SPATCLUS: an R Package for Arbitrarily Shaped Multiple Spatial Cluster Detection for Case Event Data

Christophe Demattei, Nicolas Molinari

SPATCLUS: an R Package for Arbitrarily Shaped Multiple Spatial Cluster Detection for Case Event Data

Christophe DEMATTEI [a,*], Nicolas MOLINARI [a], Jean-Pierre DAURES [a]

[a] *Laboratoire de biostatistique, d'épidémiologie et de santé publique, UFR Médecine Site Nord UPM/IURC, 640 avenue du Doyen Gaston Giraud, 34295 Montpellier cedex 5, France.*

Abstract

This paper describes an R package, named SPATCLUS, that implements a method recently proposed for spatial cluster detection of case event data. This method is based on a data transformation. This transformation is achieved by the definition of a trajectory which allows to attribute to each point a selection order and the distance to its nearest neighbour. The nearest point is searched among the points which have not yet been selected in the trajectory. Due to the trajectory effects, the distance is weighted by the expected distance under the uniform distribution hypothesis. Potential clusters are located by using multiple structural change models and a dynamic programming algorithm. The double maximum test allows to select the best model. The significativity of potential clusters is determined by Monte Carlo simulations. This method makes it possible the detection of multiple clusters of any shape.

Key words: Spatial cluster detection test, Expected distance computation, Regression model, Dynamic programming algorithm, Numerical approximations

* Laboratoire de biostatistique, d'épidémiologie et de santé publique, UFR Médecine Site Nord UPM/IURC, 640 avenue du Doyen Gaston Giraud, 34295 Montpellier cedex 5, France. Tel.: +33 467 415 921; Fax.: +33 467 542 731.

Email address: `demattei@iurc.montp.inserm.fr` (Christophe DEMATTEI).

1 Introduction

A spatial cluster is an aggregate of points in $\mathbb{R}^p$ ($p > 1$) that are grouped together in space with an abnormally high incidence, which has a low probability to have occured by chance alone. Clusters of events are often reported to health agencies and an examination of the data is sometimes required for establishing an etiologic link between exposure and cluster existence. Location and detection of spatial cluster affects severals fields such as agronomy, medicine and social sciences.

Tests for spatial clustering have received substantial attention in the literature. A large number of tests have been proposed by different scientists in the different fields mentioned above. They can be classified according to their purpose. Tests for global clustering [1–5] are used to analyse the overall clustering tendency of disease incidence in the study area. The cluster location is unknown. Cluster detection tests [6,7] are concerned with local clusters. Potential clusters are located and their significance is tested. At last, focused tests [3,4,8] are used when a pre-specified focus is supposed to be linked to disease incidence.

This paper describes the implementation in R langage of a new method of detection and inference for multiple spatial clusters [9]. This method deals with precise events within $\mathbb{R}^2$, such as spatial coordinates for the occurrence of disease cases or the geographical positions of individuals. The approach, based on transformation of the data set and a regression model, is an extension of the method presented in Molinari et al. [10] for multiple temporal clusters. This new test belongs to the class of detection tests for case event data.

The following section briefly describes the method implemented in the SPATCLUS package. It begins with data tranformation by determining a trajectory and the weighted distances. The ordered weighted distances are then used in the cluster location and detection stages. In the third section, we present a decription of the SPATCLUS package. Data input, optional parameters, output and result vizualization are detailed, main algorithms are presented and explained. The use of the exportation module in SatScan [11] format is also detailed. In the fourth section, we apply the method to both simulated and real data. The paper is concluded by a discussion.

2 Methods

The goal of the method is to test the null hypothesis which corresponds to a uniform distribution of the events. We only present here essential background. A detailed presentation of the method is given in Dematteï et al. [9].

2.1 Data transformation

Let n be the number of events occuring in A, a bounded set of $\mathbb{R}^2$ or $\mathbb{R}^3$. The spatial coordinates of those n events are i.i.d random variables denoted $X_1, \ldots, X_n$.

The data transformation consists first in the determination of a trajectory constructed from initial data $x_1, \ldots, x_n$, where x_k is a realization of X_k. An order variable, that can be seen as an order of selection for the points in the trajectory, is constructed using a recursive algorithm initiated from the first order point $x_{(1)}$ which is arbitrarily chosen (see [9] for a discussion about the choice of the first point). Then, let $x_{(k)}$ be the point with selection order k. Given $x_{(1)}, \ldots, x_{(k)}$, the point $x_{(k+1)}$ is the nearest point from $x_{(k)}$ among the $n-k$ points not yet selected. A trajectory that links successively each point to the next order point is thus defined. The algorithm used to determine the trajectory is presented in Table 1.

We can now define the distance variable $D_k = d(X_{(k)}, X_{(k+1)})$ from one point to its nearest neighbour. $d_k = d(x_{(k)}, x_{(k+1)})$ is a realization of D_k. This distance has to be weighted both to correct high distances due to the elimination process of pre-selected points and to adjust for a potential inhomogeneity in the underlying population density. The weighted distance d_k^w is defined as the ratio between the distance d_k and its expectation under H_0, the uniform distribution hypothesis. Dematteï et al. [9] have shown that the expected distance can be written

$$E_{H_0}\left[D_k / X_{(1)} = x_{(1)}, \ldots, X_{(k)} = x_{(k)}\right] = \int_0^a \left[1 - \frac{\int_{A_{k-1} \bigcap S(x_{(k)}, r)} f(x)dx}{\int_{A_{k-1}} f(x)dx}\right]^{n-k} dr, \quad (1)$$

in which $f(x)$ is the underlying density from which the n points are sampled independantly, $S(x, r)$ is the sphere centered in x with radius r, and $A_k = A \setminus \left\{\bigcup_{i=1}^{k} S(x_{(i)}, d_i)\right\}$ with the convention $A_0 = A$.

The numerical integration of $\int_0^a$ in Equation (1) is achieved by using the trapezoidal rule. Moreover, the underlying population Z, constituted by N individuals $\{z_i : i = 1, \ldots, N\}$, allows to estimate the density integrals $\int_{A_{k-1}}$ and $\int_{A_{k-1} \bigcap S(x_{(k)}, r)}$. For any set $B \subset A$, $\int_B f(x)dx$ can be approximated by $\#\{i / z_i \in B\}/N$. This integral approximation allows to adjust the computation of d_k^w for inhomogeneous population. This adjustment is important since, with rare diseases, a large study area is necessary to examine data for evidence of spatial clustering. Hence, due to a natural inhomogeneity, the density of population at risk is not constant over the study area.

2.2 Cluster location and detection

Cluster bounds can now be determined from transformed data $(k, d_k^w)_{k=1,\ldots,T}$ in which $T = n - 1$. For this purpose we consider the weighted distance regression on the selection order k. To determine the presence of m breaks (denoted by $T_1, \ldots, T_m$), the regression function taken into consideration is:

$$f(t) = \sum_{j=1}^{m+1} \overline{d}_{[T_{j-1}+1;T_j]} \times I_{[T_{j-1}+1;T_j]}(t) \tag{2}$$

with the convention $T_0 = 0$ and $T_{m+1} = T$. The notation $\overline{d}_{[T_{j-1}+1;T_j]}$ indicates the mean of d_t^w for t in $[T_{j-1} + 1; T_j]$.

The minimum percentage of points between two breaks is a parameter which have to be taken into account. Let $\epsilon \in [0; 1]$ be this parameterμ. Then, the set of possible partitions is $\Delta_\epsilon = \{(T_1, \ldots, T_m)\ ;\ \forall i = 1, \ldots, m + 1,\ card\,([T_{i-1} + 1; T_i]) \geq |T\epsilon|\}$.

Breaks (cluster bounds) are estimated by

$$(\hat{T}_1, \ldots, \hat{T}_m) = \underset{(T_1,\ldots,T_m)\in\Delta_\epsilon}{argmin} \sum_{t=1}^{T} \left(d_t^w - f(t)\right)^2\ , \tag{3}$$

and are computed efficiently using a dynamic algorithm programming presented in section 3.5.

The double maximum test proposed by Bai and Perron [12] is used to select the best model. This test allows to test the the null hypothesis of no break against an unknown number of breaks given a certain upper bound M. Once the best model is selected, a p-value is computed for each portion between two breaks by a Monte Carlo procedure.

3 Package description

In this section, the content of the package is presented and the algorithms for the data transformation and the break location are emphasized. A flow chart describing this package is presented in Figure 1. The package implements essentially the method described in the previous section and its main function is *clus*( ). Because the spatial scan statistic [7] is a reference method, the package contains also an exportation module in the SatScan format [11].

[Fig. 1 about here.]

3.1 User interface

Once R has started up, a window called "R Console" appears. Within this window, the user types his commands and R displays the results of the required computations. Each command must be written at the right side of the ">" symbol. The result of a command can be stored in a R object by using the "$<-$" assignement operator. All the functions are called in the same way. For example the command

$$resclus <- clus(data = data\_ex, pop = pop\_ex, limx = c(0,1), limy = c(0,1))$$

will analyze the case coordinate data set *data_ex* with the population coordinate data set *pop_ex*. The study area is here defined to be the unit square. The results of this analysis will be store in a R list object called *resclus*.

In order to be able to use the SPATCLUS package, the user has to type the command

$$> library(spatclus)$$

which will load the package.

3.2 Data input

In $2D$, the *clus*( ) function has 4 essential arguments that have to be specified:

data: Data frame with 2 colums giving coordinates of cases.
pop: Matrix with 2 columns giving coordinates of underlying population individuals. This matrix is called *grille* in the R programs.
limx: 2 element vector containing the study area bounds of the X-axis.
limy: 2 element vector containing the study area bounds of the Y-axis.

In $3D$, the user also has to specified the parameter **limz**, a 2 element vector containing the study area bounds of the Z-axis.

3.3 Optional parameters

The *clus*( ) function also has several optional arguments that affect the different stage of the method. Default values (DF) are given for these parameters:

- Data input:

dataincyn (DF="n"): "y" means that cases are already included in the underlying population. "n" means appends that they are not and appends *data* to *pop* .

rndm (DF=NaN): Vector that identifies the rows containing cases coordinates in the grid (only if datainc="y").

- Trajectory:

  start (DF=1): Indicates the rank of the first trajectory point in term of distance from the area edges. 1 means that the first point of the trajectory is the nearest from the edge.

- Cluster location and detection:

  m (DF=5): Maximum number of breaks.

  eps (DF=0.2): Minimum size of cluster (ratio of the total number of cases).

- Spatial scan statistic location and module of exportation in SatScan format:

  method (DF=1): 1 for multiple break clusters, 2 for spatial scan statistic location, 3 for the 2 methods.

  methk (DF=3): In the spatial scan statistic location, 1 for Bernoulli model, 2 for Poisson model, 3 for both models.

  export (DF="n"): If method = 2 or method = 3, and if export = "y", the data will be exported in "repexport" directory in SatScan software format.

  repexport (no DF): If export = "y", defines the directory in which data will be exported in SatScan software format.

3.4 Data transformation algorithm

In this section, the algorithm used for the determination of the trajectory and the distance weighting is presented. The corresponding methodology is described in section 2.1.

In the algorithm given in Table 1 and written in pseudocode, $data = \{x_1, \ldots, x_n\}$ is the set of the n case locations and $pop = \{u_1, \ldots, u_N\}$ is the set of the N individual locations that belongs to the underlying population. The trajectory is initialized by chosing $x_{(1)}$ in the *data* set, and we consider it as given in the algorithm. This choice is debated in [9]. For a better comprehension, we chose to use a set language rather than a matrix language.

[Table 1 about here.]

Some explanations are necessary for a complete understanding of the correspondance between quantities used in this algorithm and those used in Equation (1). In the k^{th} iteration of the global "counting" loop:

- after the IF block, *pop* represents A_{k-1} and $\#pop$ is used to approximate the

quantity $N \times \int_{A_{k-1}} f(x)dx$,

- in the nested "counting" loop, *rpop* represents $A_{k-1} \bigcap S(x_{(k)}, r)$ and *#rpop* is used to approximate the quantity $N \times \int_{A_{k-1} \bigcap S(x_{(k)},r)} f(x)dx$,
- the nested "counting" loop allows to compute the quantity $pas \times \left(S - \frac{1}{2}\right)$ that represents an estimation of

$$\int_0^a \left[1 - \frac{\int_{A_{k-1} \bigcap S(x_{(k)},r)} f(x)dx}{\int_{A_{k-1}} f(x)dx}\right]^{n-k} dr$$

using the trapezoidal rule,

- the last step is to store the coordinates $x_{(k)}$ of the k^{th} case of the trajectory along with its associated weighted distance d_k^w.

3.5 Break location using a dynamic programming algorithm

Consider the regression of the ordered series of the weighted distances $\{d_k^w : k = 1, \ldots, n-1\}$ on the selection order k. The regression function is given in Equation (2). In order to determine the break locations for the m-break model in Equation (3), we used the dynamic programming approach proposed by Bai and Perron [13] that permits to reduce considerably the computing time. The algorithm given in Table 2, separated in two parts, is a translation in pseudocode langage of this method.

The ϵ parameter and the optimal partition $(\hat{T}_1, \ldots, \hat{T}_m)$ are defined in section 2.2.

[Table 2 about here.]

This algorithm gives a complete description of the way to compute the break locations. In the first part, the sum of squared residuals denoted by $ssr_{i,j}$ are computed only for segments $[i; j]$ that are necessary in the m-break determination. In the second part, the optimal partition is obtained by solving the recursive problem $S_{r,j} = \min_{rh \leq i \leq j-h}[S_{r-1,i} + ssr_{i+1,j}]$ in which $S_{r,j}$ denotes the sum of squared residuals associated with the optimal partition containing r breaks using the first j observations.

3.6 Data output and plotting

The output of the *clus*( ) function is a list of objects that contains:

res: A result matrix giving, for each point ordered by its rank in the trajectory, its distance to the nearest neighbourg, the expentancy of this distance, and its

weighted distance.

pop: A matrix with 2 or 3 columns (depending on wether 2D or 3D data) giving coordinates of underlying population data points.

bc: A list of vectors of size 1 to M. The k^{th} element of the list gives the estimated breaks for the model with k breaks.

stat: A list of non corrected statistic values (F), corrected statistic value (wdm), threshold value for the WDM statistic ($wdms$), significativity ($signif$) and the number of breaks that maximizes the WDM statistic ($kmax$).

kulld.p: A vector giving the results of the spatial scan method with the Poisson model. *lambda* is the value of the spatial scan test statistic, *loglambda* is its logarithm, *cx* and *cy* are the coordinates of the circle center and *rayon* is its radius.

kulld.b: A vector giving the results of the spatial scan method with the Bernouilli model. *lambda* is the value of the spatial scan test statistic, *loglambda* is its logarithm, *cx* and *cy* are the coordinates of the circle center and *rayon* is its radius.

This list of objects can be used as argument in both plotting functions. The function *plotreg*( ) displays the selection order in the X-axis, the weighted distance in the Y-axis and draws the regression function with k breaks. The function *plotclus*( ) displays the point cloud and located cluster(s) with the k-break model. k is generally equal to the value of the *stat*$*kmax*.

3.7 Exportation module in SatScan format

In this module, the cluster location by the spatial scan statistic [7] is implemented, but p-value is not provided. For a full analysis with this method, including cluster detection via Monte Carlo replications, one can use the SatScan software [11] freely available. The SPATCLUS package allows user to export the data in a format directly usable by this software. For this purpose, one can use the following parameter values:

method = 3

methk = 1 or 2 (Bernouilli or Poisson model)

export = "*y*"

repexport = "*dir*". *dir* denotes the directory path in which the data will be exported in SatScan format.

4 Sample runs and example

4.1 Sample runs

In order to illustrate the flexibility of the method, we simulated two 200-points samples. The first sample contains two simulated potential clusters with different shapes (a parallelogram and a "L"-shaped polygon) with a density inside about 6 times higher than outside. The second sample contains four simulated potential clusters: the same than previously plus two squares. A uniform 3000-point grid was attributed to each sample in order to represent the underlying population.

We analysed those samples with $M = 8$ as maximum number of breaks and $\epsilon = 0.1$ as minimum number of points between two breaks. The critical value corresponding to these parameter values is 10.7. For the 2-cluster sample, the 4-break (2-cluster) model was selected and the $WD\ max$ statistic value was 24.2. For the 4-cluster sample, the 8-break (4-cluster) model was selected and the $WD\ max$ statistic value was 38.9. The no-cluster hypothesis was rejected is both samples and the model with 4 breaks (respectively 8 breaks) was selected. All the clusters were significant.

The regression plot and the cluster location result are presented for both samples in Figure 2.

The spatial scan statistic [7] was applied on the two samples. The exportation module was used to put data into the right format and analyze them with the SatScan software [11]. In both cases, the most likely cluster (represented by a cercle in Figure 2) was significant.

[Fig. 2 about here.]

4.2 fMRI application

A way of applying this method to functional Magnetic Resonance Imaging (fMRI) data is proposed. fMRI is a technique for determining which parts of the brain are activated under different type of experimental conditions. The standard statistical method in analysing fMRI data is based on Statistical Parametric Mapping (SPM) [14].

The aim of the application of the cluster detection method to fMRI data is to locate clusters which correspond to brain regions simultaneoulsy activated for most subjects. The process consists first in determining activation peaks for each subject by the

standard SPM method. Then the peaks of all the subjects are grouped together, which forms a $3D$ data set. Finally, the cluster detection method is applied to this data set in order to locate and detect clusters of activation peaks.

A word fluency task was given to 11 right-handed women within a classical fMRI block design with 5 control conditions (counting task) and 5 activity conditions (word fluency task) alternately. During the activation conditions, subjects had to produce silently as many words as possible beginning with a orally presented letter. The control condition consisted in counting forward from one, at a rate of about one a second.

The SPM method has been applied to each subject in order to detect significant hot spots (activation peaks) at an individual level. Each subject presents an average of 32 peaks. Then, those 354 peaks has been merged together and analysed with our method in order to determine, at a group level, which cerebral zones are activated for most of the subjects. The model with 8 breaks (4 potential clusters) was selected and the $WD\ max$ statistic value was 25.2, higher than the critical value. One of the 4 potential cluster was not significant, while the others were significant clusters ($p \leq 0.05$).

Hence, three hot spot clusters have been detected, two located in the frontal lobe and the other in the occipital lobe, each containing between 36 and 39 peaks. Those activated brain regions are represented in the Figure 3. Except for one atypical subject presenting only one peak in one of the three clusters, all the others present between 2 and 5 hot spots in each cluster. Those three clusters correspond to brain regions simultaneoulsy activated for most subjects.

Moreover, the spatial scan statistic [7] was applied to this 3D data set. The maximum spatial cluster size was initially set to 50% of population at risk (default value of SatScan). With this value, the most likely cluster groups together 261 cases among the 354 total number of cases, more than half of cases. Finally, we set this value to 30%. The most likely cluster is a sphere with centre at $(9, -5, -53)$ and radius 54.65. This significant cluster groups together 151 cases and is shown in Figure 3 by a transparent white sphere. Here, we can see that the spatial scan statistic fails: this approach detects a very large cluster which is not interpretable.

[Fig. 3 about here.]

5 Hardware and software specifications

The implemention and sample runs of this package was conducted on a 2GHz PC computer under the MandrakeLinux 9.2 distribution using the R software version 1.9.0 (CRAN, the "Comprehensive R Archive Network"). However, R runs in any OS platform (MAC, UNIX, Windows) and can be obtained freely via the different CRAN mirrors. All the mirrors URLS are available via the CRAN link on the R homepage at $http://www.r-project.org/$. Hence, the SPATCLUS package can be installed in any platform.

6 Online availability

The SPATCLUS package (link "Télécharger l'outil") and the package documentation (link "Voir la notice d'information") are available over the web via the "Thèmes de recherche" tab on the IURC biostatistical laboratory website at following URL $http://www.iurc.montp.inserm.fr/biostat/$. The package downloadable file is a ".tar.gz" archive that can be easily installed on the R software using the command "R CMD INSTALL spatclus" from source on UNIX, or "Rcmd INSTALL spatclus" on Windows. Further informations on R packages installation can be found in the "R Installation and Administration" manual available on the R homepage.

7 Discussion

This paper describes an R package that implements a new spatial cluster detection method. This description and the package documentation are complementary to help users to apply the method both easily and correctly, or for example to conduct valuable power comparisons between different methods.

The main difficulties in the implementation of the method are the distance weighting and the break location. The first algorithm presented allows to enlighten the numerical computation of the distance expectation in the weighting process. The second algorithm is a detailed version of the dynamic programming algorithm presented by Bai and Perron. This method allows to compute the break estimates using at most least-squares operations of order $O(T^2)$ for any number of breaks m. This means that it is only marginally longer to obtain the optimal partition with 8 breaks as it is with 2 breaks.

The method implemented in the SPATCLUS package has the advantage of being

very flexible. Firstly, it can be used to detect and locate several clusters, with no need to adjust for the multiple testing problem. Secondly, since the method does not need the definition of a predefined shape for potential clusters, the clusters detected can be of any shape. Moreover, since case event data are used, the method is free from map partition. Finally, a potential inhomogeneity in the underlying population distribution is taken into account through the weighting process.

References

[1] B. Ripley, Modelling spatial patterns, Journal of the Royal Statistical Society B, 39 (1977) 172–192.

[2] A.S. Whittemore, N. Friend, B.W. Brown, E.A. Holly, A test to detect clusters of disease, Biometrika, 74 (1987) 631–635.

[3] J. Cuzick, R. Edwards, Spatial clustering for inhomogeneous populations, Journal of the Royal Statistical Society B, 52 (1990) 73–104.

[4] J. Besag, J. Newell, The detection of clusters in rare diseases, Journal of the Royal Statistical Society A, 154 (1991) 143–155.

[5] T. Tango, A test for spatial disease clustering adjusted for multiple testing, Statistics in Medicine, 19 (2000) 191–204.

[6] B.W. Turnbull, E.J. Iwano, W.S. Burnett, H.L. Howe, L.C. Clark, Monitoring for clusters of disease: application to leukemia incidence in upstate New York, American Journal of Epidemiology, 132 (1990) 136–143.

[7] M. Kulldorff, A spatial scan statistic, Communications in Statistics - Theory and Methods, 26 (1997) 1481–1496.

[8] P.J. Diggle, S. Morris, T. Morton-Jones, Case-control isotonic regression for investigation of elevation in risk around a point source, Statistics in Medicine, 18 (1999) 1605–1613.

[9] C. Dematteï, N. Molinari, J.P. Daurès, Arbitrarily Shaped Multiple Spatial Cluster Detection for Case Event Data, Accepted in Computational Statistics and Data Analysis, (2006); Corrected proof available online via the DOI link *http* : *//dx.doi.org/10.1016/j.csda.2006.03.011* .

[10] N. Molinari, C. Bonaldi, J.P. Daurès, Multiple temporal cluster detection, Biometrics, 57 (2001) 577–583.

[11] M. Kulldorff and Information Managements Services, Inc. SaTScan v5.1: Software for the spatial and space-time scan statistics, *http* : *//www.satscan.org*, (2004).

[12] J. Bai, P. Perron, Estimating and testing linear models with multiple structural changes, Econometrica, 66 (1998) 47–78.

[13] J. Bai, P. Perron, Computation and analysis of multiple structural change models, Journal of Applied Econometrics, 18 (2003) 1–22.

[14] R.S.J. Frackowiak, K.J. Friston, C. Frith, R. Dolan, C.J. Price, S. Zeki, J. Ashburner and W.D. Penny, Imaging neuroscience - Theorie and analysis, in Human Brain Function, 2nd edition, part II, Academic Press, 2003.

8 Appendix

List of Figures

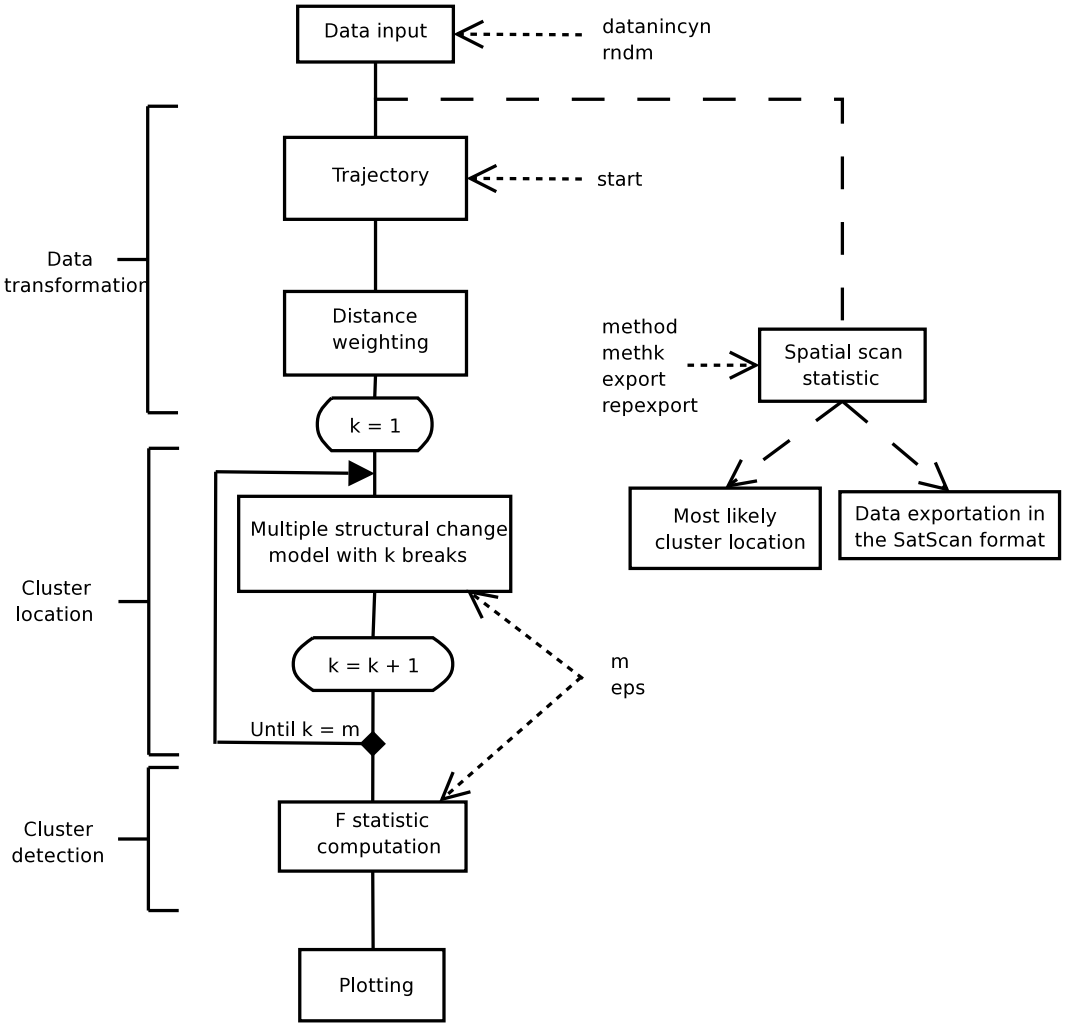

Fig. 1. Flow chart describing the package.

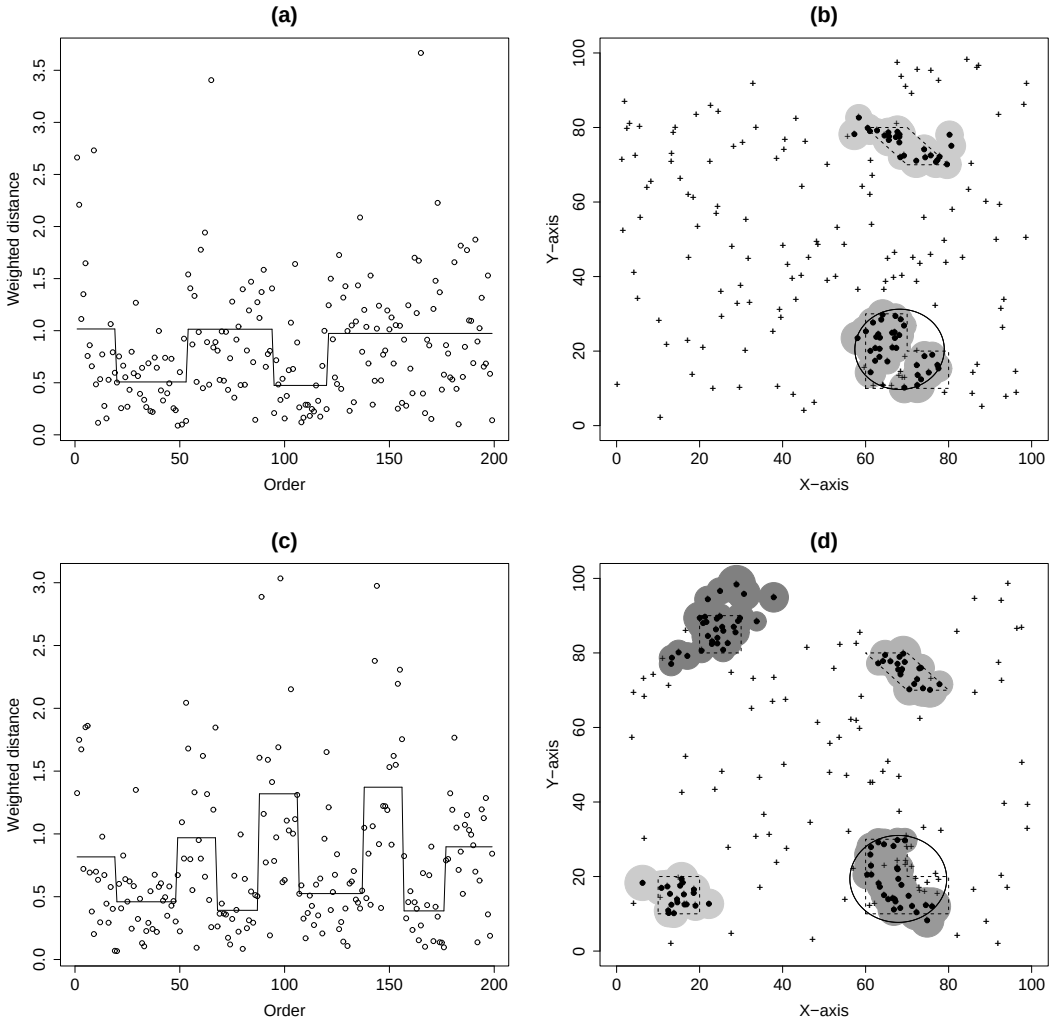

Fig. 2. Results for the 2- and 4-cluster models on simulated data . **(a)** and **(c)**: Results of the regression of distance on the order respectively for the 2 and 4-cluster model. **(b)** and **(d)**: Representation of the clusters located respectively by the 2 and 4-cluster model. Points located in the clusters are round points surrounded by a grey disc. Simulated cluster areas are represented in dotted lines. The most likely cluster located by the spatial scan statistic is represented by a cercle.

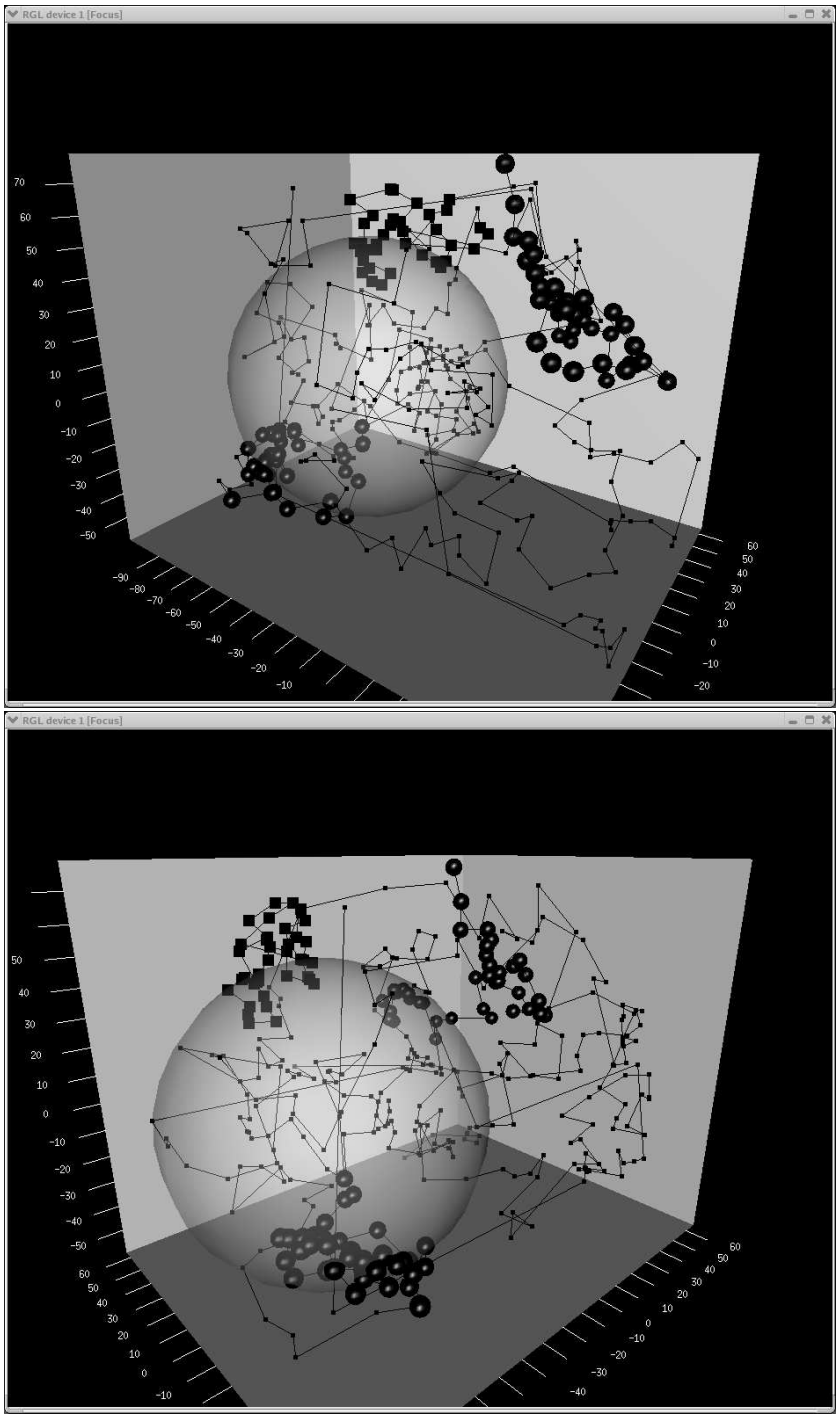

Fig. 3. 3D representation of fMRI activation peaks (protocol described in Section 4.2). At the top: right-hand side view of the brain from the front. At the bottom: right-hand side view of the brain from the back. Each peak is represented by a little black cube. A line joins two peaks that are successive in the trajectory. Points included in a significant cluster are represented by a sphere or a big black cube. The most likely cluster detected by the spatial scan statistic is represented by a transparent white sphere.

List of Tables

Table 1
Data transformation algorithm

READ $data$, pop, pas, $x_{(1)}$

FOR $k = 1$ to $n - 1$

IF $k > 1$ THEN

$pop \leftarrow pop \setminus \{u/d\left(x_{(k-1)}, u\right) \leq d\left(x_{(k-1)}, x_{(k)}\right)\}$

ENDIF

$a_k \leftarrow \max_{u \in pop} d\left(x_{(k)}, u\right)$

SET S to 0

FOR $r = 0$ to a_k by pas

SET $rpop$ to pop

$rpop \leftarrow rpop \setminus \{u/d\left(x_{(k)}, u\right) > r\}$

$S \leftarrow S + \left(1 - \frac{\#rpop}{\#pop}\right)^{n-k}$

ENDFOR

$E[d_k] \leftarrow pas \times \left(S - \frac{1}{2}\right)$

$x_{(k+1)} \leftarrow \text{argmin}_{x \in data} d\left(x_{(k)}, x\right)$

$d_k \leftarrow d\left(x_{(k)}, x_{(k+1)}\right)$

$d_k^w \leftarrow \frac{d_k}{E[d_k]}$

$data \leftarrow data \setminus \{x_{(k)}\}$

PRINT $x_{(k)}$, d_k^w

ENDFOR

Table 2
Break location algorithm

READ m, ϵ, d_1^w, d_2^w, …, d_{n-1}^w

$T \leftarrow n-1$

$h \leftarrow |T\epsilon|$

FOR $i = 1$ to T

  FOR $j = 1$ to T

    IF $j - i \geq h - 1$

      $\overline{d_{i,j}^w} \leftarrow \frac{1}{j-i+1} \sum_{k=i}^{j} d_k^w$

      $ssr_{i,j} \leftarrow \sum_{k=i}^{j} \left(d_k^w - \overline{d_{i,j}^w} \right)^2$

    ENDIF

  ENDFOR

ENDFOR

IF $m = 1$

  $\hat{T}_1 \leftarrow \text{argmin}_{h \leq j \leq T-h}[ssr_{1,j} + ssr_{j+1,T}]$

ENDIF

FOR $j = h$ to T

  $S_{0,j} \leftarrow ssr_{1,j}$

ENDFOR

IF $m > 1$

  FOR $r = 1$ to $m - 1$

    FOR $j = (r+1)h$ to $T - (m-r)h$

      $S_{r,j} \leftarrow \min_{rh \leq i \leq j-h}[S_{r-1,i} + ssr_{i+1,j}]$

      $b_{r,j} \leftarrow \text{argmin}_{rh \leq i \leq j-h}[S_{r-1,i} + ssr_{i+1,j}]$

    ENDFOR

  ENDFOR

  $S_{m,T} \leftarrow \min_{mh \leq j \leq T-h}[S_{m-1,j}]$

  $\hat{T}_m \leftarrow \text{argmin}_{mh \leq j \leq T-h}[S_{m-1,j}]$

  FOR $k = m - 1$ to 1

    $\hat{T}_k \leftarrow b_{k,\hat{T}_{k+1}}$

    PRINT $\hat{T}_k$

  ENDFOR

ENDIF