



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 the permutation of data in the form of a set of given counts at specified spatial locations to achieve particular degrees of spatial pattern; this documentation covers the software available for 32-bit operating systems such as Windows95 or WindowsNT.

There are two programs. The first achieves a degree of spatial pattern that matches the given data, the second achieves, or approaches as closely as possible, any desired degree of spatial pattern. 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 in Rothamsted Experimental Station AL5 2JQ UK the employer of its author J.N.Perry. 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 software 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 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, 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

The program **matchdtr.for** permutes a given set of data that are in the form of counts at specified spatial locations, for example numbers of moths in light-traps, numbers of plants in selected quadrats, where the spatial location of the traps and the quadrats are known. The permutations match the spatial pattern of the given counts as closely as possible, according to two criteria described in more detail below.

The techniques and notation follow exactly those outlined in the paper:

Perry, J.N. (1996). Simulation of counts in ecology and agriculture. *Computers & Electronics in Agriculture*, **15**, 93-109.

and the paper:

Perry, J.N., Bell, E.D., Smith, R.H. & Woiwod, I.P. (1996) SADIE: software to measure and model spatial pattern. *Aspects of Applied Biology*, **46**, 95-102.

In order to understand the output from the program it is essential that one or both of these papers be read.

These and earlier papers describe techniques based on the measurement of the distance to regularity of the data, and explain the construction of an associated index and test of spatial pattern. (Associated software in the same SADIE system is available for these operations). The degree of spatial pattern in the simulated patterns is assumed to be governed by this distance to regularity. The first, and simplest, criterion mentioned above concerns the accuracy of the simulations by specifying the proportion of the original distance to regularity that the distance to regularity of the simulated must be from the original to be acceptable. The parameter governing this, called EPSILO, has a real, positive, value, such that a value close to zero implies greater accuracy. For example, if EPSILO = 0.02 and the original distance to regularity = 38.1, then the distance to regularity for the simulated arrangement must be within (0.02×38.1) , i.e. 0.762, of 38.1; the allowable interval 38.1 ± 0.762 is then calculated as (37.338, 38.862). The second criterion relates to the degree to which the counts occur towards the edge of the sample area, which is known to affect the value of the distance to regularity. The simulations are constrained to try to ensure that their counts occupy the edge to a similar degree as the original. The degree to which this occurs is measured here as the distance between the centroid of the counts and the centroid of the sample units, calculated in the program as the variable DPCCC. Specifically, this second criterion specifies the proportion of the original value of DPCCC that the corresponding value for the simulated must be from the original to be acceptable. Once again, the parameter governing this, called EPSIL2, has a real, positive, value, such that a value close to zero implies greater accuracy. For example, if EPSIL2 = 0.01 and DPCCC = 4.43, then the distance corresponding to DPCCC for the simulated arrangement must be within (0.01×4.43) , i.e. 0.0443, of 4.43; the allowable interval 4.43 ± 0.0443 is then calculated as (4.3857, 4.4743).

Note that if for some reason it were desired to relax the condition that the value corresponding to the DPCCC of the simulated arrangement should be close to that of the original then this could be achieved easily by assigning EPSIL2 to be a large number, such as 10000.0D0.

Both EPSILO and EPSIL2 should be set by the user to optimum values. The smaller they are, the more accurate and well-constrained will be the resulting simulations, but the longer will be the computation time to achieve this. I would recommend a value of about 0.01 for EPSILO; an appropriate value of EPSIL2 is recommended by the program, which will be output if the program is run once as a preliminary trial.

How to use the program matchdtr.for

Channels 5 and 8 are for input. The input file for channel 5 is assumed to have the MS-DOS name: matchi5.dat; this contains the raw counts, together with their spatial coordinates. Similarly, the file for channel 8 is assumed to be called matchi8.dat; this contains parameters that control program execution. Examples of files matchi5.dat and matchi8.dat accompany the software.

Channels 6, and 7 are for output; files are produced by the program. The file matcho7.dat on channel 7 contains the simulated patterns. The file matcho6.dat on channel 6 contains a copy of the raw data, plus parameter values reflected from channel 8. The core of the program are in two algorithms, one published by myself in Perry (1996), and the other the transportation algorithm for determining the moves to regularity (this was adapted from code kindly supplied by Dr Les Proll of the University of Leeds); the output from this is unnecessary for reporting analyses and its action is made transparent to the user.

This is what you must do to run the program. First, put the n records in your data into file matchi5.dat, in the following form:

```
xcoord1 ycoord1 count1
xcoord2 ycoord2 count2
xcoord3 ycoord3 count3
. . .
. . .
. . .
. . .
xcoordn ycoordn countn
```

where the x & y coordinates should be read in as real numbers and the count as an integer, with no decimal point. No more than 2000 records can be analyzed in the version supplied.

Secondly, you should specify various control parameters in file matchi8.dat. These are: a seed for the random number generator, the number of simulations to be done, the value of EPSILO and the value of EPSILO2, in the following form:

```
iseed
nsims
epsilo
epsil2
```

where both the first two values are integer and both the last two are real. *iseed* should be less than 30,000 and *nsims* should be no greater than 1000.

For example, the file might appear thus:

```
24578
2
0.02
0.01
```

If you do not specify some of these parameters then they are set to appropriate values.

When running the program, make sure that you do not already have files with the names matcho6.dat or matcho7.dat. If these already exist from previous runs, you should rename or delete them before each new run.

That is all that is required. Good luck with the program! I will try to answer any problems you may encounter. Please read the conditions for use of this program, given at the end of this documentation, carefully.

The program is very similar in concept and execution to that of `matchdtr.for` described above. The program **desdtr.for** permutes a given set of data that are in the form of counts at specified spatial locations, for example numbers of moths in light-traps, numbers of plants in selected quadrats, where the spatial location of the traps and the quadrats are known. The permutations approach as closely as possible to a specified desired degree of spatial pattern of the given counts, according to two criteria. These criteria are essentially the same as those outlined above, except that EPSILO now governs the accuracy with which the simulated pattern achieves the specified, desired, distance to regularity, rather than that of the original data.

Again, both EPSILO and EPSIL2 should be set by the user to optimum values. The smaller they are, the more accurate and well-constrained will be the resulting simulations, but the longer will be the computation time to achieve this. This length of computation time may be very long because if a very small or large desired distance to regularity is specified, relative to that of the original data, then there will be very few, if any permutations able to satisfy the criteria for accuracy. The program will try a number of times to find a suitable permutation; each time it cannot find such a permutation it will say so, and output the stage it has got to in the algorithm for that try, which will usually be one of the best solutions possible, given the criteria. If the number of failures over all the requested simulations exceeds a user-defined value NTRIES, then the program will terminate. I would recommend a value of about 0.01 for EPSILO; an appropriate value of EPSIL2 is recommended by the program, which will be output if the program is run once as a preliminary trial.

How to use the program desdtr

Channels 5 and 8 are for input. The input file for channel 5 is assumed to have the MS-DOS name: `desi5.dat`; this contains the raw counts, together with their spatial coordinates. Similarly, the file for channel 8 is assumed to be called `desi8.dat`; this contains parameters that control program execution. Examples of files `desi5.dat` and `desi8.dat` accompany the software.

Channels 6, and 7 are for output; files are produced by the program. The file `deso7.dat` on channel 7 contains the simulated patterns. The file `deso6.dat` on channel 6 contains a copy of the raw data, plus parameter values reflected from channel 8. The core of the program are in two algorithms, one published by myself in Perry (1996), and the other the transportation algorithm for determining the moves to regularity (this was adapted from code kindly supplied by Dr Les Proll of the University of Leeds); the output from this is unnecessary for reporting analyses and its action is made transparent to the user.

This is what you must do to run the program. First, put the n records in your data into file `desi5.dat`, in the following form:

```
xcoord1 ycoord1 count1
xcoord2 ycoord2 count2
xcoord3 ycoord3 count3
. . .
. . .
. . .
. . .
. . .
```

. . .
. . .
xcoordn ycoordn countn

where the x & y coordinates should be read in as real numbers and the count as an integer, with no decimal point. No more than 2000 records can be analyzed in the version supplied.

Secondly, you should specify various control parameters in file *desi8.dat*. These are: the desired value of the distance to regularity of a simulated pattern, a seed for the random number generator, the number of simulations to be done, the value of EPSILO, the value of EPSILO2, and the maximum number of failures allowed before program termination, in the following form:

desdtr
iseed
nsims
epsilo
epsil2
ntries

where the first value is real, the next two values are integer, the next two are real, and the final value is an integer. *iseed* should be less than 30,000 and *nsims* should be no greater than 1000.

For example, the file might appear thus:

800.0
24578
2
0.02
0.01
6

If you do not specify some of these parameters then they are set to appropriate values.

When running the program, make sure that you do not already have files with the names *deso6.dat* or *deso7.dat*. If these already exist from previous runs, you should rename or delete them before each new run.

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