

The program CLUSTERXYC.EXE takes data in the form:

*x-coordinate*    *y-coordinate*    *cluster index*

where there is one such triplet per row, and the *x*- and *y*- values form a rectangular grid with distance apart  $d_x$  in the *x*- direction and  $d_y$  in the *y*- direction. The purpose of the program is to identify all patch and gap clusters, by assigning each unit either to a cluster or to a non-clustered area. A patch cluster is defined as a set of values of the cluster index that are all greater than some defined threshold,  $u$ , and that are contiguous on the grid to at least one other member of the set. Similarly, a gap cluster is defined as a set of values of the cluster index that are all less than (negatively exceed) some defined threshold,  $l$ , and that are contiguous on the grid to at least one other member of the set. An example is given below (Figure 1), for the data in the file raw\_clusters.dat, appended, with threshold values  $u=1.5$ ,  $l=-1.5$ .

Note that data in the precise format required by program CLUSTERXYC.EXE is produced by the program RBRELV13.EXE, in its output file CLUSTER.DAT (don't worry that that file has four columns instead of three, the extra column is ignored).

As well as identifying all the clusters, the program CLUSTERXYC.EXE distinguishes those units that are on boundaries of the grid, it defines cluster centroids and it examines minimum distances between the cluster centroids and between the edges of clusters. The program also supports the SURFER concept of 'blanked' (i.e. missing) data. To use the software, the data must be in a file named CLUSTERI5.DAT, and the thresholds  $u$  and  $l$  must be on rows one and two respectively of a file named CLUSTERI8.DAT. Three output files are produced: CLUSTERO6.DAT, which reflects the input data for the purpose of checking; CLUSTERO7.DAT, which contains full detailed output; and CLUSTERO9.DAT, which contains abbreviated output with reduced annotation in case the numerical output is required for inputting to some other program. As with all SADIE programs, run it by double-clicking on the .EXE file. Also, as with all SADIE programs, none of these output files should exist at the time the program is run, otherwise it will immediately crash; if any do exist they should be renamed.

It may be the case that the original data do not come from a rectangular grid, or perhaps it might be desired to produce a 'better' (finer-scaled) estimate of cluster size and/or position. In that case, contouring may be done, either to produce a rectangular grid of (estimated) cluster indices or to try to interpolate over a finer grid than the original. Many mapping, statistics, or geostatistics packages produce such interpolations, but no SADIE program exists to do this. However, if such a package has been used, and output produced in the same above format (*[x-coordinate, y-coordinate, cluster index]* in triplets, one triplet per row), then the program CLUSTERXYC.EXE may be used to identify clusters. An example is given below in the file gridded\_clusters.dat, where the original data from the example above have been kriged using a spherical model onto a 100x100 grid. The data are displayed below (Figure 2).

An example output file (renamed) is appended, for each of the two datasets above. The Fortran program is also appended in case you want to see how the program works. Don't forget to rename the input file as CLUSTERI5.DAT if you want to reproduce the output yourself.

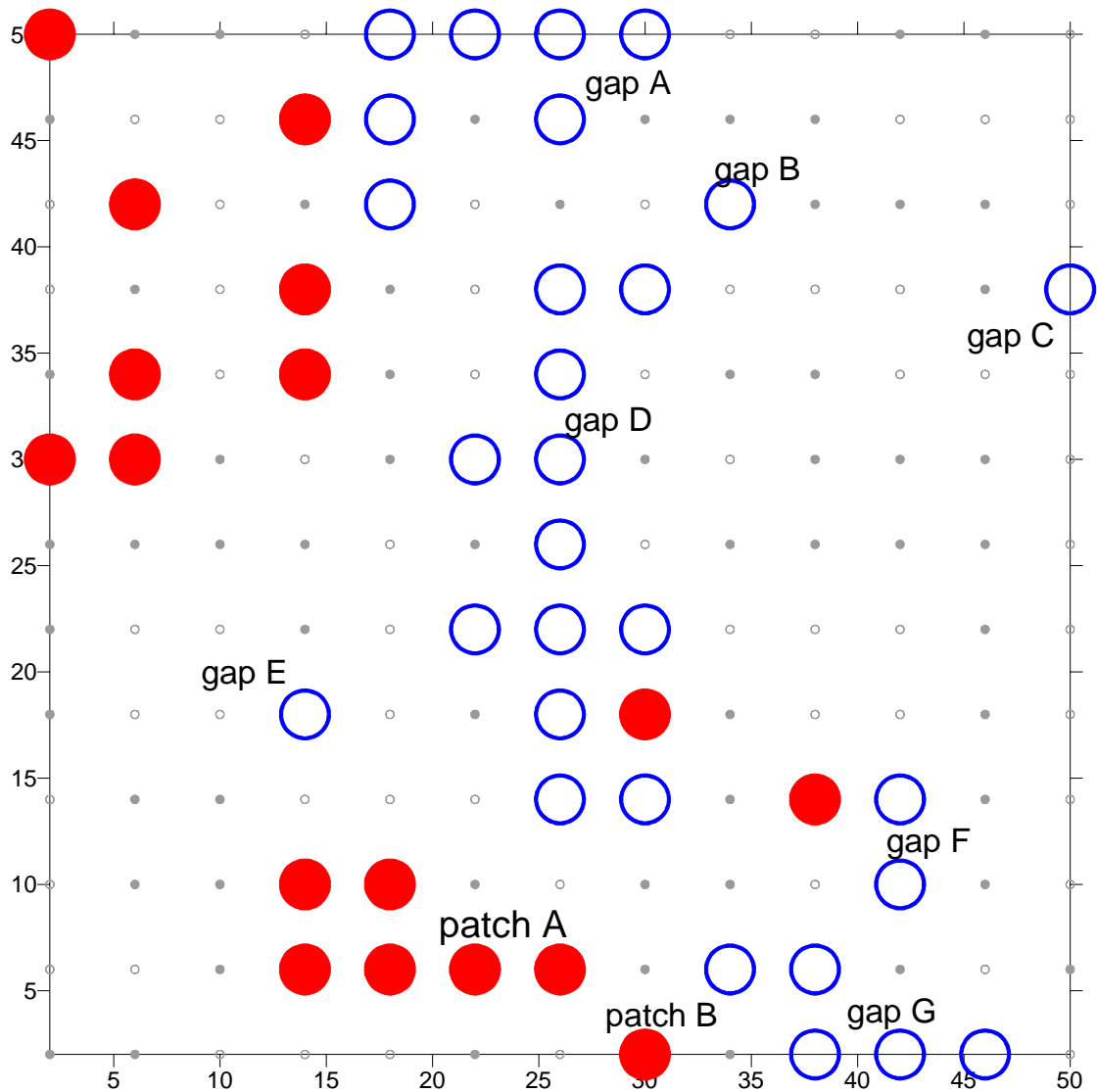


Figure 1. Data from file `raw_clusters.dat` is on a 13x13 grid, defined thus: values of  $d_x = d_y = 4$ ; minimum  $x$ -value = 2, maximum  $x$ -value = 50, minimum  $y$ -value = 2, maximum  $y$ -value = 50; threshold values  $u=1.5$ ,  $l=-1.5$ . Of the 169 cluster indices, 91 were negative and 78 positive. Of the 91 negative values, 29 negatively exceeded  $-1.5$  (large, open, blue symbols) and were, by definition, members of gap clusters. These 29 units formed the seven gap clusters shown. Note that the five members of gap G are each contiguous to at least one other member of that cluster, in the sense of having a neighbouring unit of the rectangular grid within the cluster, in either the  $x$ - or  $y$ - direction. (The unit at (38,2) is actually a nearest neighbour of two members of the cluster, the units at (42,2) and (38,6)). However, note that the unit at (42,10), while a member of a gap cluster, is not a member of gap G, since it is only diagonally adjacent to the unit at (38,6), and not a nearest neighbour in the  $x$ - or  $y$ - direction. Hence the unit at (42,10) must be a member of a separate cluster, here denoted gap F, together with the unit at (42,14). Similarly, of the 78 positive values, 17 exceeded 1.5 (large closed, red symbols) and were, by definition, members of patch clusters. These 17 units formed the nine patch clusters shown; again note that the patches labelled A and B shown are distinct.

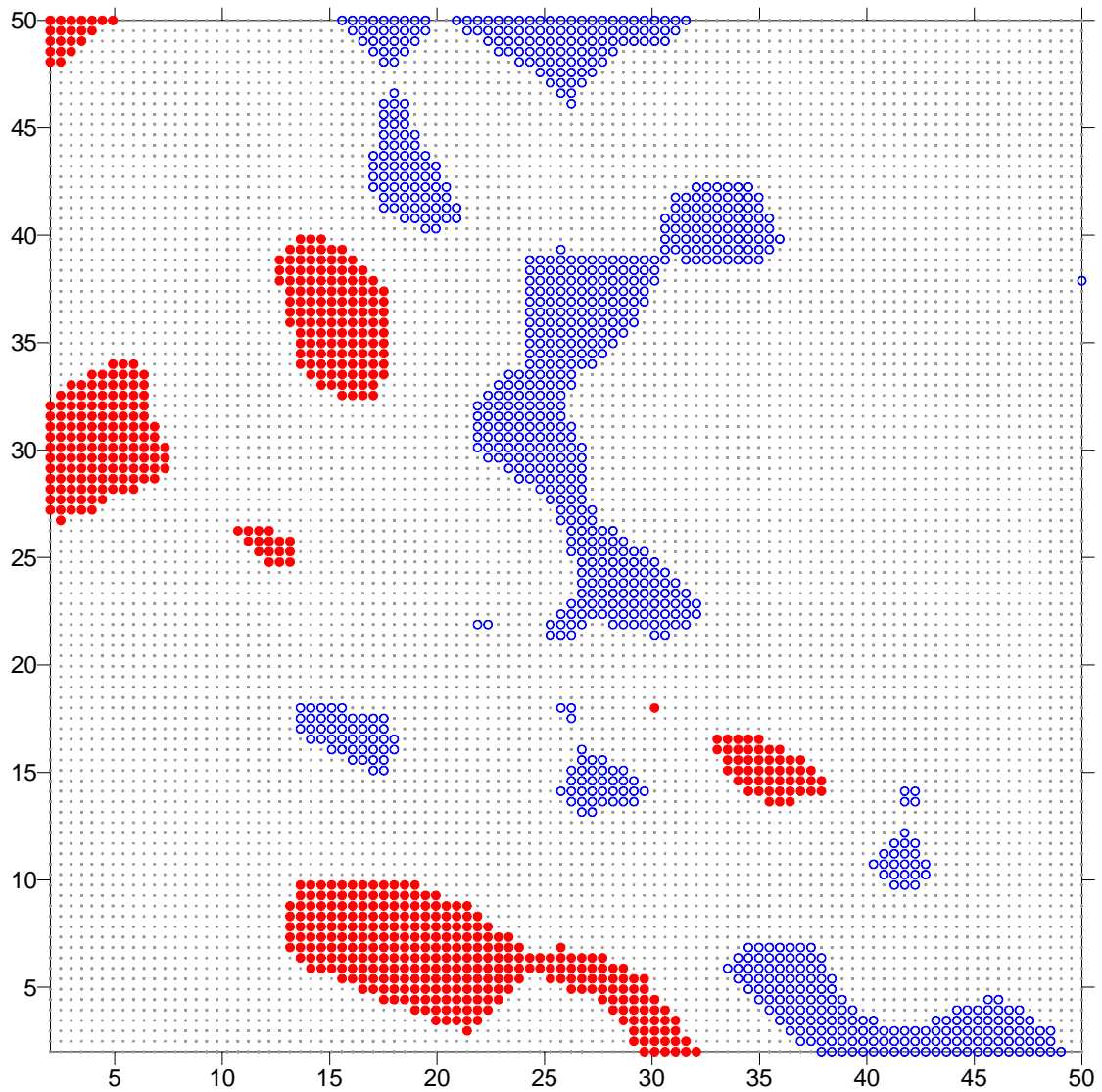


Figure 2. Data from file `gridded_clusters.dat` is on a 100x100 grid, defined thus: values of  $d_x = d_y = 4$ ; minimum  $x$ -value = 2, maximum  $x$ -value = 50, minimum  $y$ -value = 2, maximum  $y$ -value = 50; threshold values  $u=1.5$ ,  $l=-1.5$ . Of the 10000 cluster indices, 5251 were negative and 4749 positive. Of the 5251 negative values, 886 were estimated to negatively exceed  $-1.5$  (open, blue symbols) and were, by definition, members of gap clusters. These 886 units formed the twelve gap clusters shown. (Note by comparison with Figure 1 how some gaps have ‘coalesced’, thereby reducing the number of clusters, whilst even more new small clusters have been formed by interpolation, thus *increasing* the number overall.) Of the 4749 positive values, 655 were estimated to exceed 1.5 (closed, red symbols) and were, by definition, members of patch clusters. These 655 units formed the seven patch clusters shown. (Note how the number of patch clusters is *fewer* than in Figure 1.)