IACR-Rothamsted
Integrated Approach to Crop Research

Patron: Her Majesty The Queen

Harpenden, Hertfordshire AL5 2JQ
Telephone: (01582) 763133
Fax: (01582) 760981
Director of Research: **Professor I Crute**
Plant & Invertebrate Ecology Division: Acting Head: **I Denholm**

12 October 2001

Dear Colleague,

Thanks for your interest in the SADIE system of spatial analysis. Free software now exists for the SADIE methods applied to data in the form of mapped data; this documentation covers the software available for 32-bit operating systems such as Windows95 or WindowsNT. The software was developed using Microsoft Fortran Powerstation; it is supplied both as an .EXE file and as the Fortran file from which it was compiled. Further details are given in the papers referred to in the documentation.

Conditions of use

The term software means all or any part of the code supplied. The copyright in this software is vested jointly in Rothamsted Experimental Station, Harpenden, Herts. AL5 2JQ UK, the emplopyer of the author Joe N. Perry, and in The University of Tokyo, Bunkyo-ku, Tokyo 113, Japan. The software is distributed free of charge and supplied to recipients only on the following conditions all of which must be agreed by the recipients before any use of the software is made:

The recipient recognises that the code for this software **sadiem.for** contains the code of software to construct Voronoi diagrams known as VORONOI2 produced by Kokichi Sugihara & Masao Iri of the University of Tokyo who hold the copyright of VORONOI2 and that no separate use of VORONOI2 should be made by the recipient without first obtaining permission from its authors. The software other than that of VORONOI2, i.e. that part written by Joe Perry, is supplied free on the condition that you accept the conditions of use outlined under the terms of the GNU General Public License version 2 as published by the Free Software Foundation, Inc., 59 Temple Place – Suite 330, Boston, MA 02111-1307, USA. No warranty is given with this distribution and no support is offered for those using this software. While every effort has been made to ensure that this software is free of defects no guarantee can be given as to its accuracy and no liability is accepted by the author J.N. Perry or his employer or The BBSRC or IACR Rothamsted or The Lawes Trust for any damage or loss of any form caused by its use.

I hope that the software will be used mainly for research purposes and that recipients will acknowledge its supply in any publication which arises from its use. I would be interested to receive a copy of any such publication.

In accepting free copies of the software **sadiem.exe**, the recipient is deemed to accept the above conditions of use.

Best Wishes,

Yours Sincerely,

Professor Joe N. Perry DSc
joe.perry@bbsrc.ac.uk

DOCUMENTATION FOR SADIEM.FOR

The program **sadiem.for** analyzes the spatial pattern of data that are in the form of maps, where the two-dimensional coordinates of every individual are known within a defined rectangular area.

The techniques and notation follow exactly those outlined in the paper: Perry, J.N. (1995). Spatial analysis by distance indices. *Journal of Animal Ecology*, **64**, 303-314. In order to understand the output from the program it is essential that this paper be read.

The paper describes techniques based on the measurement of the distance to regularity of the data, and explains the construction of a test based on a probability P_p, and an index I_p, each based on randomized simulations of the observed data. Please note that there is an error on p.309 of the paper, column 2, para 3 line 12: the text "values of D_{rand} less than" is wrong, and should be replaced by "values of D_{rand} greater than". The algorithm that moves the points towards the 'final' (approximately) regular spatial pattern utilises Voronoi polygons. These are computed using code kindly supplied by Professor Kokichi Sugihara of the University of Tokyo. I am deeply grateful to Professor Kokichi Sugihara for his permission to incorporate that code into **sadiem.for**. The simulations involve replacement, at random positions, of the original points within the defined area. The paper introduces two diagnostic plots: an 'initial-and-final' plot and an EDF plot.

In the software, the test and index are output directly. Also output are data that enable the construction, using the user's own preferred graphical package, of both the above mentioned plots, plus a plot of the 'final', regular configuration with its associated Voronoi polygons.

How to use the program

Channels 5 and 8 are for input. The input file for channel 5 is assumed to have the MS-DOS name: smi5.dat; this contains the raw counts, together with their spatial coordinates. Similarly, the file for channel 8 is assumed to be called smi8.dat; this contains parameters that control program execution. Examples of files smi5.dat and smi8.dat accompany the software.

Channels 1, 7 and 9 are reserved for use by the Voronoi algorithm, but do not usually involve any output and no files need normally be assigned to these channels. I have never experienced any data that has given rise to any error in the Voronoi algorithm, but if an error message was output it would appear on channel 7. The program creates an output file: smo10.dat on channel 10 that contains the minimal output required for an analysis and another output file: smo11.dat on channel 11 contains data required to construct, respectively, the 'final' regular arrangement with associated Voronoi polygons, the 'initial-and-final' plot, and the EDF plot. Note that the 'final' regular arrangement with associated polygons is defined in terms of *scaled* data, i.e. is scaled to lie in the working rectangle with corners defined by (0,0), (0,*A*), (1,*A*), (1,0), as described in the paper; the arrangement and polygons will have the correct shape but not necessarily be defined in terms of the original coordinates. In *all* the other output the data is already backtransformed to the original unscaled coordinates. The program creates a third output file: smo6.dat on channel 6 that contains much output that would normally only be required if the program had failed or given unexpected results that required some further checking.

Please note that if you try to run the program with any of these output files already present, the program will fail, without giving a recognisable error message. The files smo6.dat, smo7.dat, smo10.dat, smo11.dat must be renamed or deleted before running the program.

The form of the data in channel 10 is as follows:
D, the distance to regularity of original data
Out of S, the total number of simulated sets, the number (S-R) of randomised simulated sets with distance to regularity less than D, and the number (R) of randomised simulated sets with distance to regularity greater than D
P_p
E_p, *the average distance to regularity over the simulated sets.*
I_p, *the index of aggregation.*

The form of the data in channel 11 is as follows:
Data and Voronoi polygons for scaled final arrangement. The total number of Voronoi edges in the polygonal tessellation of the scaled final arrangement, is denoted here as m and denoted on the output by the term KEGTOT. Each of the m Voronoi edges in the polygonal scaled tessellation is defined using the x- and y- coordinates of its start point and its end point, respectively. The start and end points for edge j are denoted here as (x_{sj}, y_{sj}) and (x_{ej}, y_{ej}), respectively, $j = 1,...,m$, and denoted on the output by the terms FINXS, FINYS and FINXE, FINYE, respectively.

Data

scaled x-coordinate, unit 1	*scaled y-coordinate, unit 1*	*1 (reference no. of data value)*
scaled x-coordinate, unit 2	*scaled y-coordinate, unit 2*	*2 (reference no. of data value)*
scaled x-coordinate, unit 3	*scaled y-coordinate, unit 3*	*3 (reference no. of data value)*
.	.	.
.	.	.
.	.	.
scaled x-coordinate, unit n	*scaled y-coordinate, unit n*	*n (reference no. of data value)*
:		

Voronoi edges
m (KEGTOT)

(x_{s1}, y_{s1})	(x_{e1}, y_{e1})
(x_{s2}, y_{s2})	(x_{e2}, y_{e2})
(x_{s3}, y_{s3})	(x_{e3}, y_{e3})
.	.
.	.
.	.
(x_{sm}, y_{sm})	(x_{em}, y_{em})
:	

Data for i.a.f. plot

original (actual) data

x-coordinate, unit 1	*y-coordinate, unit 1*	*1 (reference no. of data value)*
x-coordinate, unit 2	*y-coordinate, unit 2*	*2 (reference no. of data value)*
x-coordinate, unit 3	*y-coordinate, unit 3*	*3 (reference no. of data value)*
.	.	.
.	.	.
.	.	.
x-coordinate, unit n	*y-coordinate, unit n*	*n (reference no. of data value)*
:		

final arrangement (unscaled)

x-coordinate, unit 1	*y-coordinate, unit 1*	*1 (reference no. of data value)*
x-coordinate, unit 2	*y-coordinate, unit 2*	*2 (reference no. of data value)*
x-coordinate, unit 3	*y-coordinate, unit 3*	*3 (reference no. of data value)*
.	.	
.	.	
.	.	
x-coordinate, unit n	*y-coordinate, unit n*	*n (reference no. of data value)*
:		

Data for EDF plot. The distances to regularity are considered unit by unit, prior to their totalling to give the value D. For each unit are output (in column order), (i) $_k d$, the distance for the kth ranked distance (starting with the smallest) of the n data values; (ii) $_k u$, the unit number of the ordered distance $_k d$; (iii) k; (iv) $_k 2.5$, the lower 2.5% bound of the ordered distances for the simulated data with rank k; (v) $_k E$, the average of the ordered distances for the simulated data with rank k; (vi) $_k 97.5$, the upper 97.5% bound of the ordered distances for the simulated data with rank k.

$_1 d$	$_1 u$	*1*	$_1 2.5$	$_1 E$	$_1 97.5$
$_2 d$	$_2 u$	*2*	$_2 2.5$	$_2 E$	$_2 97.5$
$_3 d$	$_3 u$	*3*	$_3 2.5$	$_3 E$	$_3 97.5$
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
$_n d$	$_n u$	*n*	$_n 2.5$	$_n E$	$_n 97.5$
:					

This is what you must do to run the program. First, put the n records in your data into a file smi5.dat, in the following form:

xcoord1	*ycoord1*
xcoord2	*ycoord2*
xcoord3	*ycoord3*
.	.
.	.
.	.
.	.
xcoordn	*ycoordn*

where the x & y coordinates should be read in as real numbers. No more than 2000 records can be analyzed in the current version, but you can always change the array sizes if you have more.

Next, you should, if necessary, specify other parameters in the input file smi8.dat. On the first line of this file you may want to specify a seed for the random number generator; you can do this by specifying this value as an integer '*iseed*' in the following form:

> *iseed*

Secondly, you may wish to specify S, the number of simulations to be done, as an integer '*nsims*' in the following form:

> *nsims*

where *nsims* should be no greater than 1000. If you don't specify these two parameters, then *iseed* will be set appropriately and *nsims* set to 160. Thirdly, the program needs to know the x- and y- coordinates of the top left-hand corner of the rectangle containing the data. These may be specified as real numbers '*tlx*, *tly*' in the following form:

> *tly*   *tlx*

If you don't specify these two parameters, then *tlx* will be set to zero and *tly* to unity. Fourthly the

program needs to know the x- and y- coordinates of the bottom right-hand corner of the rectangle containing the data. These may be specified as real numbers '*brx*, *bry*' in the following form:

   bry   *brx*

If you don't specify these two parameters, then *brx* will be set to unity and *bry* to zero. Finally, you may, if you wish, specify two parameters of the Aitken acceleration routine used to speed up convergence of the algorithm that finds the 'final' regular arrangement, but I recommend that these be left unspecified and the program allowed to select appropriate values. Note that no parameters on later lines of the file may be specified without you having specified all previous parameters in addition, e.g. you may decide to specify *iseed* and allow the program to select the default values of *all* the other parameters, but you cannot specify *tly*, *tlx* without having specified *iseed* and *nsims*.

That is all that is required. Good luck with the program! I will try to answer any problems you may encounter.

Joe N. Perry   8 June 1995