALL VARIANCE indicates the variance of AVERAGEs.
“Lower” and “Upper” indicate the lower and upper critical values for a given significance level, say, 5 % (one-sided).

These numbers indicate: the lower critical value is 104.1329, the upper critical value is 187.033(the significance level is 5 %) and the average of the average nearest neighbor distances for 1000 iterations is 135.6321.

The Clark-Evans index is given by AVERAGE in the table of SANETCNNDObservedValue.csv (119.5153 in this case) divided by the last ALL AVERAGE in the table of SANETECNNDExpectedValue.csv (135.6321 in this case); consequently, 0.88.
If the observed AVERAGE is outside the range “Lower” and “Upper”, the CSR hypothesis is rejected with, say 95% confidence level.

- SANETCNNDExpectedValue.R

To visualize SANETCNNDExpectedValue.R, please see section 5.2.

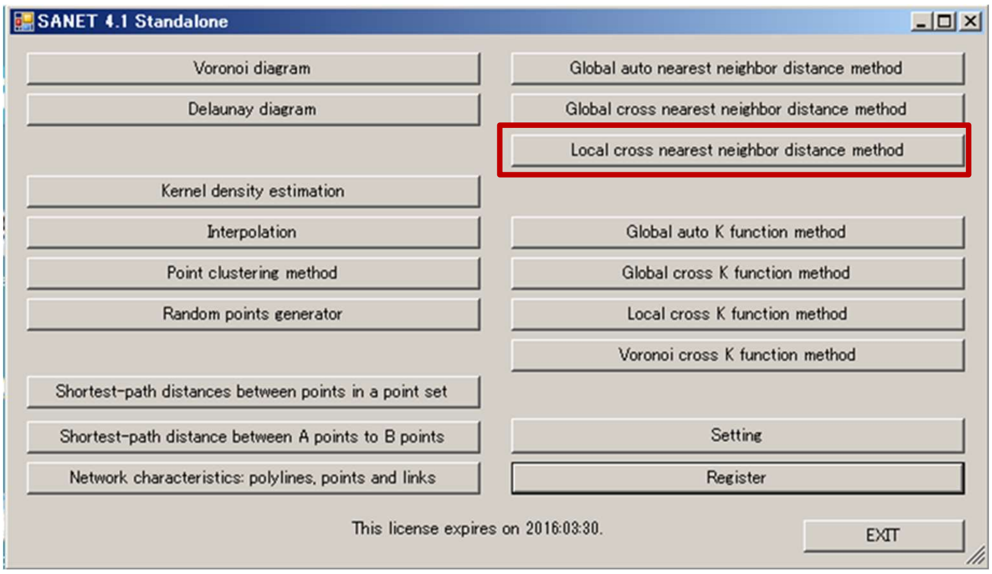
Reference

Okabe, A. and K. Sugihara (2012) *Spatial Analysis along Networks: Statistical and Computational Methods*, Chichester: John Wiley, a volume in the Wiley series of Statistics in Practice.

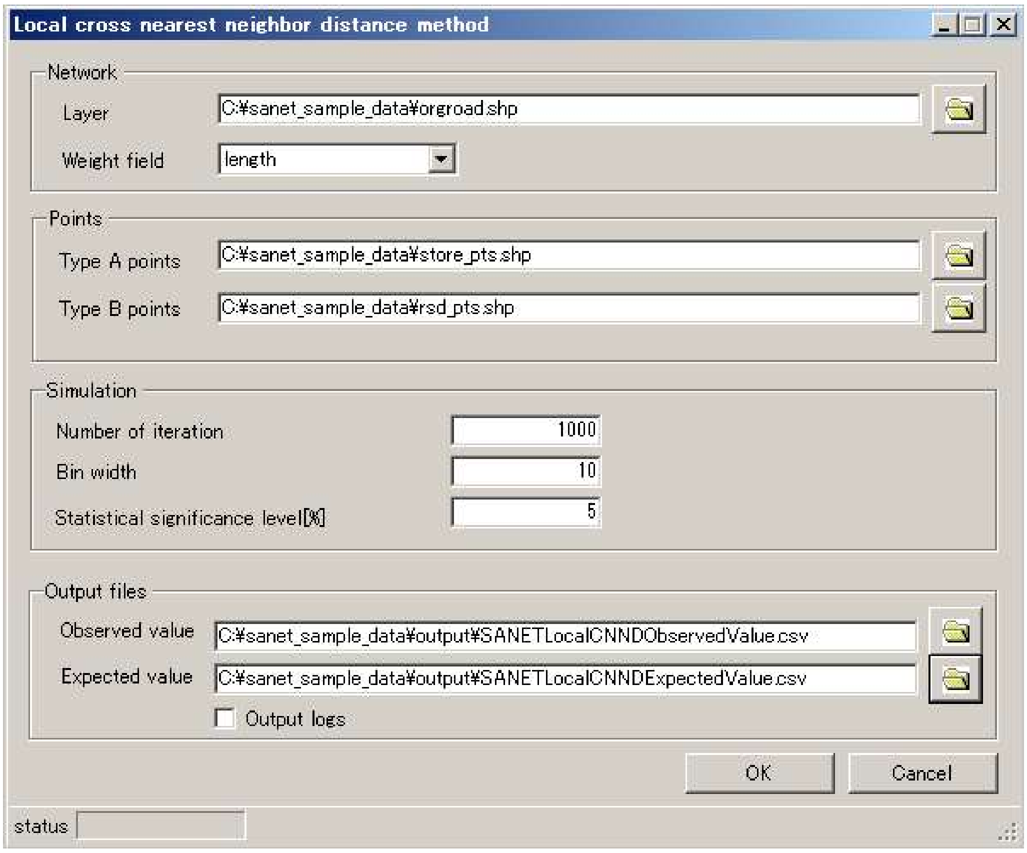
4.12 Tool 12: Local cross nearest neighbor distance method

Given two sets of points: the set of type A points, denoted by A and the set of type B points by B, consider a subset, B*i* , of set B satisfying that the nearest point from every point in B*i* is a specific point in set A, e.g., the *i*-th point in set A. This tool tests the complete spatial randomness (CSR) hypothesis in terms of the shortest-path distances from points in subset B*i* to the *i*-th point in set A. In general, it is assumed that type B points are temporal, while type A points are stable over time; for instance, the former points are restaurants and the latter points are railway stations. This method tests the effect of railway stations on the distribution of restaurants in comparison with the CSR hypothesis. The CSR hypothesis means that points are independently and identically distributed according to the uniform distribution over the network, or points follow the homogeneous binomial point process on the bounded network. A general description about the nearest neighbor distance method is provided in Chapter 5 in Okabe and Sugihara (2012); specifically, the local cross nearest neighbor distance method is shown in Section 5.2.1.

Click the “Local cross nearest neighbor distance method” in the SANET menu.



Then the following window appears.



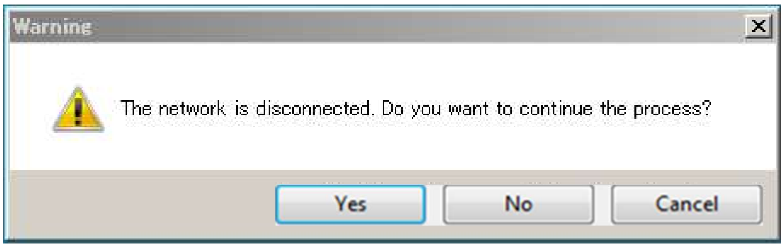
Choose  the file name of a network (e.g., orgroad.shp: street network in the sample data set).

Choose the file name of a set of type A points (e.g., store_pts.shp: store location points in the sample data set).

Choose the file name of a set of type B points (sd_pts.shp: location points in the sample data set).

Choose to store the resulting output file, and click “OK”.

If the following window does not appear, the network is completely connected. Proceed to the next step marked by *** below.



If this window appears, the given network is not completely connected. In that case, see p.55 to find out your next step to proceed.

***

When the network is completely connected or if you click “Yes” in the warning window, the program begins to run. If you do not have any trouble such as memory overflows, you obtain the following six files, three for the observed values and three for the expected values.

SANETLocalCNNDExpectedValue.csv0.csv	2015/09/15 16:12	CSV ファイル	22 KB
SANETLocalCNNDExpectedValue.csv1.csv	2015/09/15 16:12	CSV ファイル	22 KB
SANETLocalCNNDExpectedValue.csv2.csv	2015/09/15 16:12	CSV ファイル	22 KB
SANETLocalCNNDObservedValue.csv0.csv	2015/09/15 16:12	CSV ファイル	1 KB
SANETLocalCNNDObservedValue.csv1.csv	2015/09/15 16:12	CSV ファイル	1 KB
SANETLocalCNNDObservedValue.csv2.csv	2015/09/15 16:12	CSV ファイル	1 KB

The contents of the resulting files are as follows.

- SANETLocalCNNDObservedValue0.csv >> table below left
- SANETLocalCNNDObservedValue1.csv >> table below middle
- SANETLocalCNNDObservedValue2.csv >> table below right

	1	2	3
1	TypeA	TypeB	Distance
2	0	0	76.3664
3	0	1	68.20326
4	0	2	54.13666
5	0	3	61.54551
6	0	4	61.10168
7	0	5	91.89531
8	0	6	134.9839
9	0	7	60.87704
10	0	8	64.04186
11	0	9	147.3969
12	0	20	205.5262
13	0	21	239.424
14	0	28	146.3678
15	0	30	0
16	0	31	27.65748
17	0	32	156.5056
18	0	39	95.54645
19	AVERAGE	99.50448	

	1	2	3
1	TypeA	TypeB	Distance
2	1	10	274.4702
3	1	11	254.9892
4	1	12	205.3936
5	1	13	170.0136
6	1	15	140.1207
7	1	16	55.13684
8	1	17	96.84807
9	1	18	55.19432
10	1	19	87.30142
11	1	22	202.0433
12	1	23	103.8299
13	1	24	60.5299
14	1	25	20.27377
15	1	26	78.34046
16	1	27	0
17	1	29	0
18	1	41	144.9846
19	AVERAGE	114.6747	

	1	2	3
1	TypeA	TypeB	Distance
2	2	14	235.0007
3	2	33	235.0007
4	2	34	55.99832
5	2	35	108.0489
6	2	36	152.4574
7	2	37	60.18162
8	2	38	171.1144
9	2	40	208.5674
10	2	42	119.2154
11	2	43	76.07177
12	2	44	187.2046
13	2	45	106.6349
14	2	46	74.3575
15	2	47	17.92685
16	AVERAGE	129.1272	

- SANETLocalCNNDExpectedValue0.csv >> table below left
- SANETLocalCNNDExpectedValue1.csv >> table below middle
- SANETLocalCNNDExpectedValue2.csv >> table below right

	1	2
1	AVERAGE	95.69445
2	AVERAGE	117.3244
3	AVERAGE	110.5256
4	AVERAGE	81.47645
5	AVERAGE	107.9086
6	AVERAGE	74.80412
7	AVERAGE	114.2097
8	AVERAGE	122.4852
9	AVERAGE	97.76113
10	AVERAGE	[illegible]

	1	2
1	AVERAGE	120.7671
2	AVERAGE	117.305
3	AVERAGE	125.3176
4	AVERAGE	121.775
5	AVERAGE	112.3296
6	AVERAGE	144.4047
7	AVERAGE	160.2015
8	AVERAGE	126.4719
9	AVERAGE	122.7896
10	AVERAGE	[illegible]

	1	2
1	AVERAGE	143.2383
2	AVERAGE	123.816
3	AVERAGE	109.2423
4	AVERAGE	113.5618
5	AVERAGE	128.7714
6	AVERAGE	117.9935
7	AVERAGE	112.1732
8	AVERAGE	96.39701
9	AVERAGE	110.4518
10	AVERAGE	[illegible]

Reference

Okabe, A. and K. Sugihara (2012) *Spatial Analysis along Networks: Statistical and Computational Methods*, Chichester: John Wiley, a volume in the Wiley series of Statistics in Practice.

4.13 Tool 13: Global auto *K* function method

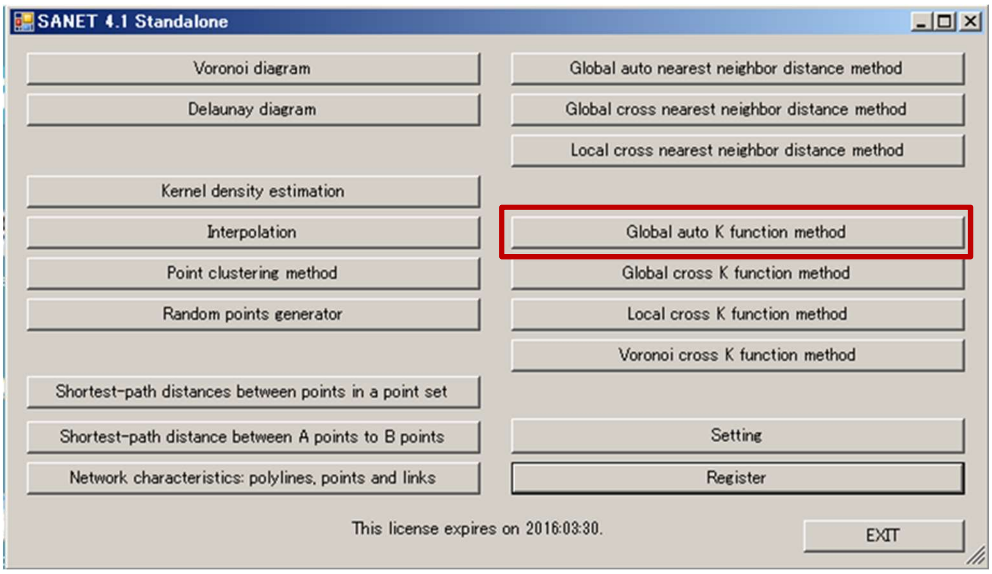
This tool tests the complete spatial randomness (CSR) hypothesis in terms of the number of points in a given point set satisfying that the shortest-path distance from every point to another point is less than a parametric shortest-path distance. The CSR hypothesis means that points are independently and identically distributed according to the uniform distribution over the network, or points follow the homogeneous binomial point process on the bounded network.

To state it explicitly, for a set of *n* points placed on a network, let $n(t \mid p_i)$ be the number of points that are within shortest-path distance t from point p_i and ρ be the density of points on the network. Then the *K* function is given

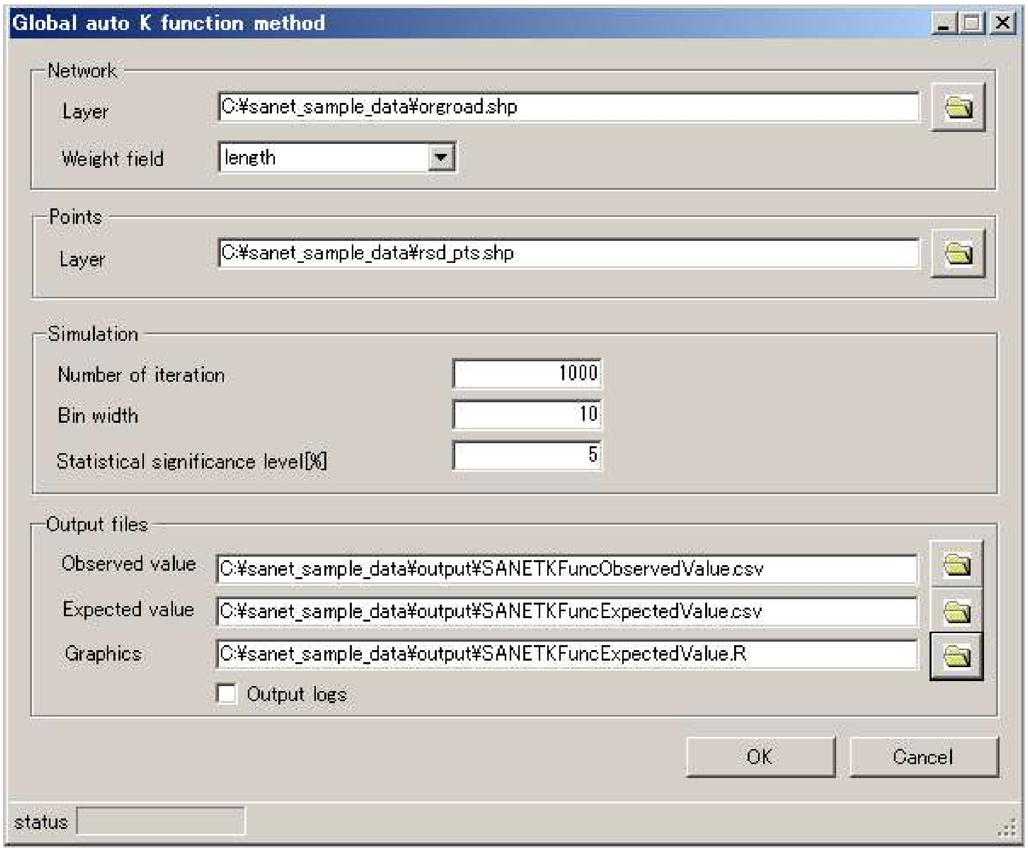
$$K(t) = \frac{1}{\rho} \frac{\sum_{i=1}^{n} n(t \mid p_i)}{n}.$$

In the literature, the *global auto K function method* is simply referred to as the *K function method*. A general review of the *K* function method is illustrated in Chapter 5 in Okabe and Sugihara (2012); specifically, the global auto *K* function method is described in Section 6.1.2, and its application in Section 12.2.3.1.

Click the "Global Auto *K* function Method" in the SANET menu.



Then the following window appears.



Choose the file name of a network (e.g., orgroad.shp: street network in the sample data set)

Choose the file name of a set of points (e.g., rsd_pts.shp: location points in the sample data set).

Fill in:

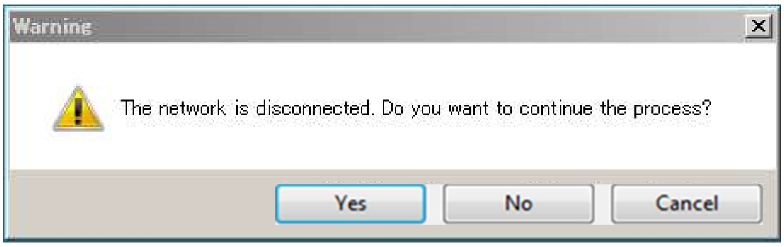
- a number of iterations for Monte Carlo simulation (a default values is 1000),
- a unit interval (a continuous distance is divided by the equal unit interval; a default value is 10) ,
- a statistical significance level (the default value is 5%).

Choose to store the output files.

If you want to obtain intermediate files, check “Output log”. Please note that the output file requires much memory.

Click “OK”.

If the following window does not appear, the network is completely connected. Proceed to the next step marked by *** below.



If this window appears, the given network is not completely connected. In that case, see p.55 to find out your next step to proceed.

***

When the network is completely connected or if you click “Yes” in the warning window, the program begins to run. If you do not have any trouble such as memory overflows, you obtain the following three files.

The contents of the resulting files are as follows.

- SANETObservedValue.csv

	A	B	C
1	FromPntII	ToPntID	Distance
2	0	1	1177.096
3	0	2	3615.931
4	0	3	2201.724
5	0	4	1864.719
6	0	5	723.519
7	0	6	1214.798
8	0	7	1393.547
9	0	8	4468.943
10	0	9	1975.764
11	0	9	1975.764

...

[illegible]	101	[illegible]	[illegible]
10295	101	91	1461.274
10296	101	92	1823.392
10297	101	93	1718.576
10298	101	94	1928.28
10299	101	95	1858.233
10300	101	96	2205.37
10301	101	98	2110.368
10302	101	99	1658.132
10303	101	100	1649.353
10304	AVERAGE	2809.084	

The first column indicates “from the *i*-th point”.

The second column indicates “to the *j*-th point ” ($i \neq j$).

The third column indicates the shortest-path distance between those points.
The last row “AVERAGE” indicates the average of shortest-path distances between any pair of points.

- SANETExpectedValue.csv

This file has two tables.

The first one is as shown below.

	A	B	C	D	E	F	G	H	I	J
1	50	100	150	200	250	300	350	400	450	50
2	0	2	4	16	40	60	94	122	164	20
3	0	2	8	24	40	68	96	122	168	22
4	0	2	8	24	42	68	100	128	174	22
5	0	2	8	26	44	70	100	136	176	22
6	0	2	10	26	44	70	102	136	178	22
7	0	2	10	26	44	70	102	136	178	22
8	0	2	10	26	46	70	102	136	180	22
9	0	2	10	26	46	72	102	138	180	22
10	0	2	12	26	46	72	102	140	182	23

The first row indicates that the unit interval is 50 and the resulting intervals are 50, 100, 150, and so on.

Each column indicates the numbers of points within the distance given by the unit interval ×*i* for 1000 Monte Carlo iterations, and those numbers are ordered from the smallest to the largest. For instance, 101 points are independently and identically generated according to the uniform distribution over the street network for 1000 times; then the points whose nearest neighbor points within 150 are 4, 8, 8, , .

The second table is as shown below. Note that the file size of this example may be too large to fit for Excel. Access may be a better choice.

1002	Simulatio	FromPntID	ToPntID	Distance
1003	0	0	1	3153.692
1004	0	0	2	2269.076
1005	0	0	3	3404.187
1006	0	0	4	2270.249
1007	0	0	5	2871.893
1008	0	0	6	1864.141
1009	0	0	7	3512.234
1010	0	0	8	1063.375
1011	0	0	9	1715.842

…

11294	0	101	90	4485.836
11295	0	101	91	3847.535
11296	0	101	92	1201.482
11297	0	101	93	209.2189
11298	0	101	94	3733.501
11299	0	101	95	4919.234
11300	0	101	96	2757.466
11301	0	101	97	190.2816
11302	0	101	98	5622.348
11303	0	101	99	1466.652
11304	0	101	100	3256.543
11305	1	0	1	685.3707
11306	1	0	2	2009.95
11307	1	0	3	1359.592
11308	1	0	4	903.1245

The first column indicates the *i*-th iteration of Monte Carlo simulation.
The second column indicates “from the *j*-th point” in the point set.
Third column indicates “to the *k*-th point in the point set.
For instance, the above table shows part of the output of the 0-th iteration, where the shortest-path distances are from the 0-th point to the *k*-th point (*k* =1, 2, …., 100).
At the end of this file, AVERAGE appears, e.g., AVERAGE 2824.017907. This implies that the average of the $n(n-1)$ shortest-path distances between any pair of points in the point set.

- SANETGraphics.R

To visualize SANETGraphics.R, please see section 5.2.

Reference

Okabe, A. and K. Sugihara (2012) *Spatial Analysis Along Networks: Statistical and Computational Methods*, Chichester: John Wiley, a volume in the Wiley series of Statistics in Practice.

4.14 Tool 14: Global cross *K* function method

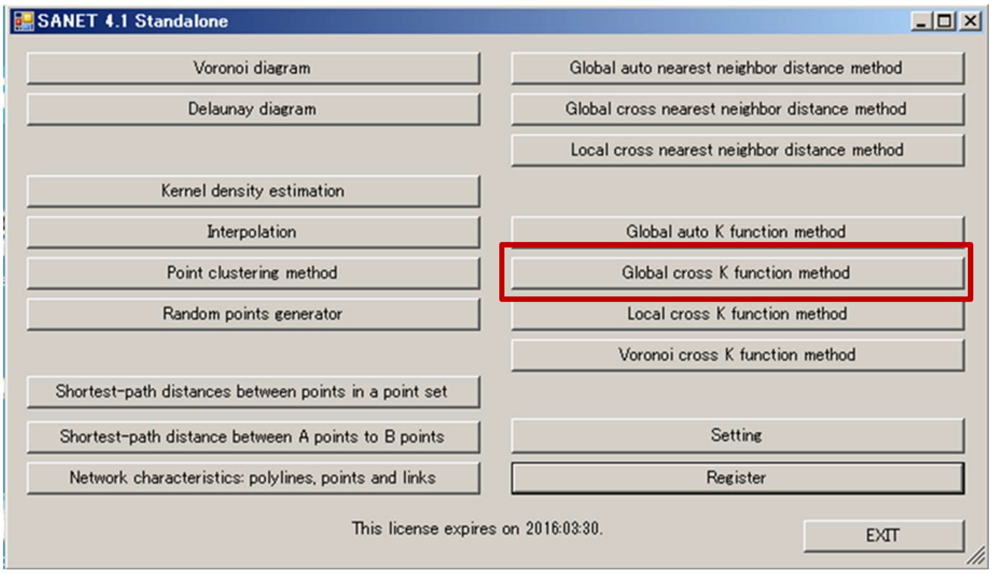
Given two sets of points, a set of type A points and that of type B points, placed on a given network, this tool tests the complete spatial randomness (CSR) hypothesis. The CSR hypothesis means that points are independently and identically distributed according to the uniform distribution over the network, or points follow the homogeneous binomial point process on the bounded network, implying that the configuration of type A points does not affect the distribution of type B points.

To state this test explicitly, consider a set of n_A type A points and that of n_B type B points, and let $n(t \mid p_{\mathrm{A}_i})$ be the number of type B points that are within shortest-path distance t from the *i*-th type A point p_{A_i}, and ρ_{B} be the density of type A points on the network. Then the *local cross K function* is given by

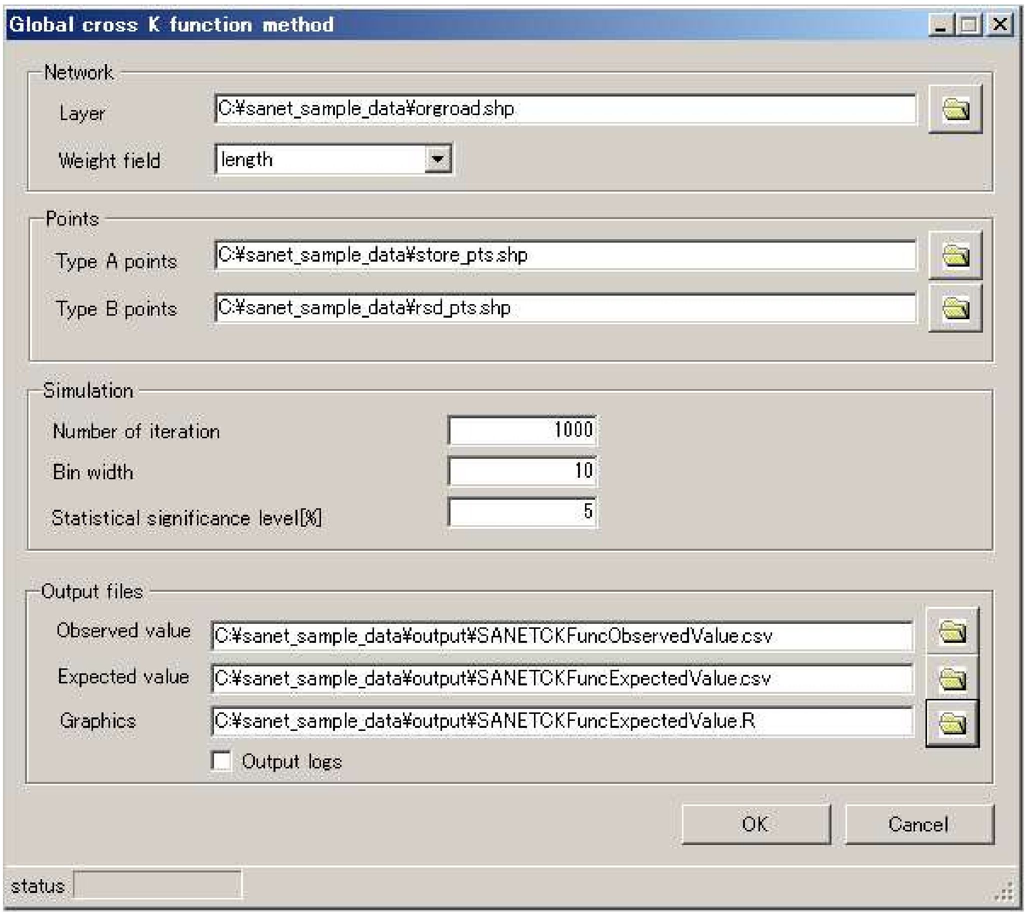
$$K_{\mathrm{AB}}(t \mid p_{\mathrm{A}_i}) = \frac{1}{\rho_{\mathrm{B}}} \frac{\sum_{i=1}^{n_{\mathrm{A}}} n(t \mid p_{\mathrm{A}_i})}{n_{\mathrm{A}}}.$$

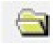
In the literature, the *global cross K function method* is simply referred to as the *cross K function method*. A general review of the cross *K* function method is illustrated in Chapter 6 in Okabe and Sugihara (2012); specifically, the global cross *K* function method is described in Section 6.2.2, and its application in Section 12.2.3.2..

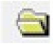
Click the “Global Cross *K* function Method” in the SANET menu.



Then the following window appears.



Choose  the file name of a network (e.g., orgroad.shp: street network in the sample data set).

Choose .

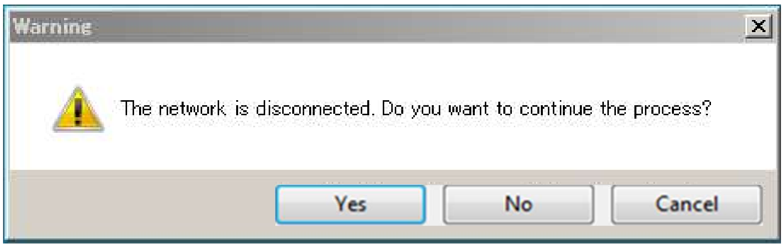
The file of type A points (e.g., store_pts.shp: store location points in the sample data set; type A points are supposed to be structural points)
The file of type B points (e.g., rsd_pts.shp: location points in the sample data set; type B points are supposed to be temporal points).
Fill in:

- the number of iterations for Monte Carlo simulation (a default values is 1000),
- a unit interval (a continuous distance is divided by the equal unit interval; a default value is 10; in this case, the resulting intervals are 10, 20, 30,…..),
- a statistical significance level (the default value is 5%; one-sided).

Choose  the out file where the resulting files are stored

If you want to obtain intermediate files, check “Output log”. Please note that the output file requires much memory.
Click “OK”.

If the following window does not appear, the network is completely connected. Proceed to the next step marked by *** below.



If this window appears, the given network is not completely connected. In that case, see p.55 to find out your next step to proceed.
***
When the network is completely connected or if you click “Yes” in the warning window, the program begins to run. If you do not have any trouble such as memory overflows, you obtain the following three files.

SANETCKFuncExpectedValue.csv	2015/09/15 22:35	CSV ファイル	149 KB
SANETCKFuncGraphics.R	2015/09/15 22:35	R ファイル	5 KB
SANETCKFuncObservedValue.csv	2015/09/15 22:35	CSV ファイル	3 KB

- SANETCKFuncObservedValue.csv

	1	2	3
1	TypeA	TypeB	Distance
2	0	0	76.3664
3	0	1	107.1152
4	0	2	54.13666
5	0	3	108.0195

…

139	2	41	197.3803
140	2	42	119.2154
141	2	43	76.07177
142	2	44	187.2046
143	2	45	115.1871
144	2	46	74.3575
145	2	47	17.92685
146	AVERAGE	251.3185	

The first column indicates "from the *i*-th point of type B".
The second column indicates "to the *j*-th point of type A".
The third column indicates the shortest-path distance between those points.
The last row "AVERAGE" indicates the average shortest-path distance.

- SANETCKFuncExpectedValue.csv

	1	2	3	4	5	6	7	8	9	10
1	10	20	30	40	50	60	70	80	90	100
2	0	0	0	0	1	2	3	3	8	11
3	0	0	0	1	1	2	3	7	9	11
4	0	0	0	1	1	2	5	7	9	11
5	0	0	0	1	2	3	5	7	9	11
6	0	0	0	1	2	3	5	7	10	11
7	0	0	0	1	2	3	5	8	10	12
8	0	0	1	1	2	4	6	8	10	12
9	0	0	1	1	2	4	6	8	10	13
10	0	0	1	1	2	4	6	8	10	13
11	0	0	1	1	2	4	6	8	10	13
12	0	0	1	2	3	4	6	8	10	13

The first row indicates that the unit interval is 10 and the resulting intervals are 10, 20, 30 and so on.
Each column indicates the numbers of points within the distance given by the unit interval×*i* for 1000 Monte Carlo iterations, and those numbers are ordered from the smallest to the largest.

- SANETCKFuncGraphics.R

To visualize SANETCKFuncGraphics.R, please see section 5.2.

Reference

Okabe, A. and K. Sugihara (2012) *Spatial Analysis Along Networks: Statistical and Computational Methods*, Chichester: John Wiley, a volume in the Wiley series of Statistics in Practice.

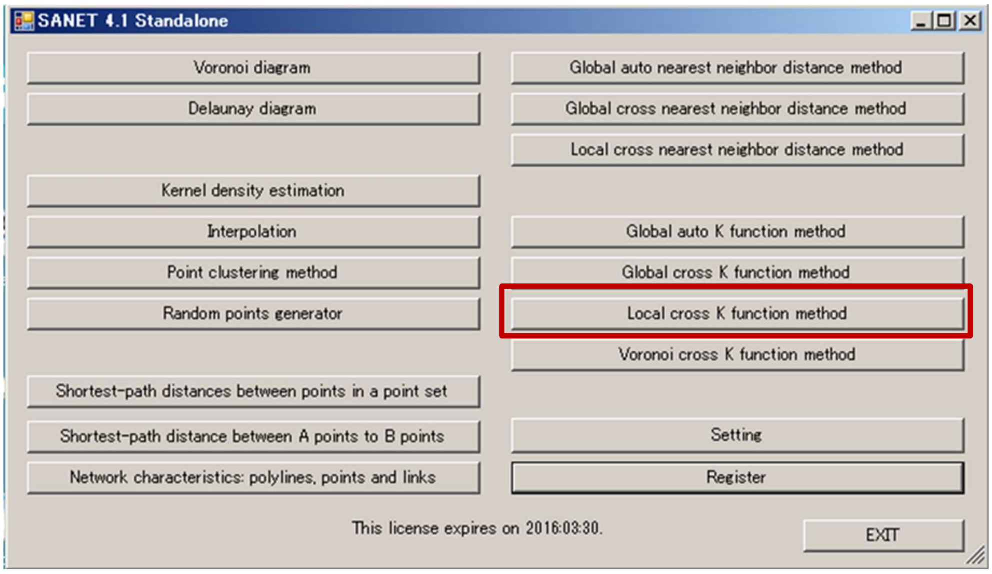
4.15 Tool 15: Local cross *K* function method

Given two sets of points, a set of type A points and that of type B points, placed on a given network, this tool tests the complete spatial randomness (CSR) hypothesis. The CSR hypothesis means that points are independently and identically distributed according to the uniform distribution over the network, or points follow the homogeneous binomial point process on the bounded network, implying that the configuration of type A points does not affect the distribution of type B points.

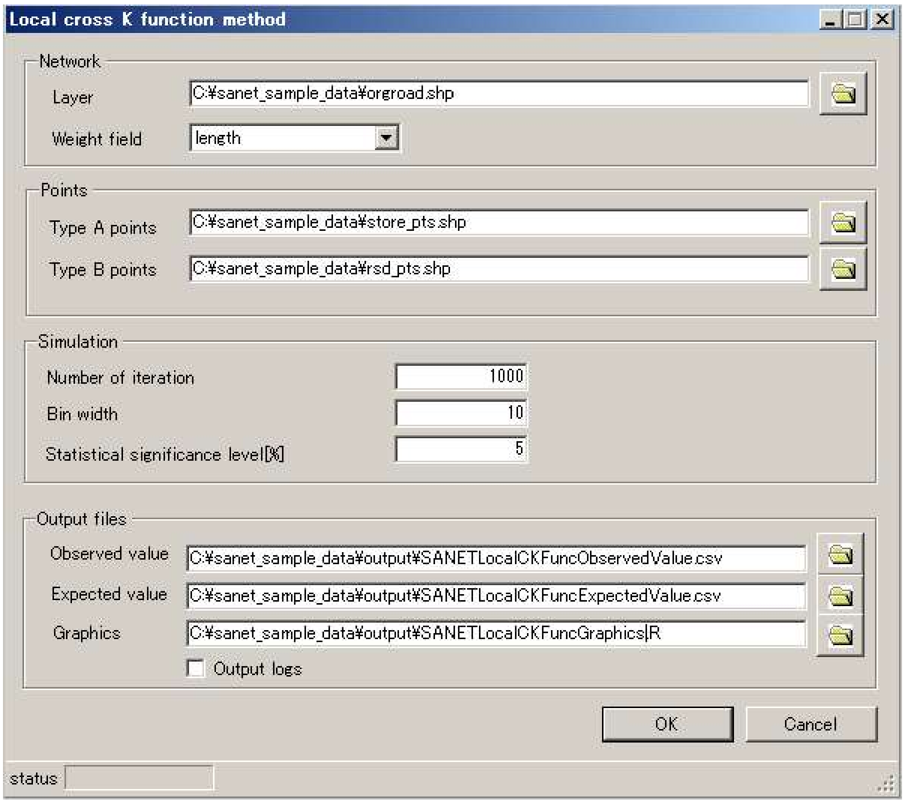
To state this test explicitly, consider a set of n_A type A points and that of n_B type B points, and let $n(t \mid p_{\mathrm{A}_i})$ be the number of type B points that are within shortest-path distance t from the *i*-th type A point p_{A_i}, and ρ_{B} be the density of type A points on the network. Then the *local cross K function* is given by

$$K_{\mathrm{AB}}(t \mid p_{\mathrm{A}_i}) = \frac{1}{\rho_{\mathrm{B}}} n(t \mid p_{\mathrm{A}_\mathrm{i}}).$$

A general review of the cross *K* function method is illustrated in Chapter 6 in Okabe and Sugihara (2012); specifically, the local cross *K* function method is described in Section 6.2.1, and its application in Section 12.2.3.4 (note that type A and type B are reversed in their book).



Click the 'Local cross *K* function method' in the SANET menu.
Then the following window appears.



Choose  the file name of a network (e.g., orgroad.shp: street network in the sample data set).

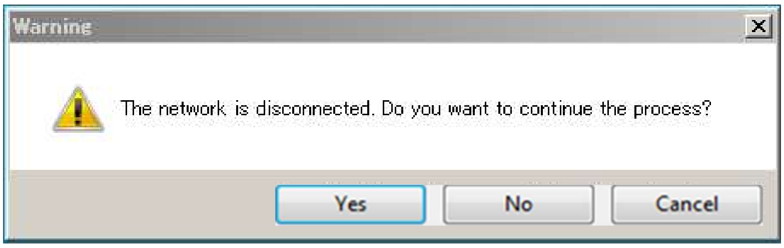
Choose : the file of type A point set (e.g., store_pts.shp: location points in the sample data set), the file of type B point set (e.g., rsd_pts.shp: store location points in the sample data set). Recall that the local cross *K* function deals with the number of type B points within a parametric shortest-path distance from the *i*-th point in the type A point set.

Fill in:

- a number of iterations for Monte Carlo simulation (a default values is 1000),
- a unit interval (a continuous distance is divided by the equal unit interval; a default value is 10; in this case, the resulting intervals are 10, 20, 30,…..; in the example, 10 was used) ,
- a statistical significance level (the default value is 5%; one-sided).

Choose  to store the resulting output files, and
Click “OK”.

If the following window does not appear, the network is completely connected. Proceed to the next step marked by *** below.



If this window appears, the given network is not completely connected. In that case, see p.55 to find out your next step to proceed.

***

When the network is completely connected or if you click “Yes” in the warning window, the program begins to run. Note that for 2 type A points and 47 type B points on the street network in the sample data set. Also note that this computation requires much output space, and so your computer memory might be overflowed. When you use the local cross K function, you are interested in particular type A points, and so you are supposed to use a small number of type A points. To shorten computational time, you may choose a long unit interval, say 100 (the default is 10).

If the memory is enough, the following files are obtained in the output file. The number of files is the number of A points multiplied by three. In the example, the number is nine (three multiplied by three).

SANETLocalCKFuncExpectedValue.csv0.csv	2015/09/15 22:12	CSV ファイル	124 KB
SANETLocalCKFuncExpectedValue.csv1.csv	2015/09/15 22:12	CSV ファイル	109 KB
SANETLocalCKFuncExpectedValue.csv2.csv	2015/09/15 22:12	CSV ファイル	128 KB
SANETLocalCKFuncGraphics.R0.r	2015/09/15 22:12	R ファイル	5 KB
SANETLocalCKFuncGraphics.R1.r	2015/09/15 22:12	R ファイル	5 KB
SANETLocalCKFuncGraphics.R2.r	2015/09/15 22:12	R ファイル	5 KB
SANETLocalCKFuncObservedValue.csv0.csv	2015/09/15 22:12	CSV ファイル	1 KB
SANETLocalCKFuncObservedValue.csv1.csv	2015/09/15 22:12	CSV ファイル	1 KB
SANETLocalCKFuncObservedValue.csv2.csv	2015/09/15 22:12	CSV ファイル	1 KB

- SANETLocalCKFuncObservedValue.csv

	A	B	C
1	ID	ToPntID	Distance
2	0	0	2992.523
3	0	1	1154.274
4	0	2	2381.217
5	0	3	2420.494
6	0	3	2420.494
7	0	4	1829.308
8	0	5	3219.903

…

43	0	41	3367.795
44	0	42	3382.072
45	0	43	2103.076
46	0	44	2091.224
47	0	45	652.6292
48	0	46	3482.345
49	AVERAGE	1821.108	

The first column indicates the *i*-th point in the type A point set.
The second column indicates the *j*-th point in the type B point set.
The last column indicates the shortest-path distance between those points.
The last row indicates the average distance fom the *i*-th A point to evey B point.

- SANETLocalCKFuncExpectedValue.csv

This file has two tables.
The first table is shown below.

	A	B	C	D
1	Simulatio	TypeA	TypeB	Distance
2	0	0	0	1443.21
3	0	0	1	2875.671
4	0	0	2	2785.1
5	0	0	3	3325.874
6	0	0	4	1031.78
7	0	0	5	1633.424
8	0	0	6	3255.872

…

2	0	0	40	500.8608
3	0	0	41	1495.888
4	0	0	42	2163.716
5	0	0	43	2172.226
6	0	0	44	257.8137
7	0	0	45	841.3883
8	0	0	46	1862.649
9	AVERAGE	1988.712		
0	1	0	0	2810.533

…

47995	999	0	41	368.4656
47996	999	0	42	2629.683
47997	999	0	43	1764.055
47998	999	0	44	4320.192
47999	999	0	45	1343.431
48000	999	0	46	2469.002
48001	AVERAGE	2171.234		
48002	ALL AVER/	2264.785		

The first column indicates the i-th iteration of Monte Carlo simulation.

The second column indicates "from the j-th point of type A".

Third column indicates "to the k-th point of type B".

The last column indicates the shortest-path distance between those points.

The last row "AVERAGE" of each simulation indicates the average of those shortest-path distances (e.g., the 9th row is the average for the first (denoted by 0) simulation.

The last row of the last simulation (e.g., 999) indicates the average of the averages of all simulations.

The second table is shown below.

48004	100	200	300	400	500	600	700	800	900	1000	1100
48005	0	0	0	0	0	0	0	0	0	1	1
48006	0	0	0	0	0	0	0	0	0	1	1
48007	0	0	0	0	0	0	0	0	1	1	1
48008	0	0	0	0	0	0	0	0	1	1	2
48009	0	0	0	0	0	0	0	0	1	1	2
48010	0	0	0	0	0	0	0	0	1	1	2
48011	0	0	0	0	0	0	0	1	1	1	2

The row indicates intervals (with a chosen unit interval). For instance, 100, 200, 300, and so on.

The column indicates the numbers of type B points within the unit interval for 1000 iterations, and those numbers are ordered from the smallest to the largest. For instance, in the 900 column, the numbers of the type A points within 900 are: 0, 0, 1, 1, … (a number of these numbers is 1000) for 1000 iterations.

- SANETLocalCKFuncGraphics.R

To visualize SANETLocalCKFuncGraphics.R, please see section 5.2.

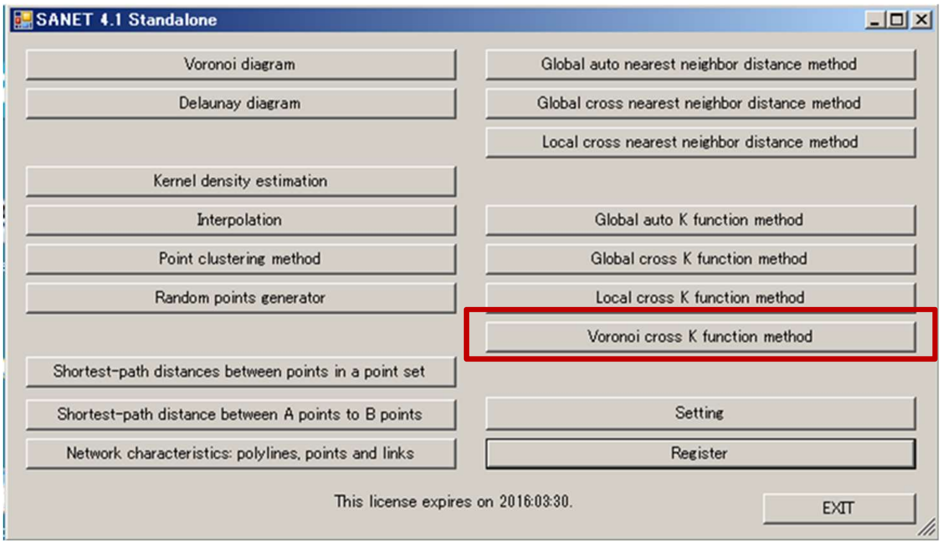
Reference

Okabe, A. and K. Sugihara (2012) *Spatial Analysis along Networks: Statistical and Computational Methods*, Chichester: John Wiley, a volume in the Wiley series of Statistics in Practice.

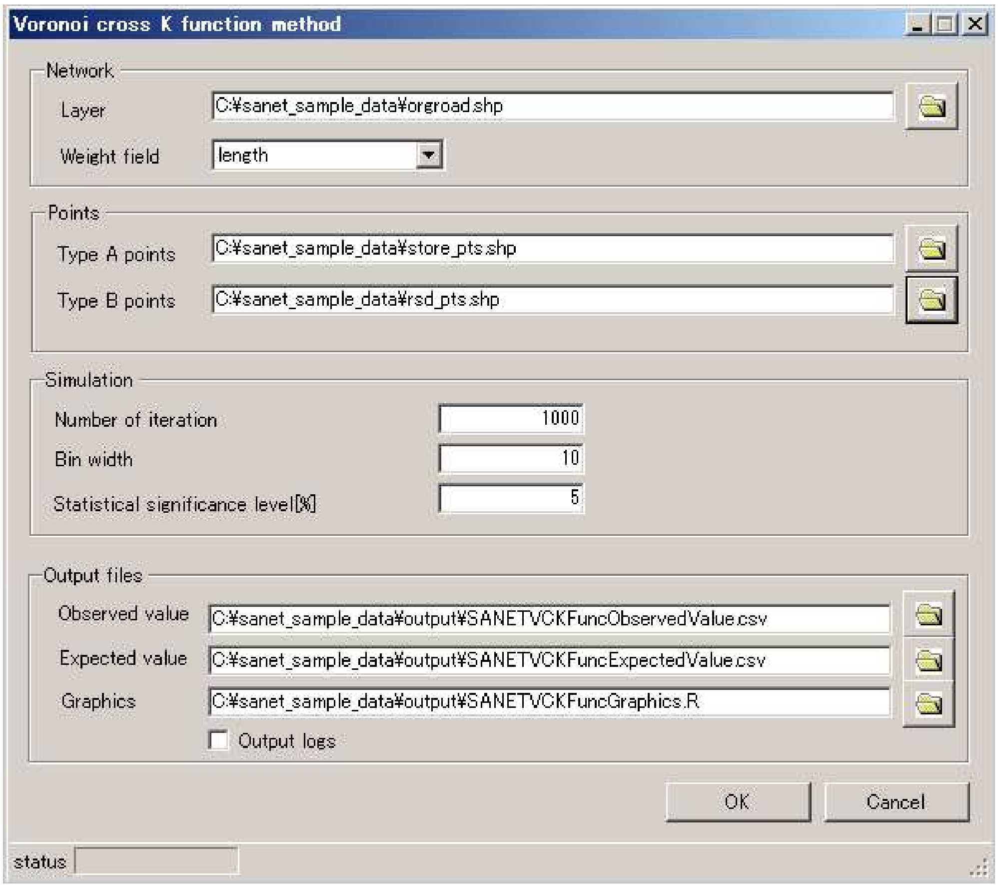
4.16 Tool 16: Voronoi cross *K* function method

Given two sets of points placed a network: a set of type A points (its number is n_A; a network Voronoi network is generated by A points; an example a network Voronoi diagram is shown in Figure 1 in Section 6.1) and that of type B points (its number is n_B), this tool tests the complete spatial randomness (CSR) hypothesis in terms of the number of type B points that are within a parametric shortest-path distance *t* from their nearest type A points. To state it a little more explicitly, consider the shortest-path distance from every type B point to its nearest type A point. Note that if a type B point is placed in the Voronoi subnetwork of the *i*-th type A point, the nearest type A point from the type B point is the *i*-th type A point. The number of the resulting shortest-path distances is n_B. Next consider a function, *K*(*t*), that indicates the number of type B points satisfying that the associated shortest-path distances are less than *t*. The tool tests the CSR hypothesis with *K*(*t*).

Click the “Voronoi Cross *K* function Method” in the SANET menu.



Then the following window appears.



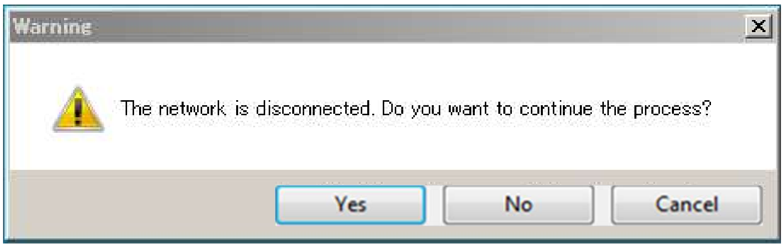
Choose  the file name of a network (e.g., orgroad.shp: street network in the sample data set).

Choose : the file of type A point set (e.g., store_pts.shp: location points in the sample data set), and the file of type B point set (e.g., rsd_pts.shp: store location points in the sample data set).

Fill in:

- a number of iterations for Monte Carlo simulation (a default values is 1000),
- a unit interval (a continuous distance is divided by the equal unit interval; a default value is 10; in this case, the resulting intervals are 10, 20, 30,…..),
- a statistical significance level (the default value is 5%).

Choose to store the resulting output files, and
Click “OK”.
If the following window does not appear, the network is completely connected. Proceed to the next step marked by *** below.



If this window appears, the given network is not completely connected. In that case, see p.55 to find out your next step to proceed.
***
When the network is completely connected or if you click “Yes” in the warning window, the program begins to run. If you do not have any trouble such as memory overflows, you obtain the following three files.

SANETVCKFuncExpectedValue.csv	2015/09/15 22:20	CSV ファイル	79 KB
SANETVCKFuncGraphics.R	2015/09/15 22:20	R ファイル	3 KB
SANETVCKFuncObservedValue.csv	2015/09/15 22:20	CSV ファイル	1 KB

- SANETVCKFuncObservedValue.csv

	1	2	3
1	FromPntID	ToPntID	Distance
2	0	0	76.3664
3	0	1	107.1152
4	0	2	54.13666
5	0	3	108.0195
6	0	4	78.20031
7	0	5	91.89531
8	0	6	134.9839
9	0	7	[illegible]

…

42	2	37	[illegible]
43	2	40	208.5674
44	2	42	119.2154
45	2	43	76.07177
46	2	44	187.2046
47	2	45	115.1871
48	2	46	74.3575
49	2	47	17.92685
50	AVERAGE	119.5153	

The first column indicates "from the *i*-th point of type A".
The second column indicates "to the *j*-th point of type B".
The third column indicates the shortest-path distance between those points.
The last row "AVERAGE" indicates the average distance.

	1	2	3	4	5	6	7	8	9	10
1	10	20	30	40	50	60	70	80	90	10
2	0	0	0	0	1	2	3	3	8	1
3	0	0	0	1	1	2	3	7	9	1
4	0	0	0	1	1	2	5	7	9	1
5	0	0	0	1	2	3	5	7	9	1
6	0	0	0	1	2	3	5	7	10	1
7	0	0	0	1	2	3	5	8	10	1
8	0	0	1	1	2	4	6	8	10	1
9	0	0	1	1	2	4	6	8	10	1
10	0	0	1	1	2	4	6	8	10	1
11	0	0	1	1	2	4	6	8	10	1

- SANETVCKFuncExpectedValue.csv

The row indicates bin lengths. For instance, 10 means the bin length is 0-10; 20 means the bin length is 10-20; and so on. While the column indicates the numbers of points in the *i*-th bin for 1000 iterations, and those numbers are ordered from the smallest to the largest.

- SANETVCKFuncGraphics.R

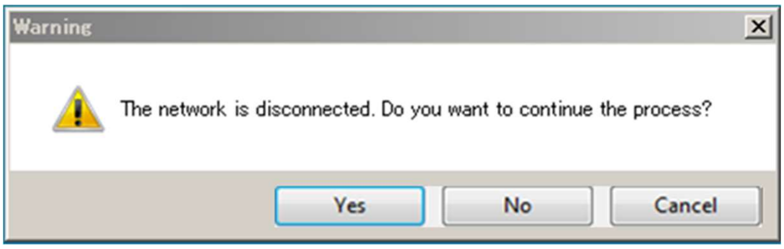
To visualize SANETVCKFuncGraphics.R, please see section 5.2.

Reference

Okabe, A. and K. Sugihara (2012) *Spatial Analysis Along Networks: Statistical and Computational Methods*, Chichester: John Wiley, a volume in the Wiley series of Statistics in Practice.

4.17 When SANET claims "The network is disconnected"

As you run SANET, you may occasionally come up with the below window;

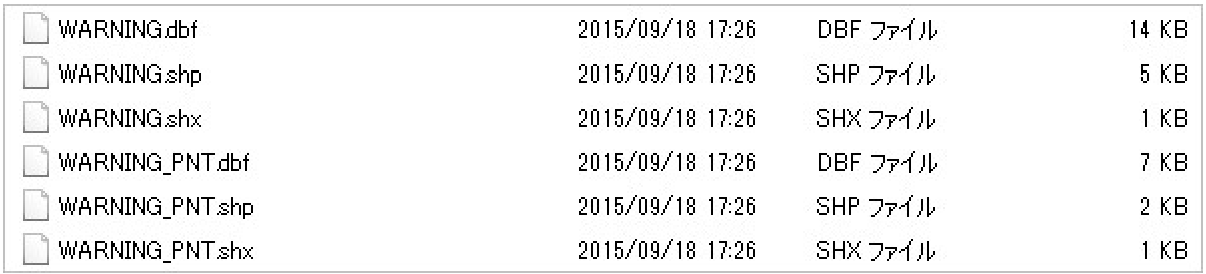


If this window appears, the given network is not completely connected.

If you say "Cancel", the SANET do nothing.

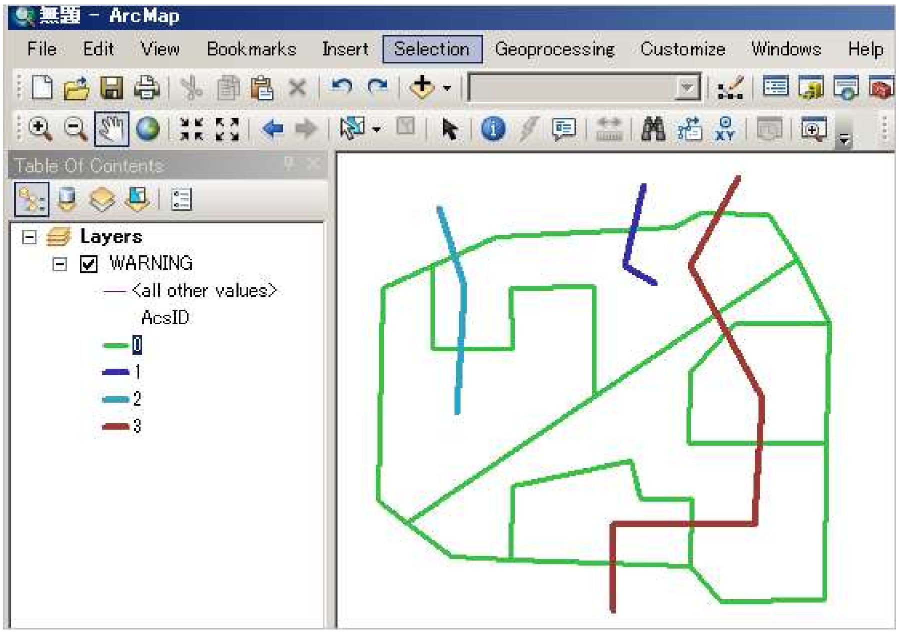
If you say "Yes", the SANET chooses the largest connected network included in the give network.

If you say "No", the SANET gives you following six file.

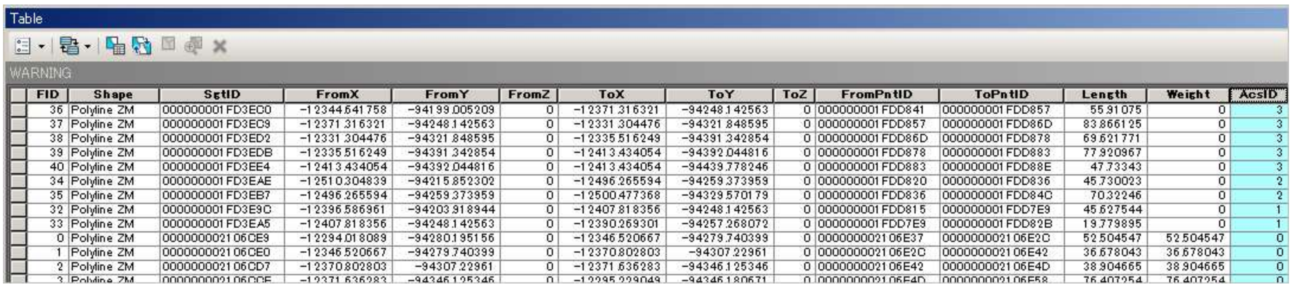


WARNING.dbf	2015/09/18 17:26	DBF ファイル	14 KB
WARNING.shp	2015/09/18 17:26	SHP ファイル	5 KB
WARNING.shx	2015/09/18 17:26	SHX ファイル	1 KB
WARNING_PNT.dbf	2015/09/18 17:26	DBF ファイル	7 KB
WARNING_PNT.shp	2015/09/18 17:26	SHP ファイル	2 KB
WARNING_PNT.shx	2015/09/18 17:26	SHX ファイル	1 KB

When you add this Shapefile on ArcGIS, for example, you will be provided with the below warning map which indicates disconnected parts by color.

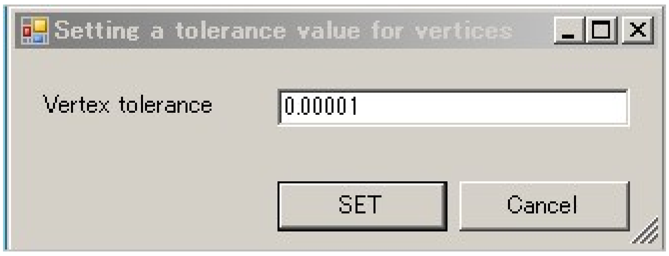


The attribute table shows disconnected links by numbers on the AcsID column.



FID	Shape	SgtID	FromX	FromY	FromZ	ToX	ToY	ToZ	FromPntID	ToPntID	Length	Weight	AcsID
36	Polyline ZM	000000001FD3EC0	-12344.641758	-94199.005209	0	-12371.316321	-94248.142563	0	000000001FDD841	000000001FDD857	55.91075	0	3
37	Polyline ZM	000000001FD3EC9	-12371.316321	-94248.142563	0	-12331.304476	-94321.848595	0	000000001FDD857	000000001FDD86D	83.866125	0	3
38	Polyline ZM	000000001FD3ED2	-12331.304476	-94321.848595	0	-12335.516249	-94391.342854	0	000000001FDD86D	000000001FDD878	69.621771	0	3
39	Polyline ZM	000000001FD3EDB	-12335.516249	-94391.342854	0	-12413.434054	-94392.044816	0	000000001FDD878	000000001FDD883	77.920967	0	3
40	Polyline ZM	000000001FD3EE4	-12413.434054	-94392.044816	0	-12413.434054	-94439.778246	0	000000001FDD883	000000001FDD88E	47.73343	0	3
34	Polyline ZM	000000001FD3EAE	-12510.304839	-94215.852302	0	-12496.265594	-94259.373959	0	000000001FDD820	000000001FDD836	45.730023	0	2
35	Polyline ZM	000000001FD3EB7	-12496.265594	-94259.373959	0	-12500.477368	-94329.57079	0	000000001FDD836	000000001FDD84C	70.32246	0	2
32	Polyline ZM	000000001FD3E9C	-12396.586961	-94203.918944	0	-12407.818356	-94248.142563	0	000000001FDD815	000000001FDD7E9	45.627544	0	1
33	Polyline ZM	000000001FD3EA5	-12407.818356	-94248.142563	0	-12390.269301	-94257.268072	0	000000001FDD7E9	000000001FDD82B	19.779895	0	1
0	Polyline ZM	000000002106CE9	-12294.018089	-94280.195156	0	-12346.520667	-94279.740399	0	000000002106E37	000000002106E2C	52.504547	52.504547	0
1	Polyline ZM	000000002106CE0	-12346.520667	-94279.740399	0	-12370.802803	-94307.22961	0	000000002106E2C	000000002106E42	36.678043	36.678043	0
2	Polyline ZM	000000002106CD7	-12370.802803	-94307.22961	0	-12371.636283	-94346.125346	0	000000002106E42	000000002106E4D	38.904665	38.904665	0
3	Polyline ZM	000000002106CCE	-12371.636283	-94346.125346	0	-12295.229049	-94346.180671	0	000000002106E4D	000000002106E58	76.407254	76.407254	0

If you think that the disconnected parts should be connected, click "Setting" in the SANET menu. Then the following window appears.

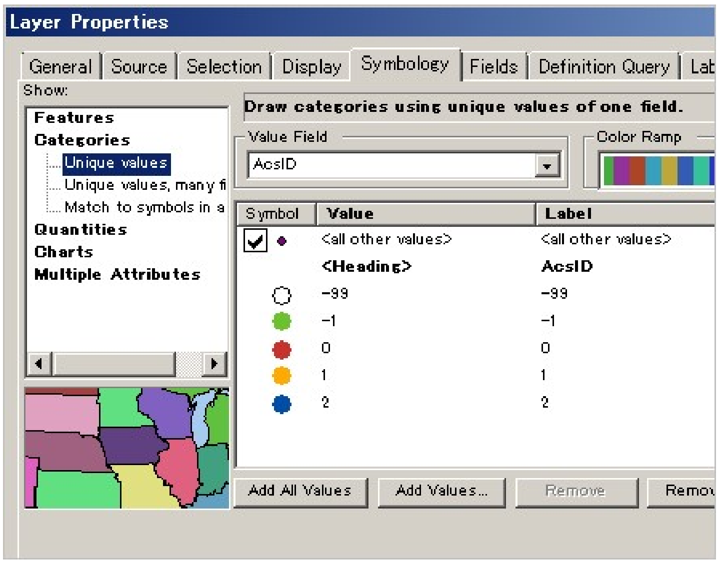


The "Vertex tolerance" means that if the distance between two nodes (vertexes) is less than 0.001(the unit depends on the scale of the map you are using, default is 0.00001), then two vertices are regarded as the same. You change the tolerance distance and try again. If your network is still disconnected, you are supposed to connect disconnected parts using tools in ArcGIS or QGIS.

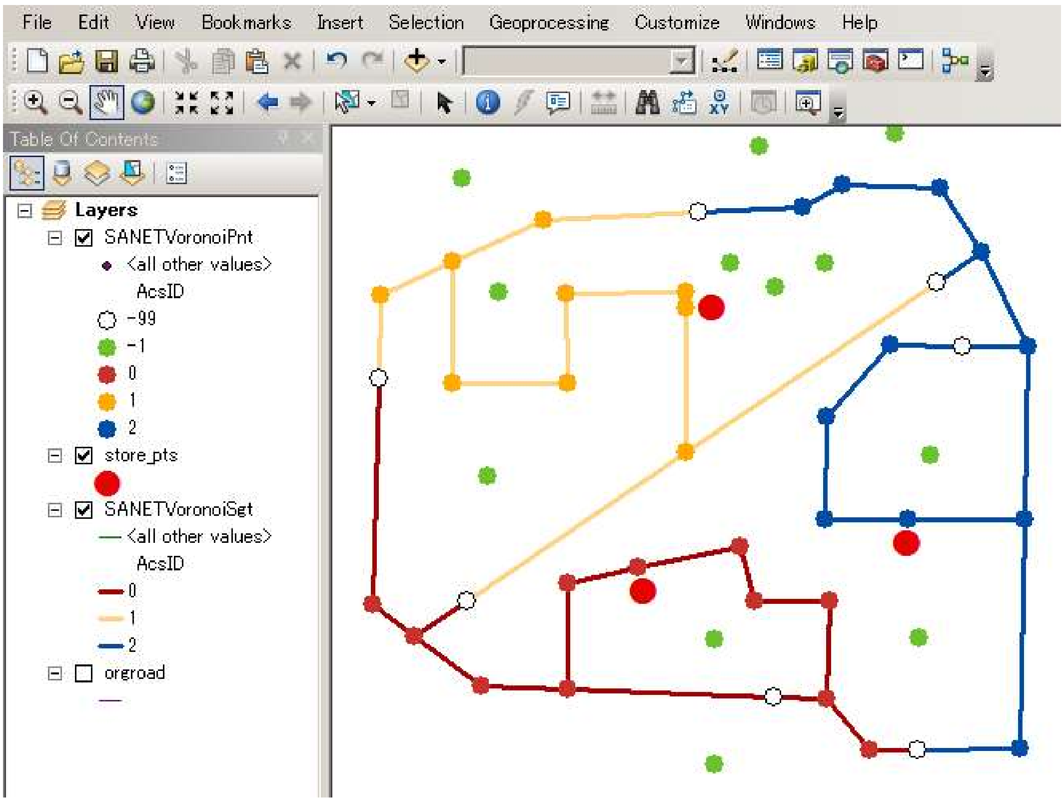
5 Visualize the SANET sample results on ArcGIS, R and QGIS

5.1 Visualize on ArcGIS and ArcScene

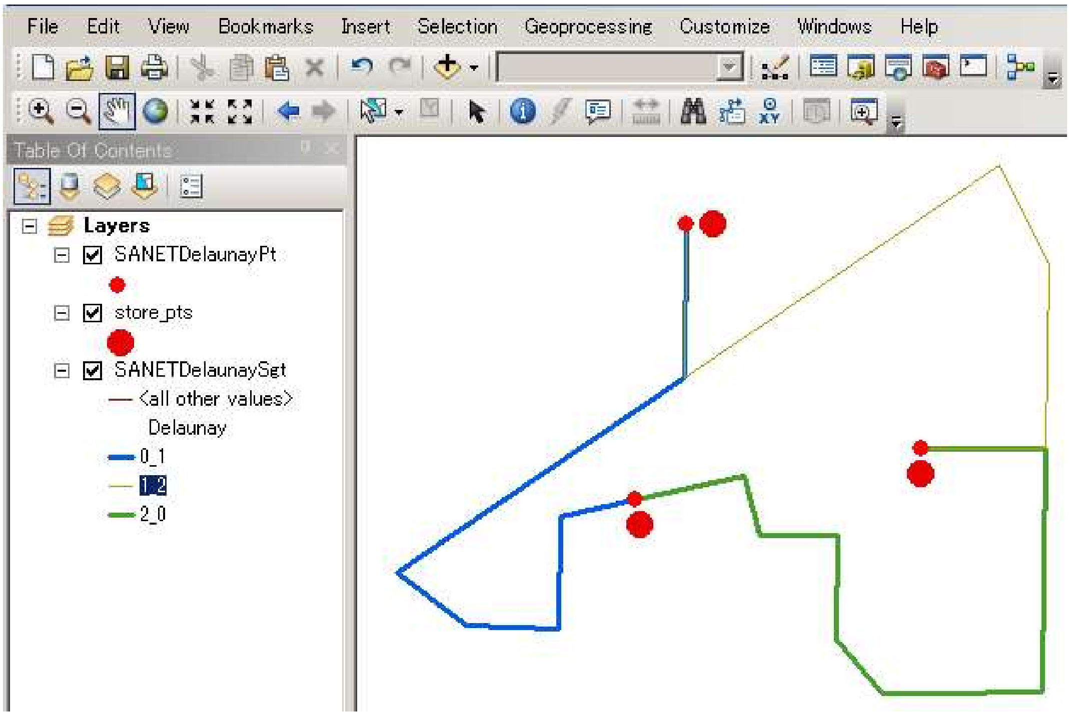
Launch ArcMap and add the Shapefile which the SANET yielded. Open the layer properties window and set “Value Field” and add values at the symbology tab to make the layer look properly. The below is an example of setting “Symbology”.



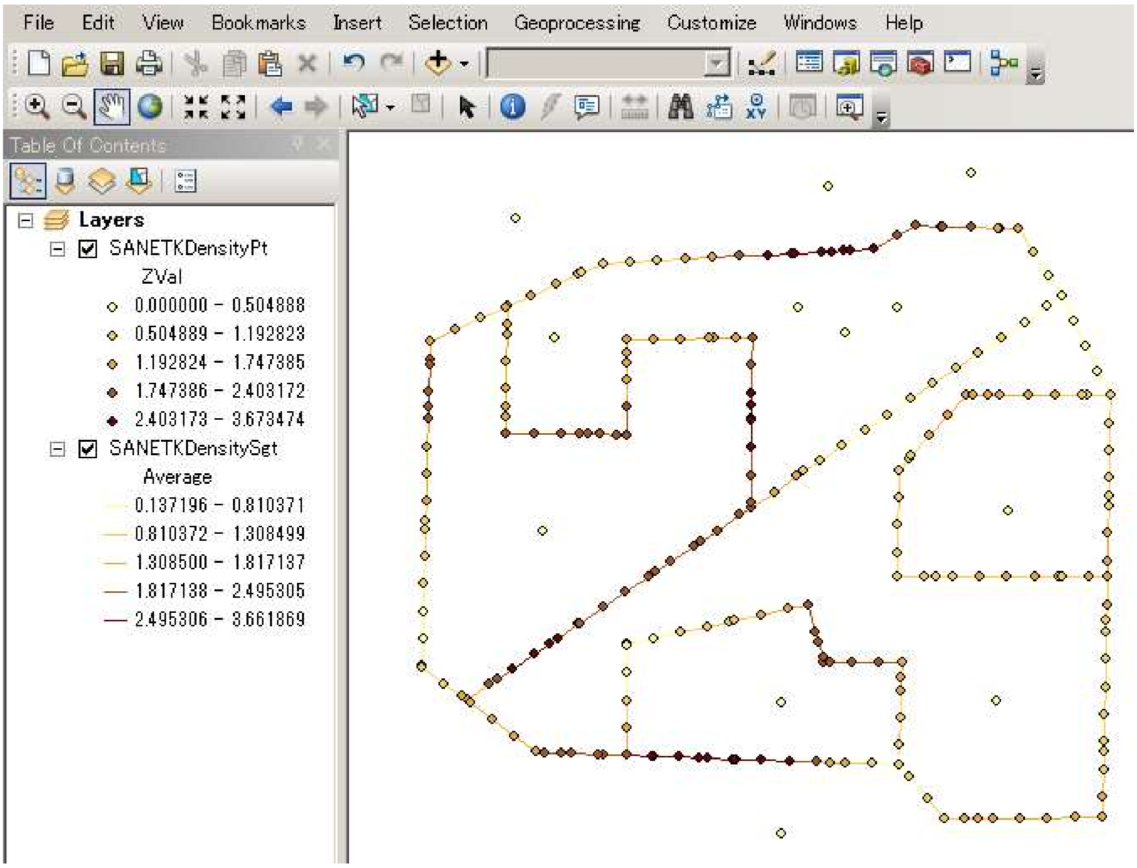
- Tool 01:Voronoi diagram



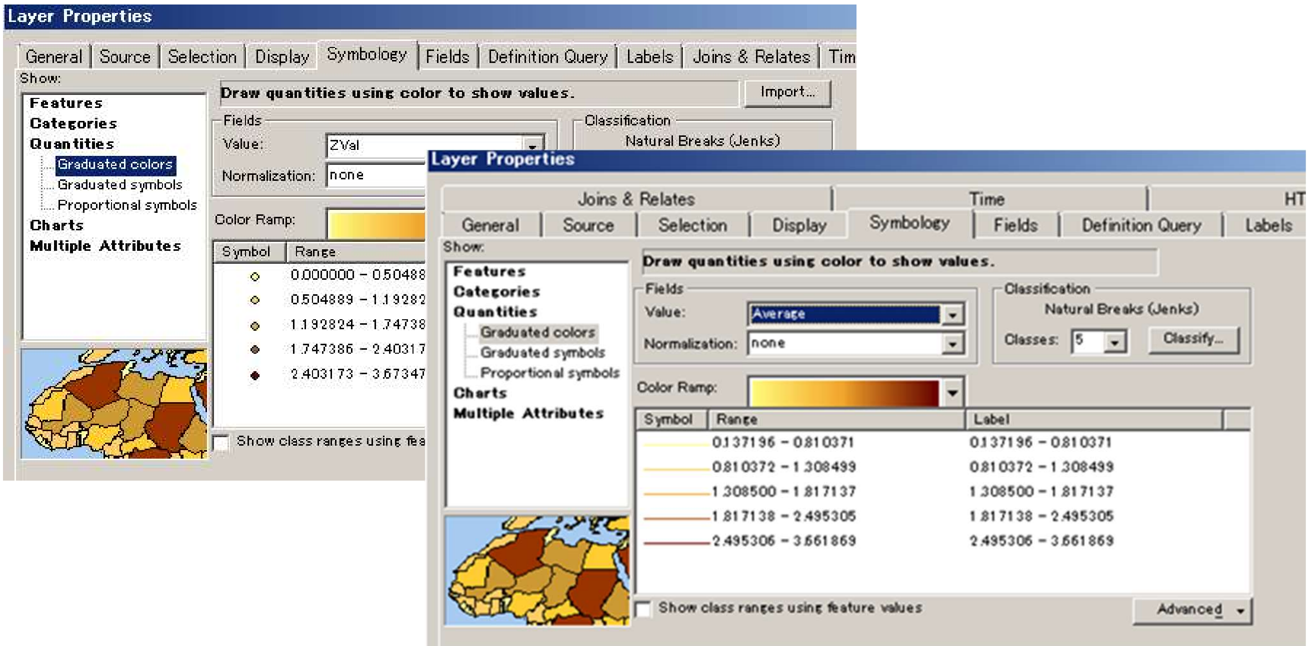
- Tool 02:Delaunay diagram



- Tool 03: Kernel density estimation

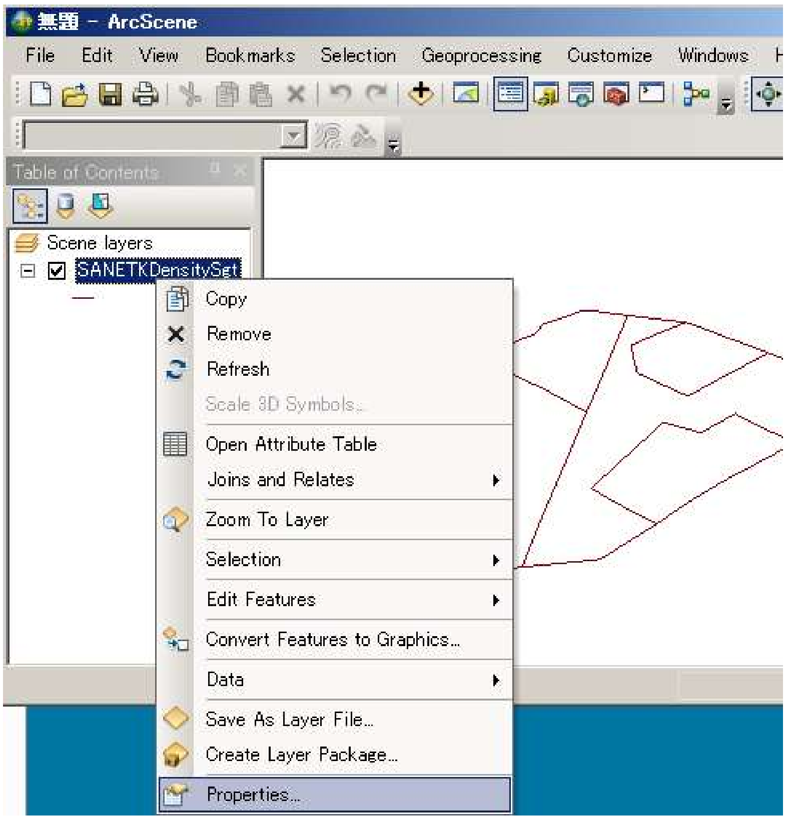


Above figure is classified as below;



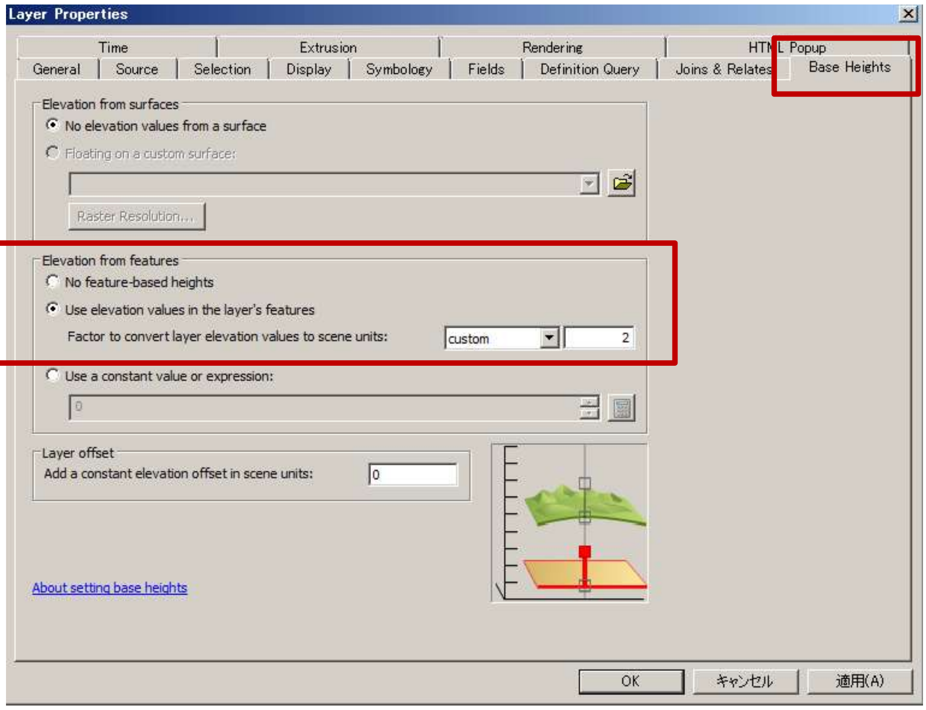
Launch ArcScene, and you can present the figure in 3D.

Add the SHAPEFILE;SANETKDensitySgt.shp using  button.

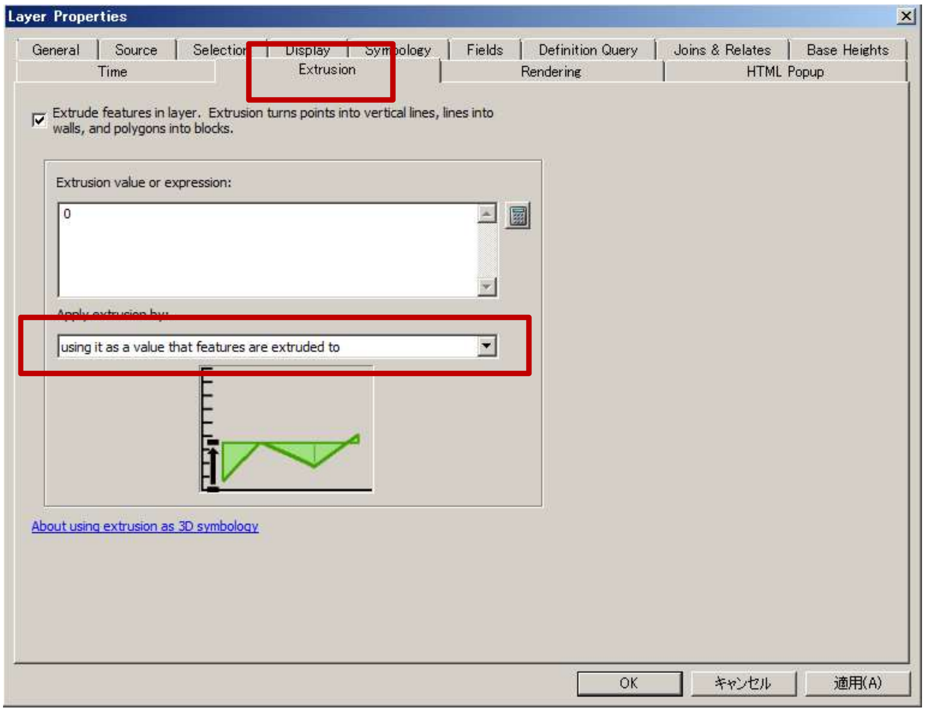


Go to layer properties to set the extrusion information.

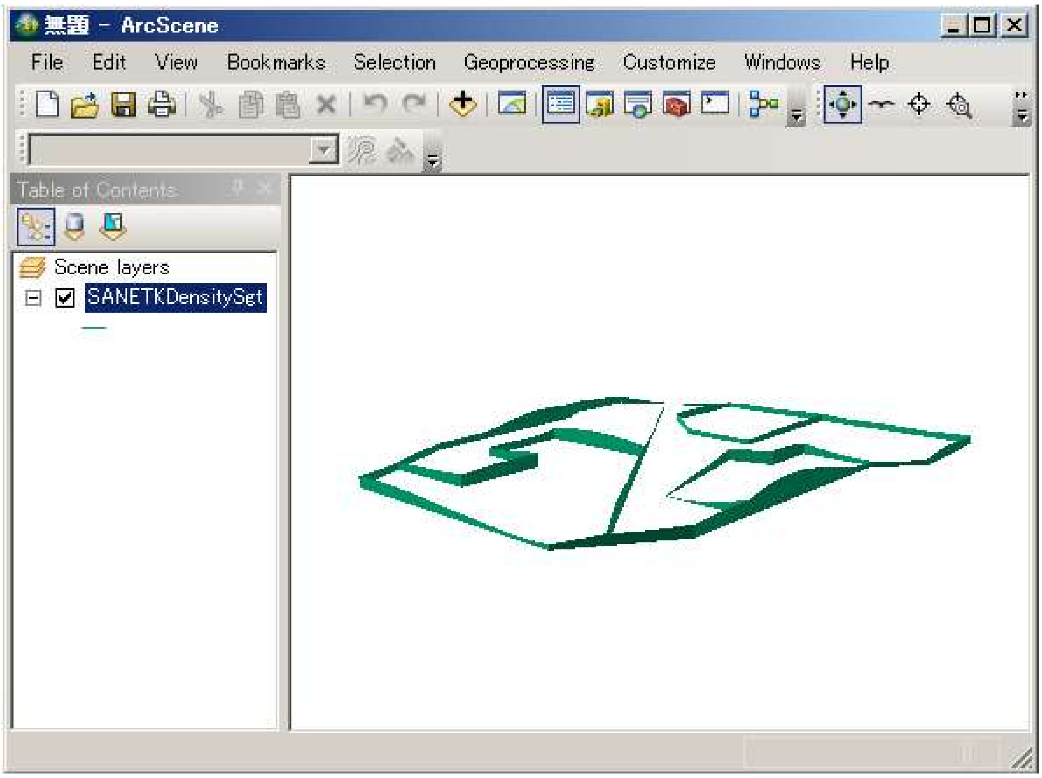
- Base Heights : Set the elevation value such as 2.0 to exaggerate a bit.



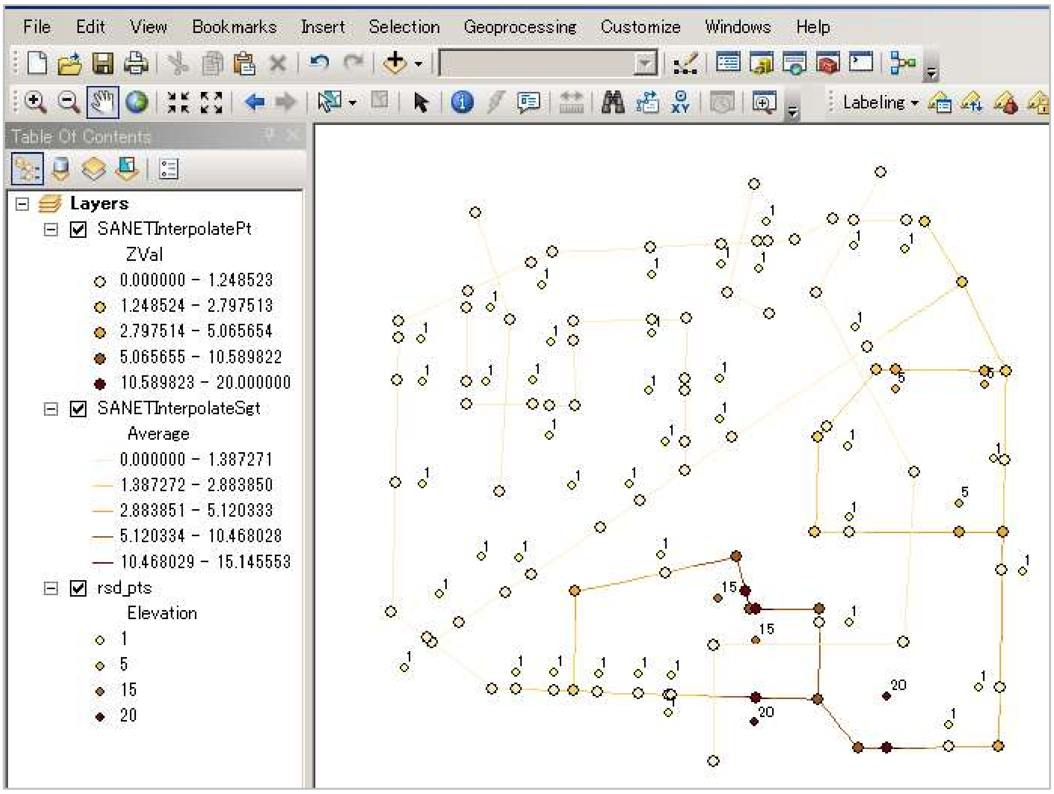
- Extrusion :



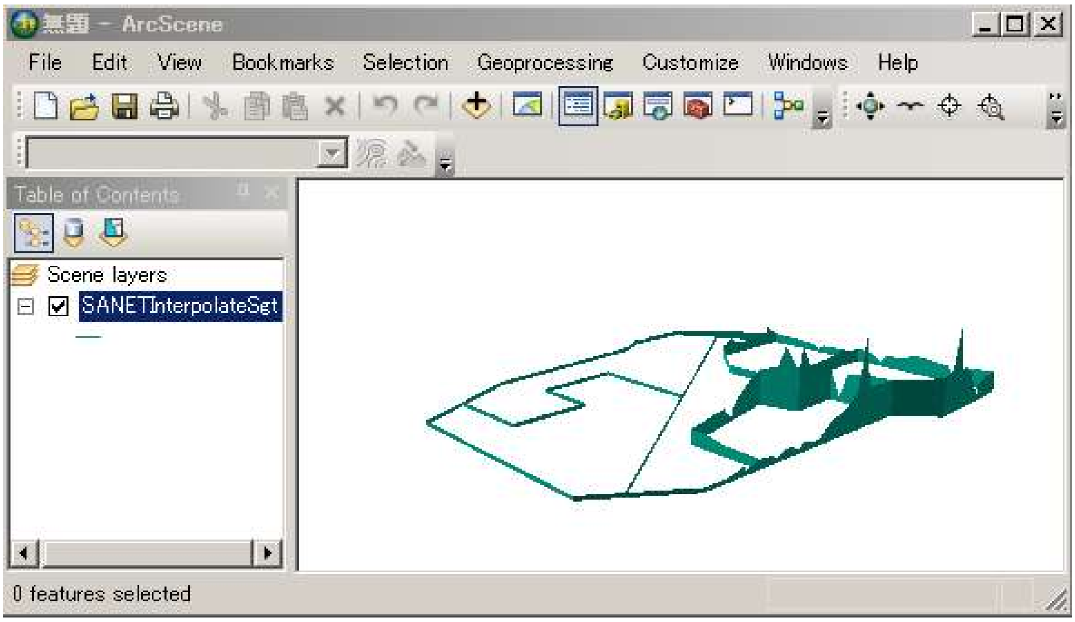
Finally, you get the below 3D presentation of Kernel Density



- Tool 04: Interpolation

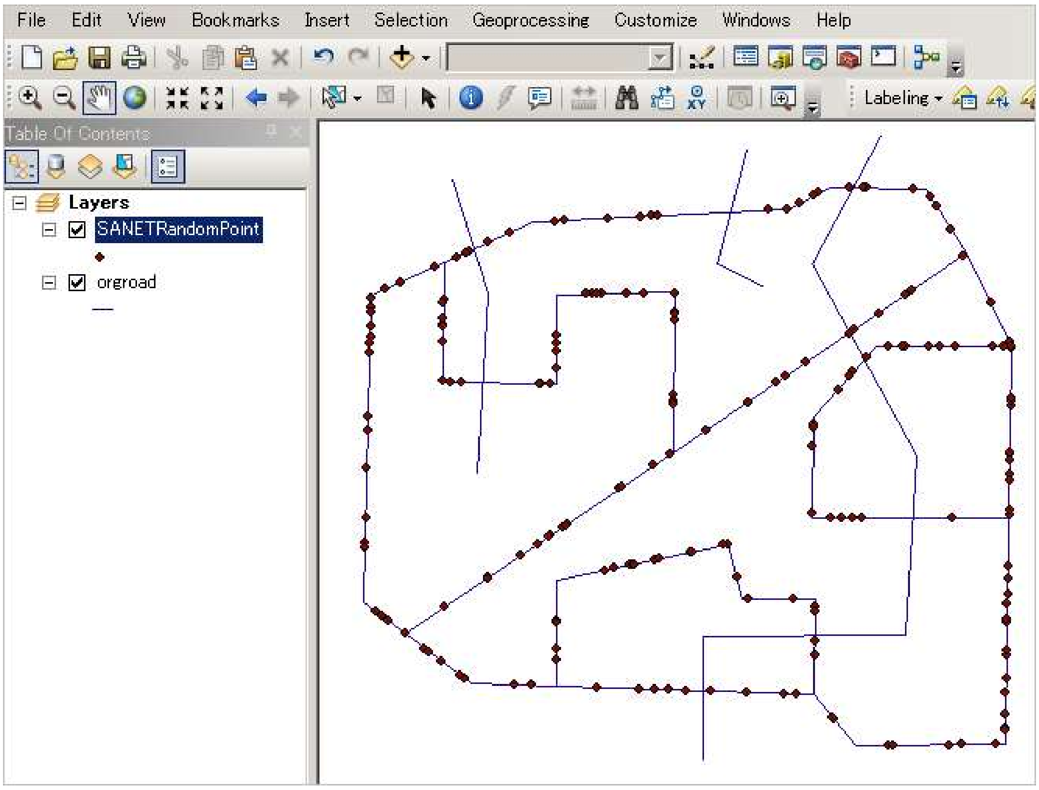


ArcScene will give you the 3D presentation of Interpolation in the same manner as you did with Tool 03 ; Kernel Density Estimation.



Here, the elevation value is set to 1.5.

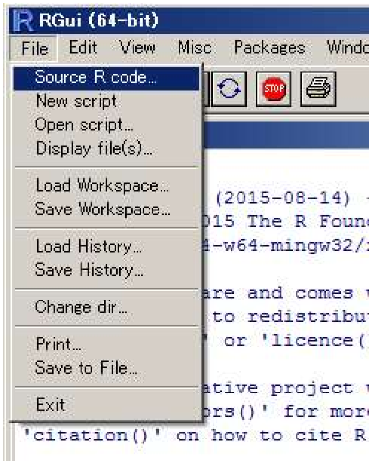
- Tool06:Random points generator



5.2 Visualize the results on R

Launch R program, and click “Files”.

Then the following window appears.

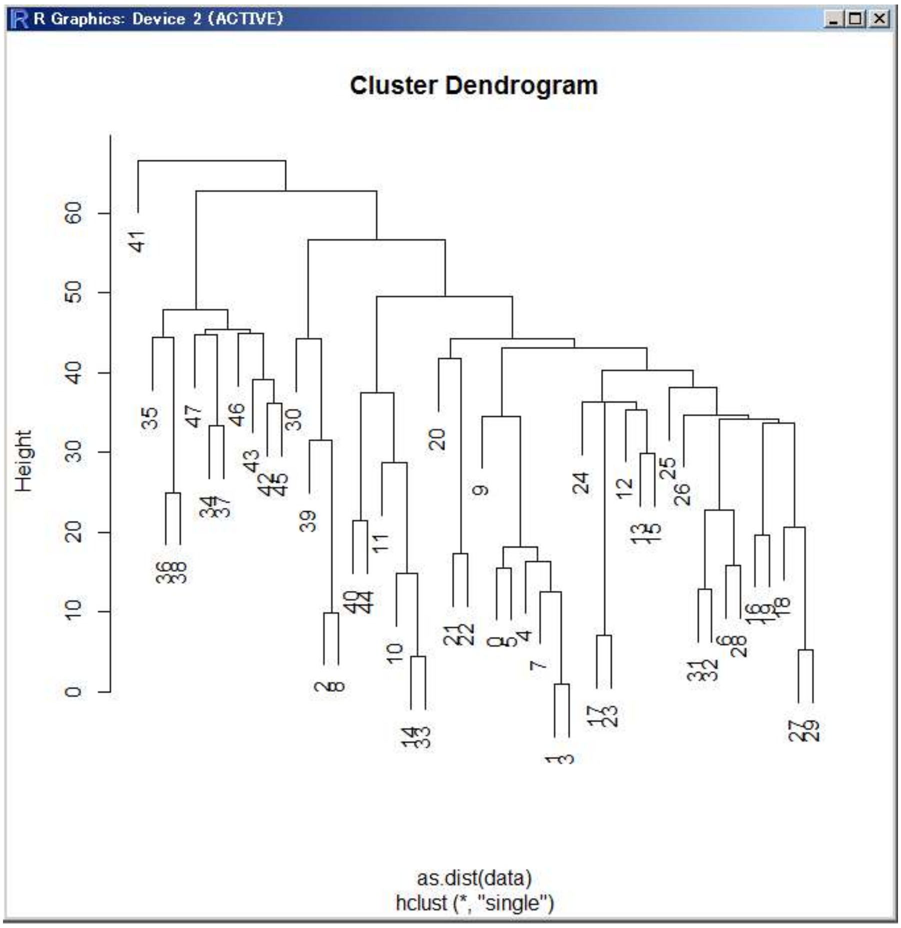


Click “Source R code”, select the resulting output file, and open it.

- Tool 05: Point clustering method

SANETClustering.R

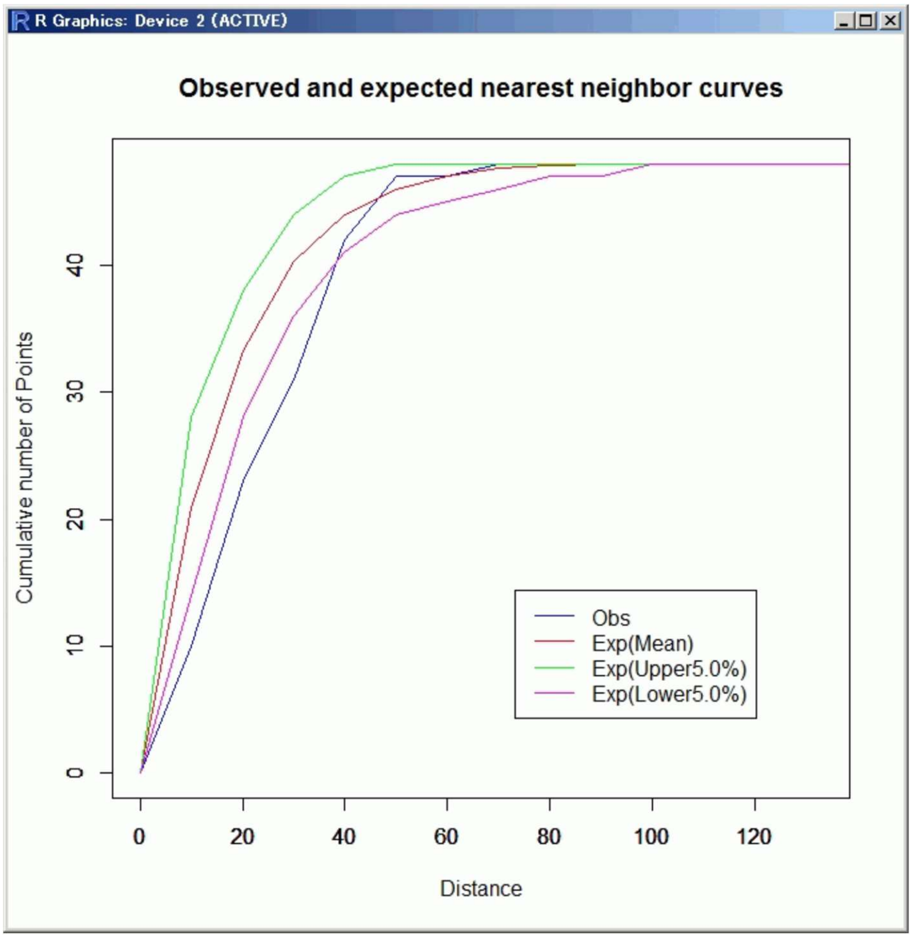
Selecting the SANET yielded R file, the following dendrogram is obtained.



- Tool 10: Global auto nearest neighbor distance method

SANETNNDExpectedValue.R

Selecting the SANET yielded R file, the following graph is obtained.

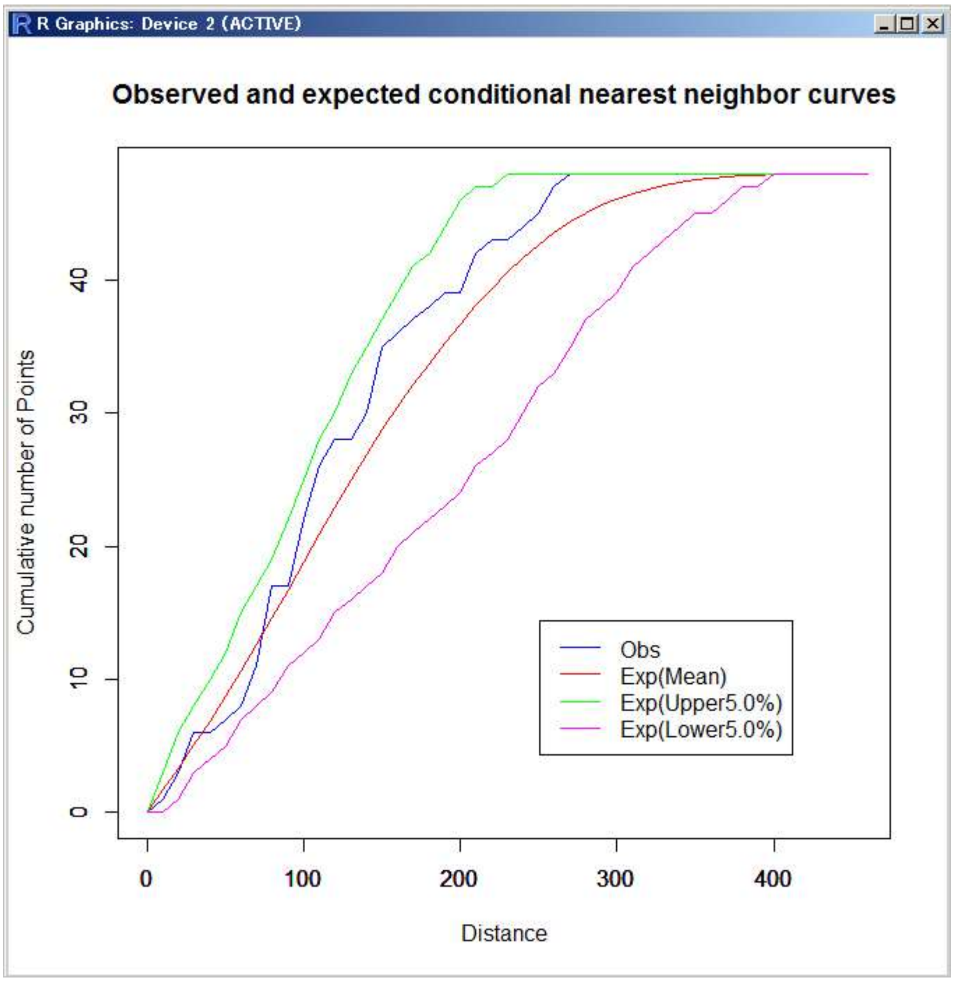


The curves are: the observed curve; the upper and lower envelop curves for the one-sided significance level 5%; and the expected curve under the CSR hypothesis. If the observed curve is in between the upper and lower envelop curves, we cannot reject the CSR hypothesis with 0.95 confidence level.

- Tool 11: Global cross nearest neighbor distance method

SANETCNNDExpectedValue.R

Selecting the SANET yielded R file, the following graph is obtained.

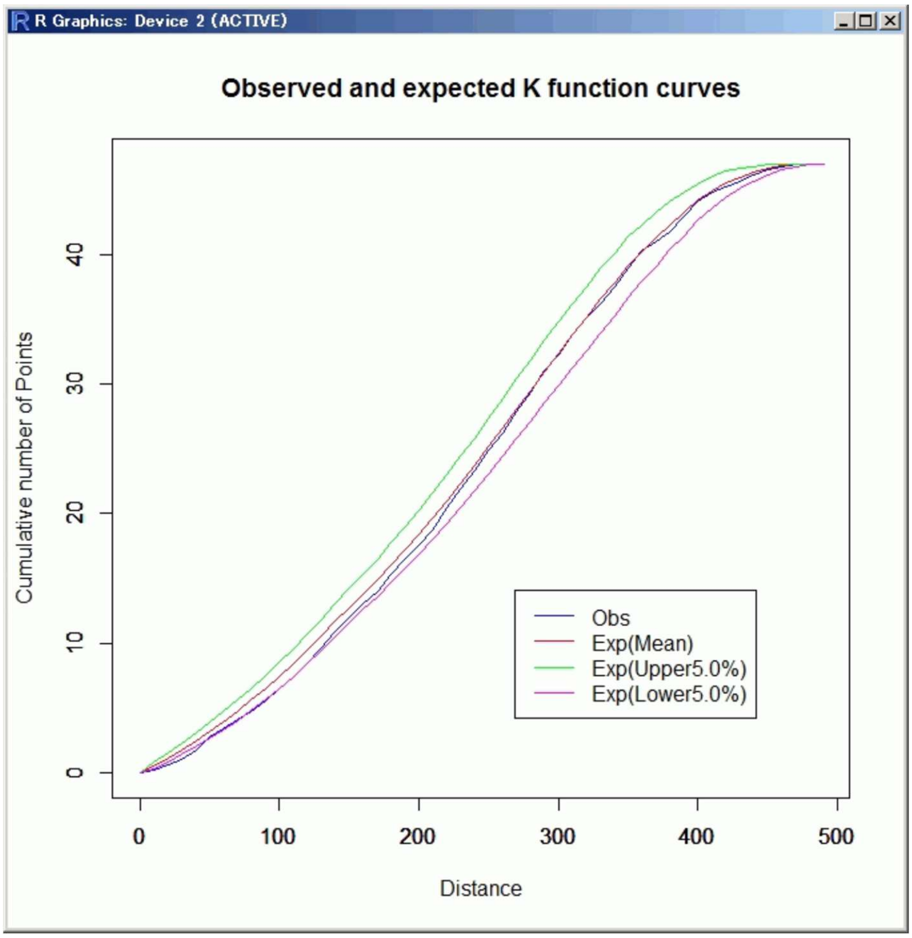


The curves are: the observed curve; the upper and lower envelop curves for significance level 5%; and the expected curve under the CSR hypothesis. If the observed curve is in between the upper and lower envelop curves, we cannot reject the CSR hypothesis with 0.95 confidence level.

- Tool 13: Global auto K function method

SANETKFuncGraphics.R

Selecting the SANET yielded R file, the following graph is obtained.

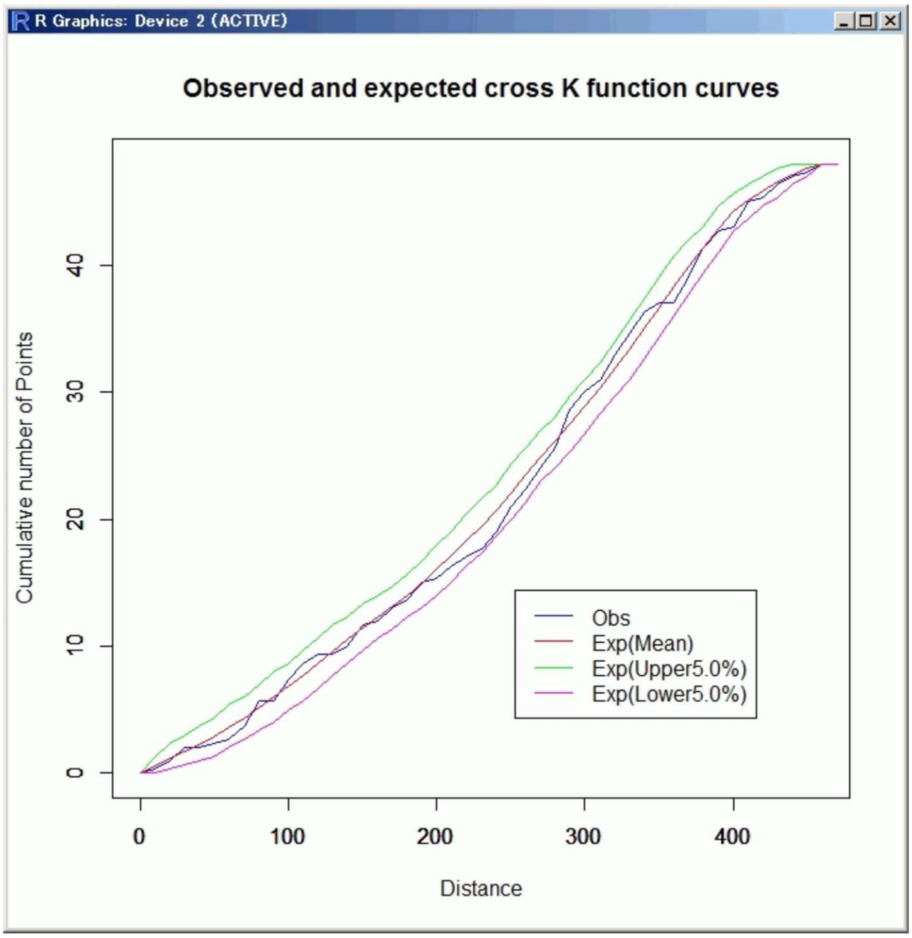


The blue curve indicates the observed curve;
the red curve indicates the mean value under the CRS hypothesis;
the green and pink curves are, respectively, the upper and lower envelop curves under the CSR hypothesis.

- Tool 14: Global cross K function method

SANETCKFuncGraphics.R

Selecting the SANET yielded R file, the following graph is obtained.

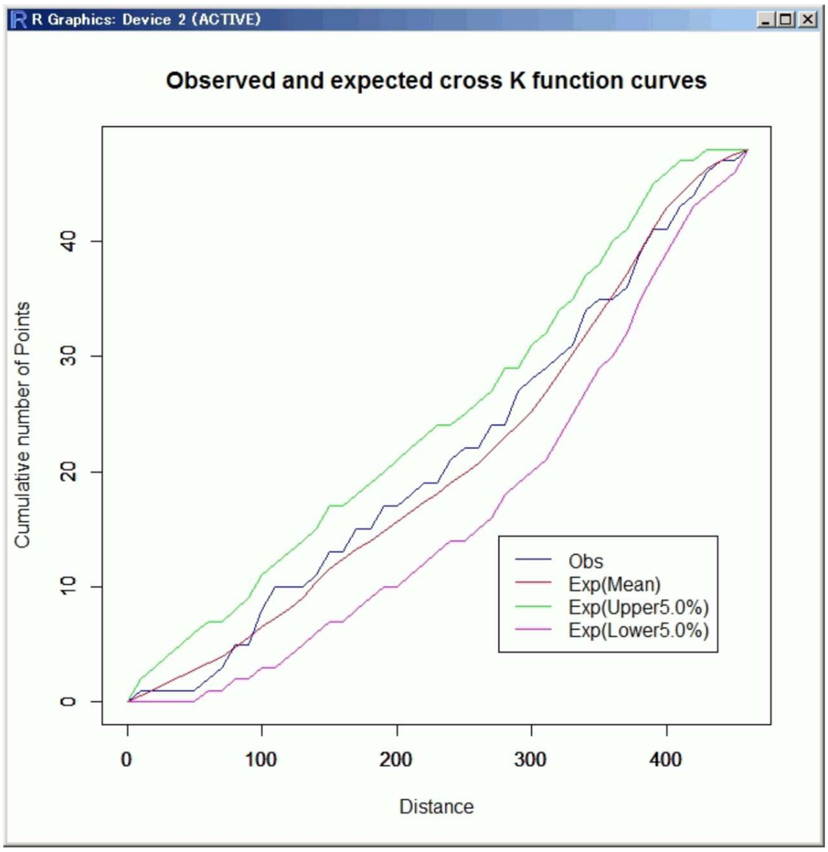


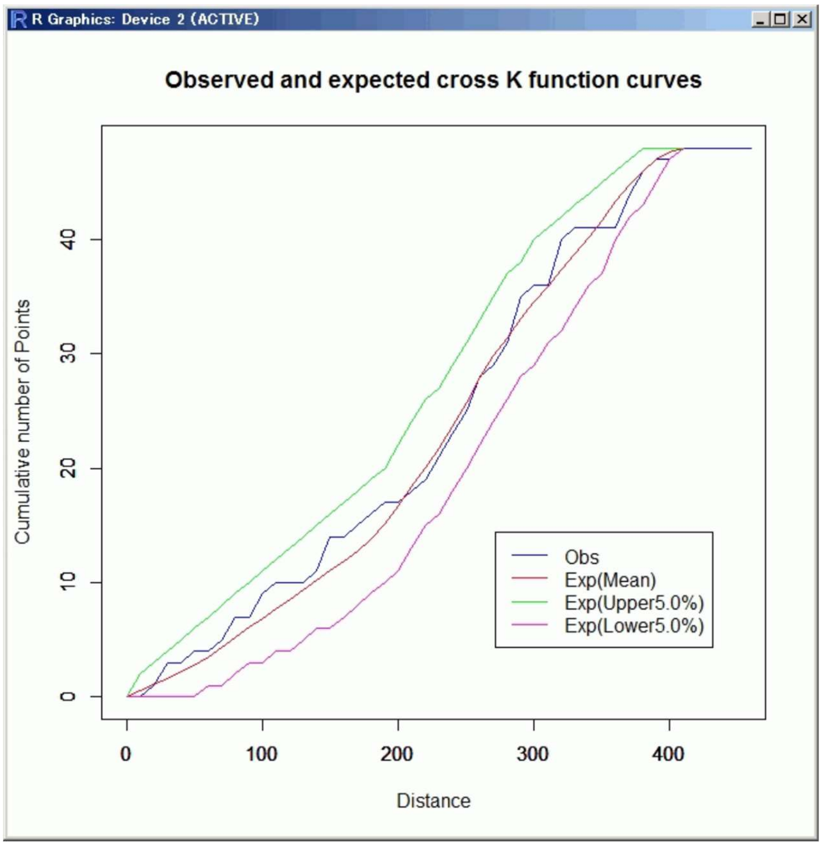
The blue curve indicates the observed curve;
The red curve indicates the mean value under the CRS hypothesis;
The green and pink curves are, respectively, the upper and lower envelop curves under the CSR hypothesis.

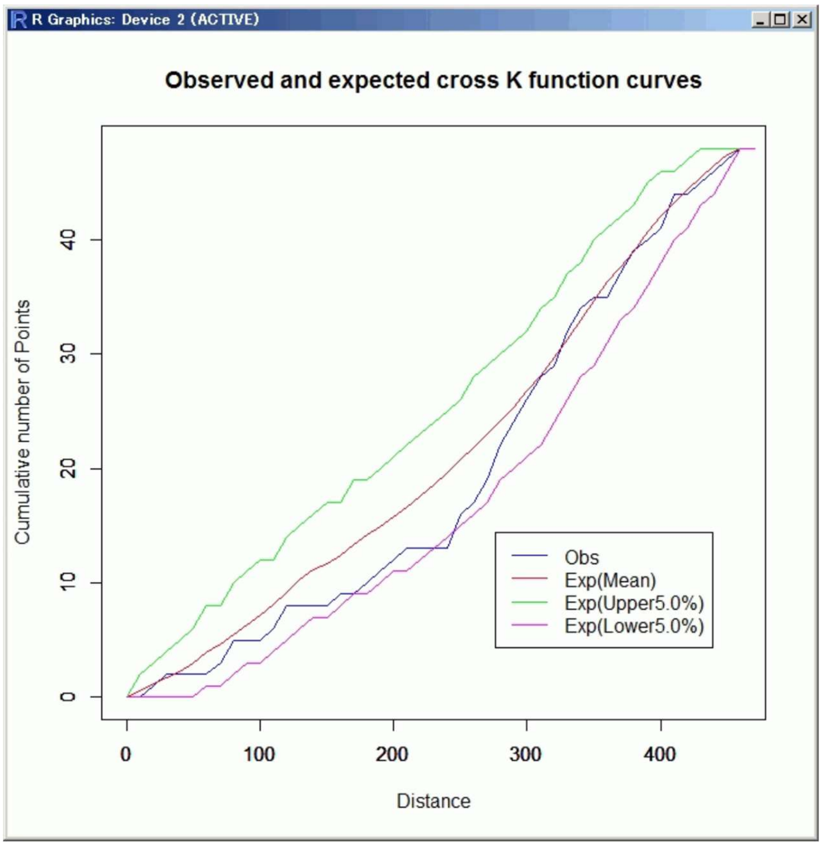
- Tool 15: Local cross K function method

SANETLocalCKFuncGraphics.R

Selecting the SANET yielded R file, the following graph is obtained.


R Graphics: Device 2 (ACTIVE)
Observed and expected cross K function curves
Cumulative number of Points
0
10
20
30
40
0
100
200
300
400
Distance
Obs
Exp(Mean)
Exp(Upper5.0%)
Exp(Lower5.0%)


R Graphics: Device 2 (ACTIVE)
Observed and expected cross K function curves
Cumulative number of Points
0
10
20
30
40
0
100
200
300
400
Distance
Obs
Exp(Mean)
Exp(Upper5.0%)
Exp(Lower5.0%)

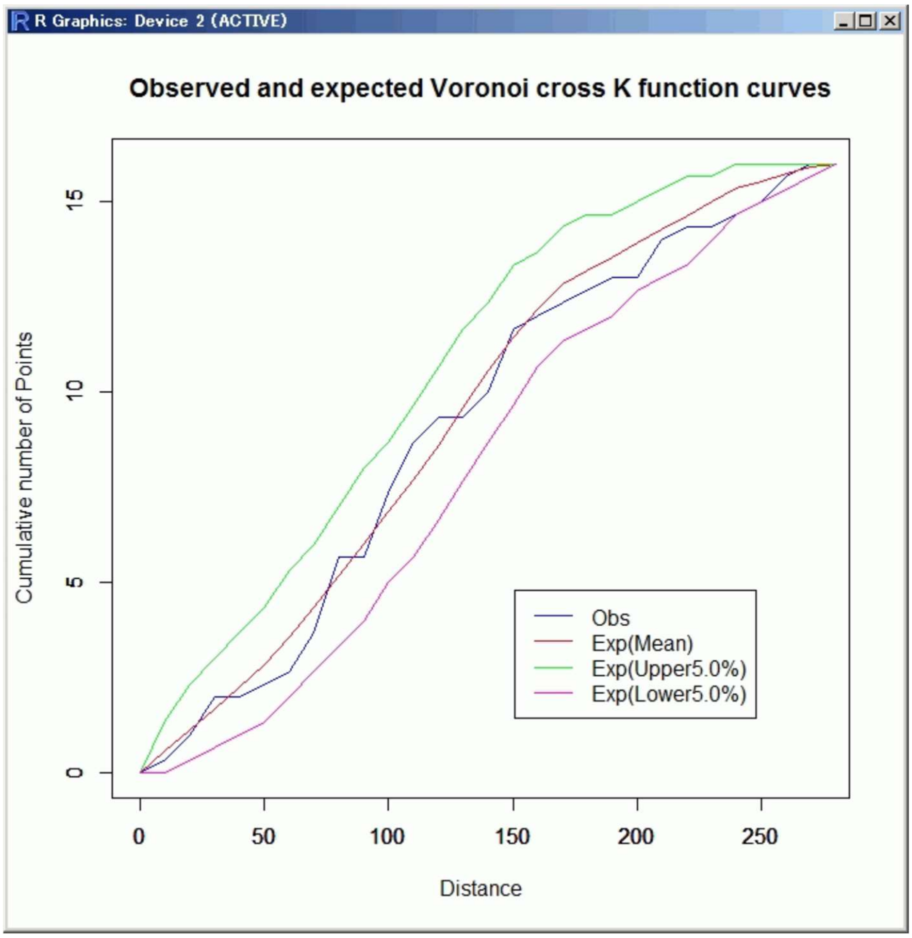


The blue curve indicates the observed curve;
the red curve indicates the mean value under the CRS hypothesis;
the green and pink curves are, respectively, the upper and lower envelop curves under the CSR hypothesis.

- Tool 16: Voronoi cross K function method

SANETVCKFuncGraphics.R

Selecting the SANET yielded R file, the following graph is obtained.



The blue curve indicates the observed curve;
the red curve indicates the mean value under the CRS hypothesis;
the green and pink curves are, respectively, the upper and lower envelop curves under the CSR hypothesis.

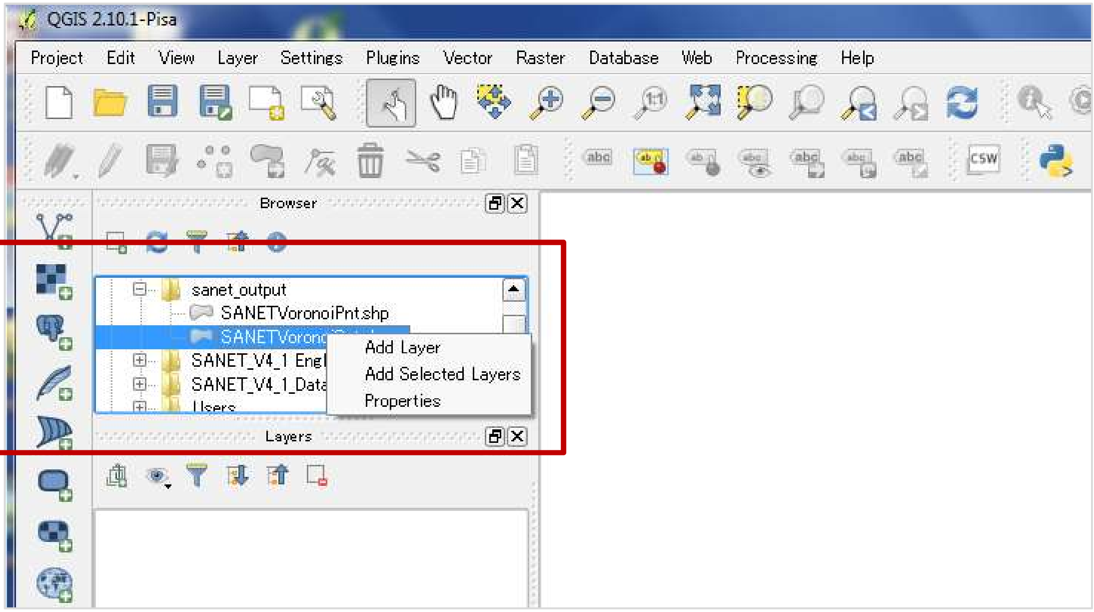
5.3 Visualize the results on QGIS

How to view SANET results on QGIS is shown here using SANETVoronoi Shapefile as an example.

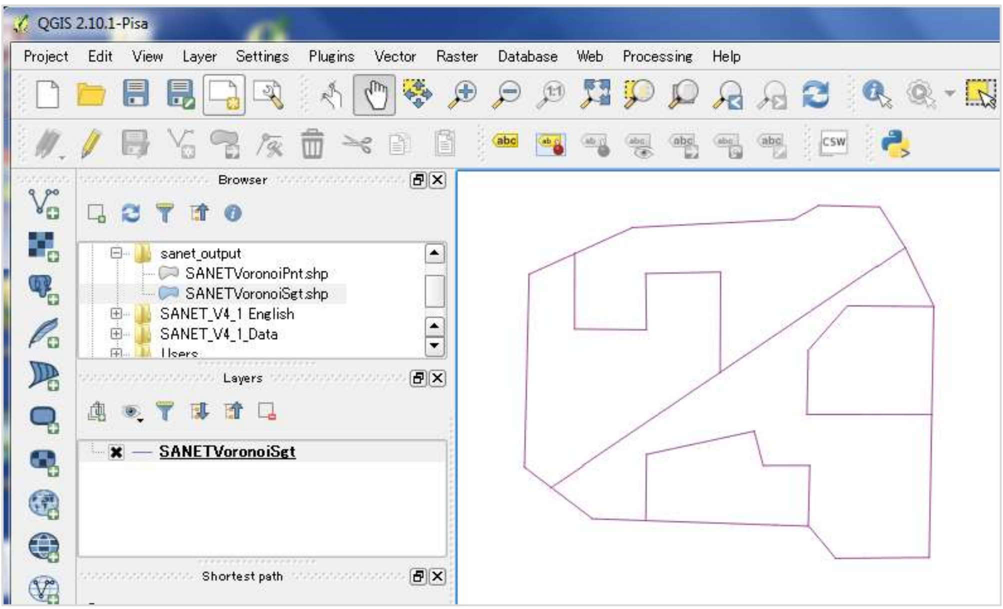
If you have already downloaded and installed QGIS on your PC, just launch QGIS desktop and add the Shapefile which the SANET yielded.

To add SANETVoronoiSgt file is as follows.

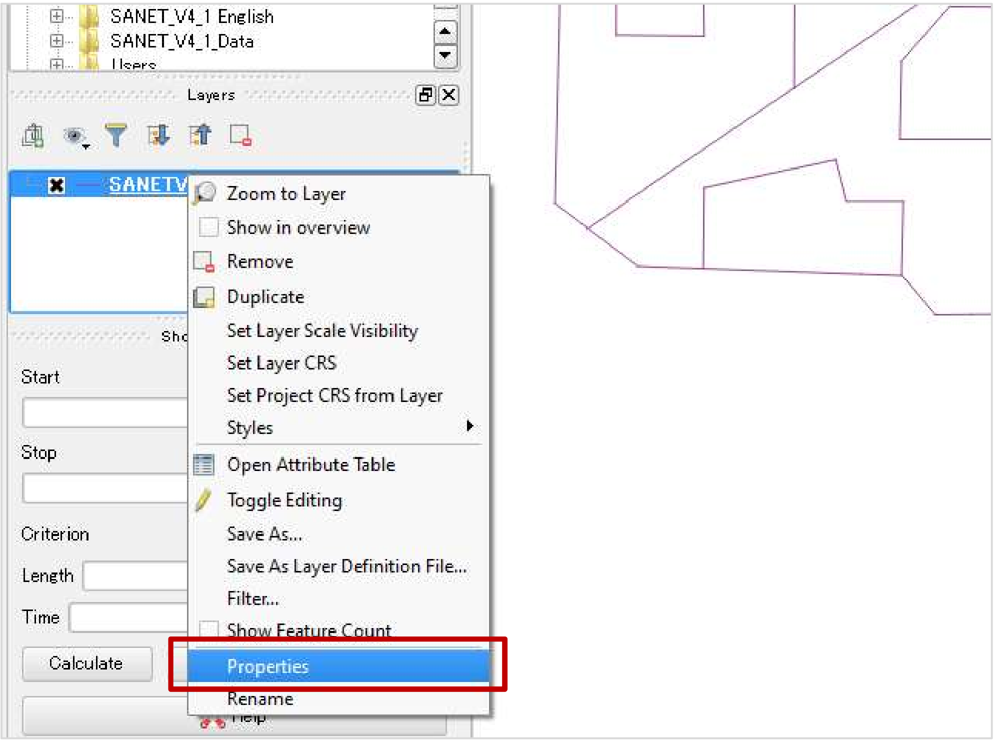
- Select the file in the directory window and right click your mouse button. Then you will find “Add Layer” in you pull-down menu to select.



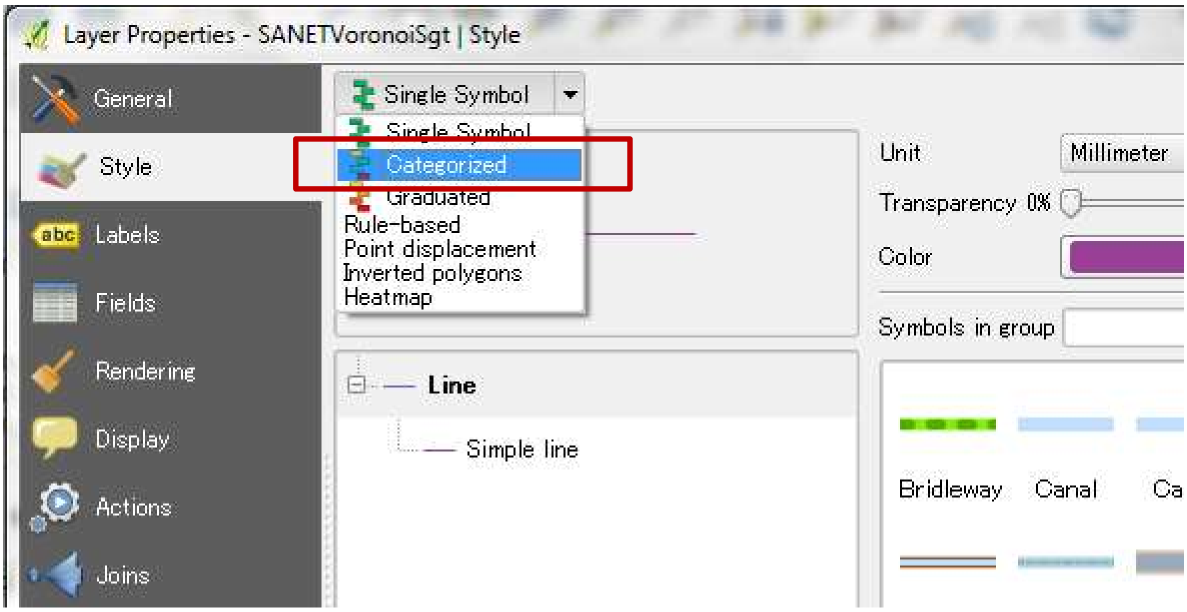
- After selecting the coordinate reference system of your Shapefile, you will get below image.



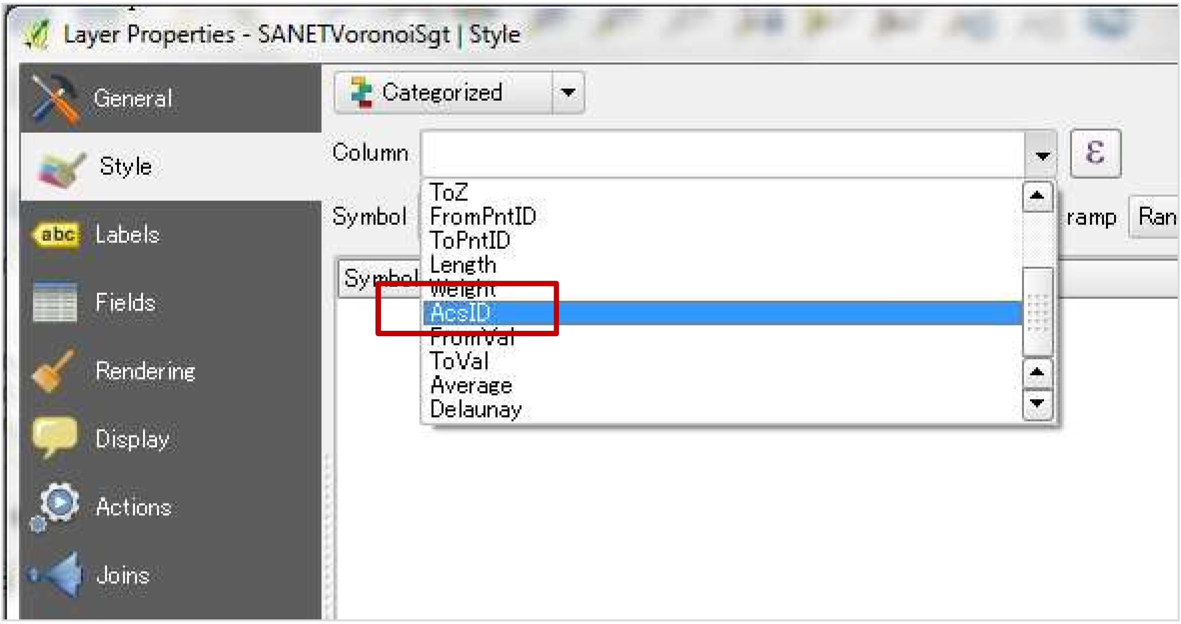
- In order to view the SANET results of voronoi diagram, open the ”Properties” window.



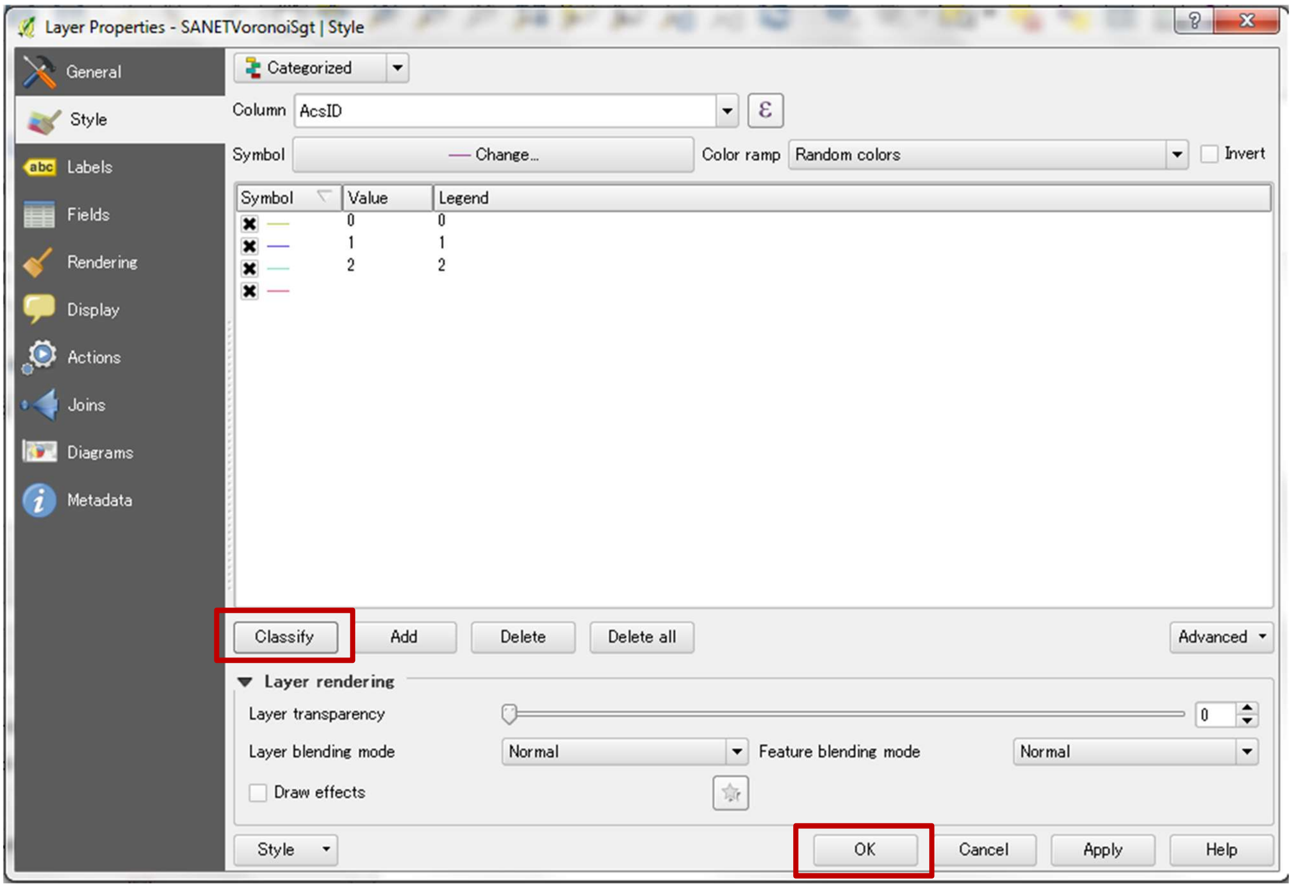
- In your Layer Properties window, select “Categorized”.



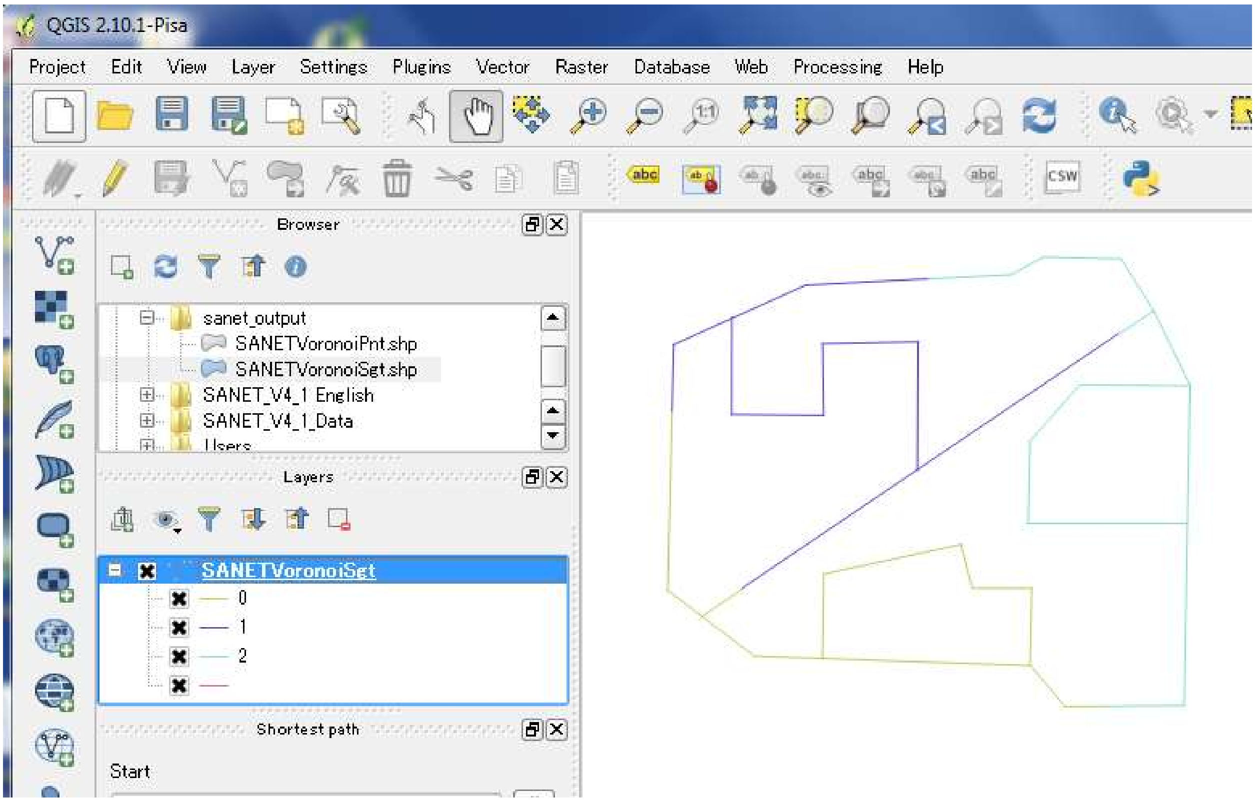
- As for Column, choose AcsID as below



- Press "Classify" button and press OK in the property window as below.

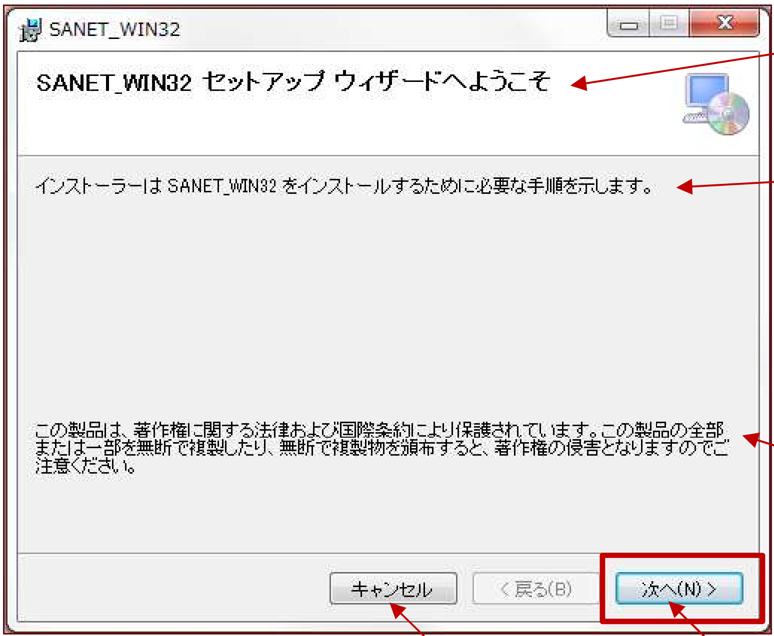


- Then, you will view the voronoi diagram of the SANET output.



◆ About Handling SANET Setup Wizard

1. Unzip the SANET_standalone10_32.zip / SANET_standalone10_64.zip
2. Double click the executable SANET file.
3. Welcome to SANET Setup Wizard



Welcome to SANET Setup Wizard

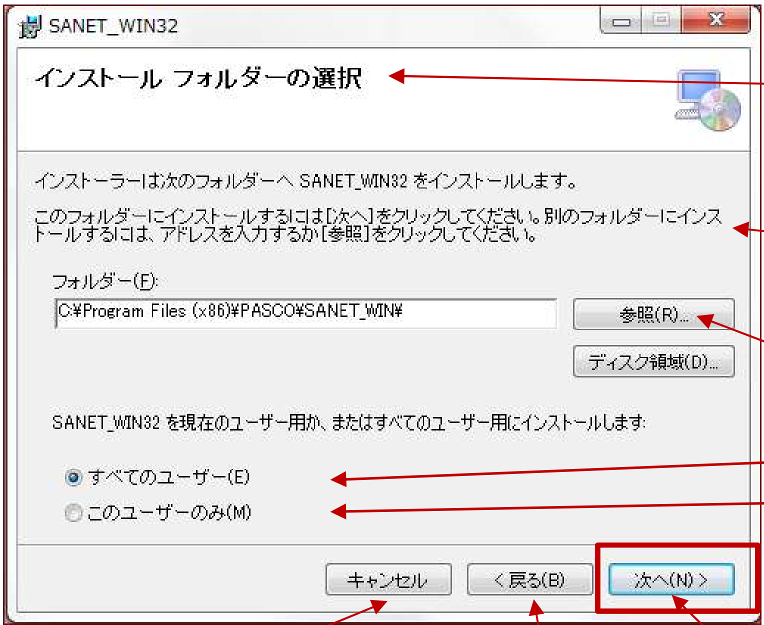
This Setup Wizard guides you the necessary steps to install SANET

This product is protected by law and international treaties on copyright. Duplicate all or part of this product without prior permission, and to distribute copies without permission, please note that it is a violation of copyright.

Cancel

NEXT

4. Select a folder to install SANET



Select a folder to install SANET

If you use a folder besides this default one, put the address below

Browse

For all users (E)

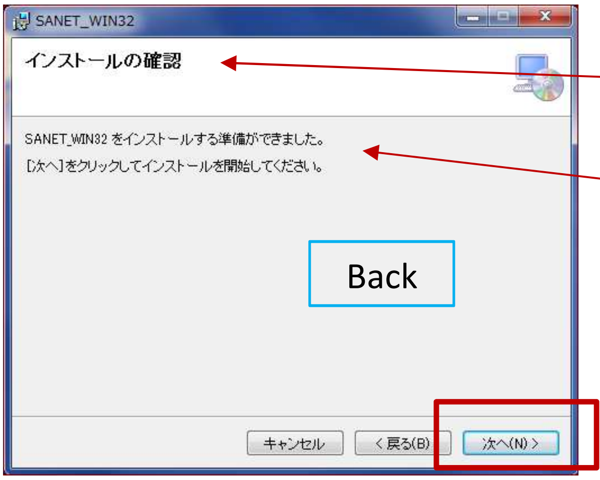
For only this user (M)

Cancel

Back

NEXT

5. Click to start installing SANET



6.

Confirm installation

Prepared to install SANET_WIN32
Please click **NEXT** to start the installation

NEXT

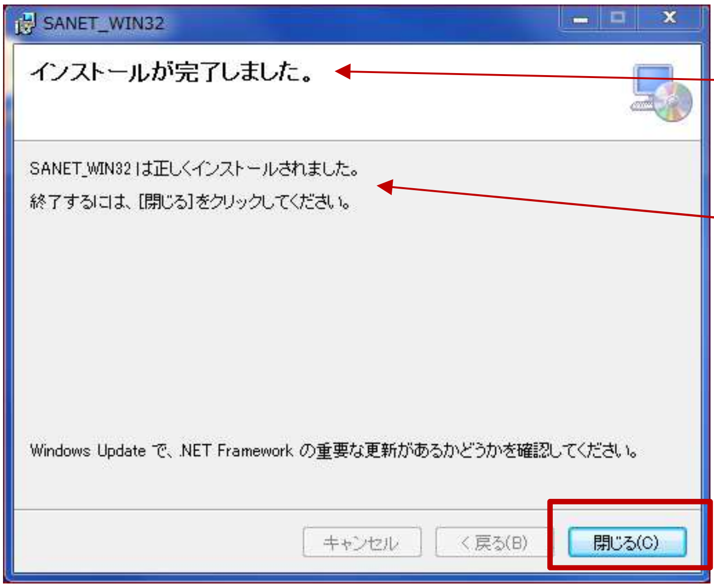
7. Installing SANET

Installing SANET

Please wait

Cancel

8. Completed installation



Completing installation on your computer

SANET_WIN32 has been successfully installed
Click **CLOSE** to finish theWizard

CLOSE

6 Manuals

6.1 Procedure for generating ‘buffer sub-networks’ of a generator point set using QGIS with GRASS

The ‘buffer subnetworks’ of a generator point set on a network means: for a given network (the gray and red line segments in Figure 1) and a given generator point set (the orange circles in Figure 1), the shortest path distance from any points on ‘buffer subnetworks’ (red line segments in Figure 1) to the nearest point in the generator point set is less than or equal to a given shortest path distance x (x=200m in Figure 1). An example is illustrated in Figure 1.

Figure 1: Buffer subnetworks (the red line segments; the orange circles are generator points)

1) Open QGIS 3.x with GRASS 7.x (x represents any number of the version of the software).

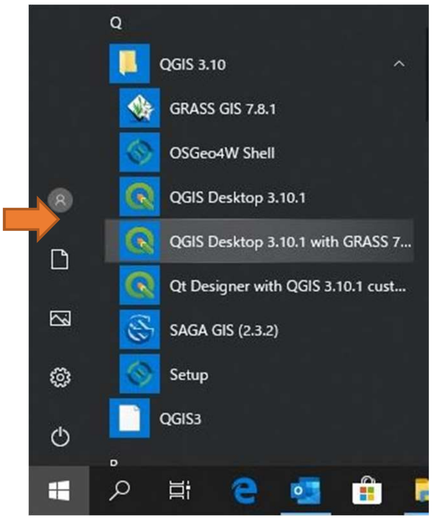


Figure 2

2) Click “New Project” icon (Ctrl+N for short-cut).

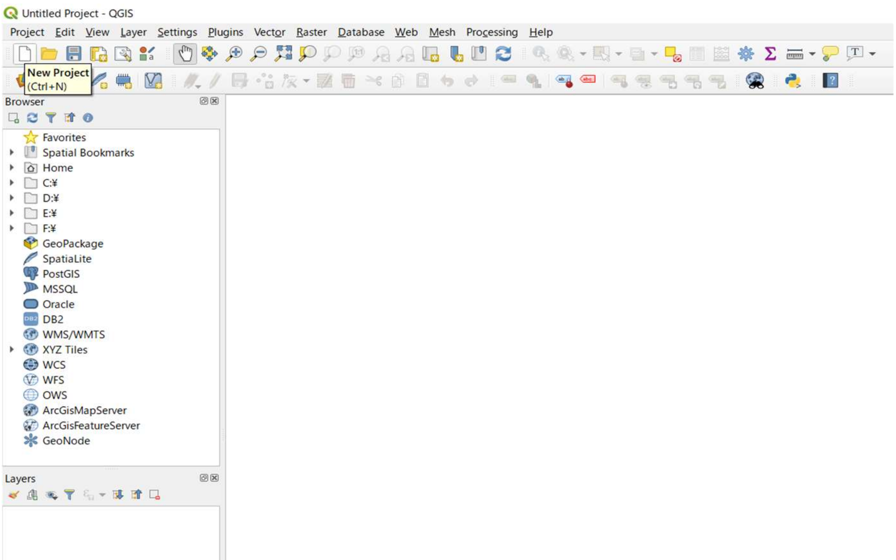


Figure 3

Then the window in Figure 4 appears.

3) Click “Open Data Source Manager” (Ctrl+L).

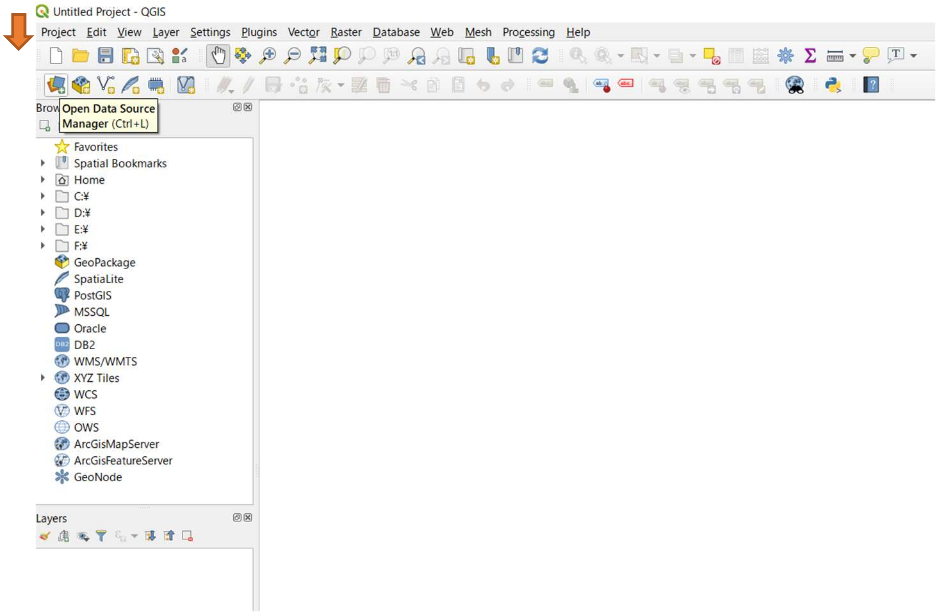


Figure 4

Then the window in Figure 5 appears.

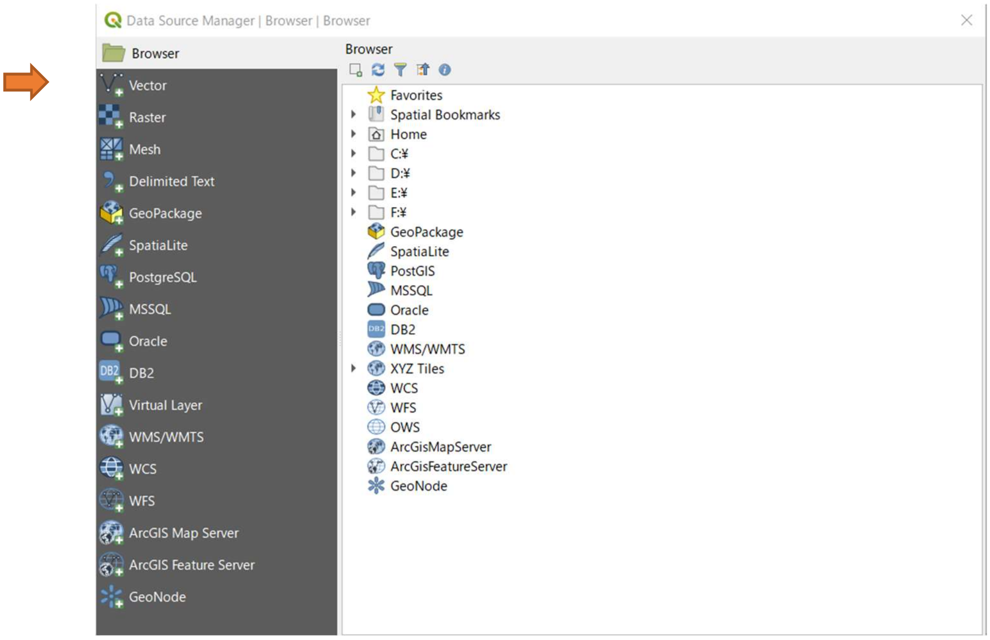


Figure 5

4) Select “Vector” in Figure 5.

Then the window in Figure 6 appears.

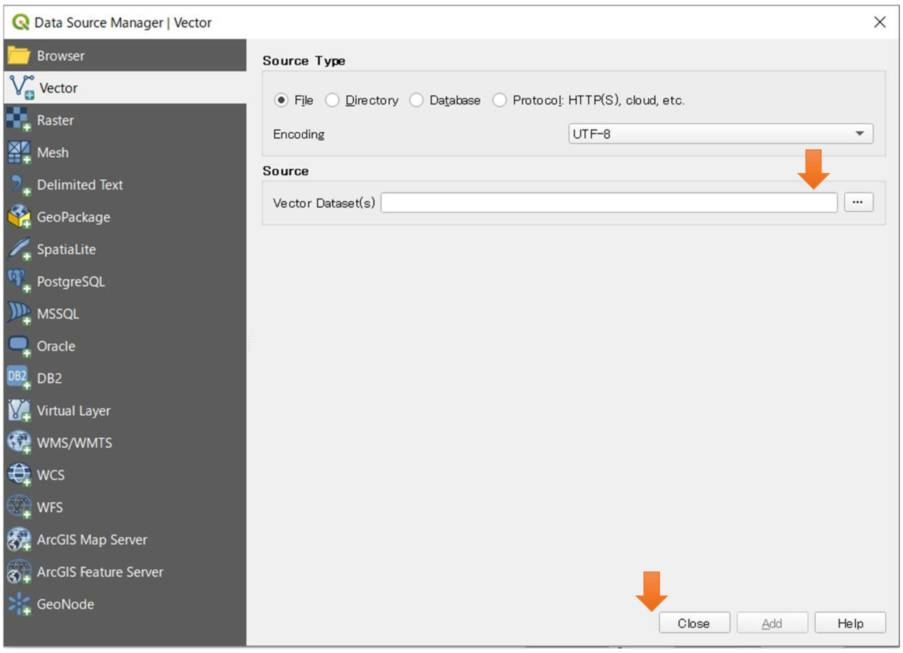


Figure 6

5) Click “Browse” icon,  in Figure 6.

Choose a shp file of network.

“Add” a shp file of a set of points that generates buffers.

Click “Add”.

If you see the files on the map canvas, click “Close” in Figure 6.

The map in Figure 7 is a sample of the screenshot after adding the files.

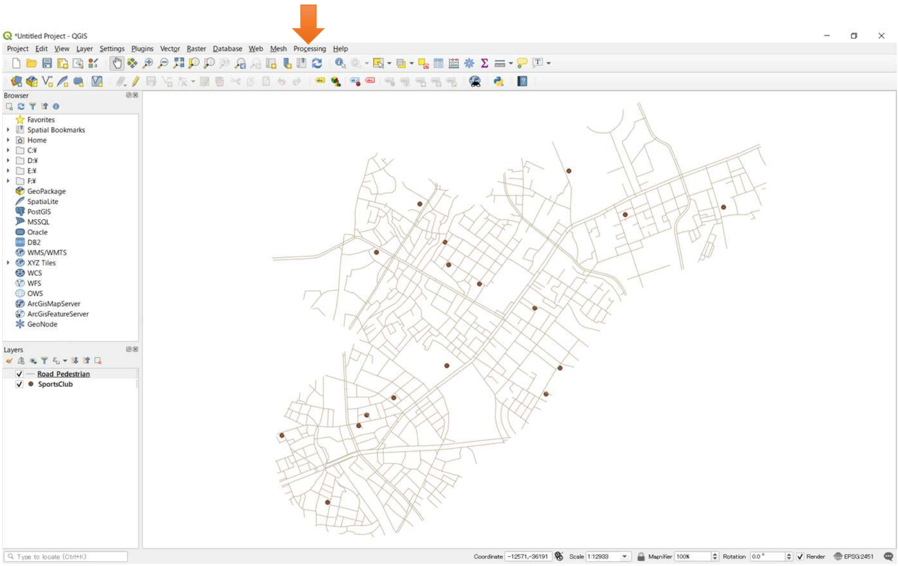


Figure 7

6) Select “Processing” > “Toolbox” indicated by the arrow in Figure 7.
Then, Processing Toolbox window in Figure 8 appears.
Enter v.net.iso in the search box and double click the tool name appeared in the result of search as in Figure 8.

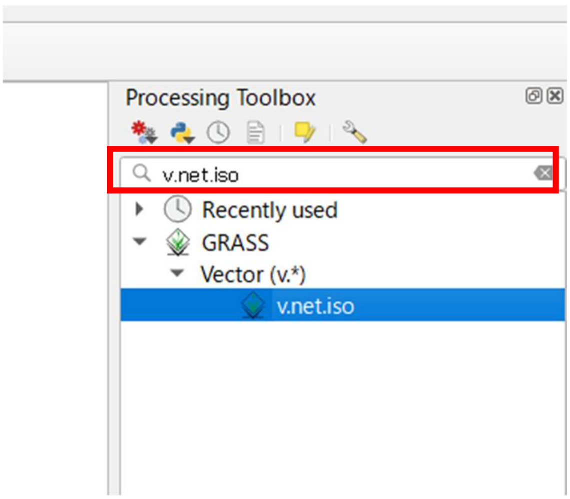


Figure 8

7) Select a network data to “input vector line layer” (in Figure 9) and
a set of generator points data to “centers point layer” (in Figure 9).

Set the value of “threshold for connecting centers to the network” and ”costs for isolines” (in Figure 9).

The costs for isolines means the list of buffer distances that you want to create; noted that you should use a comma to separate values (e.g., 100, 200, 300).

After entering those parameters, click “Run” (the arrow in Figure 9).

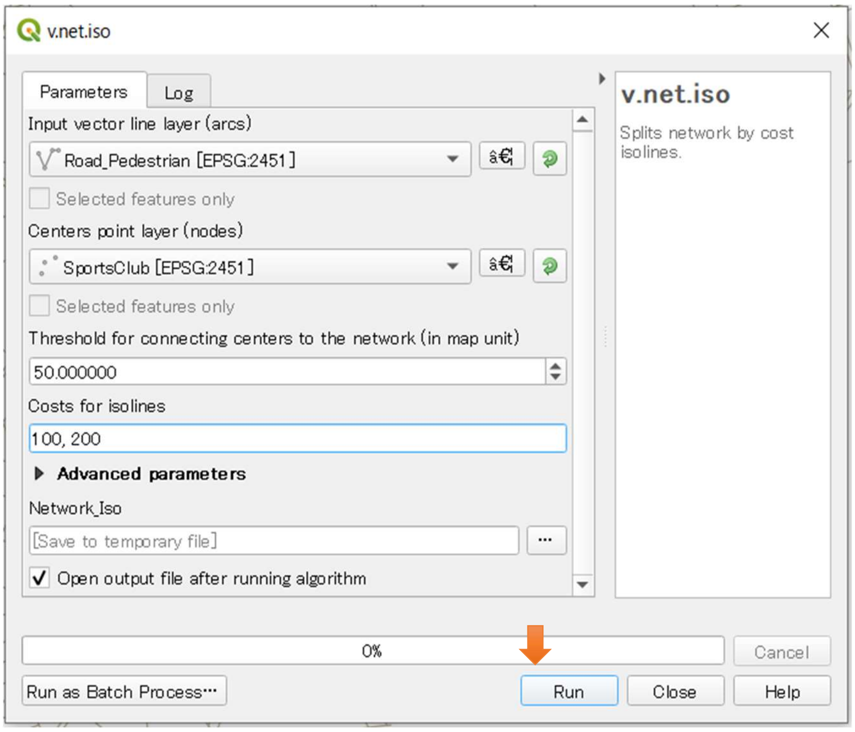


Figure 9

8) Completed the task, click “Close” (in Figure 9) to close the v.net.iso window.

9) Right click the output temporary file (the orange arrow in Figure 10), and click “Property” (the green arrow in Figure 10).

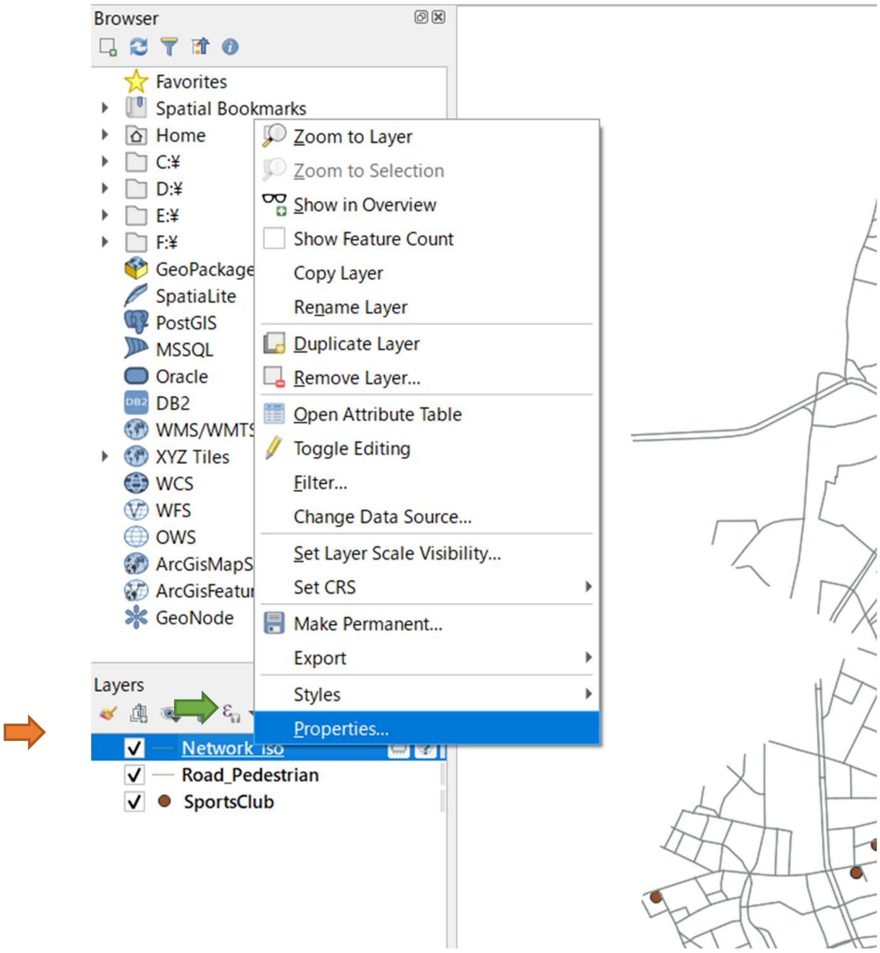


Figure 10

10) Select “Symbology” tab (the orange arrow in Figure 11), and choose “Categorized” from the box on the top (the green arrow in Figure 11).

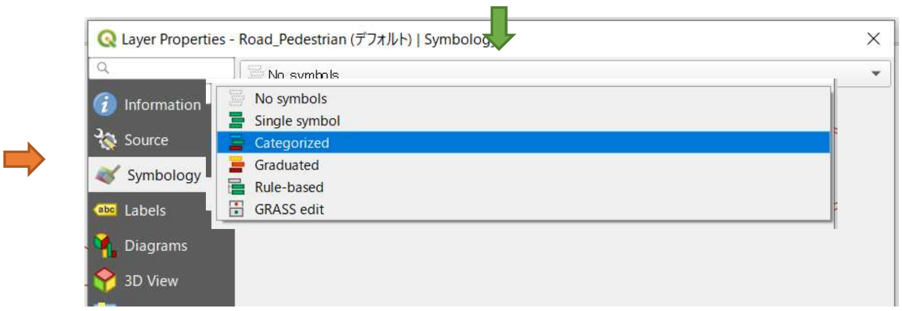


Figure 11

11) Set “cat” in Value (Figure 12).

Click “Classify” (the orange arrow in Figure 12).
Set the rules for symbols (e.g., color and width) as you like.
Click “OK” (the green arrow in Figure 12).

Note that “cat = 1” represents line segments form the buffer subnetwork in which the shortest path distance from the nearest generator point is less than 100 m; “cat = 2” is between 100 m and 200 m; “cat =3” is more than 200 m.

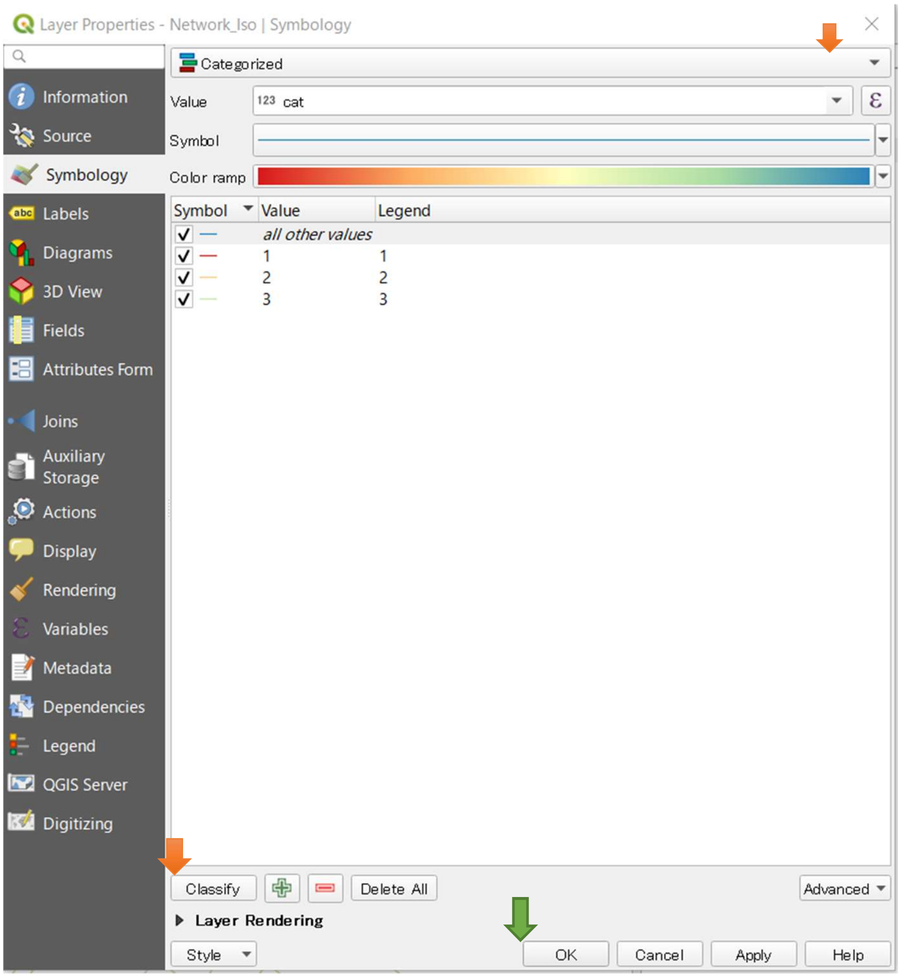


Figure 12

You obtain the map in Figure 13, showing multi-distance buffer subnetwork.

Figure 13

12) If you want to save the temporary file as a permanent shape file, right click the output temporary file (the orange arrow in Figure 14), and
click “Make Permanent” (the green arrow in Figure 14).

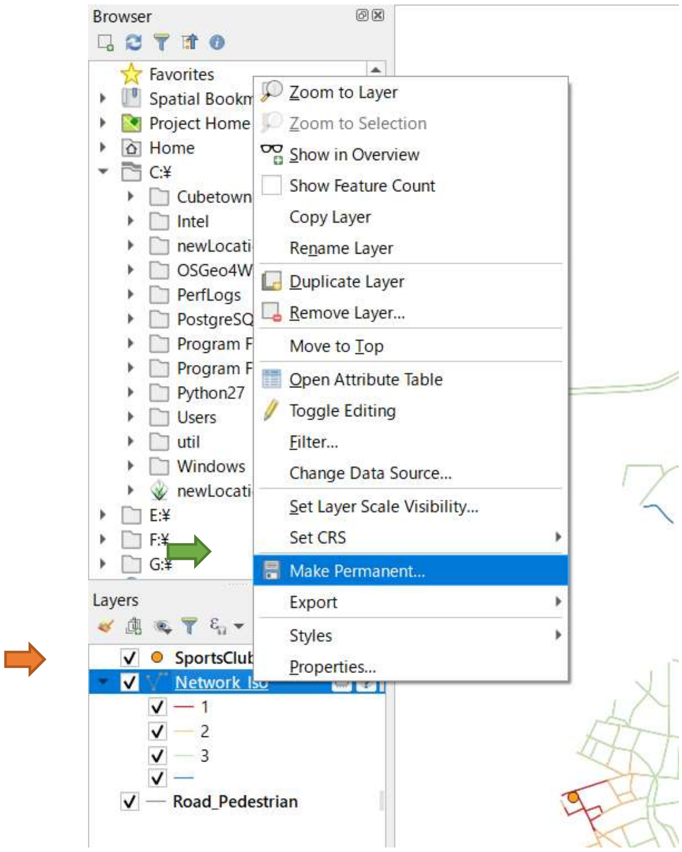


Figure 14

13) Click “Browse” icon,  in Figure 15.

Specify the destination to save and click “OK”.

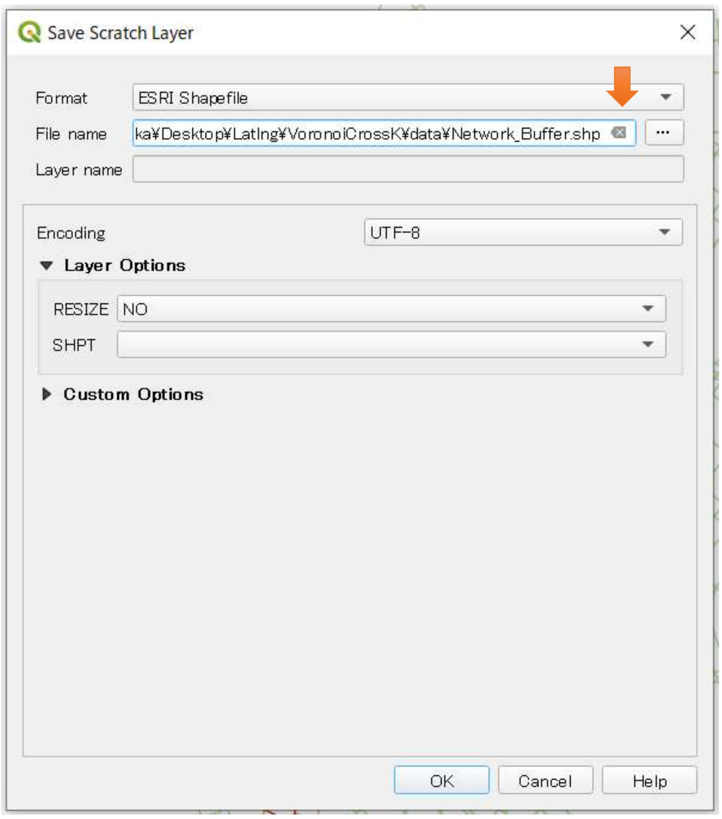


Figure 15

6.2 Procedure for counting the number of points on 'buffer sub-networks' of a generator point set

Note that from the resulting $n(x_{i+1}|P_{Aj})$, the number of points on the line segments between x_i and x_{i+1} is obtained.

To state the above computation precisely, let $N(x_i, x_{i+1})$ denote the 'buffer-ring sub-networks' of a network N in which the shortest path distance from any points on $N(x_i, x_{i+1})$ to the nearest generator point (Type A) is between x_i and x_{i+1}, where $x_i < x_{i+1}$, $x_{i+1} - x_i = c$ for $i = 0, 1, 2, \ldots$ Note that c is a parameter determined by a user. In an illustrative example here, $c =$ 10m. Visually, $N(0, x_{i+1})$ is obtained from the 'buffer sub-networks' in Figure 1 in Section 6.1, where $x_{i+1} = 200$. Therefore, 'buffer-ring sub-networks' $N(x_i, x_{i+1})$ is obtained from $N(0, x_{i+1})$¥ $N(0, x_i)$, i.e., the compliment of $N(0, x_i)$ with respect to $N(0, x_{i+1})$.

To state the above computation precisely, given a set of generator points on a given network N, let $N(x_i, x_{i+1})$ denote the sub-network of N in which the shortest path distance from any points in $N(x_i, x_{i+1})$ to the nearest generator point in given generator points is between x_i and x_{i+1}, where $x_i < x_{i+1}$ $x_{i+1} - x_i = c$ for $i = 0, 1, 2, \ldots$ Note that c is a parameter determined by a user. In an illustrative example here, $c =$ 10m.

Let $L(x_i, x_{i+1})$ be the total length of line segments constituting the sub-network $N(x_i, x_{i+1})$, and L is the total length of line segments constituting the whole network N.

In terms of the above notations, this manual describes the procedure for computing $L(x_i, x_{i+1})$.

Let l_j be link j of a given network, and $d(\partial l_j^-)$ refers to the shortest path distance between the start node of link *j* and the nearest generating point from link *j*; $d(\partial l_j^+)$ is the shortest path distance between the end node of link j and the nearest generating point.

Step 1

Run the “Voronoi diagram” tool of SANET. In more detail, see Section 4.1 Tool 01: Voronoi diagrams in User Guide/Manual for SANET Standalone.

Step 2

Copy “SANETVoronoiPnt.dbf” and “SANETVoronoiSgt.dbf” on one folder.

Step 3

Open these files in Excel.

* You may follow the four steps described below, or you may use an alternative method as you like.

3-1) Open Excel.

3-2) Select File, “Open” (the orange arrow in Figure 1).

3-3) Click “Browse” and select the folder in which you saved the files (the green arrow in Figure 1).

3-4) In the “Files of Type” drop down box, select: “All Files” (the blue arrow in Figure 1), and then choose “SANETVoronoiPnt.dbf” and “SANETVoronoiSgt.dbf”, and click “Open”.

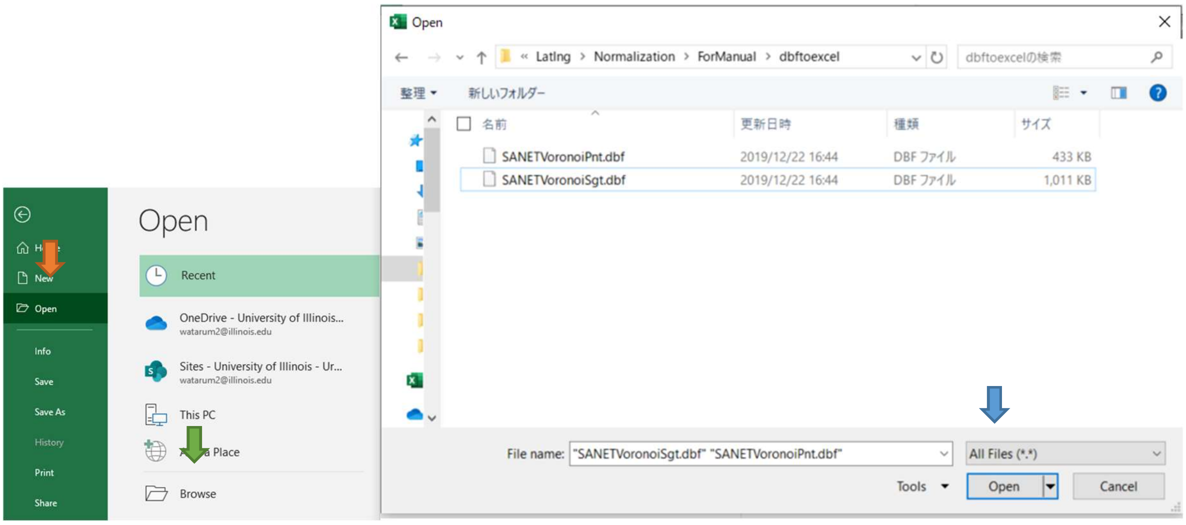


Figure 1

Step 4

Copy these two sheets on a new book (for instance, Book1). Figure2 is a sample of the screenshot after step 4).

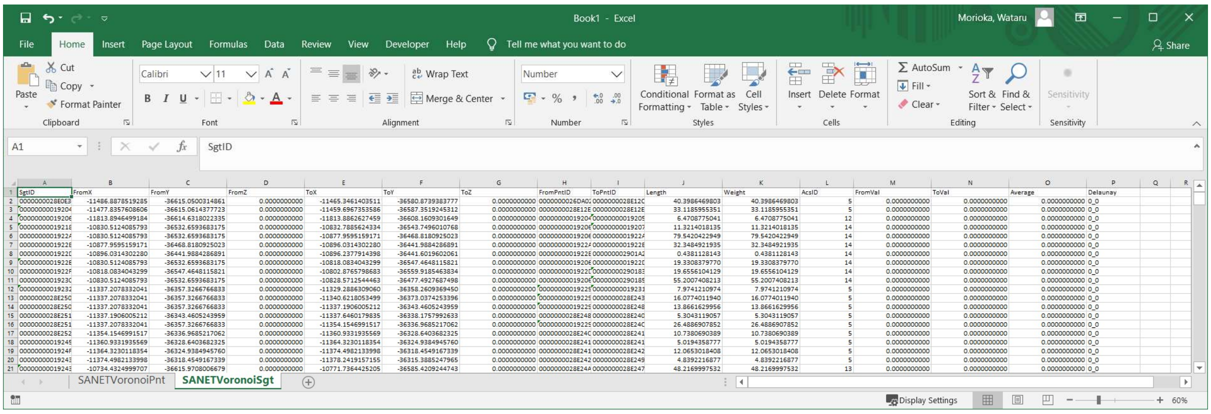


Figure 2

Step 5

In the cell Q2 (column Q and row 2) of “SANETVoronoiSgt” sheet, enter the formula to compute distance from the generator to “FromPnt” of the segment:

=VLOOKUP(H2,SANETVoronoiPnt!$A:$J,8, FALSE)

Hit the enter.

=VLOOKUP(H2,SANETVoronoiPnt!$A:$J,8, FALSE)

L	M	N	O	P	Q
	FromVal	ToVal	Average	Delaunay	
0	0.0000000000	0.0000000000	0.0000000000	0_0	1763.122
0	0.0000000000	0.0000000000	0.0000000000	0_0	
0	0.0000000000	0.0000000000	0.0000000000	0_0	
0	0.0000000000	0.0000000000	0.0000000000	0_0	

Figure 3

Step 6

Copy the formula to the other segments by left-clicking and dragging the small dot at the lower right corner of Q2 down column Q to the end row of the data.

0000000029063F1	00000000250A634	5.432680686	5.432680686	0.000000000000000	0	0	0	0_0	2468.596
000000002906378	0000000028F27F0	4.821205672	4.821205672	0.000000000000000	0	0	0	0_0	434.5112
00000000290643E	0000000028D8FF3	16.88641222	16.88641222	0.000000000000000	0	0	0	0_0	500.996
0000000029064E3	0000000028EC195	2.888190439	2.888190439	0.000000000000000	0	0	0	0_0	1246.478
0000000029064F9	0000000028EBDCD	7.808978939	7.808978939	0.000000000000000	0	0	0	0_0	1391.997
000000002906567	00000000250BA2F	21.81211395	21.81211395	0.000000000000000	0	0	0	0_0	1941.034
0000000029063E6	0000000028F2BA2	15.04838392	15.04838392	0.000000000000000	0	0	0	0_0	1549.718
0000000029065A9	0000000028DA12B	0.914670145	0.914670145	0.000000000000000	0	0	0	0_0	621.6643
00000000290630A	0000000028F2C68	24.47211596	24.47211596	0.000000000000000	0	0	0	0_0	1554.055
000000002906475	000000002622896	37.18872051	37.18872051	0.000000000000000	0	0	0	0_0	1984.225

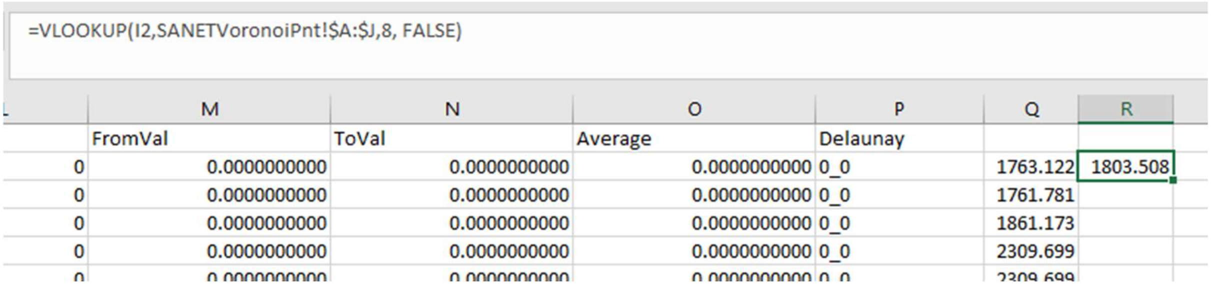
Figure 4

Step 7

In cell R2 of SANETVoronoiSgt, enter the formula to compute distance from the generator to “ToPnt” of the segment:

=VLOOKUP(I2,SANETVoronoiPnt!$A:$J,8, FALSE)

Hit the enter.



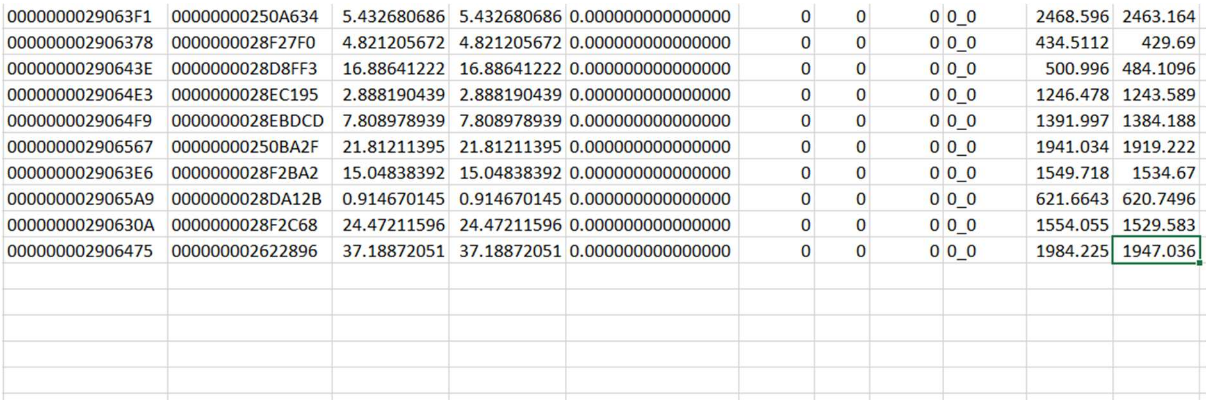
=VLOOKUP(I2,SANETVoronoiPnt!$A:$J,8, FALSE)

L	M	N	O	P	Q	R
	FromVal	ToVal	Average	Delaunay		
0	0.0000000000	0.0000000000	0.0000000000	0_0	1763.122	1803.508
0	0.0000000000	0.0000000000	0.0000000000	0_0	1761.781	
0	0.0000000000	0.0000000000	0.0000000000	0_0	1861.173	
0	0.0000000000	0.0000000000	0.0000000000	0_0	2309.699	
0	0.0000000000	0.0000000000	0.0000000000	0_0	2309.699	

Figure 5

Step 8

Copy the formula to the other segments by left-clicking and dragging the small dot at the lower right corner of R2 down column R to the end row of the data.



0000000029063F1	00000000250A634	5.432680686	5.432680686	0.000000000000000	0	0	0	0_0	2468.596	2463.164
000000002906378	0000000028F27F0	4.821205672	4.821205672	0.000000000000000	0	0	0	0_0	434.5112	429.69
00000000290643E	0000000028D8FF3	16.88641222	16.88641222	0.000000000000000	0	0	0	0_0	500.996	484.1096
0000000029064E3	0000000028EC195	2.888190439	2.888190439	0.000000000000000	0	0	0	0_0	1246.478	1243.589
0000000029064F9	0000000028EBDCD	7.808978939	7.808978939	0.000000000000000	0	0	0	0_0	1391.997	1384.188
000000002906567	00000000250BA2F	21.81211395	21.81211395	0.000000000000000	0	0	0	0_0	1941.034	1919.222
0000000029063E6	0000000028F2BA2	15.04838392	15.04838392	0.000000000000000	0	0	0	0_0	1549.718	1534.67
0000000029065A9	0000000028DA12B	0.914670145	0.914670145	0.000000000000000	0	0	0	0_0	621.6643	620.7496
00000000290630A	0000000028F2C68	24.47211596	24.47211596	0.000000000000000	0	0	0	0_0	1554.055	1529.583
000000002906475	000000002622896	37.18872051	37.18872051	0.000000000000000	0	0	0	0_0	1984.225	1947.036

Figure 6

Step 9

Copy the values of columns Q and R, and paste them to a new sheet.

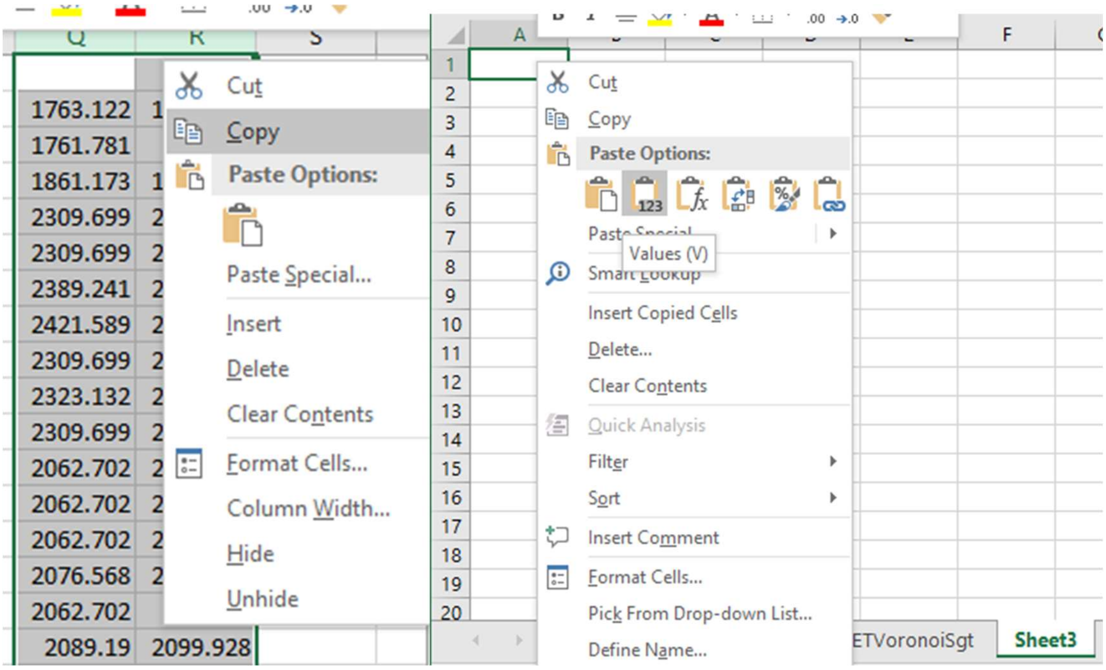


Figure 7

Step 10

Delete Cell A1 and B1 (blank spaces). Note that choosing “Shift cells up” as the option and then clicking “OK”.

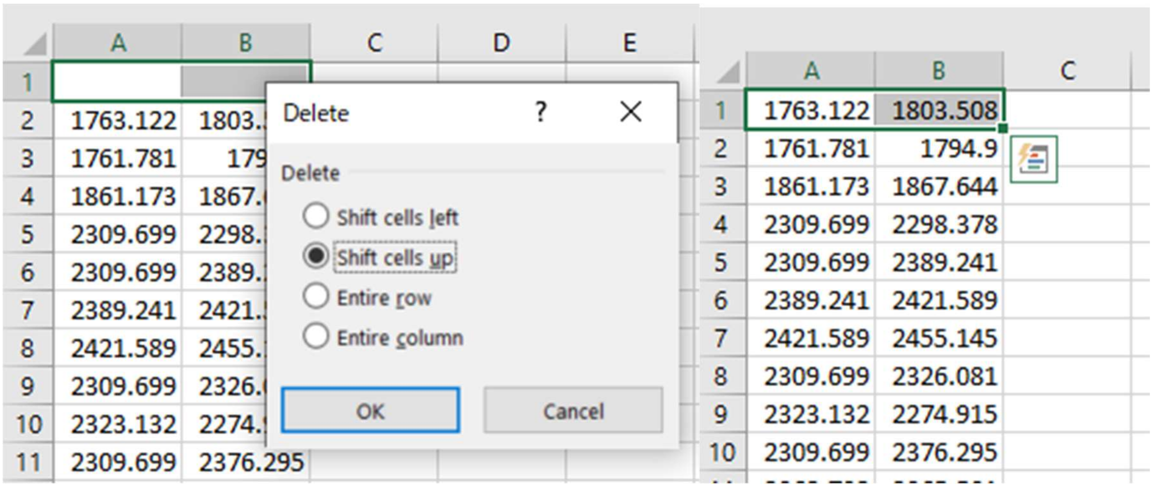


Figure 8

Step 11

Hit Alt + F11 to open Excel VBA editor.

Step 12

Add a new module in VBA editor by clicking on VBAProject (Book1), right click, Insert > Module.

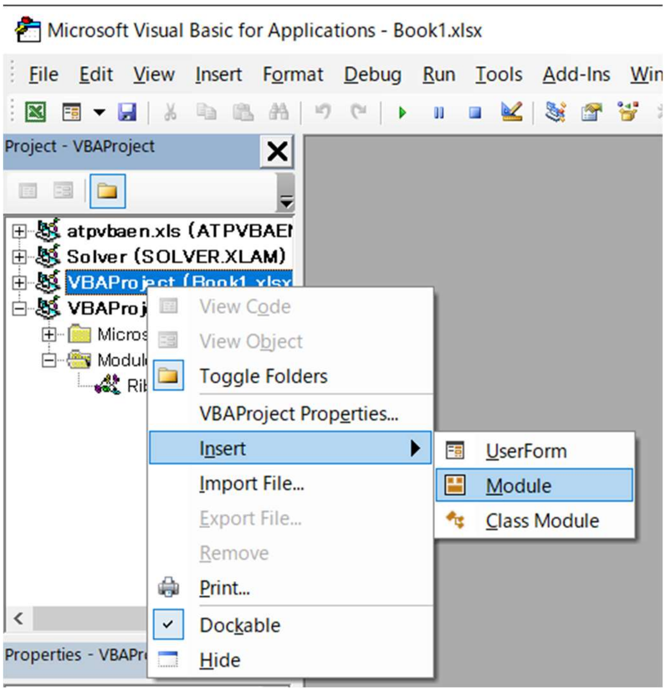


Figure 9

Step 13

Copy the following scripts and paste it into the window of Module1. The code sorts out the data. Column D will have $d(\partial l_j^{-})$: the shortest path distance between the start node of link j and the nearest generating point from link j; and column E will have $d(\partial l_j^{+})$: the shortest path distance between the end node of link j and the nearest generating point.

```
Sub Transformation()

MaxRow = Rows.Count

For i = 1 To MaxRow
    If Cells(i, 1) > Cells(i, 2) Then
        Origin = Cells(i, 2)
        Destination = Cells(i, 1)
```

```
        Else
            Origin = Cells(i, 1)
            Destination = Cells(i, 2)
        End If

        Cells(i, 4).Value = Origin
        Cells(i, 5).Value = Destination
    Next i

    End Sub
```

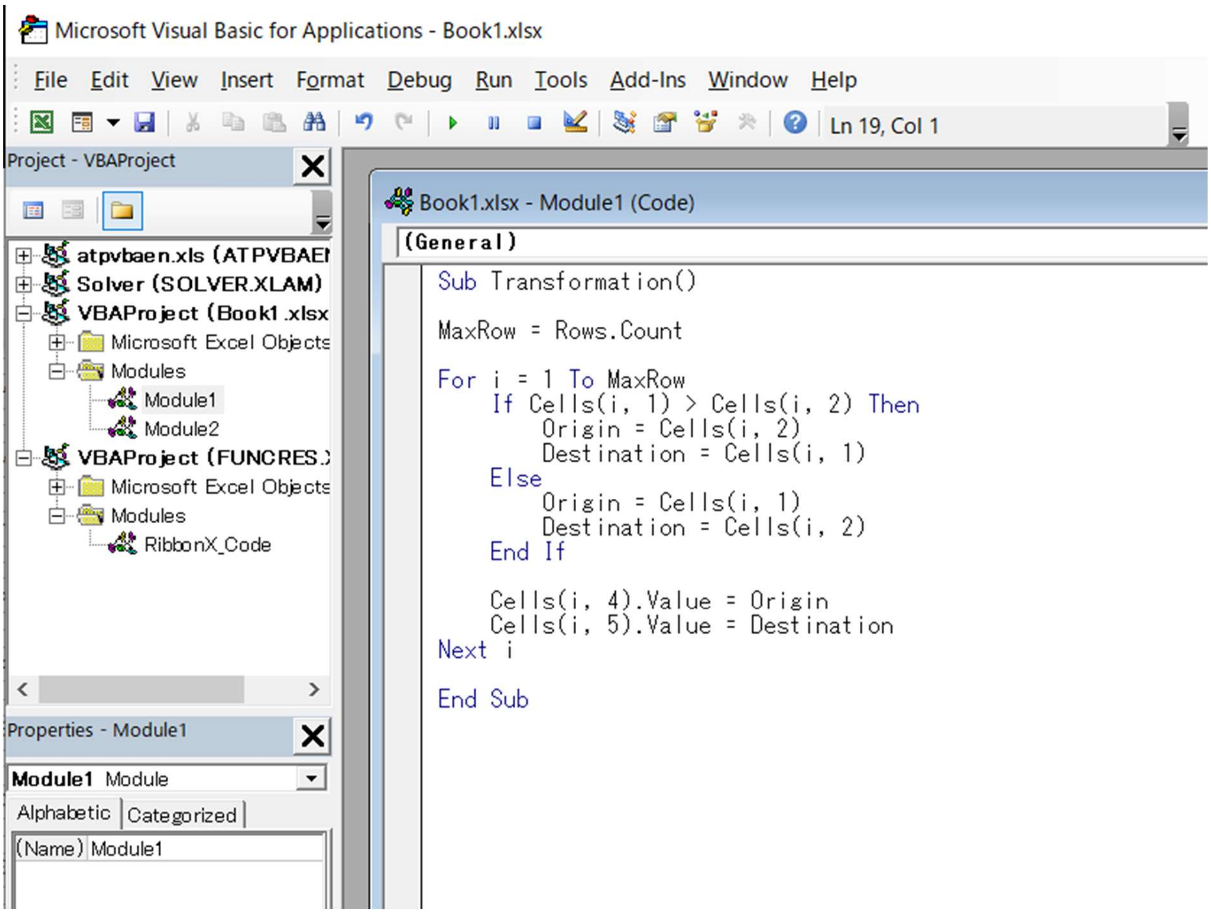


Figure 10

Step 14

Run the module by clicking “Run Sub” icon (see Figure 11). Figure 12 is an output example. You would see that each value of Column D must be smaller than each one next to it (Column E).

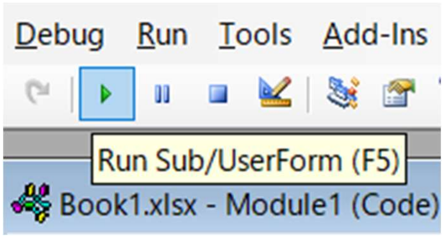


Figure 11

	A	B	C	D	E
1	1763.122	1803.508		1763.122	1803.508
2	1761.781	1794.9		1761.781	1794.9
3	1861.173	1867.644		1861.173	1867.644
4	2309.699	2298.378		2298.378	2309.699
5	2309.699	2389.241		2309.699	2389.241
6	2389.241	2421.589		2389.241	2421.589
7	2421.589	2455.145		2421.589	2455.145
8	2309.699	2326.081		2309.699	2326.081
9	2323.132	2274.915		2274.915	2323.132
10	2309.699	2376.295		2309.699	2376.295
11	2062.702	2065.501		2062.702	2065.501
12	2062.702	2046.624		2046.624	2062.702
13	2062.702	2076.568		2062.702	2076.568
14	2076.568	2081.872		2076.568	2081.872
15	2062.702	2089.19		2062.702	2089.19
16	2089.19	2099.928		2089.19	2099.928
17	2099.928	2104.948		2099.928	2104.948
18	2104.948	2117.013		2104.948	2117.013
19	2117.013	2121.852		2117.013	2121.852
20	2216.175	2263.809		2216.175	2263.809
21	2060.326	2043.344		2043.344	2060.326
22	2060.326	2075.116		2060.326	2075.116
23	2075.116	2075.754		2075.116	2075.754

Figure 12

Step 15

Create another module in the VBA editor (Module2), and copy the following scripts and paste it into the window. This code is for calculating $L(x_i, x_{i+1})$: the total length of line segments constituting the sub-network $N(x_i, x_{i+1})$. Note that this code assumes that the bin width $c =$ 10m.

```
Sub Classification()

MaxRow = Rows.Count

MaxVal = WorksheetFunction.Max(Range(Cells(1, 5), Cells(MaxRow, 5)))

MaxBin = MaxVal ¥ 10 + 1

For j = 0 To MaxBin

    For i = 1 To MaxRow

        If Cells(i, 4) <= j * 10 And Cells(i, 5) > (j + 1) * 10 Then
            Cells(j + 1, 10) = Cells(j + 1, 10) + 10

        ElseIf Cells(i, 4) <= j * 10 And Cells(i, 5) >= j * 10 And Cells(i, 5) <= (j + 1) * 10 Then
            Cells(j + 1, 10) = Cells(j + 1, 10) + (Cells(i, 5) - j * 10)

        ElseIf Cells(i, 4) > j * 10 And Cells(i, 4) <= (j + 1) * 10 And Cells(i, 5) > (j + 1) * 10 Then
            Cells(j + 1, 10) = Cells(j + 1, 10) + ((j + 1) * 10 - Cells(i, 4))

        ElseIf Cells(i, 4) > j * 10 And Cells(i, 4) <= (j + 1) * 10 And Cells(i, 5) <= (j + 1) * 10 Then
            Cells(j + 1, 10) = Cells(j + 1, 10) + (Cells(i, 5) - Cells(i, 4))
        End If
    Next i
Next j

For k = 1 To MaxBin
```

```
    Cells (k,9) = k*10
Next k

End Sub
```

Step 16

Run the module by clicking “Run Sub” icon. Note that this macro takes a lot of time (about several hours depending on data size) to complete the calculation. Figure 13 is an example of outputs.

I	J
10	20
20	27.11232
30	30
40	33.07666
50	43.23238
60	40
70	40
80	47.87207
90	50
100	57.16533
110	68.42125
120	60
130	69.38196
140	69.39657
150	79.7912
160	93.37199
170	104.5682
180	119.093
190	149.2525
200	180.9578

Figure 13

Let $P_B = \{p_{B1}, \cdots, P_{Bn}\}$ be type B points and $n(x_{i+1}|P_{Aj})$ be the number of type B points included in the sub-network within distance x_{i+1} from P_{Aj}.

In terms of the above notations, this manual describes the procedure for computing $n(x_{i+1}|P_{Aj})$.

Note that this computation uses the output of the Voronoi cross *K* function tool in SANET software package.

1) Run the “Voronoi cross *K* function method” tool of SANET. See the Section 4.16 in the SANET manual.

2) Copy “SANETVCKFuncObservedValue.csv” to another folder.

3) Open these files in Excel.

* You may follow the steps below, or you may use an alternative method as you like.

a) Open Excel.

b) Select File, “Open” (the orange arrow in Figure 1).

c) Click “Browse” and select the folder in which you saved the files (the green arrow in Figure 1).

d) In the “Files of Type” drop down box, select: “All Files” (the blue arrow in Figure 1), and then choose “SANETVCKFuncObservedValue.csv”, and click “Open”.

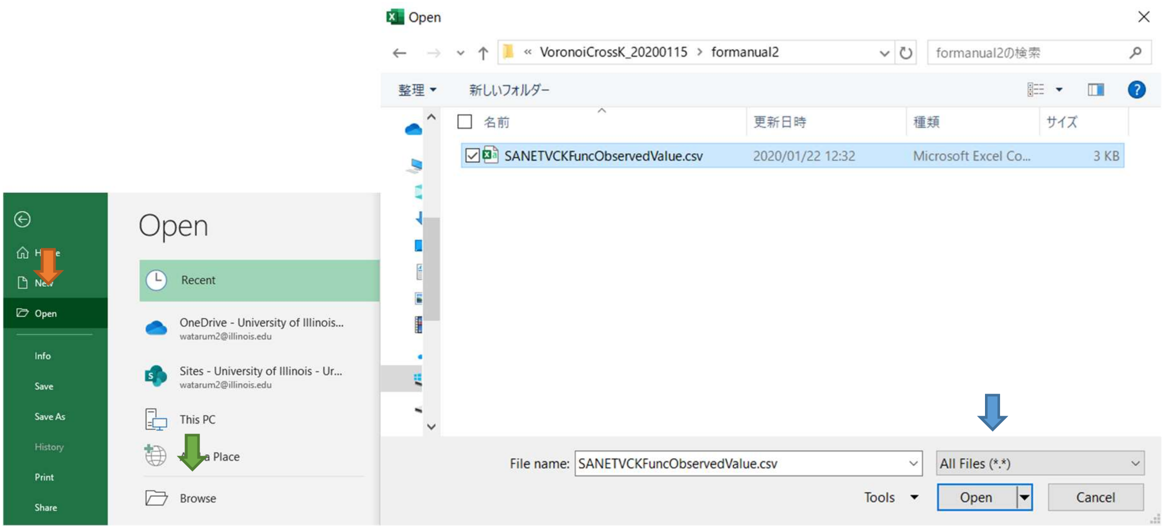


Figure 1

4) In cell E2, enter x_1 : the minimum value (meters) of bin c (e.g., 10). Next, in cell E3, enter x_2 : a double value of E2 (e.g., 20).

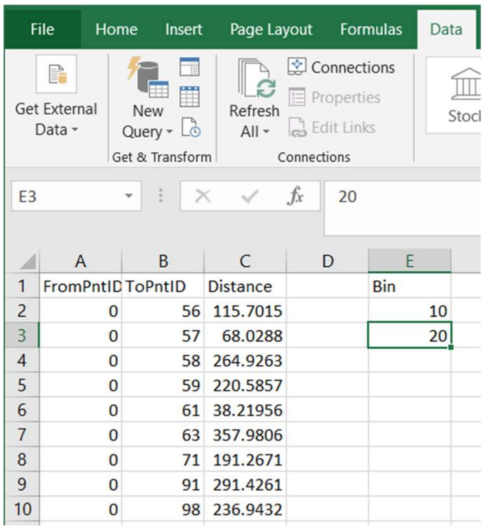
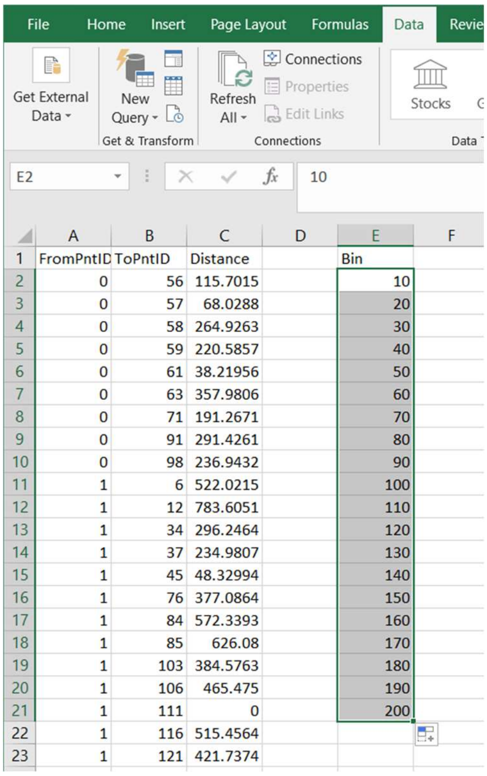


Figure 2

5) Highlight cells E2:E3. When we drag the fill handle to the below (by left-clicking and dragging the small dot at the lower right corner: AutoFill function), Excel will fill in the next cells with values having same intervals (e.g., 30, 40, 50, and so on… as shown in Figure 3).



	A	B	C	D	E
1	FromPntID	ToPntID	Distance		Bin
2	0	56	115.7015		10
3	0	57	68.0288		20
4	0	58	264.9263		30
5	0	59	220.5857		40
6	0	61	38.21956		50
7	0	63	357.9806		60
8	0	71	191.2671		70
9	0	91	291.4261		80
10	0	98	236.9432		90
11	1	6	522.0215		100
12	1	12	783.6051		110
13	1	34	296.2464		120
14	1	37	234.9807		130
15	1	45	48.32994		140
16	1	76	377.0864		150
17	1	84	572.3393		160
18	1	85	626.08		170
19	1	103	384.5763		180
20	1	106	465.475		190
21	1	111	0		200
22	1	116	515.4564		
23	1	121	421.7374		

Figure 3

6) In cell F2, enter the formula that counts $n(x_{i+1}|P_{Aj})$: the number of point-like facilities located within a specific range from generating points

=COUNTIFS($C$2:$C$131,"<”&E2)

*Note that “131” (the number of type B points) in the formula change appropriately.

Hit the enter.

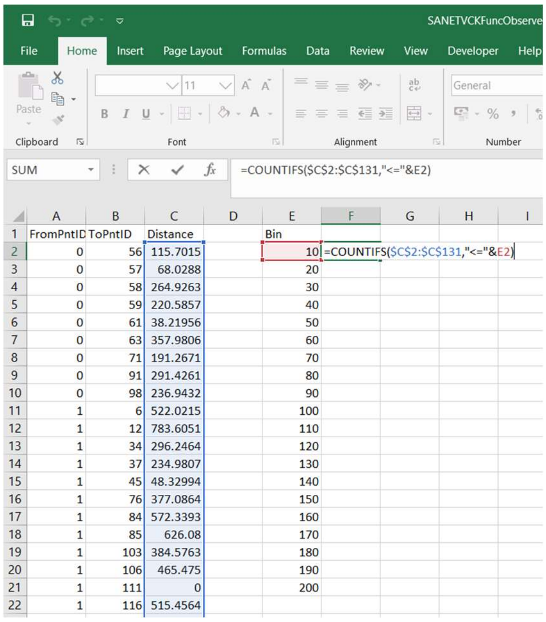


Figure 4

7) Copy the formula to the rest of bins by using AutoFilll function.

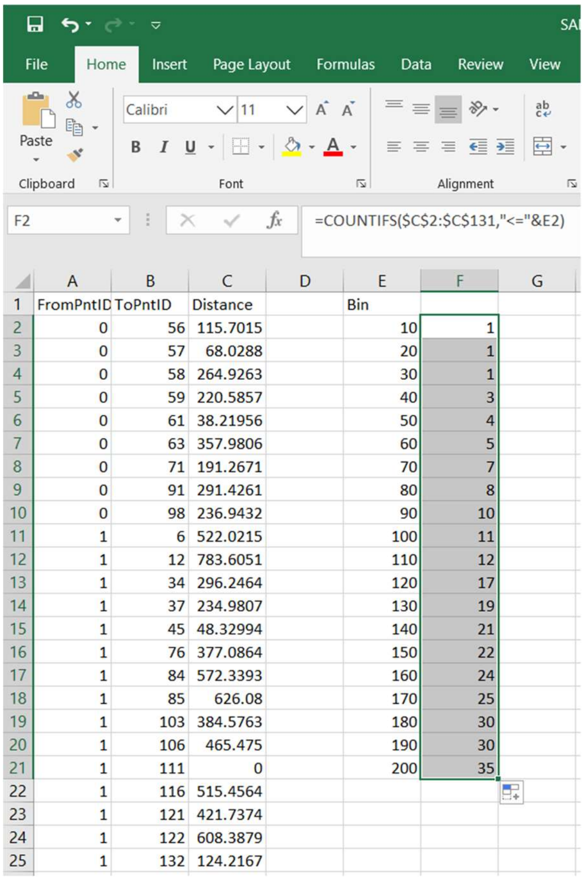


Figure 5

8) If you want to obtain the number of points on the line segments between x_i and x_{i+1}, enter the following formula in cell G2.

=COUNTIFS($C$2:$C$131,"<"&E2, $C$2:$C$131,">="&E2-10)

Hit the enter and copy the formula to the rest of bins by using AutoFilll function.

	A	B	C	D	E	F	G	H	I	J	K	L
1	FromPntID	ToPntID	Distance		Bin							
2	0	56	115.7015		10	1	=COUNTIFS($C$2:$C$131,"<"&E2, $C$2:$C$131,">="&E2-10)					
3	0	57	68.0288		20	1	0					
4	0	58	264.9263		30	1	0					
5	0	59	220.5857		40	3	2					
6	0	61	38.21956		50	4	1					
7	0	63	357.9806		60	5	1					
8	0	71	191.2671		70	7	2					
9	0	91	291.4261		80	8	1					
10	0	98	236.9432		90	10	2					
11	1	6	522.0215		100	11	1					
12	1	12	783.6051		110	12	1					
13	1	34	296.2464		120	17	5					
14	1	37	234.9807		130	19	2					
15	1	45	48.32994		140	21	2					
16	1	76	377.0864		150	22	1					
17	1	84	572.3393		160	24	2					
18	1	85	626.08		170	25	1					
19	1	103	384.5763		180	30	5					
20	1	106	465.475		190	30	0					
21	1	111	0		200	35	5					
22	1	116	515.4564									

Figure 6

6.3 Procedure for computing the total length of "buffer-ring sub-networks" of a generator point set

To state the above computation precisely, let $N(x_i, x_{i+1})$ denote the 'buffer-ring sub-networks' of a network N in which the shortest path distance from any points on $N(x_i, x_{i+1})$ to the nearest generator point (Type A point) is between x_i and x_{i+1}, where $x_i < x_{i+1}$, $x_{i+1} - x_i = c$ for $i = 0, 1, 2, \ldots$ Note that c is a parameter determined by a user. In an illustrative example here, $c =$ 10m. Visually, $N(0, x_{i+1})$ is obtained from the 'buffer sub-networks' in Figure 1 in Section 6.1, where $x_{i+1} = 200$. Therefore, the 'buffer-ring sub-networks' $N(x_i, x_{i+1})$ is obtained from $N(0, x_{i+1})$¥ $N(0, x_i)$, i.e., the compliment of $N(0, x_i)$ with respect to $N(0, x_{i+1})$.

Let $L(x_i, x_{i+1})$ be the total length of line segments constituting the buffer-ring sub-networks $N(x_i, x_{i+1})$, and L is the total length of line segments constituting the whole network N.

In terms of the above notations, this manual describes the procedure for computing $L(x_i, x_{i+1})$.

Let l_j be link j of a given network, and $d(\partial l_j^{-})$ refers to the shortest path distance between the start node of link *j* and the nearest generating point from link *j*; $d(\partial l_j^{+})$ is the shortest path distance between the end node of link j and the nearest generating point.

Step 1

Run the "Voronoi diagram" tool of SANET. In more detail, see Section 4.1 Tool 01: Voronoi diagrams in User Guide/Manual for SANET Standalone.

Step 2

Copy "SANETVoronoiPnt.dbf" and "SANETVoronoiSgt.dbf" on one folder.

Step 3

Open these files in Excel.

* You may follow the four steps described below, or you may use an alternative method as you like.

3-1) Open Excel.

3-2) Select File, "Open" (the orange arrow in Figure 1).

3-3) Click "Browse" and select the folder in which you saved the files (the green arrow in Figure 1).

3-4) In the "Files of Type" drop down box, select: "All Files" (the blue arrow in Figure 1), and then choose "SANETVoronoiPnt.dbf" and "SANETVoronoiSgt.dbf", and click "Open".

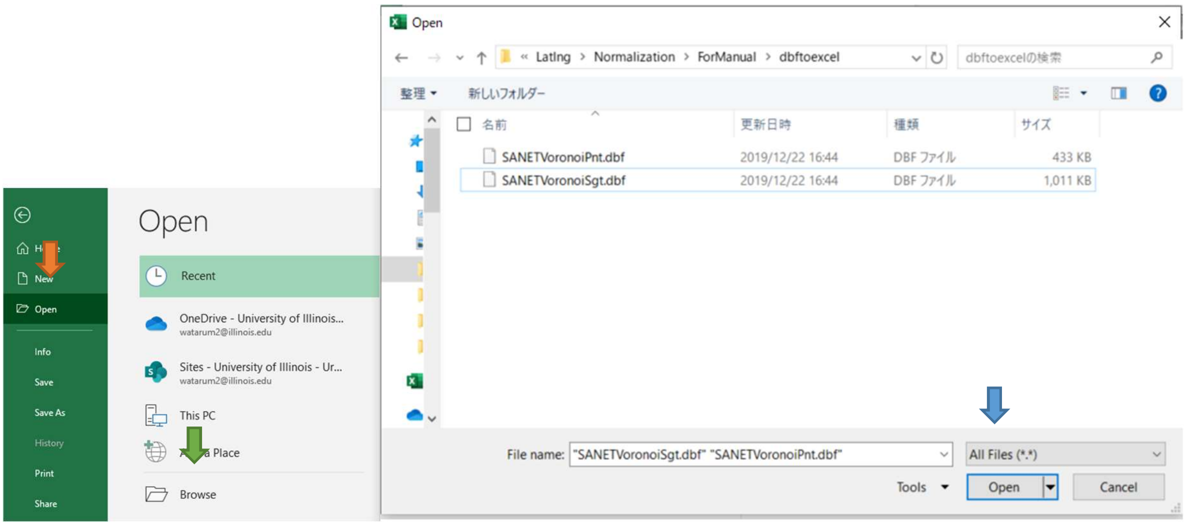


Figure 1

Step 4

Copy these two sheets on a new book (for instance, Book1). Figure2 is a sample of the screenshot after step 4).

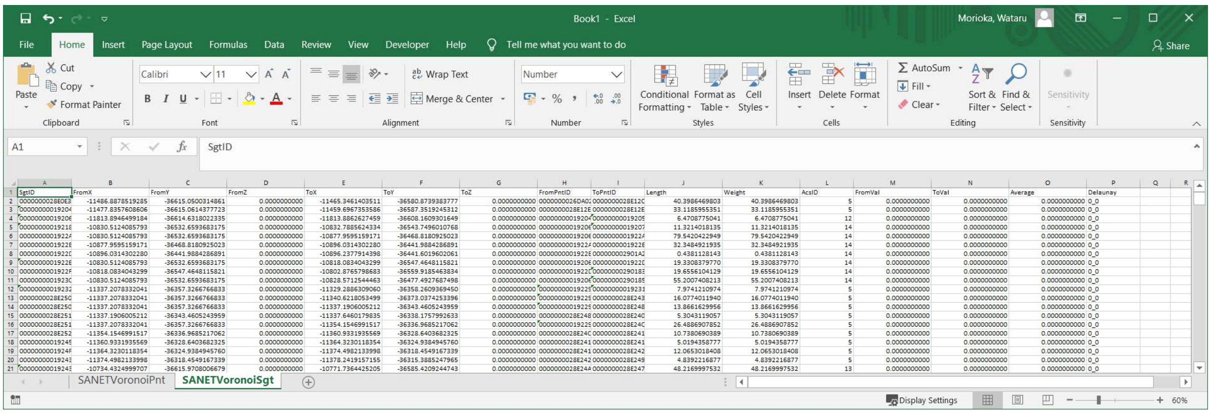


Figure 2

Step 5

In the cell Q2 (column Q and row 2) of “SANETVoronoiSgt” sheet, enter the formula to compute distance from the generator to “FromPnt” of the segment:

=VLOOKUP(H2,SANETVoronoiPnt!$A:$J,8, FALSE)

Hit the enter.

=VLOOKUP(H2,SANETVoronoiPnt!$A:$J,8, FALSE)

L	M	N	O	P	Q
	FromVal	ToVal	Average	Delaunay	
0	0.0000000000	0.0000000000	0.0000000000	0_0	1763.122
0	0.0000000000	0.0000000000	0.0000000000	0_0	
0	0.0000000000	0.0000000000	0.0000000000	0_0	
0	0.0000000000	0.0000000000	0.0000000000	0_0	

Figure 3

Step 6

Copy the formula to the other segments by left-clicking and dragging the small dot at the lower right corner of Q2 down column Q to the end row of the data.

0000000029063F1	00000000250A634	5.432680686	5.432680686	0.000000000000000	0	0	0	0_0	2468.596
000000002906378	0000000028F27F0	4.821205672	4.821205672	0.000000000000000	0	0	0	0_0	434.5112
00000000290643E	0000000028D8FF3	16.88641222	16.88641222	0.000000000000000	0	0	0	0_0	500.996
0000000029064E3	0000000028EC195	2.888190439	2.888190439	0.000000000000000	0	0	0	0_0	1246.478
0000000029064F9	0000000028EBDCD	7.808978939	7.808978939	0.000000000000000	0	0	0	0_0	1391.997
000000002906567	00000000250BA2F	21.81211395	21.81211395	0.000000000000000	0	0	0	0_0	1941.034
0000000029063E6	0000000028F2BA2	15.04838392	15.04838392	0.000000000000000	0	0	0	0_0	1549.718
0000000029065A9	0000000028DA12B	0.914670145	0.914670145	0.000000000000000	0	0	0	0_0	621.6643
00000000290630A	0000000028F2C68	24.47211596	24.47211596	0.000000000000000	0	0	0	0_0	1554.055
000000002906475	000000002622896	37.18872051	37.18872051	0.000000000000000	0	0	0	0_0	1984.225

Figure 4

Step 7

In cell R2 of SANETVoronoiSgt, enter the formula to compute distance from the generator to "ToPnt" of the segment:

=VLOOKUP(I2,SANETVoronoiPnt!$A:$J,8, FALSE)

Hit the enter.

=VLOOKUP(I2,SANETVoronoiPnt!$A:$J,8, FALSE)

L	M	N	O	P	Q	R
	FromVal	ToVal	Average	Delaunay		
0	0.0000000000	0.0000000000	0.0000000000	0_0	1763.122	1803.508
0	0.0000000000	0.0000000000	0.0000000000	0_0	1761.781	
0	0.0000000000	0.0000000000	0.0000000000	0_0	1861.173	
0	0.0000000000	0.0000000000	0.0000000000	0_0	2309.699	
0	0.0000000000	0.0000000000	0.0000000000	0_0	2309.699	

Figure 5

Step 8

Copy the formula to the other segments by left-clicking and dragging the small dot at the lower right corner of R2 down column R to the end row of the data.

0000000029063F1	00000000250A634	5.432680686	5.432680686	0.000000000000000	0	0	0	0_0	2468.596	2463.164
000000002906378	0000000028F27F0	4.821205672	4.821205672	0.000000000000000	0	0	0	0_0	434.5112	429.69
00000000290643E	0000000028D8FF3	16.88641222	16.88641222	0.000000000000000	0	0	0	0_0	500.996	484.1096
0000000029064E3	0000000028EC195	2.888190439	2.888190439	0.000000000000000	0	0	0	0_0	1246.478	1243.589
0000000029064F9	0000000028EBDCD	7.808978939	7.808978939	0.000000000000000	0	0	0	0_0	1391.997	1384.188
000000002906567	00000000250BA2F	21.81211395	21.81211395	0.000000000000000	0	0	0	0_0	1941.034	1919.222
0000000029063E6	0000000028F2BA2	15.04838392	15.04838392	0.000000000000000	0	0	0	0_0	1549.718	1534.67
0000000029065A9	0000000028DA12B	0.914670145	0.914670145	0.000000000000000	0	0	0	0_0	621.6643	620.7496
00000000290630A	0000000028F2C68	24.47211596	24.47211596	0.000000000000000	0	0	0	0_0	1554.055	1529.583
000000002906475	000000002622896	37.18872051	37.18872051	0.000000000000000	0	0	0	0_0	1984.225	1947.036

Figure 6

Step 9

Copy the values of columns Q and R, and paste them to a new sheet.

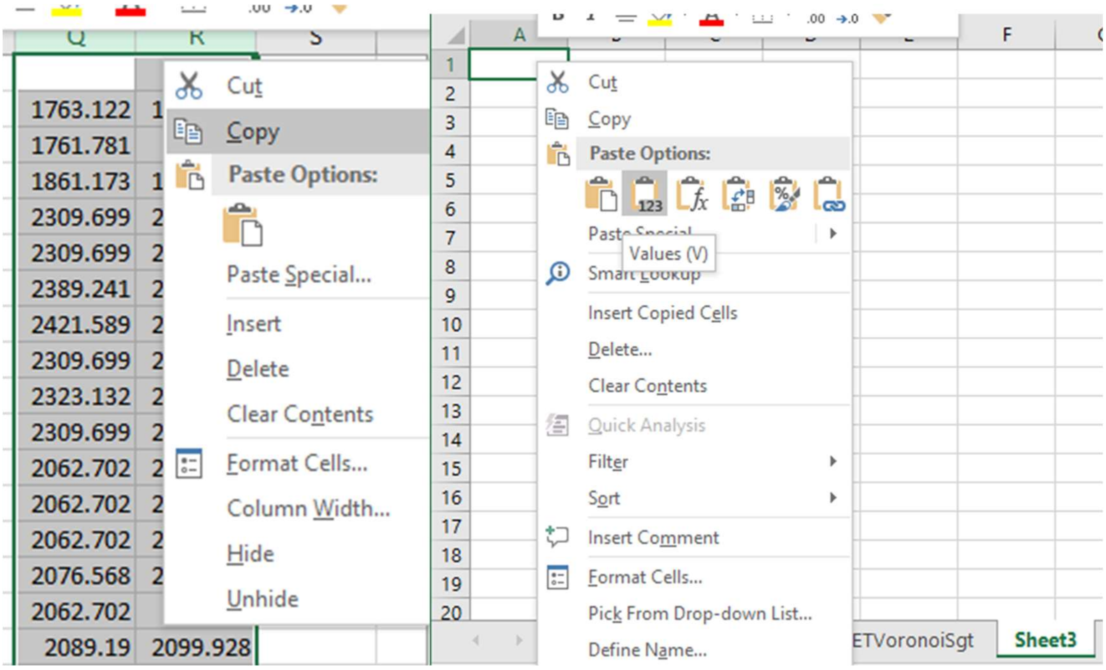


Figure 7

Step 10

Delete Cell A1 and B1 (blank spaces). Note that choosing “Shift cells up” as the option and then clicking “OK”.

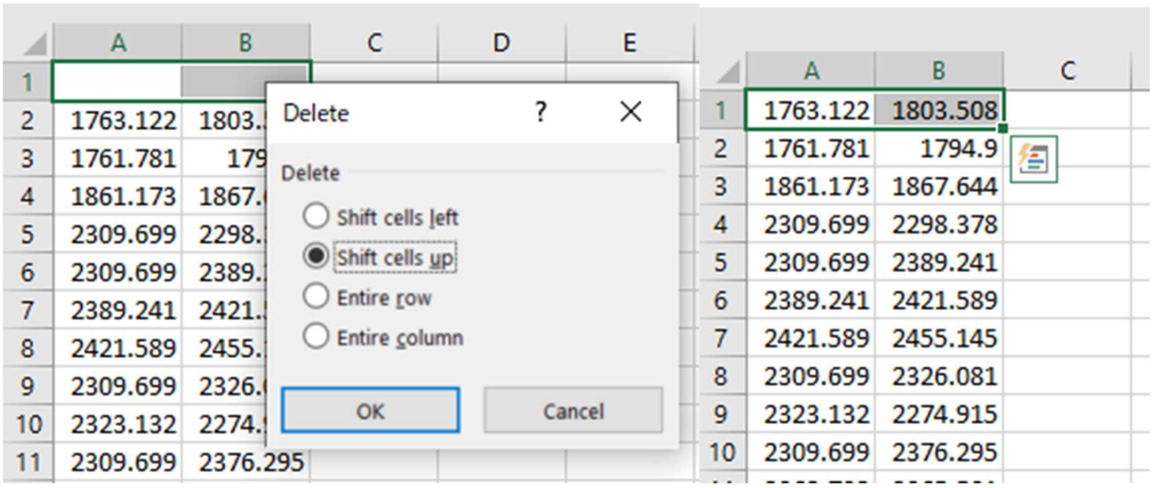


Figure 8

Step 11

Hit Alt + F11 to open Excel VBA editor.

Step 12

Add a new module in VBA editor by clicking on VBAProject (Book1), right click, Insert > Module.

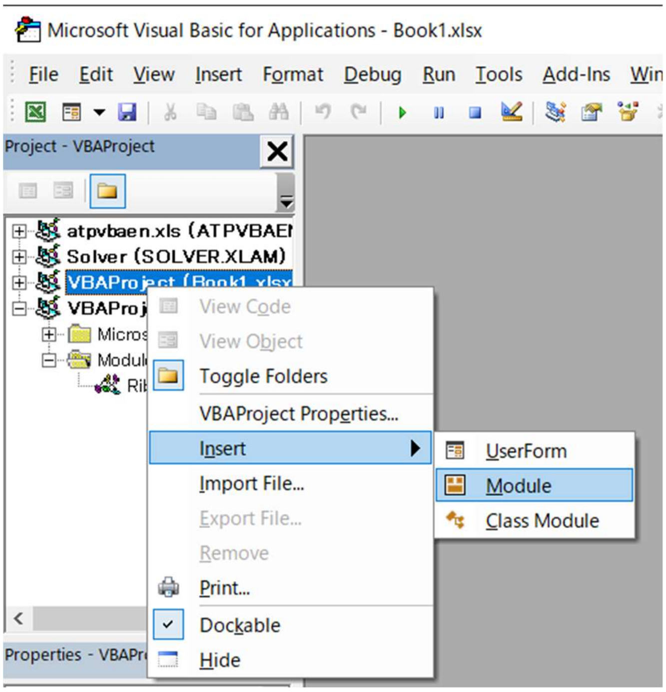


Figure 9

Step 13

Copy the following scripts and paste it into the window of Module1. The code sorts out the data. Column D will have $d(\partial l_j^-)$: the shortest path distance between the start node of link j and the nearest generating point from link j; and column E will have $d(\partial l_j^+)$: the shortest path distance between the end node of link j and the nearest generating point.

```
Sub Transformation()

MaxRow = Rows.Count

For i = 1 To MaxRow
    If Cells(i, 1) > Cells(i, 2) Then
        Origin = Cells(i, 2)
        Destination = Cells(i, 1)
```

```
        Else
            Origin = Cells(i, 1)
            Destination = Cells(i, 2)
        End If

        Cells(i, 4).Value = Origin
        Cells(i, 5).Value = Destination
    Next i

    End Sub
```

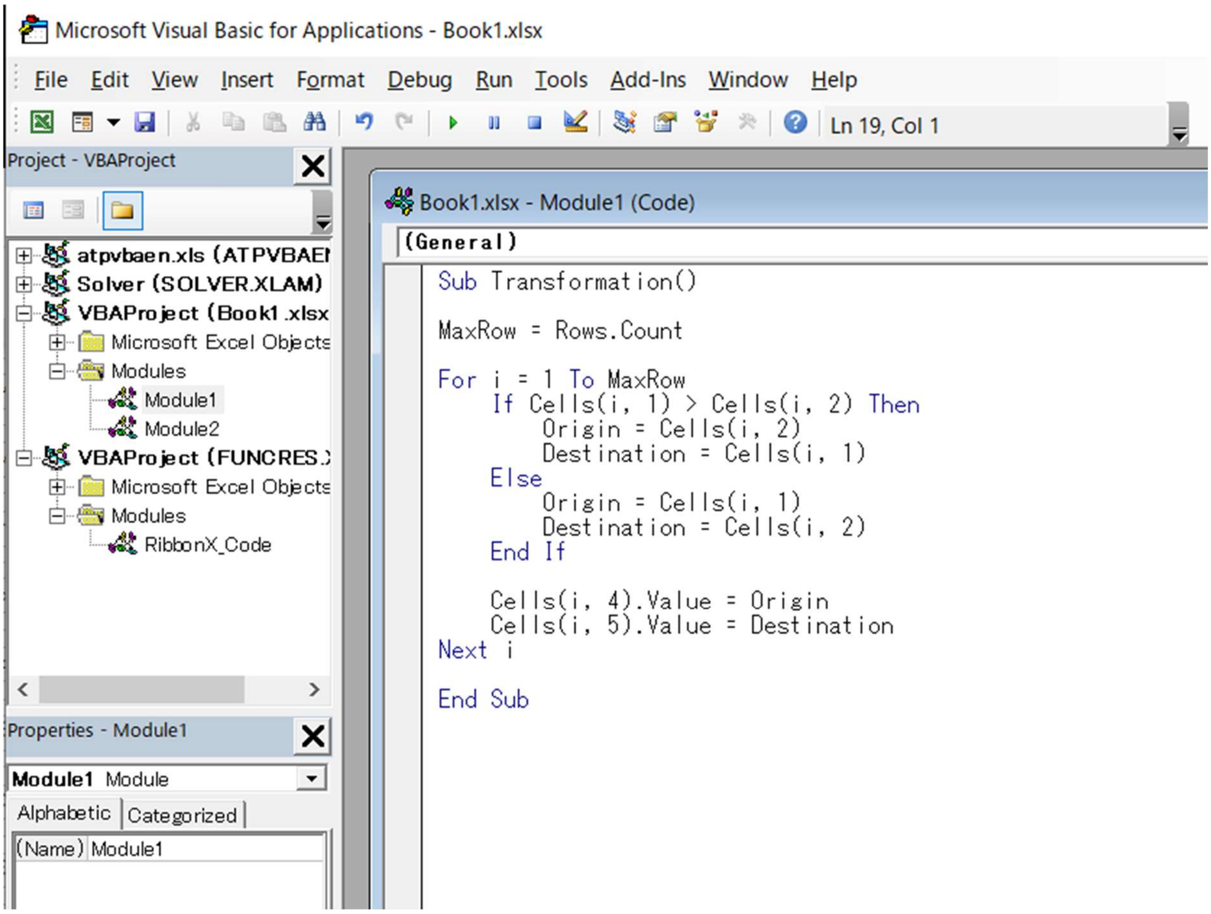


Figure 10

Step 14

Run the module by clicking “Run Sub” icon (see Figure 11). Figure 12 is an output example. You would see that each value of Column D must be smaller than each one next to it (Column E).

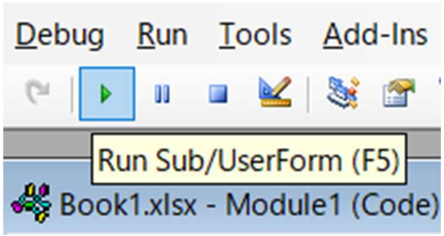


Figure 11

	A	B	C	D	E
1	1763.122	1803.508		1763.122	1803.508
2	1761.781	1794.9		1761.781	1794.9
3	1861.173	1867.644		1861.173	1867.644
4	2309.699	2298.378		2298.378	2309.699
5	2309.699	2389.241		2309.699	2389.241
6	2389.241	2421.589		2389.241	2421.589
7	2421.589	2455.145		2421.589	2455.145
8	2309.699	2326.081		2309.699	2326.081
9	2323.132	2274.915		2274.915	2323.132
10	2309.699	2376.295		2309.699	2376.295
11	2062.702	2065.501		2062.702	2065.501
12	2062.702	2046.624		2046.624	2062.702
13	2062.702	2076.568		2062.702	2076.568
14	2076.568	2081.872		2076.568	2081.872
15	2062.702	2089.19		2062.702	2089.19
16	2089.19	2099.928		2089.19	2099.928
17	2099.928	2104.948		2099.928	2104.948
18	2104.948	2117.013		2104.948	2117.013
19	2117.013	2121.852		2117.013	2121.852
20	2216.175	2263.809		2216.175	2263.809
21	2060.326	2043.344		2043.344	2060.326
22	2060.326	2075.116		2060.326	2075.116
23	2075.116	2075.754		2075.116	2075.754

Figure 12

Step 15

Create another module in the VBA editor (Module2), and copy the following scripts and paste it into the window. This code is for calculating $L(x_i, x_{i+1})$: the total length of line segments constituting the sub-network $N(x_i, x_{i+1})$. Note that this code assumes that the bin width $c =$ 10m.

```
Sub Classification()

MaxRow = Rows.Count

MaxVal = WorksheetFunction.Max(Range(Cells(1, 5), Cells(MaxRow, 5)))

MaxBin = MaxVal ¥ 10 + 1

For j = 0 To MaxBin

    For i = 1 To MaxRow

        If Cells(i, 4) <= j * 10 And Cells(i, 5) > (j + 1) * 10 Then
            Cells(j + 1, 10) = Cells(j + 1, 10) + 10

        ElseIf Cells(i, 4) <= j * 10 And Cells(i, 5) >= j * 10 And Cells(i, 5) <= (j + 1) * 10 Then
            Cells(j + 1, 10) = Cells(j + 1, 10) + (Cells(i, 5) - j * 10)

        ElseIf Cells(i, 4) > j * 10 And Cells(i, 4) <= (j + 1) * 10 And Cells(i, 5) > (j + 1) * 10 Then
            Cells(j + 1, 10) = Cells(j + 1, 10) + ((j + 1) * 10 - Cells(i, 4))

        ElseIf Cells(i, 4) > j * 10 And Cells(i, 4) <= (j + 1) * 10 And Cells(i, 5) <= (j + 1) * 10 Then
            Cells(j + 1, 10) = Cells(j + 1, 10) + (Cells(i, 5) - Cells(i, 4))
        End If
    Next i
Next j

For k = 1 To MaxBin
```

```
    Cells (k,9) = k*10
Next k

End Sub
```

Step 16

Run the module by clicking “Run Sub” icon. Note that this macro takes a lot of time (about several hours depending on data size) to complete the calculation. Figure 13 is an example of outputs.

I	J
10	20
20	27.11232
30	30
40	33.07666
50	43.23238
60	40
70	40
80	47.87207
90	50
100	57.16533
110	68.42125
120	60
130	69.38196
140	69.39657
150	79.7912
160	93.37199
170	104.5682
180	119.093
190	149.2525
200	180.9578

Figure 13

6.4 Exact statistical test for Voronoi cross *K* function method on networks using outputs of the tools in SANET software package

This manual instructs a procedure for achieving the exact statistical test of global Voronoi cross *K* function method. The global Voronoi cross *K* function method is explained in detail in Section 6.2.3 in Chatper 6 of Okabe and Sugihara (2012).

To begin with, let us introduced a few notations. We consider a network N, on which two types of points are placed, i.e., Type A points (generator points of the network Voronoi) and Type B points (activity points).

Note that the notations conform to the SANET software. Okabe and Sugihara (2012) treated the terms in the opposite (i.e., Type A: activity points, Type B: generator points).

Let $N(x_i, x_{i+1})$denote the sub-network of N in which the shortest path distance from any points on $N(x_i, x_{i+1})$ to the nearest point in Type B points is between x_i and x_{i+1}, where $x_i < x_{i+1}$, $x_{i+1} - x_i = c$ for $i = 0, 1, 2, \ldots$ Note that c is a parameter determined by a user. In an illustrative example in Figure 1, c = 10 m.

Let $L(x_i, x_{i+1})$ be the total length of line segments constituting the sub-network $N(x_i, x_{i+1})$, and L is the total length of line segments constituting the whole network N.

Let $m(x_i, x_{i+1})$ be the number of type B points on the sub-network $N(x_i, x_{i+1})$.

Suppose that Type B points are distributed on N according to the complete spatial randomness, CSR. Then the probability that a Type B point is located on $N(x_i, x_{i+1})$ is given by $p_i = L(x_i, x_{i+1})/L$, and $m(x_i, x_{i+1})$ follows the binomial distribution with parameter $L(x_i, x_{i+1})/L$.

The data of $L(x_i, x_{i+1})$ are obtained from running the Voronoi diagram tool in SANET. This task is explained by the manual "A procedure for computing the length of line segments of a network satisfying that the shortest path distance from a point on the line segments of a network to its nearest generator point among given generator points placed on the network is between x_i and x_{i+1}".

Step 1

Compute the total length L of network N by summing $L(x_i, x_{i+1})$ for $i = 0, 1, 2, \ldots$. You may use AutoSum function as shown in Figure 1.

I	J	K	L
Bin	Length	Probability	
10	343.4549		
20	395.2408		
30	463.1867		
40	500.218		
50	590.0724		

========[Omit]========

840	138.9082		
850	143.2698		
860	130.7604		
870	74.9661		
880	33.43163		
890	25.54627		
900	20		
910	20		
920	17.0496		
930	10		
940	10		
950	10		
960	3.765532		
	=SUM(J2:J97)		
	SUM(**number1**, [number2], ...)		

Figure 1

Step 2

To compute, first, type in the equation in cell K2 (Figure2) and drag as shown in Figure3.

I	J	K	L
Bin	Length	Probability	
10	343.4549	=SUM($J$2:J2)/$J$98	
20	395.2408		

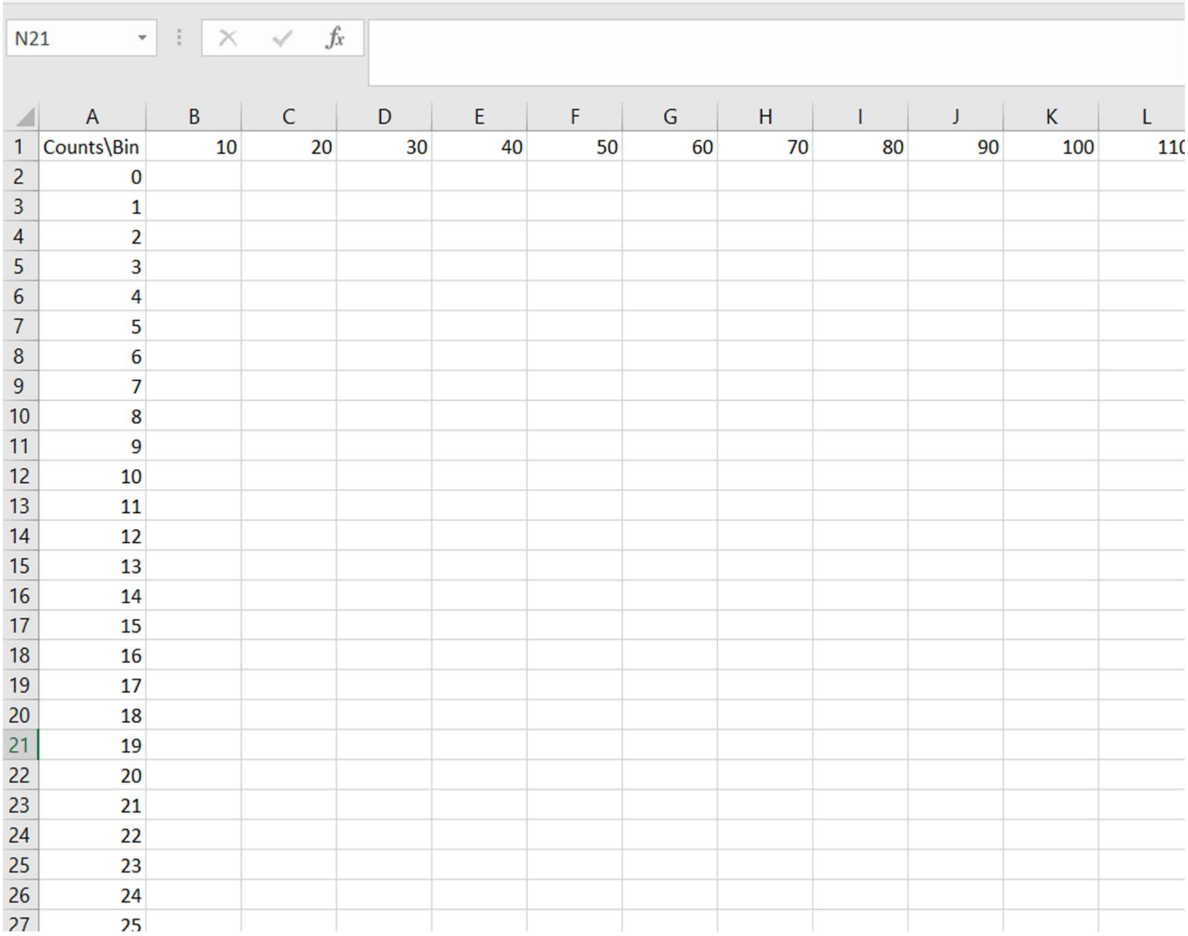
Figure 2

Bin	Length	Probability
10	343.4549	0.003739
20	395.2408	0.008041
30	463.1867	0.013083
40	500.218	0.018529
50	590.0724	0.024952
60	658.2757	0.032118
70	714.2047	0.039892
80	789.9798	0.048492
90	873.915	0.058005
100	1073.829	0.069695
110	1254.99	0.083356
120	1286.821	0.097364
130	1342.8	0.111981
140	1456.309	0.127834
150	1540.961	0.144609
160	1629.962	0.162352
170	1706.882	0.180933
180	1740.582	0.19988
190	1835.952	0.219866
200	1829.098	0.239777

Figure 3

Step 3

Create a new sheet and set the framework shown in Figure 4.


N21

	A	B	C	D	E	F	G	H	I	J	K	L
1	Counts\Bin	10	20	30	40	50	60	70	80	90	100	110
2	0											
3	1											
4	2											
5	3											
6	4											
7	5											
8	6											
9	7											
10	8											
11	9											
12	10											
13	11											
14	12											
15	13											
16	14											
17	15											
18	16											
19	17											
20	18											
21	19											
22	20											
23	21											
24	22											
25	23											
26	24											
27	25											

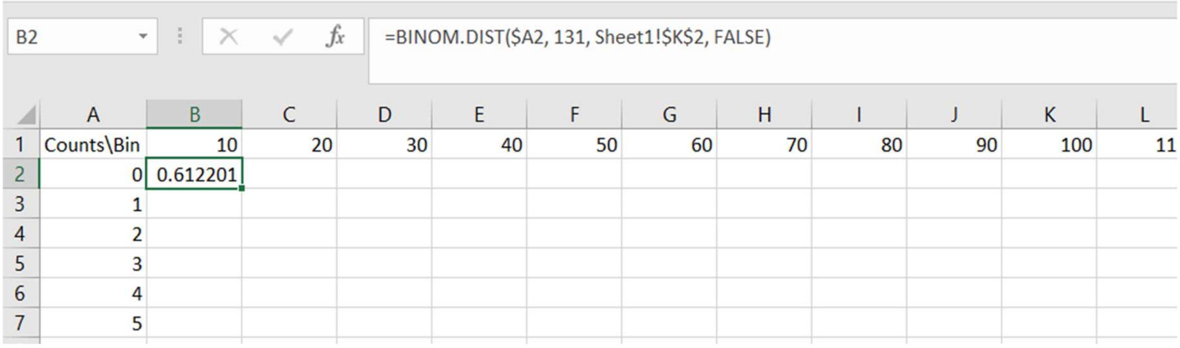
Figure 4

Step 4

In cell B2, type in the following formula to calculate the probabilities of the binomial distribution for $i = 10, 20, \ldots$

=BINOM.DIST($A2, 131, Sheet1!$K$2, FALSE)

*Note that "131" (the number of type A points) and "Sheet1" (the sheet name storing the probabilities of each bin width) in the formula change appropriately. In this case, the resulting value in cell B2 of Figure 5 represents the probability to locate no type A points within 10m from their nearest type B points under CSR hypothesis.



B2 =BINOM.DIST($A2, 131, Sheet1!$K$2, FALSE)

	A	B	C	D	E	F	G	H	I	J	K	L
1	Counts\Bin	10	20	30	40	50	60	70	80	90	100	11
2	0	0.612201										
3	1											
4	2											
5	3											
6	4											
7	5											

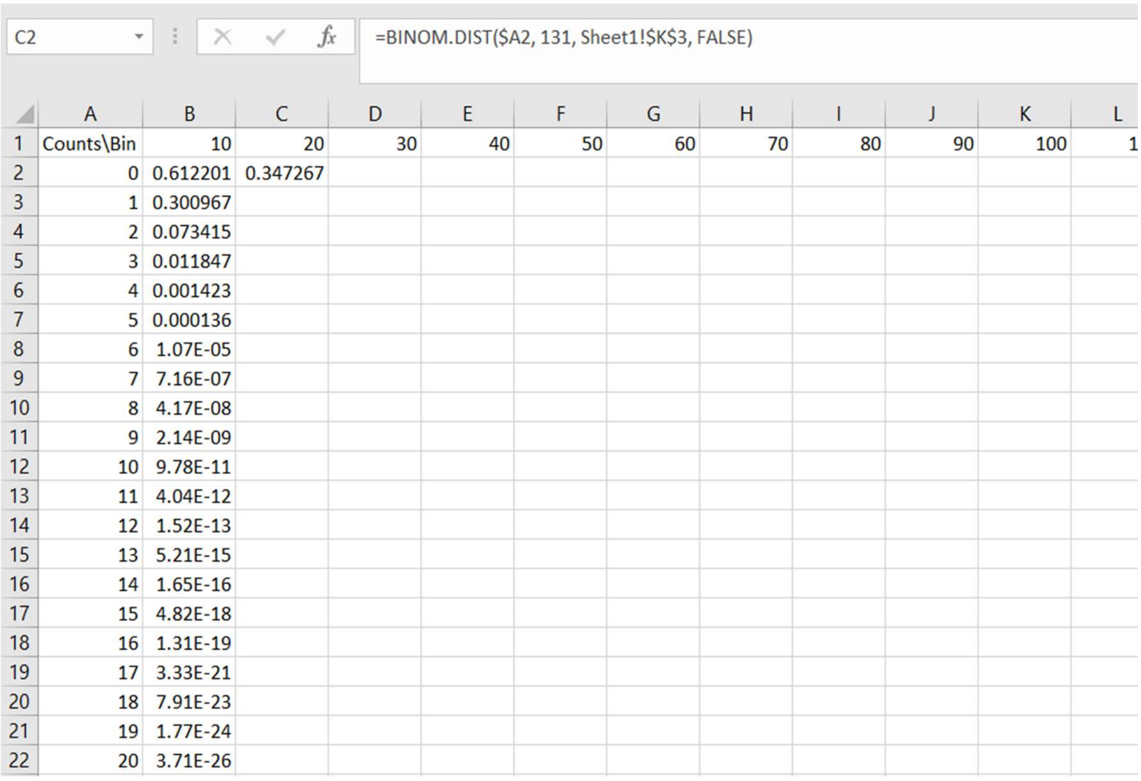
Figure 5

Step5

Copy the formula to the rest of rows (Column B) by using AutoFilll function. And, in cell C2, type in the following formula to calculate the probability.

=BINOM.DIST($A2, 131, Sheet1!$K$3, FALSE)

* Change the value from $2 in the formula described at Step 4 to $3 (colored in Red).



C2 =BINOM.DIST($A2, 131, Sheet1!$K$3, FALSE)

	A	B	C	D	E	F	G	H	I	J	K	L
1	Counts\Bin	10	20	30	40	50	60	70	80	90	100	1
2	0	0.612201	0.347267									
3	1	0.300967										
4	2	0.073415										
5	3	0.011847										
6	4	0.001423										
7	5	0.000136										
8	6	1.07E-05										
9	7	7.16E-07										
10	8	4.17E-08										
11	9	2.14E-09										
12	10	9.78E-11										
13	11	4.04E-12										
14	12	1.52E-13										
15	13	5.21E-15										
16	14	1.65E-16										
17	15	4.82E-18										
18	16	1.31E-19										
19	17	3.33E-21										
20	18	7.91E-23										
21	19	1.77E-24										
22	20	3.71E-26										

Figure 6

Repeat Step 5 until all bins and counts are filled.

* The value colored in Red in Step 5 should be added one by one every time changing the column. Figure 7 shows the sample output.

	A	B	C	D	E	F	G	H	I	J	K	L
1	Counts\Bin	10	20	30	40	50	60	70	80	90	100	110
2	0	0.612201	0.347267	0.178132	0.086292	0.03651	0.013891	0.00483	0.001486	0.000398	7.76155E-05	1.11761E-05
3	1	0.300967	0.368778	0.309351	0.213407	0.122393	0.060387	0.026287	0.009921	0.003214	0.000761716	0.000133136
4	2	0.073415	0.194316	0.266566	0.26187	0.203587	0.130251	0.070996	0.032865	0.012864	0.003709195	0.000786948
5	3	0.011847	0.067734	0.151954	0.212579	0.224025	0.185854	0.126844	0.072021	0.034062	0.011948721	0.003077167
6	4	0.001423	0.017571	0.064462	0.128421	0.183453	0.197353	0.168652	0.117453	0.067117	0.02864473	0.008954422
7	5	0.000136	0.003618	0.021706	0.061579	0.119245	0.166342	0.17799	0.152039	0.104975	0.054506937	0.020682722
8	6	1.07E-05	0.000616	0.006043	0.024413	0.064082	0.115916	0.155305	0.162717	0.135744	0.085752029	0.039496939
9	7	7.16E-07	8.92E-05	0.00143	0.00823	0.029284	0.068688	0.115231	0.148082	0.149263	0.11471751	0.064137474
10	8	4.17E-08	1.12E-05	0.000294	0.002408	0.011616	0.035329	0.074211	0.116974	0.142463	0.133209361	0.090402407
11	9	2.14E-09	1.24E-06	5.33E-05	0.000621	0.004062	0.016022	0.042141	0.081472	0.11989	0.136386278	0.112351543
12	10	9.78E-11	1.23E-07	8.61E-06	0.000143	0.001268	0.006486	0.021362	0.050656	0.090066	0.124653318	0.12464513
13	11	4.04E-12	1.09E-08	1.26E-06	2.97E-05	0.000357	0.002368	0.009763	0.028397	0.061006	0.102723512	0.124682196
14	12	1.52E-13	8.87E-10	1.67E-07	5.61E-06	9.14E-05	0.000786	0.004057	0.014472	0.037565	0.076956125	0.113381157
15	13	5.21E-15	6.58E-11	2.02E-08	9.69E-07	2.14E-05	0.000239	0.001543	0.006751	0.021174	0.052774015	0.094380209
16	14	1.65E-16	4.5E-12	2.26E-09	1.54E-07	4.62E-06	6.67E-05	0.00054	0.0029	0.01099	0.033323257	0.072338809
17	15	4.82E-18	2.84E-13	2.33E-10	2.27E-08	9.21E-07	1.73E-05	0.000175	0.001153	0.005278	0.019472216	0.051310045
18	16	1.31E-19	1.67E-14	2.24E-11	3.11E-09	1.71E-07	4.16E-06	5.28E-05	0.000426	0.002356	0.010576128	0.033828041
19	17	3.33E-21	9.17E-16	2.01E-12	3.97E-10	2.96E-08	9.33E-07	1.48E-05	0.000147	0.000982	0.005359805	0.020809527
20	18	7.91E-23	4.71E-17	1.69E-13	4.75E-11	4.8E-09	1.96E-07	3.9E-06	4.74E-05	0.000383	0.002543049	0.011984806
21	19	1.77E-24	2.27E-18	1.33E-14	5.33E-12	7.3E-10	3.87E-08	9.64E-07	1.44E-05	0.00014	0.00113306	0.00648175
22	20	3.71E-26	1.03E-19	9.89E-16	5.63E-13	1.05E-10	7.19E-09	2.24E-07	4.1E-06	4.83E-05	0.000475351	0.003300781

Figure 7

Step 6

In order to visualize the magnitude of probabilities, you may want to use "data bars" with conditional formatting. Select the cells which you would like apply the rules. On the home tab, click “Conditional Formatting”. Point to Data Bars, and then click “More rules” (Figure 8).

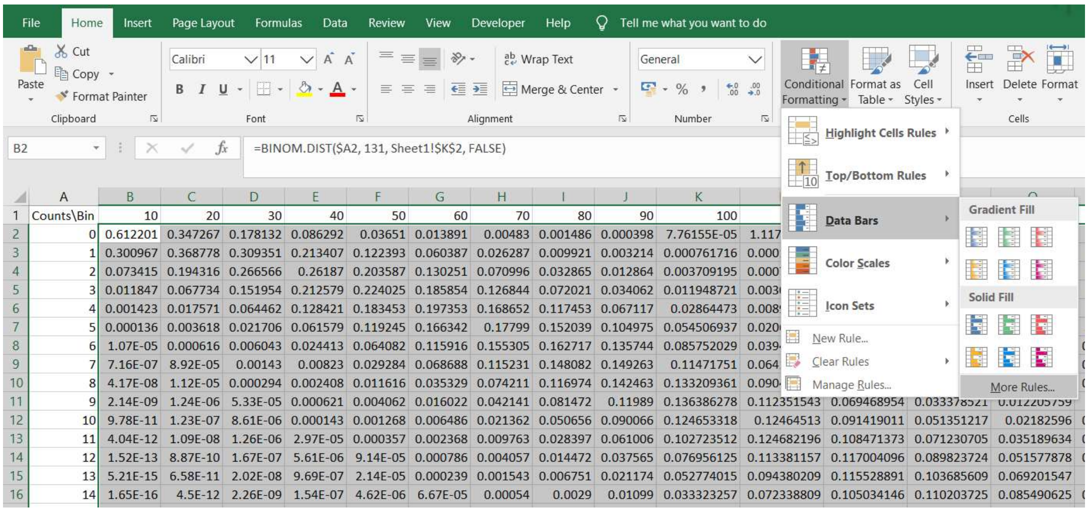


Figure 8

As Figure 9 shows, select “Number” from type drop down from Maximum and enter 1 in the value. Click “OK”.

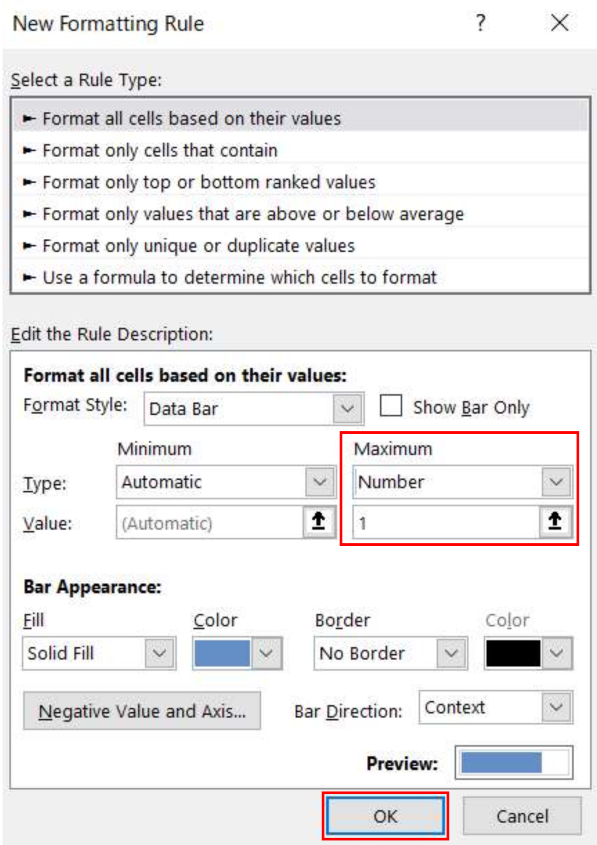


Figure 9

You obtain the bar charts with values in Figure 10, showing probabilities of a result of the Network Voronoi Cross K function method.

	A	B	C	D	E	F	G	H	I	J	K	L
1	Counts\Bin	10	20	30	40	50	60	70	80	90	100	110
2	0	0.612201	0.347267	0.178132	0.086292	0.03651	0.013891	0.00483	0.001486	0.000398	7.76155E-05	1.11761E-05
3	1	0.300967	0.368778	0.309351	0.213407	0.122393	0.060387	0.026287	0.009921	0.003214	0.000761716	0.000133136
4	2	0.073415	0.194316	0.266566	0.26187	0.203587	0.130251	0.070996	0.032865	0.012864	0.003709195	0.000786948
5	3	0.011847	0.067734	0.151954	0.212579	0.224025	0.185854	0.126844	0.072021	0.034062	0.011948721	0.003077167
6	4	0.001423	0.017571	0.064462	0.128421	0.183453	0.197353	0.168652	0.117453	0.067117	0.02864473	0.008954422
7	5	0.000136	0.003618	0.021706	0.061579	0.119245	0.166342	0.17799	0.152039	0.104975	0.054506937	0.020682722
8	6	1.07E-05	0.000616	0.006043	0.024413	0.064082	0.115916	0.155305	0.162717	0.135744	0.085752029	0.039496939
9	7	7.16E-07	8.92E-05	0.00143	0.00823	0.029284	0.068688	0.115231	0.148082	0.149263	0.11471751	0.064137474
10	8	4.17E-08	1.12E-05	0.000294	0.002408	0.011616	0.035329	0.074211	0.116974	0.142463	0.133209361	0.090402407
11	9	2.14E-09	1.24E-06	5.33E-05	0.000621	0.004062	0.016022	0.042141	0.081472	0.11989	0.136386278	0.112351543
12	10	9.78E-11	1.23E-07	8.61E-06	0.000143	0.001268	0.006486	0.021362	0.050656	0.090066	0.124653318	0.12464513
13	11	4.04E-12	1.09E-08	1.26E-06	2.97E-05	0.000357	0.002368	0.009763	0.028397	0.061006	0.102723512	0.124682196
14	12	1.52E-13	8.87E-10	1.67E-07	5.61E-06	9.14E-05	0.000786	0.004057	0.014472	0.037565	0.076956125	0.113381157
15	13	5.21E-15	6.58E-11	2.02E-08	9.69E-07	2.14E-05	0.000239	0.001543	0.006751	0.021174	0.052774015	0.094380209
16	14	1.65E-16	4.5E-12	2.26E-09	1.54E-07	4.62E-06	6.67E-05	0.00054	0.0029	0.01099	0.033323257	0.072338809
17	15	4.82E-18	2.84E-13	2.33E-10	2.27E-08	9.21E-07	1.73E-05	0.000175	0.001153	0.005278	0.019472216	0.051310045
18	16	1.31E-19	1.67E-14	2.24E-11	3.11E-09	1.71E-07	4.16E-06	5.28E-05	0.000426	0.002356	0.010576128	0.033828041
19	17	3.33E-21	9.17E-16	2.01E-12	3.97E-10	2.96E-08	9.33E-07	1.48E-05	0.000147	0.000982	0.005359805	0.020809527
20	18	7.91E-23	4.71E-17	1.69E-13	4.75E-11	4.8E-09	1.96E-07	3.9E-06	4.74E-05	0.000383	0.002543049	0.011984806
21	19	1.77E-24	2.27E-18	1.33E-14	5.33E-12	7.3E-10	3.87E-08	9.64E-07	1.44E-05	0.00014	0.00113306	0.00648175

Figure 10