Technical Bulletin
1999 No. 3

IRRI Biodiversity Software Series. III. BOUNDARY: A Program for Detecting Boundaries in Ecological Landscapes

Developed by W.J. Zhang and K.G. Schoenly

IRRI
INTERNATIONAL RICE RESEARCH INSTITUTE

The International Rice Research Institute (IRRI) was established in 1960 by the Ford and Rockefeller Foundations with the help and approval of the Government of the Philippines. Today IRRI is one of the 16 nonprofit international research centers supported by the Consultative Group on International Agricultural Research (CGIAR). The CGIAR is sponsored by the Food and Agriculture Organization of the United Nations, the International Bank for Reconstruction and Development (World Bank), the United Nations Development Programme (UNDP), and the United Nations Environment Programme (UNEP). Its membership comprises donor countries, international and regional organizations, and private foundations.

As listed in its most recent Corporate Report, IRRI receives support, through the CGIAR, from a number of donors including UNDP, World Bank, European Union, Asian Development Bank, Rockefeller Foundation, and the international aid agencies of the following governments: Australia, Belgium, Canada, People's Republic of China, Denmark, France, Germany, India, Indonesia, Islamic Republic of Iran, Japan, Republic of Korea, The Netherlands, Norway, Peru, Philippines, Spain, Sweden, Switzerland, Thailand, United Kingdom, and United States.

The responsibility for this publication rests with the International Rice Research Institute.

IRRI Technical Bulletins

The IRRI Technical Bulletin is a rapid means of presenting results of research on a specialized technical subject such as the development of experimental methods, specialized software, or other solutions to complex research problems.

Mailing Address: MCPO Box 3127, 1271 Makati City, Philippines
Phone: (63-2) 845-0563, 844-3351 to 53
Fax: (63-2) 891-1292, 845-0606
Email: IRRI@CGIAR.ORG
Telex: (ITT) 40890 Rice PM: (CWT) 14519 IRILB PS
Cable: RICEFOUND MANILA
URL: http://www.cgiar.org.irri
Riceweb: http://www.riceweb.org
Riceworld: http://www.riceworld.org
Courier address: Suite 1099, Pacific Bank Building
6776 Ayala Avenue, Makati
Metro Manila, Philippines
Tel. (63-2) 891-1236, 891-1174, 891-1258, 891-1303

Suggested citation:
Zhang WJ, Schoenly KG. 1999. IRRI Biodiversity Software Series. III. BOUNDARY: a program for detecting boundaries in ecological landscapes. IRRI Technical Bulletin No. 3. Manila (Philippines): International Rice Research Institute. 17 p.

Cover: All photos from IRRI archives except for photo of rat, which was taken by N.Q. Hung.

ISBN 971-22-0135-X
ISSN

Technical Bulletin
1999 No. 3

IRRI Biodiversity Software Series. III. BOUNDARY: A Program for Detecting Boundaries in Ecological Landscapes

Developed by W.J. Zhang and K.G. Schoenly

IRRI
INTERNATIONAL RICE RESEARCH INSTITUTE

BOUNDARY: A Program for Detecting Boundaries in Ecological Landscapes

Developed by W.J. Zhang and K.G. Schoenly

Like other ecosystems, agricultural landscapes function as a semipermeable and fluid patchwork of habitats that harbor thousands of species. For many organisms, habitat structure is an important determinant of feeding success, survival, and reproduction (Bell et al 1991). Boundaries between habitat patches function as filters, barriers, and conduits that influence the direction and spread of biological and physical materials (Forman 1995). For example, although the presence of a road may not limit the movement of vertebrates, it can be a devastating barrier to insects (Samways 1994). Studies on the abiotic and biotic factors that influence the nature, location, and number of boundaries in ecological landscapes (Hansen et al 1988, Holland et al 1991) are encapsulated in the concept of "boundary dynamics" (Wiens et al 1985). Over broad spatial and temporal scales, boundaries brought about by agricultural activities show how societies have used landscapes as environments have changed (Hansen et al 1988).

Historically, agricultural research has focused on crop fields at the individual and aggregate levels but not on the larger surrounding landscape of which they are part. Consequently, studies of boundary dynamics in agroecosystems are few. Rice scientists, however, have begun to explore relationships between rice habitats and their surrounding nonrice habitats. For example, Yu et al (1996) showed that the egg parasitoids *Anagrus* spp. of rice leafhoppers and *Oligosita* spp. of planthoppers live in nearby *Echinochloa*- and *Leptochloa*-rich habitats, respectively, while parasitizing other leaf- and planthopper species there. Moreover, in a study of the distribution and abundance of 63 ant species living on bunds in irrigated fields, Way et al (1998) showed that the aggressive ant, *Solenopsis geminata,* preferred undisturbed bunds, and that it preyed on immature hemipterans, lepidopterans, and eggs of the golden apple snail (*Pomacea canaliculata*), an introduced and emerging pest of rice in Asia. Finally, Barrion et al (unpublished data) found that plowing grasses next to rice, a familiar farmer practice, brought a 7-fold increase in spider populations compared with unplowed controls and that predator populations increased faster in legume–rice systems than in rice–rice systems in Iloilo. As an ecorational feature of rice integrated pest management (IPM), the presence, strength, and physical nature of boundary effects between crop and noncrop habitats have implications for improving biological pest control through habitat manipulation, modification, and conservation.

Research issues

What constitutes a boundary to humans, such as a road or levee in an agricultural landscape, may not be perceived in the same way, if at all, by other organisms (e.g., microbes, insects, lizards, birds). Likewise, insects and lizards, owing to their small size, may detect additional boundaries not seen by humans. For example, if a levee restricts movements of certain organisms, species identities and abundances on both sides of the levee will be different enough to constitute a boundary, at least for these species. On the other hand, if a levee does not restrict organism movements, species identities and abundances will be similar on both sides of the levee, suggesting that these taxa "see" the landscape as

a single habitat. Detecting such taxon-specific boundaries (i.e., those independent of human perceptions) can help researchers learn which biocontrol linkages are active, diluted, or inactive in the presence of different boundaries imposed by farmers (throughout the text, taxa and species are used synonymously). Other, specific questions in rice pest management that can be answered by boundary detection analysis include the following:

1. Do boundaries perceived by herbivores and natural enemies coincide with each other and with human-imposed or perceived boundaries?
2. Which farmer-imposed boundaries (e.g., road, levee, rice-uncultivated edge, pesticide-sprayed alley) stop or restrict movements of herbivores and natural enemies?
3. Do taxon-imposed boundaries for different functional groups (e.g., herbivores, ground predators) remain fixed over a cropping cycle or do they change in location, number, strength, or width?
4. On balance, which noncrop habitats that border crop fields minimize boundary effects, harbor fewer pests, and share the largest fraction of natural enemies of crop pests?

Technical information

Two versions of BOUNDARY were written for use in MS-DOS (QBASIC™) and Windows™ (DELPHI-3™) environments to serve different operating systems in use at national agricultural research stations. Both versions use the same algorithm of Cornelius and Reynolds (1991) but vary in screen appearance and have different data/memory capacities. The QBASIC version of the program was developed using the Microsoft QBASIC language that runs under MS-DOS and Windows platforms and requires a maximum of 8 kb of system memory. The Windows-based version was created using developer tools and Windows interfaces contained within the DELPHI-3 development kit. The source code in the DELPHI-3 version is Object PASCAL™, which is Borland's object-oriented extension to PASCAL. The DELPHI-3 program requires 347 kb of memory to run. The DELPHI-3 version runs under advanced Windows environments such as Windows 95 and later.

Boundary detection

Botanists have used statistical methods for detecting boundaries in landscapes for decades to delineate biotic zones and realms (van der Maarel 1990, Ludwig and Cornelius 1987). Typically, such methods rely on the use of a species distance (or dissimilarity) computed from pairwise comparison of species lists. More sophisticated probability-based boundary detection methods (Cornelius and Reynolds 1991) have been combined with gradient-directed transects (Gillison and Brewer 1985) to capture changes along soil-water-plant gradients and to quantify boundary patterns at different spatial scales for different functional groups of organisms.

The program BOUNDARY uses the boundary detection algorithm of Cornelius and Reynolds (1991) for one-dimensional (transect) records. This method uses sliding-window analysis (SWA) and Monte Carlo simulation to plot, detect, and statistically verify taxon-defined boundaries. Because different taxa respond differently to the same landscape even at the same scale of resolution (Ludwig and Cornelius 1987), we recommend that this program be run separately for different functional groups of taxa (e.g., flying arthropods, ground-moving arthropods, vegetation, etc.) at the same window size.

The scale at which boundaries are detected also depends on the window size. Consequently, BOUNDARY allows you to select different window sizes for each analysis (Fig. 1). Which and how many window sizes should researchers use? One guiding principle is for you to choose a range of window sizes that varies between minimizing the effect of noise on the results (minimum window size) and detecting the same boundaries at a smaller resolution (maximum

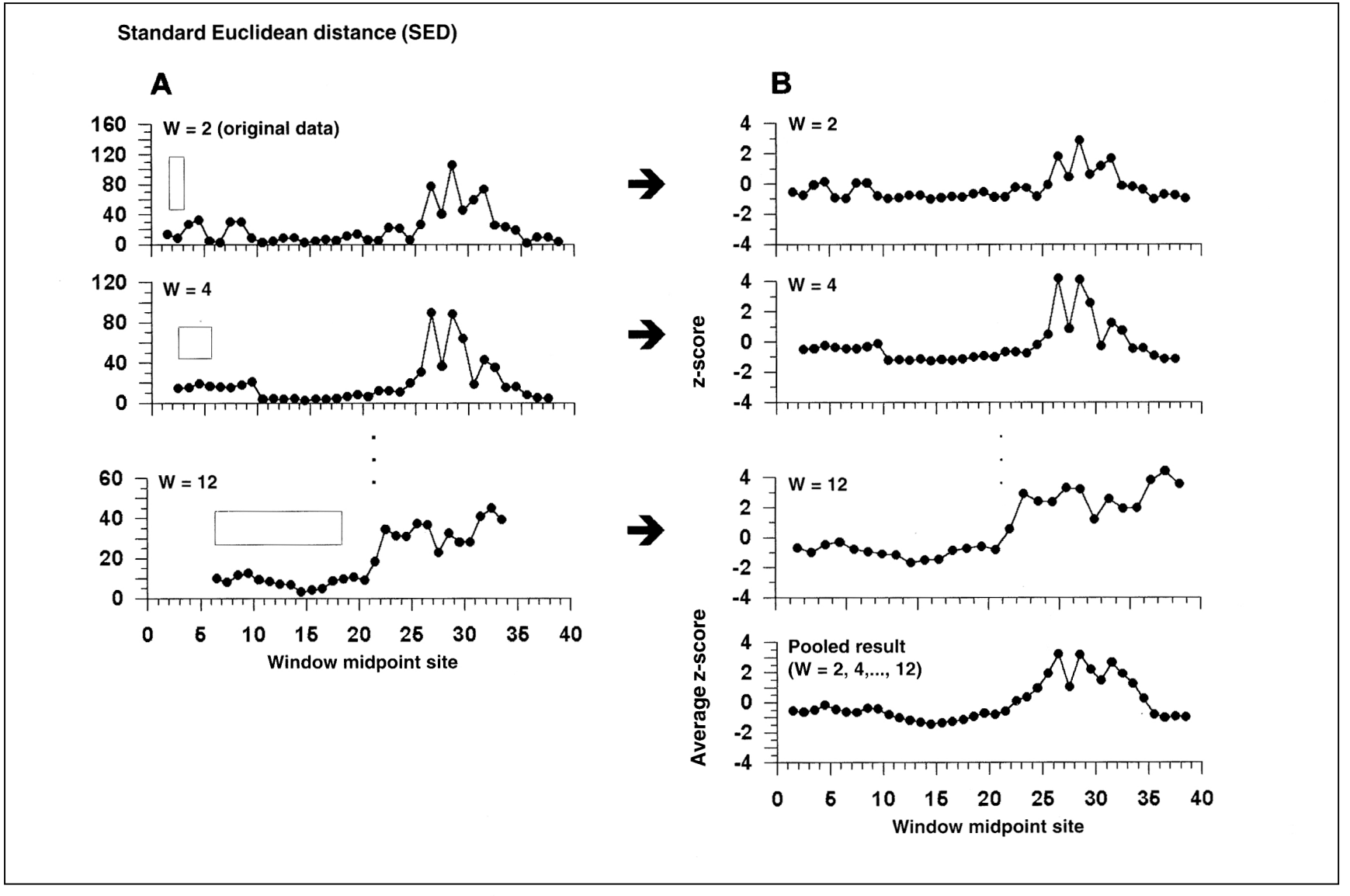


Fig. 1. Sliding window analysis and z-transformations of Euclidean distances for detritivorous invertebrates along a 39-site transect at IRRI at 50 days after transplanting: (A) plots of Euclidean distance for different window widths (W), (B) plots of z-transformations of Euclidean distances.

window size). For examples and further suggestions, we recommend that you consult Cornelius and Reynolds (1991).

BOUNDARY begins by placing a window containing data over adjacent sites along a transect and then dividing the window into two halves. The window has a predetermined width through which the data are "viewed" during the analysis (Fig. 1A). For each half-window, the mean abundance of each taxon in the sample is calculated (as in a moving average) and the two window halves are analyzed quantitatively using a dissimilarity index. The program then moves the window, one sample site at a time, along the transect and repeats the process to the end of the transect.

In BOUNDARY, you can use either of two quantitative distance measures of dissimilarity: standard Euclidean distance (SED) or Manhattan distance (MD). A large value in SED or MD for two neighboring sites suggests the presence of a boundary (peaks in Fig. 1A) that restricts species movements. In contrast, a small value in SED or MD suggests that organism movements are unimpeded at this site or that a human-imposed boundary at this site does not restrict species movements.

Calculation of sliding-window dissimilarities

Standard Euclidean distance and Manhattan distance are calculated as:

$$SED = [1/S \sum_{i=1}^{s} (X_{ik} - X_{jk})^2]^{1/2}$$

$$MD = 1/S \sum_{i=1}^{s} | X_{ik} - X_{jk} |$$

where X_{ik} and X_{jk} are sampled abundances of taxon k at neighboring sites i and j and S is species richness. Appendix 1 also shows these mathematical steps in detail.

Note that SED and MD correct for effects of sample size (species richness) by dividing the sum of the squared differences by S. Krebs (1999, p. 382) calls this version of SED the average Euclidean distance. In their boundary detection algorithm, Cornelius and Reynolds (1991) used the uncorrected version of SED, also called Euclidean distance.

The effectiveness of SED and MD in detecting boundaries is a function of the level of heterogeneity within landscape units. By systematically varying window size, different levels of spatial heterogeneity can be identified within a single data set (Fig. 1). SED is a quantitative measure of dissimilarity and it gives more weight to abundant taxa than to rare taxa. Consequently, we advise users of this program to consider the context of this weighting function for your particular research issue. In rice ecosystems, pest populations are often the most abundant herbivores, whereas their most effective arthropod biocontrol agents are among the most abundant natural enemies (K. Schoenly et al, unpublished data). Therefore, from an IPM and insect biocontrol standpoint, the use of SED or MD is scientifically justified.

Bootstrap procedures

Following the algorithm of Cornelius and Reynolds (1991), BOUNDARY uses a bootstrap procedure to estimate the mean SED and its standard deviation for calculating z-transformed SEDs (Fig. 1B) and to determine which boundaries are statistically significant. Bootstrapping reshuffles species abundances among transect sites. This has the advantage of maintaining the correlation structure of the data matrix (Fig. 2B). This step is repeated, at least 1,000 times, and the (expected) mean SED and its standard deviation are calculated for each midpoint site along the transect (Fig. 2C, 1–3). An overall mean SED and its standard deviation, taken over all transect positions, are calculated from the expected mean SEDs (Fig. 2C, 4). For each transect midpoint, a z-transformed SED is calculated using the original (observed) SED, the overall mean (bootstrapped) SED, and its standard deviation (Figs. 1B and 2C, 5–7). Cornelius and Reynolds (1991) reported that simulated runs, derived from ED dissimilarities, show that the distribution of EDs passed normality tests in most cases, indicating that the overall mean ED and its standard deviation are reasonable estimators of central tendency and disper-

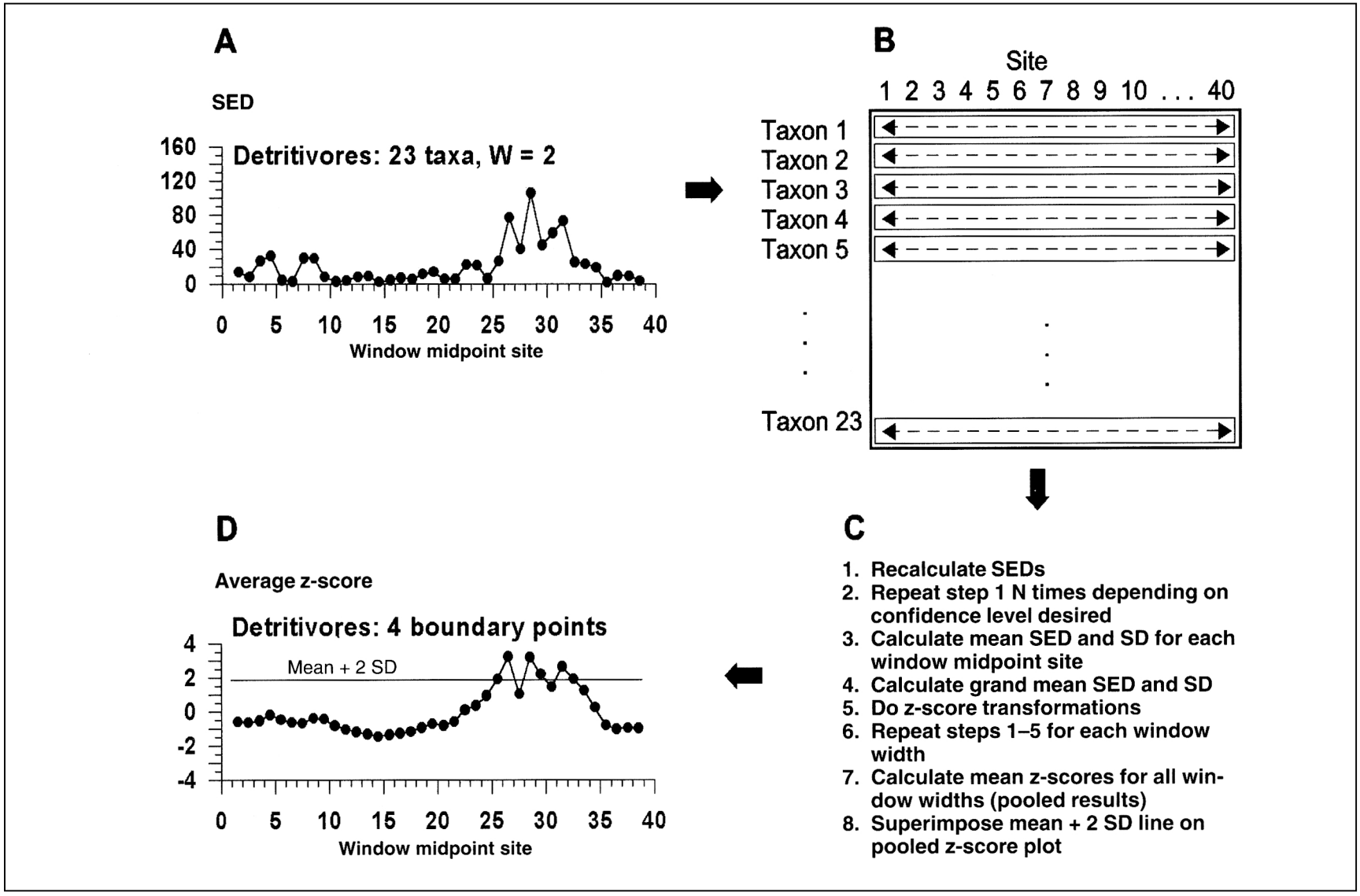


Fig. 2. Bootstrap procedure for statistically verifying boundaries in transect data based on the algorithm of Cornelius and Reynolds (1991): (A) plot observed SEDs for each window width (W), (B) reshuffle species abundances, (C) determine statistical significance, (D) plot observed and expected z-scores.

sion. These z-transformed dissimilarities permit different window sizes to be pooled and averaged to give a scale-independent profile of taxon-defined boundaries in an ecological landscape (Fig. 2D). Mathematical details of these bootstrap procedures are shown in Appendices 2–4.

Figure 2D shows the final plot of z-scores after pooling six different window widths (2–12). The threshold for statistically verifying which midpoint locations function as boundaries for these taxa is indicated by the horizontal (mean + 2 SD) line on the plot. Cornelius and Reynolds (1991) claim that dissimilarity peaks that extend above 2 standard deviations (z-score = 2) are statistically significant (at the nominal 0.05 level) and that this judgment is conservative for most cases. For the data in Figure 2D, detritivorous invertebrates partition the landscape into two zones, one at transect sites 1 to 26 and the other at sites 33 to 39, with a wide boundary in between.

Input file format

Data for BOUNDARY must be in the form of a data table or matrix in which taxa (immature or adult) correspond to rows and transect sampling sites correspond to columns. Cells of the matrix contain integers corresponding to sampled abundances separated by one or more spaces for clarity. The data file required by BOUNDARY must be a space-delimited text file. The DELPHI-3 version, unlike the QBASIC version, requires that input files have the extension ".txt" because this is a default extension for input files in the input file dialog boxes of BOUNDARY.

The space-delimited text format

The space-delimited format has been incorporated to make it easier to export data from spreadsheet files, such as Microsoft Excel or Corel Quattro Pro. The first row of each space-delimited text file consists of a numerical label for each transect site, separated by spaces (Fig. 3). Subsequent rows contain the taxon ID number, derived from the master lists, mastern.xls and mastera.xls, followed by the sampled abundance for that taxon in each transect site, all separated by spaces.

```
       1    2    3    4    5    6    7
 727  15    4    0    0    0    0    4
-618  10    0    0    0    0    0    0
-270   9    2    0    0    2    1    5
1000   8    0    8    8   11    2   29
 378   7    1    0    0    0    2    3
-620   7    0    1    0    0    4    5
-378   6    0    0    0    0    0    0
1079   6    1   10    2    2    5   20
 914   5    1    2    4    8    5   20
1094   5    0    0    0    0    0    0
-149   4    0    3    0    0    0    3
-581   4    0    3    0    2    0    5
```

Fig. 3. Sample data matrix containing the space-delimited file format.

Using Excel to create space-delimited text files

The following procedure describes how to create a space-delimited .txt file using Microsoft Excel 97:

1. Using the mouse, highlight the matrix you wish to create as a text file.
2. Choose the **Edit/Copy** command to copy the contents of the matrix in the clipboard.
3. Choose the **File/New** command to create a new file, then click **Edit/Paste** to transfer the contents of the clipboard to the new file.
4. If the file to be saved has 50–100 data columns, use the **Format/Column/Width** option to change the width of the first column (Taxon ID number) to seven spaces. Highlight the remaining columns with the mouse and again click **Format/Column/Width** and enter four spaces for these columns. This step will ensure that the width of this data matrix does not exceed 240 spaces, the maximum width allowed by Excel.
5. Choose the **File/Save As/Formatted Text (space-delimited)(*.prn)** and click **Save**.
6. In the file name box, type in a file name. Change the ".prn" extension (entered by default by MS Excel) to ".txt".
7. A warning box appears informing you that the "selected file type will save only the active sheet." Click **Save**.

If your spreadsheet software is other than Excel, consult your user manual for instructions on how to create space-delimited text files.

Program installation

The QBASIC version of the boundary detection program (BOUNDARY.BAS) requires the executable file qbasic.exe to run. You should copy these files into the same MS-DOS or Windows directory on the hard drive. The DELPHI-3 version (BOUNDARY.EXE) is a stand-alone executable file that does not require .dll (dynamic link library) files. As an ordinary executable file, BOUNDARY.EXE can be copied into MS-DOS, Windows, and other environments that have screen resolution of 800 × 600 pixels or more.

Starting BOUNDARY

For the QBASIC version, double-click **qbasic.exe**, then click **File/Open** to see and select **BOUNDARY.BAS**. After viewing the program text, click **Run** and follow the data input instructions. In the DELPHI-3 version, double-click the program icon in Windows Explorer or in the Program Manager to run the program. After a few seconds, a red window pane appears containing different-colored buttons and options for data input. You can find additional details by clicking the **Help** button along the bottom panel.

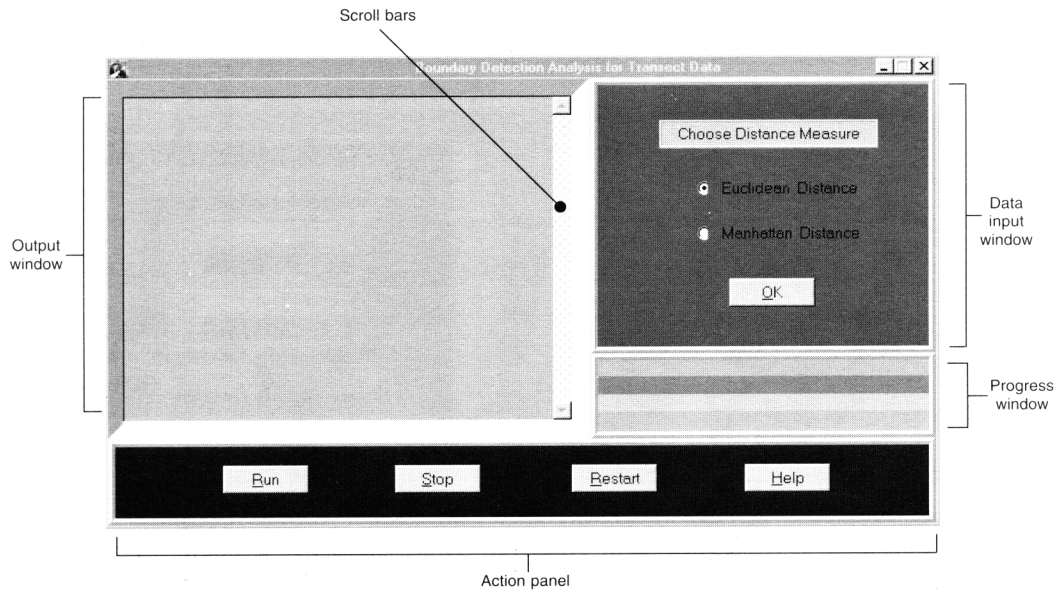
Running the program

Because of space limitations, our tutorial below will show only the DELPHI-3 (Windows) version of BOUNDARY. The QBASIC version lacks the familiar Windows format but retains inputting instructions and the boundary detection algorithm like those of the DELPHI-3 version.

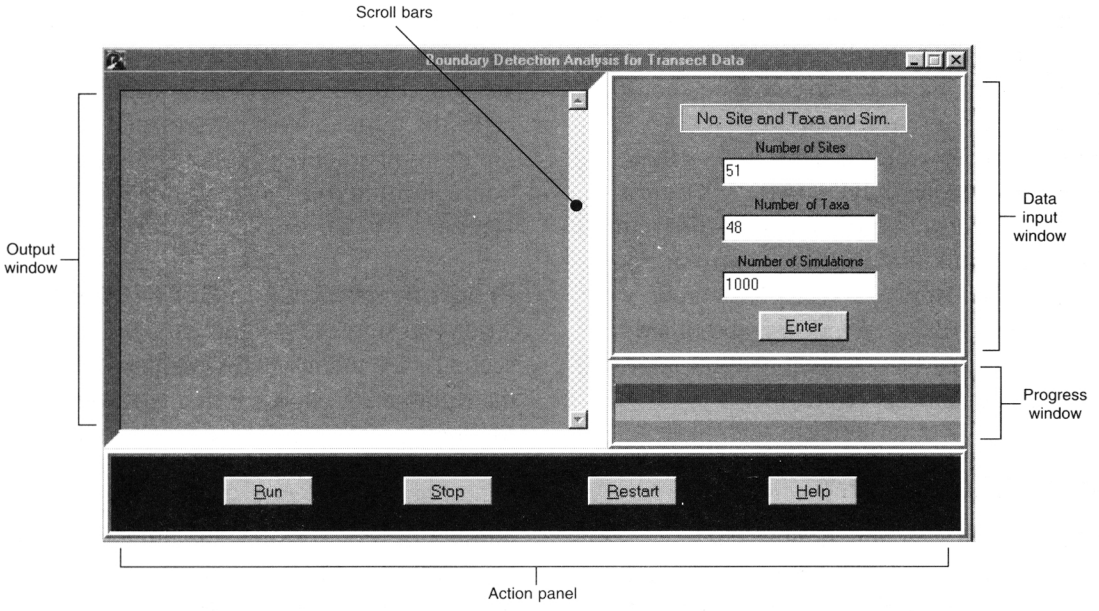
Program windows in BOUNDARY

After you start the program and wait several seconds, the main window appears (Fig. 4). In the right panel, choose either Euclidean or Manhattan distance for computing site-pair dissimilarities. In the second program window, the middle-right panel requires you to enter the number of transect sites, the number of taxa, and the number of Monte Carlo simulations desired (e.g., 1,000) and click **Enter** (Fig. 5). In the third program window, the middle-right panel requires you to specify the minimum window width (must be an even number), the number of window widths (e.g., 6), and the window width step expressed as an even number (e.g., 2). Then click **Enter** (Fig. 6). The middle-right panel of the fourth window requires you to specify the path and file name of the transect data set to be analyzed (Fig. 7).

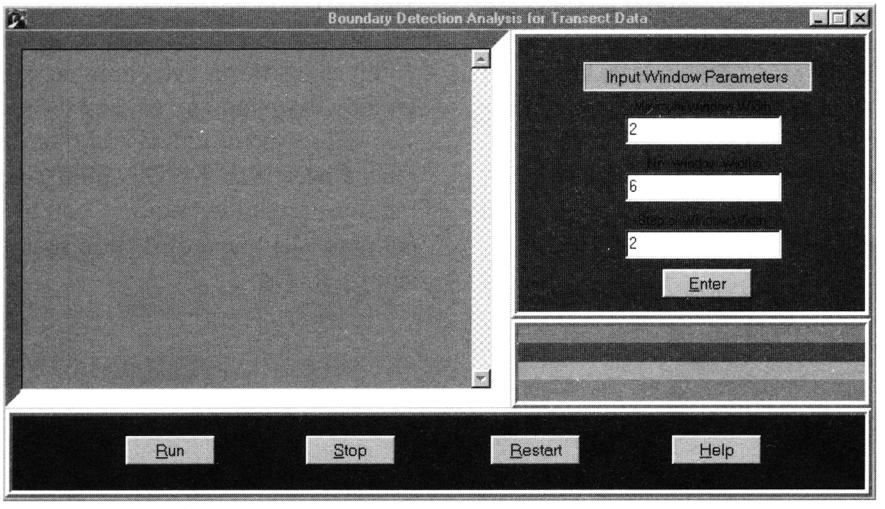
Fig. 4.



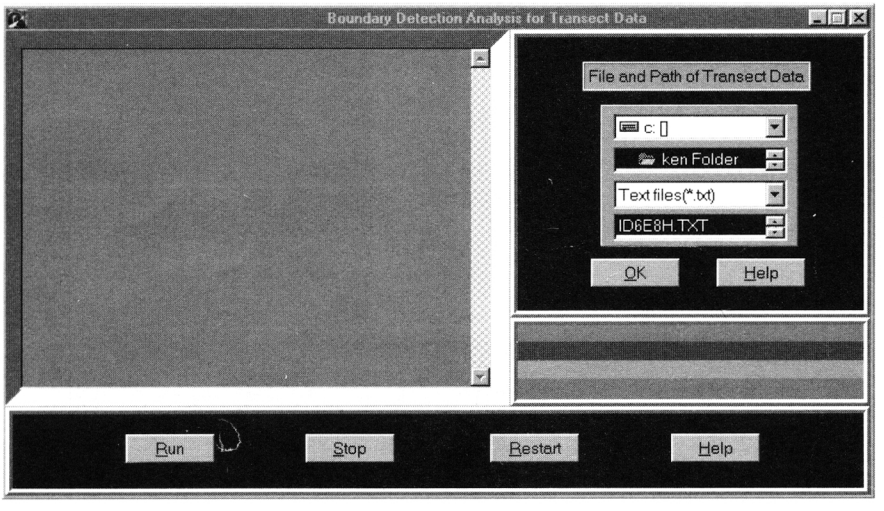
Figs. 5, 6, and 7.



5



6



7

8

After inputting the final data details, click **Run** to run BOUNDARY. Hint statements, activated when the mouse pointer approaches an input box, specify the nature of the parameter needed for each step of input. The taskbar at the bottom is an action panel that includes options to **Run** the program (after data inputting is complete), **Stop** it (during program execution), **Restart** the program, and seek **Help**. In the right corner of the window, two buttons familiar to Windows users let you **Minimize** the window but keep the program running (_) and **Close** the window so you can exit the program (×).

While running, BOUNDARY will display approximate running time and remaining time on the time bar. Running time refers to the real time BOUNDARY has taken to run the program from execution, whereas remaining time refers to the approximate time left to completion in minutes and seconds (Fig. 8). After execution, you can scan the results on the display by scrolling up on the scroll bar.

Output files

After program execution, the interface for saving the results will appear automatically. At this time, you can save the output as a series of text and graph files. The first file that you can save is the text file of all boundary results. Output in this file starts with results from the minimum (W = 2) window width (SEDs and z-score transformations) and continues to the last (maximum) window width (W = 12), followed by the pooled z-scores for all window widths (Fig. 9). The output for each window width also shows the sites that cross thresholds of varying statistical significance (mean + 1, 2, and 3 standard deviations). You can direct results to a printer (Fig. 10) or save the graph by specifying a unique file name in the next set of panels (Fig. 11).

Fig. 8.

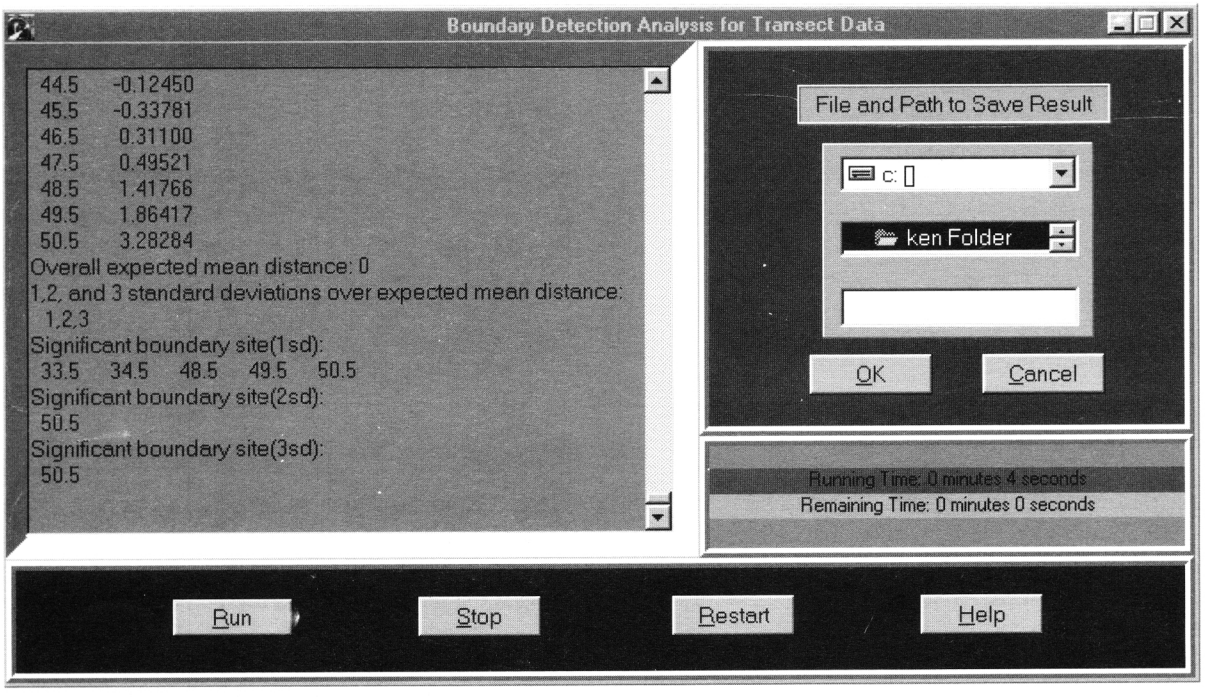


Fig. 9. Sample output of boundary detection results.

```
w=2
Distances
   1.5          0.66144
   2.5          0.54006
   3.5          0.87797
              .
              .
              .
  50.5          7.31722
Overall expected mean distance:
   1.93803
1,2,and 3 standard deviations over
expected mean distance:
   1.64914,   3.29828,   4.94742
Significant boundary site(1sd):
  32.5  33.5  39.5  49.5  50.5
Significant boundary site(2sd):
  32.5      50.5
Significant boundary site(3sd):
  50.5
Z-scores
   1.5        -0.77410
   2.5        -0.84769
   3.5        -0.64279
              .
              .
              .
  50.5         3.26181
w=4
Distances
   2.5          0.58630
   3.5          0.59073
   4.5          0.50000
              .
              .
              .
  49.5          3.29615
Overall expected mean distance:
  1.50370
1,2,and 3 standard deviations over
expected mean distance:
  0.99211,   1.98422,   2.97634
Significant boundary site(1sd):
  32.5  33.5  34.5  39.5  48.5  49.5
Significant boundary site(2sd):
  32.5      34.5      48.5
Significant boundary site(3sd):
None
Z-scores
   2.5        -0.92469
   3.5        -0.92023
   4.5        -1.01168
              .
              .
              .
  49.5         1.80670
w=6
Distances
   3.5          0.44876
   4.5          0.44876
   5.5          0.63099
              .
              .
              .
   48.5          2.79011
Overall expected mean distance:
    1.27279
1,2,and 3 standard deviations over
expected mean distance:
    0.73752,   1.47503,   2.21255
Significant boundary site(1sd):
   33.5  34.5  35.5  47.5  48.5
Significant boundary site(2sd):
   48.5
Significant boundary site(3sd):
None
Z-scores
   3.5        -1.11729
   4.5        -1.11729
   5.5        -0.87021
              .
              .
              .
  48.5         2.05734
w=8
Distances
   4.5          0.47735
   5.5          0.52167
   6.5          0.90355
              .
              .
              .
  47.5          2.17945
Overall expected mean distance:
   1.12529
1,2,and 3 standard deviations over
expected mean distance:
   0.59761,   1.19523,   1.79284
Significant boundary site(1sd):
   29.5  33.5  34.5  43.5  46.5 47.5
Significant boundary site(2sd):
None
```

```
Significant boundary site(3sd):
None
Z-scores
   4.5       -1.08421
   5.5       -1.01006
   6.5       -0.37104
             .
             .
             .
  47.5        1.76394
w=10
Distances
   5.5         0.73258
   6.5         0.89629
   7.5         0.81854
             .
             .
             .
  46.5         1.62275
Overall expected mean distance:
   1.01665
1,2,and 3 standard deviations over
expected mean distance:
   0.50621,   1.01242,   1.51862
Significant boundary site(1sd):
  28.5  33.5  34.5  43.5  46.5
Significant boundary site(2sd):
None
Significant boundary site(3sd):
None
Z-scores
   5.5       -0.56119
   6.5       -0.23778
   7.5       -0.39138
             .
             .
             .
  46.5        1.19733
w=12
Distances
   6.5         0.77691
   7.5         0.72048
   8.5         0.54645
             .
             .
             .
  45.5         1.55140
Overall expected mean distance:
   0.93430
1,2,and 3 standard deviations over
expected mean distance:
   0.45043,   0.90086,   1.35129
Significant boundary site(1sd):
  24.5  27.5  28.5  33.5  34.5  45.5
Significant boundary site(2sd):
  27.5
Significant boundary site(3sd):
None
Z-scores
   6.5       -0.34943
   7.5       -0.47470
   8.5       -0.86107
             .
             .
             .
  45.5         1.37001
Pooled distances
   1.5       -0.77410
   2.5       -0.88619
   3.5       -0.89344
             .
             .
             .
  50.5         3.26181
Overall expected mean distance: 0
1,2, and 3 standard deviations over
expected mean distance:
   1,2,3
Significant boundary site(1sd):
33.5  34.5  48.5  49.5  50.5
Significant boundary site(2sd):
  50.5
Significant boundary site(3sd):
  50.5
```

Fig. 10.

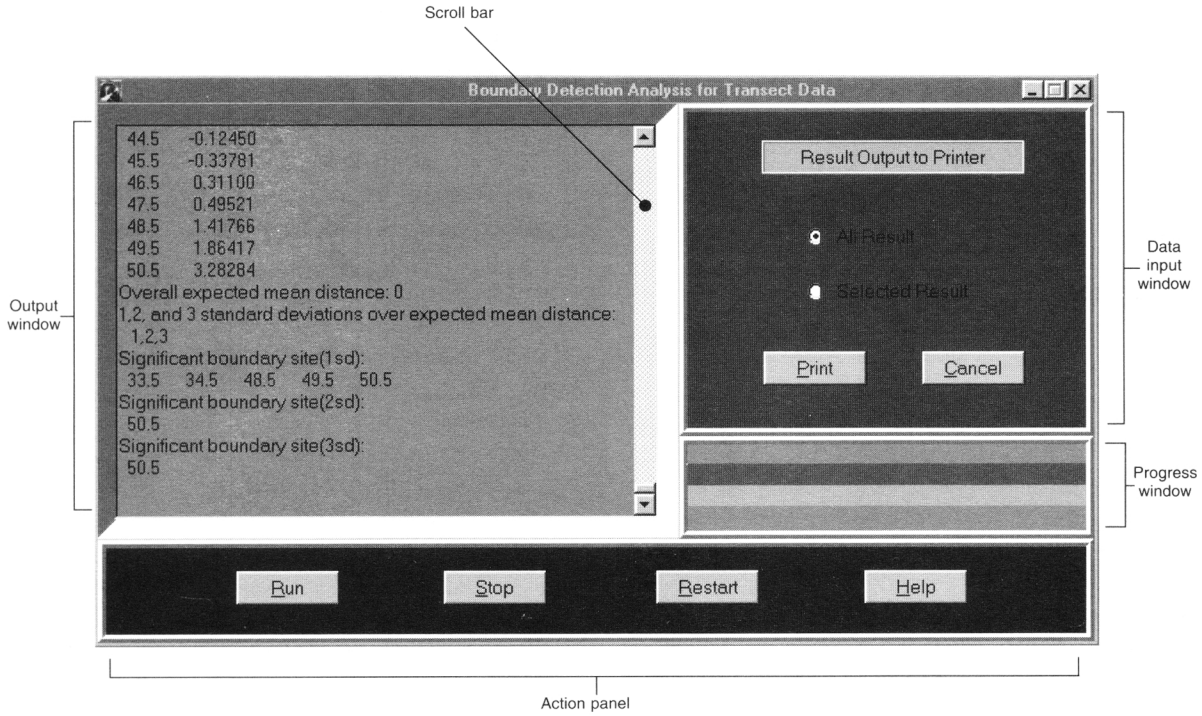
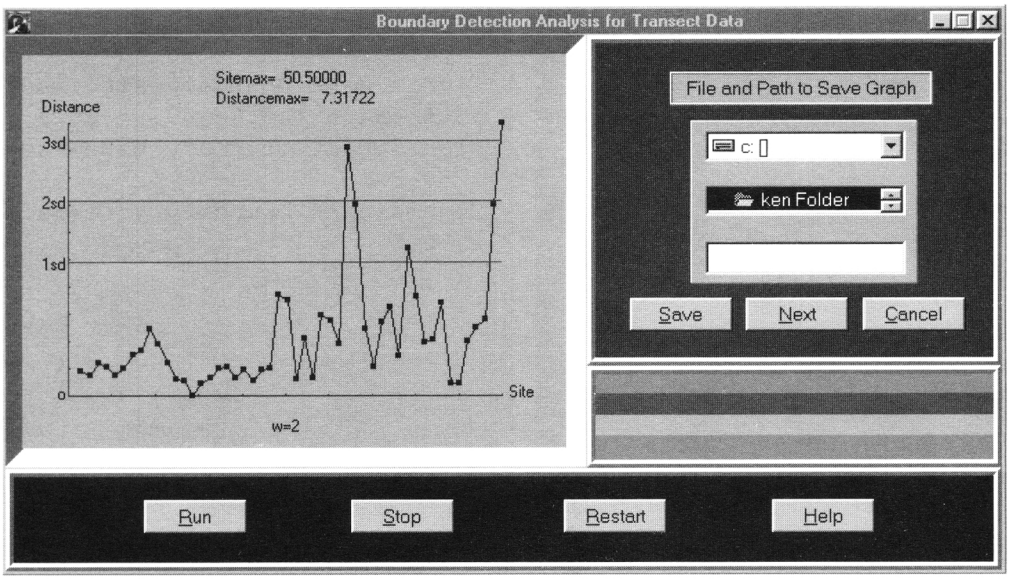

Scroll bar
Boundary Detection Analysis for Transect Data
44.5 -0.12450
45.5 -0.33781
46.5 0.31100
47.5 0.49521
48.5 1.41766
49.5 1.86417
50.5 3.28284
Overall expected mean distance: 0
1,2, and 3 standard deviations over expected mean distance:
1,2,3
Significant boundary site(1sd):
33.5 34.5 48.5 49.5 50.5
Significant boundary site(2sd):
50.5
Significant boundary site(3sd):
50.5
Result Output to Printer
All Result
Selected Result
Print
Cancel
Run
Stop
Restart
Help
Output window
Data input window
Progress window
Action panel

Fig. 11.


Boundary Detection Analysis for Transect Data
Sitemax= 50.50000
Distancemax= 7.31722
Distance
3sd
2sd
1sd
0
Site
w=2
File and Path to Save Graph
c: []
ken Folder
Save
Next
Cancel
Run
Stop
Restart
Help

Case study: 1996 transect, IRRI upland farm

Two pairs of perpendicular arranged transects, one pair oriented along N–S (NS1–36, NS1–39) and the other along E–W directions (EW1–51, EW1–52) and traversing 6 ha of rice fields and uncultivated habitats in the IRRI upland farm, were established in 1996. Transect sites were spaced at approximately 5-m intervals and were sampled for invertebrates and vegetation every 2 wk in the growing season using a suction sampler (and 0.16-m^2 enclosure) and pin-frame sampler, respectively, during both the dry and wet seasons. Edges in the landscape were distinguishable as roads (o), borders between rice and natural vegetation (r-w), and rice bunds (b).

Results of a sliding-window analysis of abundance data for EW1, calculated from standard Euclidean distance, were pooled for six window widths (2–12) and plotted for each of three crop stages (8, 50, and 64 DT) and one postharvest date (114 DT). At each end of EW1, boundaries (peaks) for vegetation (flora) were detected coinciding with abrupt changes in species composition between natural and cultivated (rice) vegetation (Fig. 12). During the crop cycle, however, boundaries were statistically significant (0.05) only at the east end (sites 1–12) of the transect. A gradual amplitude increase in several peaks in the right-most three-fourths of the transect coincided with re-vegetation of the bunds (Fig. 12).

Herbivorous arthropods did not exhibit the same boundary locations as the vegetation (Fig. 12). In contrast, boundary analyses for predaceous arthropods detected identical locations as vegetation at or near the border between rice and surrounding vegetation, but only at the east end of the transect. Thus, herbivorous arthropods tend to integrate the entire rice landscape as either a single or pair of habitats within the rice plots, whereas predaceous arthropods perceived the border between rice and surrounding native grasses as different habitats.

Execution Errors in BOUNDARY

If a problem develops during execution of these programs, an error will be displayed on the lower right panel. The list below includes some common errors and their explanation, which may help in troubleshooting:

Error	Explanation
Divided by zero with integer!	An integer value divided by zero
Divided by zero with floating point!	A floating point value divided by zero
Range check error!	Integer value exceeds defined range
Floating point overflow!	A floating point value exceeds upper limit
Floating point underflow!	A floating point value falls below lower limit
File cannot be accessed!	File not open, or read-only, etc.
Input or output error!	A nonfile, file has no data or too few or too many taxa or too few or too many transect sites
Invalid floating point operation!	Square root of negative value, etc.
Math or other error!	Not enough memory on hard disk, etc.

Other features of BOUNDARY

Using the Stop button

You can stop a program at any time during execution (after data entry) by clicking the **Stop** button near the bottom of the program window. If **Stop** is clicked, the program will return to its first program window (i.e., first parameter input window).

Saving output

The program window(s) for saving results will appear automatically following program execution. As each file save option appears, you can save any or all of the files by simply entering the path and a unique file name in the file-naming box. Likewise, if you wish to ignore certain files, simply click the **Cancel** button.

Printing results

After saving one or more files, the program box for printing appears. If you wish to print all

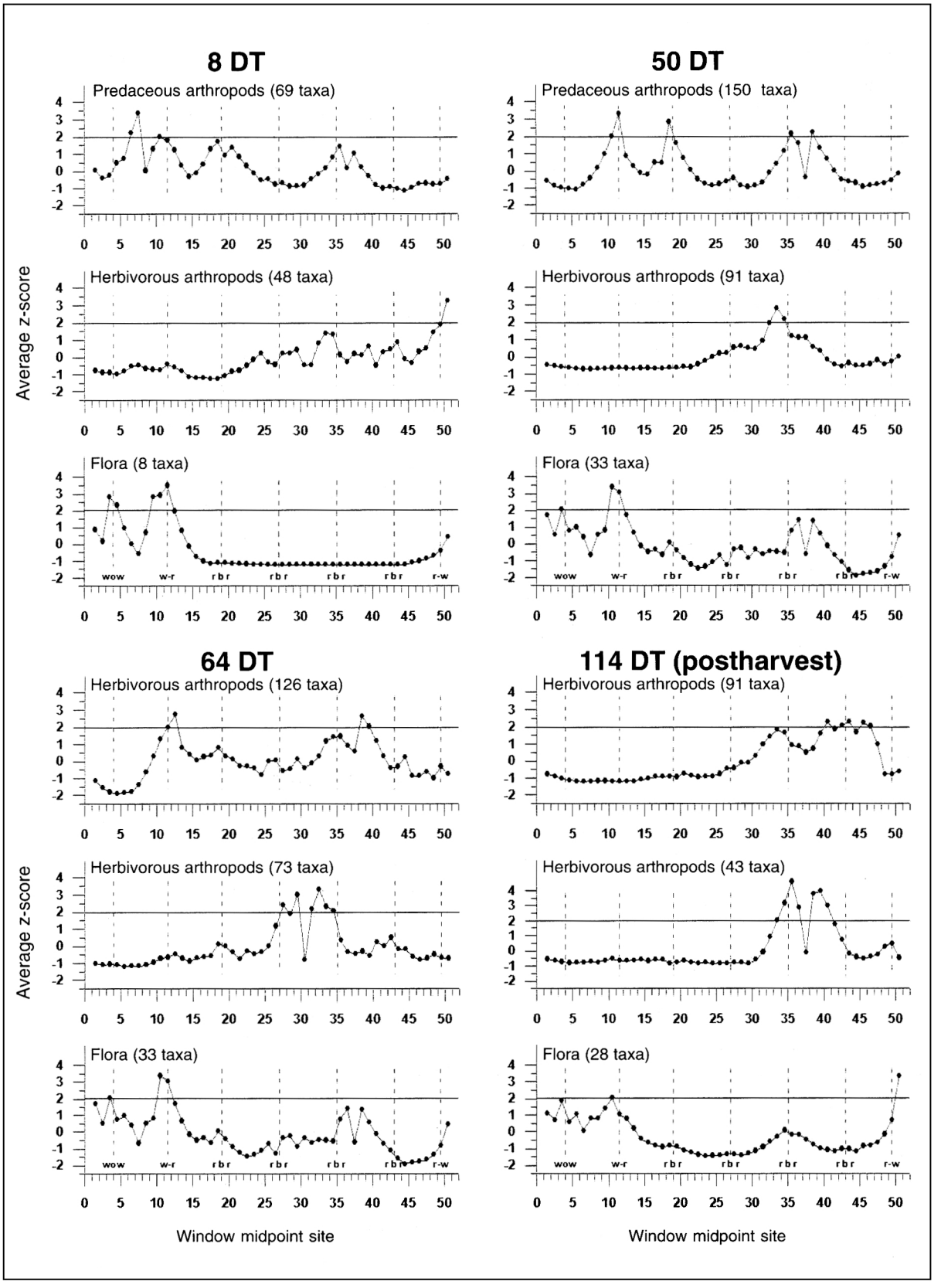


Fig. 12. Pooled z-score results of a sliding-window analysis of abundance data for vegetation, herbivores, and predators along a 51-site transect at IRRI at 8, 50, 64, and 114 days after transplanting.

results, then click the **All Results** and **Print** buttons. If you wish to print a partial set of results, then click the **Selected Results** and **Print** buttons. If no printing is desired, simply click the **Cancel** button.

Using the Restart and Help buttons

To return a program to the starting program window, you only need to click the **Restart** button. BOUNDARY.EXE comes with a **Help** button, located near the bottom of the program window. Information in **Help** includes the mathematical steps of the boundary detection algorithm of Cornelius and Reynolds (1991), which can be scrolled up and down using the vertical scroll bar in the Windows interface. The QBASIC version (BOUNDARY.BAS) also includes a help text.

Disclaimer

This software was tested on IBM-compatible PCs in 1998-99 using field data collected at IRRI in 1996. In this current version, we have made every effort to test BOUNDARY thoroughly and have corrected known programming errors. Nevertheless, a computer program subjected to repeated use by different users on different machines will invariably reveal additional errors. Should you uncover what you believe is a new programming bug, please advise us, preferably by e-mail (b.hardy@cgiar.org), and send us the following: (1) a description of the problem, (2) your computer model and processor, and (3) a copy of the data set you were using. We also welcome suggestions on how BOUNDARY can be improved.

References

Bell SS, McCoy ED, Mushinsky HR. 1991. Habitat structure: the physical arrangements of objects in space. New York (USA): Chapman and Hall.

Cornelius JM, Reynolds JF. 1991. On determining the statistical significance of discontinuities within ordered ecological data. Ecology 72:2057-2070.

Forman RTT. 1995. Land mosaics: the ecology of landscapes and regions. Cambridge University Press.

Gillison AN, Brewer KRW. 1985. The use of gradient directed transects or gradsects in natural resource surveys. J. Environ. Manage. 20:103-127.

Hansen AJ, de Castro F, Naiman RJ. 1988. Ecotones: what and why? Biol. Int. (Special Issue) 17:9-46.

Holland MM, Risser PG, Naiman RJ. 1991. Ecotones: the role of landscape boundaries in the management and restoration of changing environments. New York (USA): Chapman and Hall.

Krebs CJ. 1999. Ecological methodology. 2nd edition. New York (USA): Benjamin/ Cummings.

Ludwig JA, Cornelius JM. 1987. Locating discontinuities along ecological gradients. Ecology 68:448-450.

Samways MJ. 1994. Insect conservation biology. New York (USA): Chapman and Hall.

van der Maarel E. 1990. Ecotones and ecoclines are different. J. Veg. Sci. 1:135-138.

Way MJ, Islam Z, Heong KL, Joshi RC. 1998. Ants in tropical irrigated rice: distribution and abundance, especially of *Solenopsis geminata* (Hymenoptera: Formicidae). Bull. Entomol. Res. 88:467-476.

Wiens JA, Crawford CS, Gosz JR. 1985. Boundary dynamics: a conceptual framework for studying landscape ecosystems. Oikos 45:421-427.

Yu X, Heong KL, Cui H, Barrion AT. 1996. Role of non-rice habitats for conserving egg parasitoids of rice planthoppers and leafhoppers. In: Hokyo N, Norton G, editors. Pest management strategies in Asian monsoon agroecosystems. Kyushu National Agricultural Experiment Station, Ministry of Agriculture, Forestry, and Fisheries, Kyushu, Japan. p 63-77.

Acknowledgments

This project was supported by the Asian Development Bank, through RETA 5711, "Exploiting Biodiversity for Sustainable Rice Pest Management." We thank the participants of the three lighthouse sites from the Philippines (L. Flor, A. Velilla), China (J. Tang), and Vietnam (L.P. Lan) for beta testing these programs. For the data in the 1996 field study we cited and analyzed in this manual, we thank I. Domingo and V. Magalit for supervising the field study and for encoding the data in Excel spreadsheets; L. Datoon, A. Rivera, E. Hernandez, and E. Revilla for processing the invertebrate samples; E. Rico, R. Abuyo, J. Reyes, and B. Aquino for collecting the field samples; A.T. Barrion for his taxonomic expertise; and A. Pulpulaan for verifying the spreadsheet records against the raw data sheets. We also thank V. Salazar for producing the cover page for this software series, and the staff of IRRI's Communication and Publications Services, particularly B. Hardy, G. Reyes, and B. Lazaro.

Appendix 1. Calculation of split moving-window dissimilarities.*

Consider a data series of $j = 1, 2, 3, \ldots N$ ordered positions with $i = 1, 2, 3, \ldots V$ measurement variables at each position. X_{ij} is the abundance of the ith variable at the jth position along the series. Consider a window of width Q that brackets sequential positions along the series, as in the calculation of moving averages. A series of length N will have $N - Q$ sequential windows of width Q. If Q is a non-zero, even integer $<N$, then a window of width Q can be split into two equal halves, designated by W_A and W_B, with each half window having $Q/2$ sequential positions. The location of each window can be uniquely defined by the location of its window midpoint $k + 0.5$, where $k = Q/2, Q/2 + 1, Q/2 + 2, \ldots N - Q/2$. For each window midpoint location, the average of each variable i in each window half is given by

$$\bar{W}_{Ak+0.5,i} = \frac{\sum_{j=k-Q/2}^{k} X_{ij}}{Q/2} \qquad j = k - Q/2 + 1$$

and

$$\bar{W}_{Bk+0.5,i} = \frac{\sum_{j=k+1}^{k+Q/2} X_{ij}}{Q/2} \qquad N - Q + 1$$

A dissimilarity/similarity index ($DS_{k+0.5}$) can be calculated between each of the resulting $N - Q$ pairs of average vectors. For each window midpoint location, the standard Euclidean distance between average, half-window vectors is given by

$$DS_{k+0.5} = \left[\sum_{i=1}^{v} \left(\bar{W}_{Ak+0.5,i} - \bar{W}_{Bk+0.5,i} \right)^2 \right]^{1/2} \qquad N - Q + 1$$

* Corrections of mathematical errors in Cornelius and Reynolds (1991) are shown in the right margins in boldface.

Appendix 2. Calculation of overall expected mean dissimilarity and standard deviation.

Using Monte Carlo simulation techniques, an expected mean dissimilarity can be estimated for an SMW dissimilarity array calculated for a given window width Q. Each data vector is randomly repositioned along the data series for l = 1, 2, 3, ... M times, and SMW dissimilarities (as in Appendix 1) are calculated for each of the re-ordered data sets, resulting in an array of dissimilarities, $DR_{k+0.5,l}$. The mean expected dissimilarity and standard deviation for each k + 0.5 window midpoint location along the series is given by

$$\overline{DS}_{k+0.5} = \frac{\sum_{l=1}^{M} DR_{k+0.5,l}}{M}$$

and

$$SD_{k+0.5} = \frac{\left[\sum_{l=1}^{M} \left(DR_{k+0.5,l} - \overline{DS}_{k+0.5}\right)^2\right]^{1/2}}{M-1}$$

and the overall expected mean dissimilarity and average standard deviation for the series at window width Q is given by

$$\overline{DS}_{\cdot} = \frac{\sum_{k=Q/2}^{N-Q/2} \overline{DS}_{k+0.5}}{N-Q} \qquad N-Q+1$$

and

$$\overline{SD} = \frac{\sum_{k=Q/2}^{N-Q/2} SD_{k+0.5}}{N-Q} \qquad N-Q+1$$

$$SD_{k+0.5} = \sqrt{\frac{\sum_{l=1}^{M} \left(DR_{k+0.5,l} - \overline{DS}_{k+0.5}\right)^2}{M-1}}$$

Appendix 3. Z-score transformation of SMW dissimilarities.

For each window midpoint location, the SMW dissimilarity estimate from a window of a given width can be transformed to a standardized variable, or Z-score ($DZ_{k+0.5}$), relative to the overall expected mean dissimilarity and standard deviation for that window width (from Appendix 2) by

$$DZ_{k+0.5} = \frac{DS_{k+0.5} - \overline{DS}_{\cdot}}{\overline{SD}}$$

Appendix 4. Calculation of pooled SMW distances.

For each window midpoint location, a pooled SMW dissimilarity estimate can be calculated from the individual Z-score-transformed SMW dissimilarities from s = 1, 2, 3, ... T different window widths by

$$\overline{DZ}_{k+0.5} = \frac{\sum_{s=1}^{T} DZ_{k+0.5,s}}{T}$$