

GeoBUGS User Manual

Version 1.2, September 2004

Andrew Thomas¹

Nicky Best²

Dave Lunn²

Richard Arnold³

David Spiegelhalter⁴

¹ Rolf Nevanlinna Institute
P.O. Box 4 (Yliopistonkatu 5)
FIN-00014 University of Helsinki
Finland

² Department of Epidemiology & Public Health,
Imperial College School of Medicine,
Norfolk Place,
London W2 1PG, UK

³ School of Mathematical and Computing Sciences
Victoria University,
P. O. Box 600, Wellington,
New Zealand

⁴ MRC Biostatistics Unit,
Institute of Public Health,
Robinson Way,
Cambridge CB2 2SR, UK

e-mail: bugs@mrc-bsu.cam.ac.uk [general]
ant@ni.helsinki.fi [technical]

internet: <http://www.mrc-bsu.cam.ac.uk/bugs>

Contents

Introduction

Changes from GeoBUGS 1.1Beta

Importing map polygon files ⇌

Exporting maps from GeoBUGS into Splus format

Producing adjacency matrices

Editing adjacency matrices

Producing maps

User-specified cut-points and shading

Identifying individual areas on a map

Copying and saving maps

Spatial distributions ⇔ ⇐

Temporal distributions ⇔ ⇐

Examples ⇔ ⇐

Appendix 1: Technical details of Structured Multivariate Gaussian and Conditional Autoregressive (CAR) models and hyperprior specification ⇔ ⇐

Appendix 2: Technical details of the Poisson-gamma Spatial Moving Average convolution model

References

Introduction [\[top\]](#)

GeoBUGS is an add-on module to WinBUGS which provides an interface for:

- * [producing maps](#) of the output from disease mapping and other spatial models
- * [creating](#) and [manipulating](#) adjacency matrices that are required as input for the [conditional autoregressive models](#) available in WinBUGS 1.4 for carrying out spatial smoothing.

Version 1.2 of GeoBUGS contains map files for

- * Districts in Scotland (called **Scotland**)
- * Wards in a London Health Authority (called **London_HA**)
- * Counties in Great Britain (called **GB_Counties**)
- * Departements in France (called **France**)
- * Nomoi in Greece (called **Grecenomoï**)
- * Districts in Belgium (called **Belgium**)
- * Communes in Sardinia (called **Sardinia**)
- * Subquarters in Munich (called **Munich**)
- * A 15 x 15 regular grid (called **Elevation**)
- * Wards in West Yorkshire (UK) (called **WestYorkshire**)
- * A 4 x 4 regular grid (called **Forest**)
- * A grid of 750 m² grid cells covering the town of Huddersfield and surroundings in northern England (called **Huddersfield_750m_grid**)

A list of the area IDs for each map and the order in which the areas are stored in the map file can be obtained using the [export Splus](#) command.

GeoBUGS 1.2 also has facilities for importing user-defined maps reading polygon formats from [Splus](#), [ArcInfo](#) and [Epimap](#), plus a link to a program written by Yue Cui for importing [ArcView](#) shape files.

Changes from GeoBUGS 1.1Beta [\[top\]](#)

- * New distributions:
 - spatial.disc
 - pois.conv
 - mv.car
- * New examples:
 - spatial moving average model applied to forest biodiversity and disease mapping examples
 - multivariate spatial modelling using multivariate intrinsic CAR and shared component models for mapping multiple diseases
 - temporal smoothing of daily air pollution measurements using a random walk prior
- * Problems and bugs fixed:
 - no restriction on dimension of vector that can be fitted using the spatial.exp distribution when used in conjunction with spatial.exp.pred.uni (previously vector was restricted to have length < 100)
 - sum-to-zero constraint in car.normal and car.l1 distributions is fixed (previous method for imposing the constraint did not always give a mean of exactly zero)
 - problem with selecting areas using adjacency tool for regular grid maps is fixed

Importing map polygon files [\[top\]](#)

Polygon files can be imported into GeoBUGS from a variety of other packages:

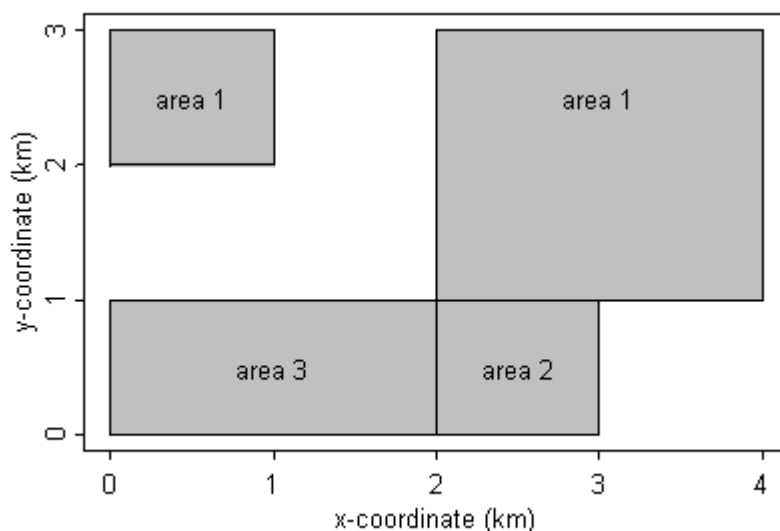
- * [Splus](#)
- * [ArcInfo](#)
- * [EpiMap](#)
- * [ArcView](#)

Import files are text files containing:

- * Number of regions in the map
- * List of labels for each region, with corresponding ID number
- * List of x and y co-ordinates for each polygon, plus the polygon label

The different GeoBUGS import formats are designed to follow as closely as possible the format in which Splus, ArcInfo and EpiMap export polygons, respectively. However, some manual editing of the polygon files exported from these various packages is also necessary before they can be read into GeoBUGS.

The following simple map is used to illustrate the different import formats. The map contains 3 areas (labelled area 1, area 2 and area 3); area 1 consists of 2 separate polygons, while areas 2 and 3 consist of one polygon each.



Splus format [\[top\]](#)

```
map:3
xScale: 1000
yScale: 1000
```

```
1 area1
2 area2
3 area3
```

```
area1 0 2
area1 1 2
area1 1 3
area1 0 3
NA NA NA
area1 2 1
area1 4 1
area1 4 3
area1 2 3
NA NA NA
area2 0 0
area2 2 0
area2 2 1
area2 0 1
NA NA NA
area3 2 0
area3 3 0
area3 3 1
area3 2 1
END
```

The Splus import file is in three parts:

The first line contains the key word 'map' (lower case) followed by a colon and an integer, N, where N is the number of distinct areas in the map (note that one area can consist of more than one polygon). The 2nd and 3rd lines are *optional*, and can be used to specify the units for the map scale. By default, GeoBUGS assumes that the polygon coordinates are measured in metres. If the coordinates are measured in kilometres, say, then specify xScale and yScale to be 1000. GeoBUGS will then multiply all polygon co-ordinates by xScale and yScale as appropriate before storing the map file. If xScale and yScale are not specified, then the default units (metres) are assumed.

The next part of the import file is a 2 column list giving:

(column 1) the numeric ID of the area - this must be a unique integer between 1 and N; the areas should be labelled in the same order as the corresponding data for that area appears in the model.

(column 2) the area label - this must start with a character, and can be a maximum of 79 alphanumeric characters (no spaces allowed)

The final part of the import file is a 3 column list giving the co-ordinates of the polygons. The format is:

(col 1) the label of the area to which the polygon belongs

(col 2) x-coordinate

(col 3) y-coordinate

The polygon coordinates can be listed either clockwise or anticlockwise. Polygons should be separated by a row of NA's

The import file should end with the key word: END

Note: Brad Carlin has a link on his web page to an Splus program called poly.S to extract polygons for any state in the United States in the appropriate format for loading into GeoBUGS (<http://www.biostat.umn.edu/~brad/software.html>)

ArcInfo format [\[top\]](#)

```
map:3
xScale: 1000
yScale: 1000
```

```
1 area1
2 area2
3 area3
```

```
regions
99 area1
103 area1
210 area2
211 area3
END
```

```
99 0 0
0 2
1 2
1 3
0 3
END
```

```
103 0 0
2 1
4 1
4 3
2 3
END
```

```
210 0 0
0 0
2 0
2 1
0 1
END
```

```
211 0 0
2 0
3 0
3 1
2 1
END
END
```

The ArcInfo import file is in four parts:

The first 2 parts are the same as the Splus format.

The third part begins with a line containing the key word 'regions' (lower case). Below this is a 2 column list giving:

(column 1) an integer label corresponding to the integer label for one of the polygons listed in the final part of the import file. Each polygon should have a unique integer label, but this can be arbitrary (i.e. labels don't need to start at 1 or be consecutive). If using the ArcInfo command UNGENERATE to export a set of polygons, this is the integer label that ArcInfo automatically attaches to each polygon.

(column 2) the area label to which the polygon with that integer ID belongs. Note, if an area contains more than one polygon, then each polygon ID should be listed on a separate line and given the same area label (e.g., area1 in the above example).

There should be as many rows in this part of the file as there are polygons. This will be equal to or greater than the number of distinct areas in the map.

The final part of the import file gives the co-ordinates of the polygons. The format for each polygon is:

(row 1, column 1) the integer ID for the polygon (this should correspond to one of the integer IDs listed in the previous part of the import file).

(row 1, columns 2 and 3) if the polygon file has been exported directly from ArcInfo, these 2 numbers usually give the centroid of the polygon. However, they are not used by GeoBUGS, so can be arbitrary.

Subsequent rows contain a 2-column list of numbers giving the x- and y-coordinates of the poly. The polygon coordinates can be listed either clockwise or anticlockwise.

Polygons should be separated by a line containing the key word END.

The final row of the import file should also contain the key word END

Epimap format [\[top\]](#)

map:3

xScale: 1000

yScale: 1000

1 area1

2 area2

3 area3

area1, 4

0, 2

1, 2

1, 3

0, 3

area1, 4

2, 1

4, 1

4, 3

2, 3

area2, 4

0, 0

2, 0

2, 1

0, 1

area3, 4

2, 0

3, 0

3, 1

2, 1

END

The Epimap import file is in three parts:

The first 2 parts are the same as the Splus format.

The third gives the polygon co-ordinates. The format for each polygon is:

(row 1, column 1) the label of the area to which the polygon belongs.

(row 1, column 2) the number of vertices in the polygon (note the comma separator)
Subsequent rows contain a 2-column list of numbers giving the x- and y-coordinates of the poly, separated by a comma. The polygon coordinates can be listed either clockwise or anticlockwise.

The final row of the import file should contain the key word END

ArcView format [\[top\]](#)

GeoBUGS does not have an option for loading ArcView shape files directly. However, Ms Yue Cui at the University of Minnesota has written programs in Splus and R for converting shape files into the GeoBUGS Splus format so that they can be loaded in GeoBUGS (<http://www.biostat.umn.edu/~yuecui/>).

Loading a polygon file into GeoBUGS [\[top\]](#)

Open the polygon file as a separate text file in WinBUGS 1.4Beta and select the appropriate import option from the **Map** menu. (To try loading the example map files above, first copy them to a separate file, and focus the window containing this file). If the map has been loaded correctly, a **Save As** dialog box will appear, prompting you to enter a name for the map file. By default, the map file will be saved in the `Maps\Map` subdirectory of your WinBUGS14Beta program, with a `.map` extension. You can view this map by selecting **Open** from the **File** menu (go to the `Maps\Map` subdirectory and select file type: `map file (*.map)`). You will need to exit WinBUGS and re-start before the new map will appear on the pull-down list of available maps in the **Map Tool** and **Adjacency Tool** of the **Map** menu.

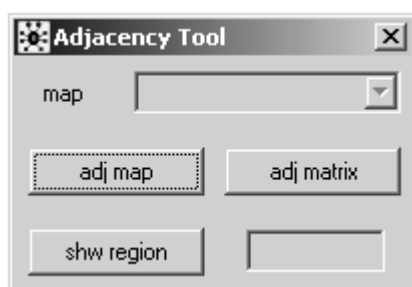
Exporting maps from GeoBUGS into Splus format [\[top\]](#)

Focus the window containing the map in GeoBUGS and select **Export Splus** from the **Map** menu. This will write the map in Splus format to a new window. This command can be used to obtain the list of area IDs and the order in which they are specified in the GeoBUGS map (see top part of export file).

Producing adjacency matrices [\[top\]](#)

GeoBUGS includes an option to produce a data file containing the adjacency matrix for any map loaded on the system. This file is in a format required by the [*car.normal*](#), [*car.11*](#) and [*mv.car*](#) conditional autoregressive distributions available in the WinBUGS 1.4 program.

- * Select the **Adjacency Tool** option from the **Map** menu.



- * Select the name of the map you wish to draw from the pull-down menu labelled **Map** and click on **adj map**. The selected map will then appear in a window.
- * Typing the ID number of a region in the bottom white box and clicking **shw region** will cause the specified region to be highlighted in red on the map; its neighbours (defined to be any region adjacent to the red region) are highlighted in green. A region and its neighbours can also be highlighted by positioning the mouse cursor over the required region on the map and clicking with the left button.
- * Click on the **adj matrix** button to produce a text file containing the adjacency matrix in a form suitable for

loading as data into WinBUGS for use with the *car.normal*, *car.I1* and *mv.car* distributions. (See appendix 1 for further details about these three distributions).

Note: when calculating which areas are adjacent to which others, GeoBUGS includes a 'tolerance' zone of 0.1 metres. This tolerance zone should not lead to spurious neighbours unless you forget to appropriately scale your distance units in the polygon file using the xScale and yScale options, or your map covers a tiny geographic region (in which case, artificially re-scaling the distance units for your map should overcome any problems).

Editing adjacency matrices [\[top\]](#)

To remove a region from the set of neighbours for a specific area:

- * Highlight the specific area in red on the map;
- * Place the mouse cursor over the region to be removed from the set of neighbours for the red area;
- * Hold down the Ctrl key while clicking with the left mouse button. The removed area will no longer be highlighted in green.

To add a region to the set of neighbours for a specific area:

- * Highlight the specific area in red on the map;
- * Place the mouse cursor over the region to be added to the set of neighbours for the red area;
- * Hold down the Ctrl key while clicking with the left mouse button. The additional area will then become highlighted in green.

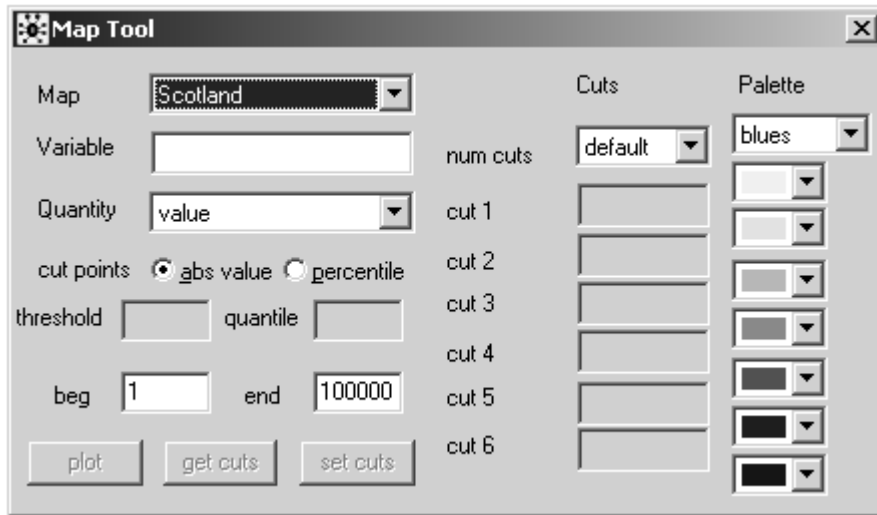
Once you have finished editing the set of neighbours for each region on your map, create the new adjacency matrix by clicking on the **adj matrix** button.

Producing maps [\[top\]](#)

Note: in order to produce a map of the mean or other summary statistic of the posterior distribution of a stochastic variable, you must have already set a samples or summary monitor for that parameter and have carried out some updates.

To produce a map:

- * Select the **Mapping Tool** option from the **Map** menu.

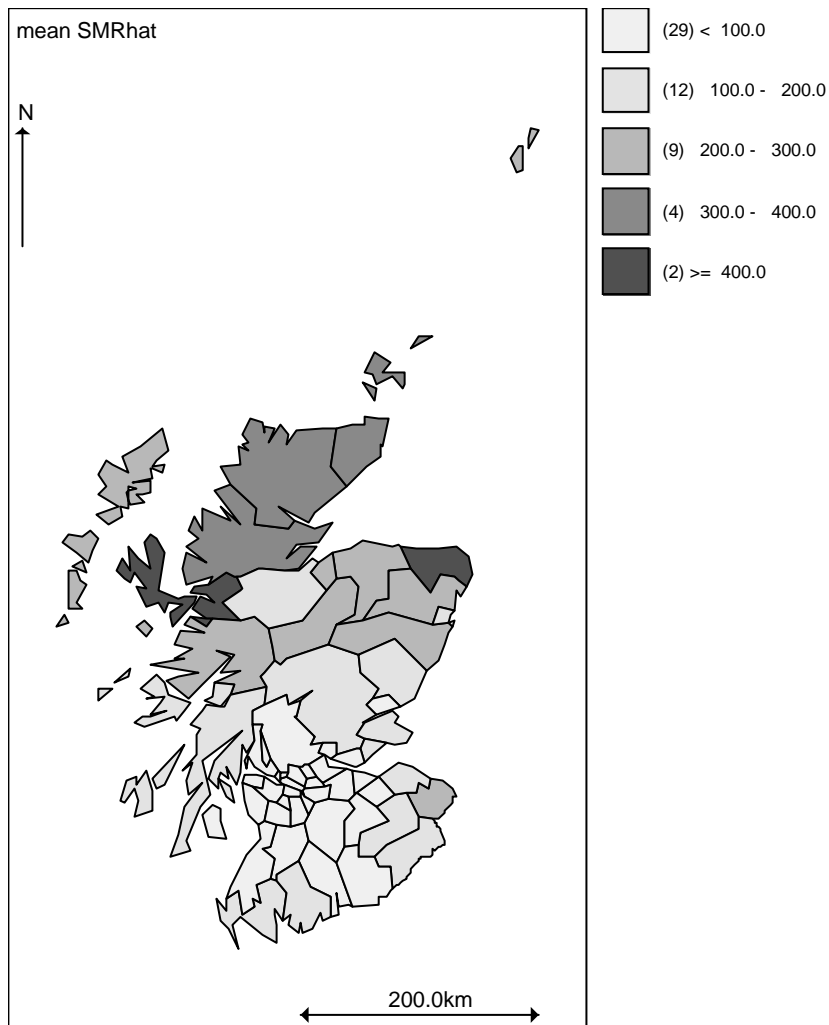


- * Select the name of the map you wish to draw from the pull-down menu labelled **Map**.
- * Type the name of the variable to be mapped in the white box labelled **Variable**.
- * If the variable is *data* (e.g. the raw SMR, expected counts E, or a covariate) pick the **value** option in the pull down menu labelled **Quantity** and then click the **plot** button: a map shaded according to the values of the variable will now appear.
- * If the variable is a *stochastic* quantity (e.g. the relative risks) there are various options which you can select from the **Quantity** menu:
 - if you have monitored the variable by setting a *summary monitor*, then you must select the **mean(summary)** option from this menu, as only the posterior means are stored by the summary monitor;
 - if you have monitored the variable by setting a *samples monitor* (which stores the full posterior sample), you can select any of the remaining options from the Quantity menu:
 - mean(sample)** will map the posterior means of the variable;
 - percentile** will plot posterior quantiles of the variable - if you select this option, you must then type the required percentile in the box labelled **quantile**;
 - prob greater** will map the posterior probability that the value of the variable is greater than or equal to the specified threshold, which you must type in the box labelled **threshold**;
 - prob less** will map the posterior probability that the value of the variable is less than or equal to the specified threshold, which you must type in the box labelled **threshold**;

When you have selected the quantity you want to map, click the **plot** button to display the map.

- * The numbers in brackets shown on the map legend give the number of areas classified in each category on the map.

Fig 1. GeoBUGS map of SMRs for Scottish Lip Cancer data (see [Examples](#) section for further details)



User-specified cut-points and shading [\[top\]](#)

GeoBUGS can work with two kinds of cut-points: absolute value cut-points and percentile cut-points. For absolute cut-points, GeoBUGS chooses a default set of breaks based on the absolute value of the variable to be mapped: these are chosen to give equally spaced intervals. For percentile cutpoints, GeoBUGS chooses the 10th, 50th and 90th percentiles of the empirical distribution of the variable to be mapped. The default shading is blue-scale.

To edit the colours for shading the map:

- * Select the **custom** option from the **Palette** menu in the top right of the **Map Tool**.
- * Clicking with the left mouse button on the arrow by each colour will bring up a menu of alternative colours that can be selected.
- * After you have selected the new colour scheme, click on the **set cuts** button to "refresh" the currently selected map or click on the **plot** button to produce a new map.
- * To reset the colour scheme to blue shades, select **blues** from the Palette menu and click on the **set cuts** button again.

To edit the absolute value cutpoints:

- * Select the required number of cutpoints (a maximum of 6 cutpoints is currently allowed) from the menu labelled **num cuts** under the **Cuts** menu.

- * Type the required values of the cutpoints in the appropriate boxes labelled cut 1, cut 2 etc.
- * Click on the **set cuts** button to "refresh" the currently selected map or click on the **plot** button to produce a new map.

To produce maps using cutpoints based on percentiles:

- * Select the **percentile** option rather than **abs value** option for the cutpoints on the left of the **Map Tool**.
- * Click on the **plot** button to produce a new map. The default is to set the cutpoints to the 10th, 50th and 90th percentiles of the empirical distribution of values to be mapped.
- * To display the absolute values corresponding to these percentiles on the map legend:
 - reselect the **abs value** option for cutpoints;
 - click on the **get cuts** button - the absolute values corresponding to the percentiles should now be displayed in the **Cuts** boxes on the right of the **Map Tool**
 - click on the **set cuts** button - the legend labels on the map should now display the absolute values of the cutpoints.

To use the same set of cutpoints for multiple maps:

- * Select the window with the map containing the cutpoints you wish to use.
- * Click on the **get cuts** button - the cutpoints used for the selected map should now be displayed in the **Cuts** boxes on the right of the **Map Tool**.
- * Select the window with the map whose cutpoints you wish to change.
- * Click on the **set cuts** button - the map should now be updated using the new cutpoints.

Some current limitations:

- * It is not possible to save user-defined colour schemes once you quit GeoBUGS.

Identifying individual areas on a map [\[top\]](#)

The index, label and value of an individual area on the map can be found by placing the cursor over the area of interest on the map and clicking with the left mouse button. The index (i.e. ID number i of the area, where $i=1,\dots,\text{Number of areas}$), area label (given in the polygon file) and value of variable currently being mapped for the selected area will be shown in left of the grey bar at the bottom of the WinBUGS program window.

Copying and saving maps [\[top\]](#)

Maps produced using the GeoBUGS map tool can be copied and pasted into other Microsoft Windows software such as Word and PowerPoint. In WinBUGS 1.4, this can only be done if the map is plotted in the *log file* (but not in a separate window) - select 'Output options' from the 'Options' menu and click on 'log' rather than 'window' before plotting the map. To select the map, click anywhere on the map to focus it (a blue border should then appear around the figure); then select 'Copy' from the 'Edit' menu (or Ctrl-C). Then paste into the appropriate Word or PowerPoint file etc. To save the map as a postscript file, you will need to install a postscript print driver on your PC, then select 'Print' from the 'File' menu, check the 'print to file' box, and then select 'Print'.

If you produce the map in a *window* rather than the *log file*, you can save the map as a WinBUGS .odc document; this will allow you to re-open the map within WinBUGS 1.4 and re-edit the cutpoints and colours if you wish. Unfortunately there is a bug which means that log files containing maps cannot be saved as .odc files and re-opened.

Spatial distributions [\[top\]](#)

See appendices for further technical details about the various spatial distributions implemented in GeoBUGS 1.2.

car.normal and *car.l1* [\[top\]](#)

The intrinsic Gaussian CAR prior distribution is specified using the distribution ***car.normal*** for the vector of random variables $\mathbf{S} = (S_1, \dots, S_N)$. A robust version of this model is available, which assumes a double exponential (Laplace) rather than Gaussian distribution: this is called ***car.l1***. The syntax for these distributions is as follows:

```
S[1:N] ~ car.normal(adj[], weights[], num[], tau)
S[1:N] ~ car.l1(adj[], weights[], num[], tau)
```

where:

adj[] : A vector listing the ID numbers of the adjacent areas for each area (this is a sparse representation of the full adjacency matrix for the study region, and can be generated using the ***Adjacency Tool*** from the ***Map*** menu in *GeoBUGS*).

weights[] : A vector the same length as ***adj[]*** giving *unnormalised* weights associated with each pair of areas. For the CAR model described above, taking $C_{ij} = 1$ (equivalently $W_{ij} = 1/n_i$) if areas i and j are neighbours and 0 otherwise, gives a vector of 1's for ***weights[]***.

num[] : A vector of length N (the total number of areas) giving the number of neighbours n_i for each area.

tau : A scalar argument representing the precision (inverse variance) parameter of the Gaussian CAR prior, or the inverse scale parameter of the Laplace prior (for the latter model, the variance = $2 / \tau^2$).

The first 3 arguments must be entered as data (it is currently not possible to allow the ***weights*** to be unknown); the final variable ***tau*** is usually treated as unknown and so is assigned a prior distribution. The data variables ***num*** and ***adj*** may be created by the ***adj matrix*** option of the *GeoBUGS Adjacency Tool* as described [above](#). The variable ***weights*** must be created by the user, and must be a vector the same length as ***adj***. A common choice is to set all the ***weights*** equal to 1 since this gives the standard Besag, York and Mollie (1991) CAR model (see section on [intrinsic CAR models](#) in Appendix 1 for further discussion of weights). The easiest way to do this is to create a loop in your *WinBUGS* model code:

```
for(j in 1:sumNumNneigh) { weights[j] <- 1 }
```

where ***sumNumNneigh*** is the length of ***adj*** and is also output by the ***adj matrix*** option of the *GeoBUGS Adjacency Tool*.

Important things to check when using the ***car.normal*** or ***car.l1*** distributions:

- * The ***car.normal*** and ***car.l1*** distributions use unnormalised weights (see section on [intrinsic CAR models](#) in Appendix 1).
- * An area cannot be specified as its own neighbour so make sure the ID number of the area itself does not appear in as one of its list of neighbours in the ***adj*** vector. *GeoBUGS* does *not* check for this, so it is the user's responsibility.
- * The weights must be symmetric ($W_{ij} = W_{ji}$). *GeoBUGS* does carry out a check for this and returns an error message if non-symmetric weights are detected.
- * Take care with priors on tau, and be prepared to carry out sensitivity analysis to this choice.
- * The ***car.normal*** and ***car.l1*** distributions are parameterised to include a sum-to-zero constraint on the random effects. This means that you must also include a separate intercept term in your model, which **MUST** be assigned an improper uniform prior using the ***dflat()*** distribution.
- * Because the ***car.normal*** and ***car.l1*** distributions are improper, they can only be used as prior distributions, and not as a likelihood.

car.proper [\[top\]](#)

The proper Gaussian CAR prior distribution is specified using the distribution **car.proper** for the *vector* of random variables $\mathbf{S} = (S_1, \dots, S_N)$. The syntax for this distributions is as follows:

$$S[1:N] \sim \text{car.proper}(\text{mu}[], \text{C}[], \text{adj}[], \text{num}[], \text{M}[], \text{tau}, \text{gamma})$$

where:

mu[] : A vector giving the mean for each area (this can either be entered as data, assigned a prior distribution, or specified deterministically within the model code).

C[] : A vector the same length as **adj[]** giving *normalised* weights associated with each pair of areas (see sections on [conditional specification](#) and [proper CAR priors](#) in Appendix 1). Note that this differs from the intrinsic **car.normal** or **car.11** distributions, where unnormalised weights should be specified.

adj[] : A vector listing the ID numbers of the adjacent areas for each area (this is a sparse representation of the full adjacency matrix for the study region, and can be generated using the **Adjacency Tool** from the **Map** menu in *GeoBUGS*).

num[] : A vector of length N (the total number of areas) giving the number of neighbours n_i for each area.

M[] : A vector of length N giving the diagonal elements M_{ii} of the conditional variance matrix (see sections on [conditional specification](#) and [proper CAR priors](#) in Appendix 1)

tau : A scalar parameter representing the overall precision (inverse variance) parameter.

gamma : A scalar parameter representing the overall degree of spatial dependence. This parameter is constrained to lie between bounds given by the inverse of the minimum and maximum eigenvalues of the matrix $\mathbf{M}^{-1/2} \mathbf{C} \mathbf{M}^{1/2}$ (see [appendix 1](#)). *GeoBUGS 1.2* contains two deterministic functions for calculating these bounds (or they can be calculated externally to *GeoBUGS* and input by the user):

min.bound(C[], adj[], num[], M[])

max.bound(C[], adj[], num[], M[])

where the arguments are the same as the corresponding vectors used as arguments to the **car.proper** distribution.

Important things to check when using the **car.proper** distribution:

- * **C**, **adj**, **num** and **M** must be entered as data (it is currently not possible to allow **C** to be unknown); **num** and **adj** may be created by the **adj matrix** option of the *GeoBUGS Adjacency Tool* as described [above](#). The [Lips example](#) shows a (slightly clumsy) way of creating the **C** and **M** vectors within the WinBUGS model code; alternatively, these can be created externally to *GeoBUGS* and read in as data.
- * The **car.proper** distribution uses normalised weights **C** (see section on [proper CAR priors](#) in Appendix 1).
- * An area cannot be specified as its own neighbour so make sure the ID number of the area itself does not appear in as one of its list of neighbours in the **adj** vector. *GeoBUGS* does *not* check for this, so it is the user's responsibility.
- * The symmetry constraint $C_{ij} M_{jj} = C_{ji} M_{ii}$ must be satisfied. *GeoBUGS* does carry out a check for this and returns an error message if lack of symmetry is detected.
- * Take care with priors on tau, and be prepared to carry out sensitivity analysis to this choice.

- * Take care with priors on gamma: you must ensure that the prior is constrained between the appropriate bounds. Besag, York and Mollie (1991) suggest that gamma needs to be close to its upper bound before it reflects reasonable spatial dependence, so you may want to try informative priors to represent this, and be prepared to carry out sensitivity analysis.

spatial.exp and **spatial.disc** [\[top\]](#)

Bayesian Gaussian kriging models (multivariate Gaussian distribution with covariance matrix expressed as a parametric function of distance between pairs of points - e.g. see Diggle, Tawn and Moyeed, 1998 and [Appendix 1](#)) can be specified using the distributions **spatial.exp** or **spatial.disc** for the *vector* of random

variables $\mathbf{S} = (S_1, \dots, S_N)$. The syntax for these distributions is as follows:

```
S[1:N] ~ spatial.exp(mu[], x[], y[], tau, phi, kappa)
S[1:N] ~ spatial.disc(mu[], x[], y[], tau, alpha)
```

where:

mu[]: A vector giving the mean for each area (this can either be entered as data, assigned a prior distribution, or specified deterministically within the model code).

x[] and **y[]**: Vectors of length N giving the x and y coordinates of the location of each point, or the centroid of each area

tau: A scalar parameter representing the overall precision (inverse variance) parameter.

Two options are available for specifying the form of the covariance matrix: the powered exponential function and the 'disc' function (see section on [Joint Specification](#) in Appendix 1).

The **powered exponential function** is implemented using the **spatial.exp** distribution and has 2 parameters:

phi: A scalar parameter representing the rate of decline of correlation with distance between points. *Note that the magnitude of this parameter will depend on the units in which the x and y coordinates of each location are measured (e.g. metres, km etc.).*

kappa: A scalar parameter controlling the amount of spatial smoothing. This is constrained to lie in the interval [0, 2).

The **disc function** is implemented using the **spatial.disc** distribution and has 1 parameter:

alpha: A scalar parameter representing the radius of the 'disc' centred at each (x, y) location. *Note that the magnitude of this parameter will depend on the units in which the x and y coordinates of each location are measured (e.g. metres, km etc.).*

Warning: These models can be very *slow* for even moderate sized datasets (the algorithm is of order N^3)! Experience to date also suggests that it may be better to *hierarchically centre* this model. For example, consider the following two alternative parameterisations of the same model:

Non-hierarchically centred

```
for (i in 1:N){
  y[i] ~ dnorm(S[i], gamma)
  mu[i] <- alpha+beta*z[i]
}
S[1:N] ~ spatial.exp(mu[], x[], y[], tau, phi,1)
```

Hierarchically centred

```
for (i in 1:N){
  y[i] ~ dnorm(S[i], gamma)
  S[i] <- alpha+beta*z[i] + W[i]
  mu[i] <- 0
}
W[1:N] ~ spatial.exp(mu[], x[], y[], tau, phi,1)
```

In some simulated examples, the non-hierarchically centred parameterisation has produced incorrect results, while the hierarchically centred parameterisation gives sensible answers. This may be a feature of the single-site updating schemes used in WinBUGS, so interpret your results with care! (Thanks to Alan Gelfand, Shanshan Wu and Alex Schmidt for noting this problem).

Experience also suggests that there is often very little information in the data about the values of the parameters of the powered exponential (i.e. phi and kappa) or disc (i.e. alpha) functions. We therefore recommend that reasonably informative priors are used, or that the values are fixed a priori, based on either substantive knowledge or exploratory analysis using e.g. variograms.

spatial.pred and **spatial.unipred** [\[top\]](#)

Spatial interpolation or prediction at arbitrary locations can be carried out using the ***spatial.pred*** or ***spatial.unipred*** functions, in conjunction with fitting either the ***spatial.exp*** or ***spatial.disc*** model to a set of observed data. ***spatial.pred*** carries out joint or simultaneous prediction at a set of target locations, whereas ***spatial.unipred*** carries out single site prediction. The difference is that the single site prediction yields marginal prediction intervals (i.e. ignoring correlation between prediction locations) whereas joint prediction yields simultaneous prediction intervals for the set of target locations (which will tend to be narrower than the marginal prediction intervals). The predicted means should be the same under joint or single site prediction. The disadvantage of joint prediction is that it is very slow (the computational time is of order P^3 , where P is the number of prediction sites). The syntax for these predictive distributions is:

Joint prediction:

```
T[1:P] ~ spatial.pred(mu.T[], x.T[], y.T[], S[])
```

Single site prediction:

```
for(j in 1:P) {
  T[j] ~ spatial.unipred(mu.T[j], x.T[j], y.T[j], S[])
}
```

where:

P : Scalar giving the number of prediction locations

$\mu.T[]$: vector of length P (or scalar for single site version) specifying the mean for each prediction location (this should be specified in the same way as the mean for the observed data S).

$x.T[]$ and **$y.T[]$** : Vectors of length P (or scalars for single site version) giving the x and y coordinates of the location of each prediction point

S : The vector of observations to which either the ***spatial.exp*** or ***spatial.disc*** model has been fitted.

pois.conv [\[top\]](#)

A conjugate Poisson-gamma spatial moving average distribution can be specified for non-negative counts defined on a spatial lattice (i.e. discrete geographical partition), using the distribution ***pois.conv***. This is a discrete version of the Poisson-gamma random field model of Wolpert and Ickstadt (1998) and Best et al (2000a). The basic syntax for this distribution is as follows:

```
S ~ dpois.conv(mu[])
```

where **S** is a non-negative *scalar* parameter defined at some (usually spatial) location, and **$\mu[]$** is a *vector* of length J representing the influence of a set of gamma distributed latent parameters at each of J different locations on the value of S . Hence **$\mu[]$** must be defined as a convolution of gamma random variables:

```
for (j in 1 : J) {
  mu[j] <- gamma[j] * k[j]
  gamma[j] ~ dgamma(a, b)
}
```

where **$k[j]$** is a spatial kernel or spatial 'weight' depending on some measure of distance or spatial proximity between the j th latent location and the location of S , and **a** and **b** are hyperparameters to be specified (see [Appendix 2](#) for further discussion of this distribution, including suitable kernel functions). Usually, **$k[]$** is calculated externally and read into WinBUGS as data; alternatively, if $k[]$ depends on unknown parameters, it may be defined as part of the BUGS code and re-computed at each MCMC iteration. However, this is likely to slow down the sampling within WinBUGS by many orders of magnitude, so is not recommended for models with large numbers of latent parameters (i.e. J large).

More typically, the distribution will be used for each element of a *vector* of counts defined on a spatial lattice of N regions, using the following syntax:

```
for (i in 1:N) {
  S[i] ~ dpois.conv(mu[i, ])
  for (j in 1 : J) {
```

```

      mu[i, j] <- gamma[j] * k[i, j]
    }
  }

```

where the latent $\gamma[j]$ parameters are defined as above, and $k[.,j]$ is now an $N \times J$ matrix with elements $k[i,j]$ representing the 'weight' or contribution of the latent gamma random variable at location j to the expected value of S at location i .

Conditional on μ , the $S[i]$ have independent Poisson distributions with mean $= \sum_j \mu[i, j]$.

Note that the model may be extended to include observed covariates as well as latent variables in the Poisson mean - see [example 6](#).

mv.car [\[top\]](#)

The multivariate intrinsic Gaussian CAR prior distribution is specified using the distribution ***mv.car*** for the $p \times N$ matrix of random variables \mathbf{S} , where columns of \mathbf{S} represent the spatial units (areas) and rows represent the variables (it is important to ensure the dimensions of \mathbf{S} are specified the correct way round). The syntax for this distribution is as follows:

$$S[1:p, 1:N] \sim \text{mv.car}(\text{adj}[], \text{weights}[], \text{num}[], \text{omega}[,])$$

where:

adj[] : A vector listing the ID numbers of the adjacent areas for each area (this is a sparse representation of the full adjacency matrix for the study region, and can be generated using the ***Adjacency Tool*** from the ***Map*** menu in ***GeoBUGS***).

weights[] : A vector the same length as ***adj[]*** giving *unnormalised* weights associated with each pair of areas. For the CAR model described above, taking $C_{ij} = 1$ (equivalently $W_{ij} = 1/n_i$) if areas i and j are neighbours and 0 otherwise, gives a vector of 1's for ***weights[]***.

num[] : A vector of length N (the total number of areas) giving the number of neighbours n_i for each area.

omega[,] : A $p \times p$ matrix representing the precision (inverse variance) matrix of the multivariate intrinsic Gaussian CAR prior.

The first 3 arguments must be entered as data (it is currently not possible to allow the ***weights*** to be unknown); the final variable ***omega*** is usually treated as unknown and so is assigned a prior distribution (which must be a Wishart distribution). The data variables ***num*** and ***adj*** may be created by the ***adj matrix*** option of the ***GeoBUGS Adjacency Tool*** as described [above](#). The variable ***weights*** must be created by the user, and must be a vector the same length as ***adj***. A common choice is to set all the ***weights*** equal to 1 since this gives the multivariate equivalent of the standard Besag, York and Mollie (1991) CAR model (see sections on [intrinsic CAR models](#) and [multivariate intrinsic CAR models](#) in Appendix 1 for further discussion of weights). The easiest way to do this is to create a loop in your ***WinBUGS*** model code:

```

for(j in 1:sumNumNneigh) { weights[j] <- 1 }

```

where ***sumNumNneigh*** is the length of ***adj*** and is also output by the ***adj matrix*** option of the ***GeoBUGS Adjacency Tool***.

Important things to check when using the ***mv.car*** distribution:

- * The ***mv.car*** distribution uses unnormalised weights, as for the ***car.normal*** distribution.
- * An area cannot be specified as its own neighbour so make sure the ID number of the area itself does not appear in as one of its list of neighbours in the ***adj*** vector. ***GeoBUGS*** does *not* check for this, so it is the user's responsibility.
- * The weights must be symmetric ($W_{ij} = W_{ji}$). ***GeoBUGS*** does carry out a check for this and returns an error message if non-symmetric weights are detected.
- * Take care with priors on ***omega***, and be prepared to carry out sensitivity analysis to this choice.
- * The ***mv.car*** distribution is parameterised to include a sum-to-zero constraint on the random effects. This

means that you must also include separate intercept terms in your model for each of the p variables, which MUST be assigned improper uniform priors using the **dflat()** distribution.

- * An alternative unconstrained version of the multivariate CAR prior is available in WinBUGS 1.4, called **mv.car.uncon**. The syntax is the same as for **mv.car**.
- * Because the **mv.car** (and **mv.car.uncon**) distribution is improper, it can only be used as a prior distribution, and not as a likelihood.
- * Please regard the **mv.car** and **mv.car.uncon** distributions as beta-test versions. If you encounter any problems using either distribution, please report these to n.best@imperial.ac.uk

Temporal distributions [\[top\]](#)

Using car.normal as a random walk prior for temporal smoothing [\[top\]](#)

In one dimension, the intrinsic Gaussian CAR distribution reduces to a Gaussian random walk (see e.g. Fahrmeir and Lang, 2001). Assume we have a set of temporally correlated random effects θ_t , $t=1, \dots, T$ (where T is the number of equally-spaced time points). In the simplest case of a random walk of order 1, RW(1), we may write

$$\begin{aligned} \theta_t \mid \theta_{-t} &\sim \text{Normal} \left(\theta_{t+1}, \phi \right) && \text{for } t = 1 \\ &\sim \text{Normal} \left((\theta_{t-1} + \theta_{t+1})/2, \phi / 2 \right) && \text{for } t = 2, \dots, T-1 \\ &\sim \text{Normal} \left(\theta_{t-1}, \phi \right) && \text{for } t = T \end{aligned}$$

where θ_{-t} denotes all elements of $\boldsymbol{\theta}$ except the θ_t . This is equivalent to specifying

$$\theta_t \mid \theta_{-t} \sim \text{Normal} \left(\sum_k C_{tk} \theta_k, \phi M_{tt} \right) \quad \text{for } t = 1, \dots, T$$

where $C_{tk} = W_{tk} / W_{t+}$, $W_{t+} = \sum_k W_{tk}$ and $W_{tk} = 1$ if $k = (t-1)$ or $(t+1)$ and 0 otherwise; $M_{tt} = 1/W_{t+}$. Hence the RW(1) prior may be fitted using the **car.normal** distribution in WinBUGS, with appropriate specification of the weight and adjacency matrices, and num vector (see [example 9](#))

A second order random walk prior is defined as

$$\begin{aligned} \theta_t \mid \theta_{-t} &\sim \text{Normal} \left(2\theta_{t+1} - \theta_{t+2}, \phi \right) && \text{for } t = 1 \\ &\sim \text{Normal} \left((2\theta_{t-1} + 4\theta_{t+1} - \theta_{t+2})/5, \phi / 5 \right) && \text{for } t = 2 \\ &\sim \text{Normal} \left((-\theta_{t-2} + 4\theta_{t-1} + 4\theta_{t+1} - \theta_{t+2})/6, \phi / 6 \right) && \text{for } t = 3, \dots, T-2 \\ &\sim \text{Normal} \left((-\theta_{t-2} + 4\theta_{t-1} + 2\theta_{t+1})/5, \phi / 5 \right) && \text{for } t = T-1 \\ &\sim \text{Normal} \left(-\theta_{t-2} + 2\theta_{t-1}, \phi \right) && \text{for } t = T \end{aligned}$$

Again this is equivalent to specifying

$$\theta_t \mid \theta_{-t} \sim \text{Normal} \left(\sum_k C_{tk} \theta_k, \phi M_{tt} \right) \quad \text{for } t = 1, \dots, T$$

where C_{tk} is defined as above, but this time with $W_{tk} = -1$ if $k = (t-2)$ or $(t+2)$, $W_{tk} = 4$ if $k = (t-1)$ or $(t+1)$ and t in $(3, T-2)$, $W_{tk} = 2$ if $k = (t-1)$ or $(t+1)$ and $t = 2$ or $T-1$, $W_{tk} = 0$ otherwise; $M_{tt} = 1/W_{t+}$.

Note that if the observed time points are not equally spaced, it is necessary to include missing values (NA) for the intermediate time points (see [example 9](#)).

Examples [\[top\]](#)

1. Conditional Autoregressive (CAR) models for disease mapping: Lip cancer in Scotland [\[top\]](#)

The rates of lip cancer in 56 counties in Scotland have been analysed by Clayton and Kaldor (1987) and Breslow and Clayton (1993). The form of the data includes the observed and expected cases (expected numbers based on the population and its age and sex distribution in the county), a covariate measuring the percentage of the population engaged in agriculture, fishing, or forestry, and the "position" of each county expressed as a list of adjacent counties.

County	Observed cases O_i	Expected cases E_i	Percentage in agric. x_i	SMR	Adjacent counties
1	9	1.4	16	652.2	5,9,11,19
2	39	8.7	16	450.3	7,10
...
56	0	1.8	10	0.0	18,24,30,33,45,55

We note that the extreme SMRs (Standardised Mortality Ratios) are based on very few cases.

We may smooth the raw SMRs by fitting a random-effects Poisson model allowing for spatial correlation, using the intrinsic conditional autoregressive (CAR) prior proposed by Besag, York and Mollie (1991). For the lip cancer example, the model may be written as:

$$O_i \sim \text{Poisson}(\mu_i)$$
$$\log \mu_i = \log E_i + \alpha_0 + \alpha_1 x_i / 10 + b_i$$

where α_0 is an intercept term representing the baseline (log) relative risk of disease across the study region, x_i is the covariate "percentage of the population engaged in agriculture, fishing, or forestry" in district i , with associated regression coefficient α_1 and b_i is an area-specific random effect capturing the residual or unexplained (log) relative risk of disease in area i . We often think of b_i as representing the effect of latent (unobserved) risk factors.

To allow for spatial dependence between the random effects b_i in nearby areas, we may assume a CAR prior for these terms. Technical details, including parameterisation and a discussion of suitable hyperpriors for the parameters of this model, are given in [appendix 1](#). The [car.normal](#) distribution may be used to fit this model. The code for the lip cancer data is given below:

Model

```
model {  
  
  # Likelihood  
  for (i in 1 : N) {  
    O[i] ~ dpois(mu[i])  
    log(mu[i]) <- log(E[i]) + alpha0 + alpha1 * X[i]/10 + b[i]  
    RR[i] <- exp(alpha0 + alpha1 * X[i]/10 + b[i])      # Area-specific relative risk (for maps)  
  }  
  
  # CAR prior distribution for random effects:  
  b[1:N] ~ car.normal(adj[], weights[], num[], tau)  
  for(k in 1:sumNumNeigh) {
```

```

weights[k] <- 1
}

# Other priors:
alpha0 ~ dflat()
alpha1 ~ dnorm(0.0, 1.0E-5)
tau ~ dgamma(0.5, 0.0005)      # prior on precision
sigma <- sqrt(1 / tau)        # standard deviation
}

```

Data → click on one of the arrows to open data ←

Note that the data for the adjacency matrix (variables **adj**, **num** and **SumNumNeigh**) have been generated using the **adj matrix** option of the **Adjacency Tool** menu in *GeoBUGS*. By default, this treats islands as having no neighbours, and so the three areas representing the Orkneys, Shetland and the Outer Hebrides islands in Scotland have zero neighbours. You can edit the adjacency map of Scotland to include these areas as neighbours if you wish. The **car.normal** distribution sets the value of b_i equal to zero for areas i that are islands. Hence the posterior relative risks for the Orkneys, Shetland and the Outer Hebrides in the present example will just depend on the overall baseline rate α_0 and the covariate x_i . Alternatively, you could specify a *convolution* prior for the area-specific random effects (Besag, York and Mollie 1991) which partitions the overall random effect for each area into the sum of a spatial component plus a non-spatial component. In this model, the islands will just have a non-spatial term for the random effect. See [example on lung cancer in a London Health Authority](#) for details of this model.

Inits → click on one of the arrows to open initial values ←

Note that the initial values for elements 6, 8 and 11 of the vector **b** are set to NA since these correspond to the three islands (Orkneys, Shetland and the Outer Hebrides). The values of **b** are set to zero by the **car.normal** prior for these 3 areas, and so they are not stochastic nodes.

2. Convolution priors: Lung cancer in a London Health Authority [\[top\]](#)

This example has been modified from a London Health Authority annual report. The theme of the report was inequality in health and the aim was to investigate links between poverty and ill health. To investigate this issue, the Health Authority carried out a series of disease mapping studies at census ward level. There are 44 wards in the Health Authority in this example. The data are simulated observed and expected counts of lung cancer incidence in males aged 65 and over living in the Health Authority region; a ward level index of socio-economic deprivation is also available.

We fit the following model, allowing a convolution prior for the random effects:

$$\begin{aligned}
 O_i &\sim \text{Poisson}(\mu_i) \\
 \log \mu_i &= \log E_i + \alpha + \beta \text{depriv}_i + b_i + h_i
 \end{aligned}$$

where depriv_i is the deprivation covariate, b_i are spatial random effects assigned a CAR prior, and we introduce a second set of random effects h_i for which we assume an exchangeable Normal prior. The random effect for each area is thus the sum of a spatially structured component b_i and an unstructured component h_i . This is termed a convolution prior (Besag, York and Mollie 1991; Mollie 1996). Besag, York and Mollie argue that this model is more flexible than assume only CAR random effects, since it allows the data to decide how much of the residual disease risk is due to spatially structured variation, and how much is unstructured over-dispersion.

The code for this model is given below.

Model

```

model {

for (i in 1 : N) {
  # Likelihood
  O[i] ~ dpois(mu[i])
  log(mu[i]) <- log(E[i]) + alpha + beta * depriv[i] + b[i] + h[i]
  RR[i] <- exp(alpha + beta * depriv[i] + b[i] + h[i])    # Area-specific relative risk (for maps)

  # Exchangeable prior on unstructured random effects
  h[i] ~ dnorm(0, tau.h)
}

# CAR prior distribution for spatial random effects:
b[1:N] ~ car.normal(adj[], weights[], num[], tau.b)
for(k in 1:sumNumNeigh) {
  weights[k] <- 1
}

# Other priors:
alpha ~ dflat()
beta ~ dnorm(0.0, 1.0E-5)
tau.b ~ dgamma(0.5, 0.0005)
sigma.b <- sqrt(1 / tau.b)
tau.h ~ dgamma(0.5, 0.0005)
sigma.h <- sqrt(1 / tau.h)
}

```

Data → click on one of the arrows to open data ←

Inits → click on one of the arrows to open initial values ←

3. Proper CAR model: Lip cancer revisited [\[top\]](#)

This example illustrates the use of the proper CAR distribution (*car.proper*) rather than the intrinsic CAR distribution for the area-specific random effects in the Lip cancer example (9.1). We use the definition of the **C** and **M** matrices proposed by Cressie and colleagues (see section on [proper CAR models](#) in Appendix 1).

Model

```

model {

# Set up 'data' to define spatial dependence structure
# =====
for(i in 1 : N) {
  m[i] <- 1/E[i]    # scaling factor for variance in each cell
}

# The vector C[] required as input into the car.proper distribution is a vector representation
# of the weight matrix with elements Cij. The first J1 elements of the C[] vector contain the
# weights for the J1 neighbours of area i=1; the (J1+1) to J2 elements of the C[] vector contain
# the weights for the J2 neighbours of area i=2; etc.
# To set up this vector, we need to define a variable cumsum, which gives the values of J1,
# J2, etc.; we then set up an index matrix pick[, ] with N columns corresponding to the
# i=1,...,N areas, and with the same number of rows as there are elements in the C[] vector
# (i.e. sumNumNeigh). The elements C[ (cumsum[i]+1):cumsum[i+1] ] correspond to

```

```

# the set of weights Cij associated with area i, and so we set up ith column of the matrix pick[, ]
# to have a 1 in all the rows k for which cumsum[i] < k <= cumsum[i+1], and 0's elsewhere.
# For example, let N=4 and cumsum=c(0,3,5,6,8), so area i=1 has 3 neighbours, area i=2 has 2
# neighbours, area i=3 has 1 neighbour and area i=4 has 2 neighbours. The the matrix pick[, ] is:
#       pick
#       1, 0, 0, 0,
#       1, 0, 0, 0,
#       1, 0, 0, 0,
#       0, 1, 0, 0,
#       0, 1, 0, 0,
#       0, 0, 1, 0,
#       0, 0, 0, 1,
#       0, 0, 0, 1,
#
# We can then use the inner product (inprod(,)) function in WinBUGS and the kth row of pick to
# select which area corresponds to the kth element in the vector C[]; likewise, we can use inprod(, )
# and the ith column of pick to select the elements of C[] which correspond to area i.
#
# Note: this way of setting up the C vector is somewhat convoluted!!!! In future versions, we hope the
# GeoBUGS adjacency matrix tool will be able to dump out the relevant vectors required. Alternatively,
# the C vector could be created using another package (e.g. Splus) and read into WinBUGS as data.
#
cumsum[1] <- 0
for(i in 2:(N+1)) {
  cumsum[i] <- sum(num[1:(i-1)])
}
for(k in 1 : sumNumNeigh) {
  for(i in 1:N) {
    pick[k,i] <- step(k - cumsum[i] - epsilon) * step(cumsum[i+1] - k)
    # pick[k,i] = 1 if cumsum[i] < k <= cumsum[i+1]; otherwise, pick[k,i] = 0
  }
  C[k] <- sqrt(E[adj[k]] / inprod(E[], pick[k,])) # weight for each pair of neighbours
}
epsilon <- 0.0001

# Model
# =====

# Likelihood
for (i in 1 : N) {
  O[i] ~ dpois(mu[i])
  log(mu[i]) <- log(E[i]) + S[i]

  RR[i] <- exp(S[i]) # Area-specific relative risk
  theta[i] <- alpha
}

# Proper CAR prior distribution for spatial random effects:
S[1:N] ~ car.proper(theta[], C[], adj[], num[], m[], prec, gamma)

# Other priors:
alpha ~ dnorm(0, 0.0001)
prec ~ dgamma(0.5, 0.0005) # prior on precision
v <- 1/prec # variance
sigma <- sqrt(1 / prec) # standard deviation

gamma.min <- min.bound(C[], adj[], num[], m[])
gamma.max <- max.bound(C[], adj[], num[], m[])
gamma ~ dunif(gamma.min, gamma.max)

```

```
}
```

Data → click on one of the arrows to open data ←

Initial values → Click on one of the arrows for the initial values ←

4. Bayesian kriging and spatial prediction: Surface elevation [\[top\]](#)

The data for this example are taken from Diggle and Riberio (2000) (and came originally from Davis (1973)). The data file contains the variables height, x and y, giving surface elevation at each of 52 locations (x, y) within a 310-foot square. The unit of distance is 50 feet, whereas one unit of height represents 10 feet of elevation. A Gaussian kriging model can be fitted to these data in WinBUGS 1.4 using either the [*spatial.exp*](#) or [*spatial.disc*](#) distributions. The data file also contains a set of locations x.pred and y.pred representing a 15 x 15 grid of points at which we wish to predict surface elevation. Predictions can be obtained using either the [*spatial.pred*](#) or [*spatial.unipred*](#) predictive distributions in WinBUGS 1.4

Model

```
model {  
  
  # Spatially structured multivariate normal likelihood  
  height[1:N] ~ spatial.exp(mu[], x[], y[], tau, phi, kappa)      # exponential correlation function  
  # height[1:N] ~ spatial.disc(mu[], x[], y[], tau, alpha)      # disc correlation function  
  
  for(i in 1:N) {  
    mu[i] <- beta  
  }  
  
  # Priors  
  beta ~ dflat()  
  tau ~ dgamma(0.001, 0.001)  
  sigma2 <- 1/tau  
  
  # priors for spatial.exp parameters  
  phi ~ dunif(0.05, 20)      # prior range for correlation at min distance (0.2 x 50 ft) is 0.02 to 0.99  
                             # prior range for correlation at max distance (8.3 x 50 ft) is 0 to 0.66  
  kappa ~ dunif(0.05, 1.95)  
  
  # priors for spatial.disc parameter  
  # alpha ~ dunif(0.25, 48)      # prior range for correlation at min distance (0.2 x 50 ft) is 0.07 to 0.96  
                                 # prior range for correlation at max distance (8.3 x 50 ft) is 0 to 0.63  
  
  # Spatial prediction  
  
  # Single site prediction  
  for(j in 1:M) {  
    height.pred[j] ~ spatial.unipred(beta, x.pred[j], y.pred[j], height[])  
  }  
  
  # Only use joint prediction for small subset of points, due to length of time it takes to run  
  for(j in 1:10) { mu.pred[j] <- beta }  
  height.pred.multi[1:10] ~ spatial.pred(mu.pred[], x.pred[1:10], y.pred[1:10], height[])  
  
}
```

Data → Click on one of the arrows for the data ←

Initial values

- Click on one of the arrows for initial values for spatial.exp model ←
- Click on one of the arrows for initial values for spatial.disc model ←

The GeoBUGS map tool can be used to produce an approximate map of the posterior mean predicted surface elevation. It is not possible to map point locations using GeoBUGS. However, a map file called **elevation** is already loaded in the GeoBUGS Map Tool --- this is a 15 x 15 grid with the centroids of the grid cell corresponding to the locations of each of the prediction points. You can use this to produce a map of the posterior mean (or other posterior summaries) of the vector of predicted values **height.pred**.

5. Poisson-gamma spatial moving average (convolution) model: Distribution of hickory trees in Duke forest [\[top\]](#)

Ickstadt and Wolpert (1998) and Wolpert and Ickstadt (1998) analyse the spatial distribution of hickory trees (a subdominant species) in a 140 metre x 140 metre research plot in Duke Forest, North Carolina, USA, with the aim of investigating the regularity of the distribution of trees: spatial clustering would reveal that the subdominant species is receding from, or encroaching into, a stand recovering from a disturbance, while regularity would suggest a more stable and undisturbed recent history.

The data comprise counts of trees, Y_i , in each of $i=1, \dots, 16$ 30m x 30m quadrats (each having area $A_i = 900$ m²) on a 4x4 grid covering the research plot (excluding a 10m boundary region to minimise edge effects), together with the x and y co-ordinates of the centroid of each plot (relative to the south-west corner of the research plot). Ickstadt and Wolpert (1998) fit a Poisson-gamma spatial moving average model as follows:

$$\begin{aligned} Y_i &\sim \text{Poisson}(\mu_i) & i = 1, \dots, 16 \\ \mu_i &= A_i * \lambda_i \\ \lambda_i &= \sum_j k_{ij} \gamma_j \end{aligned}$$

where γ_j can be thought of as latent unobserved risk factors associated with locations or sub-regions of the study region indexed by j , and k_{ij} is a kernel matrix of 'weights' which depend on the distance or proximity between latent location j and quadrat i of the study region (see [Appendix 2](#) for further details of this convolution model). Ickstadt and Wolpert (1998) take the locations of the latent γ_j to be the same as the partition of the study region used for the observed data i.e. $j = 1, \dots, 16$ with latent sub-region j being the same as quadrat i of the study region. Note that it is not necessary for the latent partition to be the same as the observed partition - see [example 6](#). The γ_j are assigned independent gamma prior distributions

$$\gamma_j \sim \text{Gamma}(\alpha, \beta) \quad j = 1, \dots, 16.$$

Ickstadt and Wolpert (1998) consider two alternative kernel functions: an adjacency-based kernel and a distance-based kernel. The adjacency based kernel is defined as:

$$\begin{aligned} k_{ij} &= 1 && \text{if } i = j \\ k_{ij} &= \exp(\theta_2)/n_i && \text{if } j \text{ is a nearest neighbour of } i \text{ (only north-south and east-west neighbours considered)} \\ &&& \text{and } n_i \text{ is the number of neighbours of area } i \\ k_{ij} &= 0 && \text{otherwise} \end{aligned}$$

The distance based kernel is defined as:

$$k_{ij} = \exp(-[d_{ij} / \exp(\theta_2)]^2) \quad \text{where } d_{ij} \text{ is the distance between the centroid of areas } i \text{ and } j$$

For both kernels, the parameter $\exp(\theta_2)$ reflects the degree of spatial dependence or clustering in the data, with large values of θ_2 suggesting spatial correlation and irregular distribution of hickory trees.

Ickstadt and Wolpert (1998) elicit expert prior opinion concerning the unknown hyperparameters $\theta_0 = \log(\alpha)$, $\theta_1 = -\log(\beta)$ and θ_2 and translate this information into Normal prior distributions for θ_0 , θ_1 and θ_2 .

This model may be implemented in WinBUGS 1.4 using the [pois.conv](#) distribution. The code is given below.

Model

```

model {

# likelihood
for (i in 1:N) {
  Y[i] ~ dpois.conv(mu[i,])
  for (j in 1:J) {
    mu[i, j] <- A[i] * lambda[i, j]
    lambda[i, j] <- k[i, j] * gamma[j]
  }
}

# priors (see Ickstadt and Wolpert (1998) for details of prior elicitation)
for (j in 1:J) {
  gamma[j] ~ dgamma(alpha, beta)
}
alpha <- exp(theta0)
beta <- exp(-theta1)

theta0 ~ dnorm(-0.383, 1)
theta1 ~ dnorm(-5.190, 1)
theta2 ~ dnorm(-1.775, 1)      # prior on theta2 for adjacency-based kernel
# theta2 ~ dnorm(1.280, 1)    # prior on theta2 for distance-based kernel

# compute adjacency-based kernel
for (i in 1:N) {          # Note N = J in this example (necessary for adjacency-based kernel)
  k[i, i] <- 1
  for (j in 1:J) {
    d[i, j] <- sqrt((x[i] - x[j])*(x[i] - x[j]) + (y[i] - y[j])*(y[i] - y[j])) # distance between areas i and j
                                                    # (only needed to help compute
                                                    # nearest neighbour weights;
                                                    # alternatively, read matrix w from file)

    w[i, j] <- step(30.1 - d[i, j])          # nearest neighbour weights:
                                                    # areas are 30 sq m, so any pair of areas with
                                                    # centroids > 30m apart are not nearest neighbours
                                                    # (0.1 added to account for numeric imprecision in d)
  }
  for (j in (i+1):J) {
    k[i, j] <- w[i, j] * exp(theta2) / (sum(w[i,]) - 1)
    k[j, i] <- w[j, i] * exp(theta2) / (sum(w[j,]) - 1)
  }
}

# alternatively, compute distance-based kernel
# for (i in 1:N) {
#   k[i, i] <- 1
#   for (j in 1:J) {
#     d[i, j] <- sqrt((x[i] - x[j])*(x[i] - x[j]) + (y[i] - y[j])*(y[i] - y[j])) # distance between areas i and j
#   }
#   for (j in (i+1):J) {
#     k[i, j] <- exp(-pow(d[i, j]/exp(theta2), 2))
#     k[j, i] <- exp(-pow(d[j, i]/exp(theta2), 2))
#   }
# }

```



```

# }
# }

# summary quantities for posterior inference
for (i in 1:N) {
  density[i] <- sum(lambda[i, ])          # smoothed density of hickory trees (per sq metre) in area i
  obs.density[i] <- Y[i]/A[i]           # observed density of hickory trees (per sq metre) in area i
}
spatial.effect <- exp(theta2)           # large values indicate strong spatial dependence;
                                         # spatial.effect -> 0 indicates spatial independence
}

```

Data → click on one of the arrows to open data ←

Initial values → click on one of the arrows to open initial values ←

Results

Assuming adjacency-based kernel (equivalent to Ickstadt and Wolpert's model M_S):

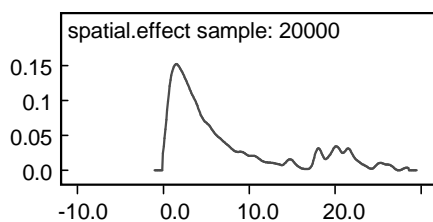
node	mean	sd	MC error	2.5%	median	97.5%	start	sample
spatial.effect	0.2856	0.1793	0.009916	0.04003	0.2588	0.7088	10001	10000
theta0	-0.6121	0.4562	0.02313	-1.519	-0.6084	0.228	10001	10000
theta1	-5.111	0.5088	0.0179	-6.046	-5.133	-4.05	10001	10000
theta2	-1.487	0.7497	0.04143	-3.218	-1.352	-0.3441	10001	10000

Ickstadt and Wolpert report a posterior mean (sd) of 0.281 (0.156) for the spatial effect, $\exp(\theta_2)$, from their analysis using a 4x4 partition of the study region (Table 1.3).

Assuming distance-based kernel (equivalent to Ickstadt and Wolpert's model M_D):

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
spatial.effect	8.042	7.474	0.5676	0.5371	4.684	23.9	20001	20000
theta0	-0.297	0.5198	0.02996	-1.533	-0.2463	0.5766	20001	20000
theta1	-5.276	0.522	0.01989	-6.197	-5.308	-4.157	20001	20000
theta2	1.563	1.108	0.06682	-0.6216	1.544	3.174	20001	20000

Ickstadt and Wolpert report a posterior mean (sd) of 7.449 (6.764) for the spatial effect, $\exp(\theta_2)$, from their analysis using a 4x4 partition of the study region (Table 1.3), and posterior means of -0.02, -5.28 and 1.54 respectively for theta0, theta1 and theta2. Ickstadt and Wolpert noted that the posterior for the spatial effect was multi-modal, as seen in the density plot below:



Note however that convergence of this distance-based model is not very good, and the results reported above are not particularly stable (the adjacency-based model appears to converge much better). This may be partly due to the rather strong prior information about theta2 for the distance-based model which, as noted by Ickstadt and Wolpert, appears to conflict with the data.

6. Poisson-gamma spatial moving average (convolution) model: Ecological

regression of air pollution and respiratory illness in children [\[top\]](#)

Best et al (2000b) applied the Poisson-gamma convolution model to a small-area ecological regression analysis of traffic-related air pollution and respiratory illness in children living in the Huddersfield region of northern England. They carried out two analyses, one using a partition of the study region into 427 administrative enumeration districts, and the other partitioning the region into 605 regular 750m² grid cells. Here we consider the latter partition of the study region. The following data were available:

Y_i = number of cases of self-reported frequent cough amongst children aged 7-9 in each of the $i=1, \dots, 605$ areas (750 m² grid cells)

pop_i = (estimated) population of 7-9 year old children in each area, based on pro-rated 1991 enumeration district census counts. (Note that this leads to <1 child per area in some areas, due to the pro-rata split; counts are also divided by 100, to give rates per 100 children).

$NO_{2,i}$ = average annual nitrogen dioxide concentration in each area, measured in $\mu\text{g m}^{-3}$ above the baseline concentration for the study region of approximately $20\mu\text{g m}^{-3}$

Best et al (2000b) fitted an identity-link spatial moving average Poisson regression model:

$$\begin{aligned} Y_i &\sim \text{Poisson}(\mu_i) \\ \mu_i &= pop_i * \lambda_i \\ \lambda_i &= \beta_0 + \beta_1 NO_{2,i} + \beta_2 \sum_j k_{ij} \gamma_j \end{aligned}$$

where γ_j can be thought of as latent unobserved risk factors associated with locations or sub-regions of the study region indexed by j . These locations or sub-regions are typically defined by the user, and are chosen to represent a partition of the region that is appropriate for capturing unmeasured spatial variation in the disease rate. Best et al (2000b) define the latent γ_j parameters on a 12×8 grid of approx 3km² quadrats covering the Huddersfield study region. k_{ij} are then the elements of a 605×96 Gaussian kernel matrix, with

$$k_{ij} = 1 / (2\pi\rho^2) \exp(-d_{ij}^2 / 2\rho^2)$$

where d_{ij} represents the Euclidean distance between the centroid of area i of the study region and the location of the j^{th} latent factor. Note that if the latent factors are defined on sub-regions rather than points, it is preferable to integrate the above kernel over the latent sub-region (i.e. over all possible distances between the centroid of area i and all locations within the latent sub-region) - the Splus / R function given in the data file below implements this integration when computing the kernel matrix. $\rho > 0$ is a distance scale governing how rapidly the kernel weights (i.e. the influence of the j^{th} latent factor on the disease risk in the i^{th} area) decline with increasing distance. Best et al (2000b) fix $\rho = 1$ km. By fixing ρ , the kernel matrix can be pre-computed outside of WinBUGS (see link in the data file for Splus / R code for doing this), thus simplifying the implementation. It is possible to treat ρ as uncertain (see [example 5](#)), but this entails calculating the elements k_{ij} at each MCMC iteration within the BUGS code: this leads to big increases in computation time for all but very small study regions and latent grids. (Note that since the overall scale parameter of the Gaussian kernel is common to all elements k_{ij} it can be factored out of the external calculation of the kernel, and included as an uncertain parameter (here called β_2) in the BUGS model.

Independent gamma prior distributions are assumed for the uncertainty quantities β_0 , β_1 , β_2 and γ_j as this enables the MCMC sampler to exploit conjugacy with the Poisson likelihood.

See Best et al (2000b) for full details and interpretation of the model, and [Appendix 2](#) for further technical information about the Poisson-gamma model.

The code for this model is given below.

Model

model {

```

# Likelihood
for (i in 1 : I) {
  Y[i] ~ dpois.conv(mu[i, ])      # convolution likelihood:
                                  #   Y[i] ~ Poisson( sum_j { mu[i, j] } )
                                  # where mu[i, j] = pop[i] * lambda[i, j] and
                                  # pop[i] = (standardised) population (in 100's) in area i
                                  # lambda[i, j] = disease rate in area i attributed to jth 'source', where
                                  # the sources include both observed and latent covariates.

  for (j in 1 : J) {
    k1[i, j] <- ker[i, j] / prec      # rescale kernel matrix
    lambda[i, j] <- beta2 * gamma[j] * k1[i, j] # jth 'source' (jth latent spatial grid cell)
  }
  lambda[i,J+1] <- beta0              # (J+1)th 'source' (overall intercept; represents background rate)
  lambda[i,J+2] <- beta1 * NO2[i]    # (J+2)th 'source' (effect of observed NO2 covariate in area i)
                                      # Note: if additional covariates have been measured, these
                                      # should be included in the same way as NO2, giving terms
                                      # lambda[i, J+3], lambda[i, J+4],.....etc.

  for (j in 1 : J+2) {
    mu[i, j] <- lambda[i, j] * pop[i]
  }
}

# Priors
# See Table 22.1 in Best et al. (2000b) for further details.

# Priors for latent spatial grid cell (gamma) parameters:
#
# Assume gamma_j ~ gamma(a, t)
#
# Prior shape parameter for gamma distn on gamma[j]'s will change in proportion to the size of
# the latent grid cells, to ensure aggregation consistency. The prior inverse scale parameter for
# gamma distn on gamma[j]'s stays constant across different partitions, but depends on the
# units used to measure the area of the latent grid cells: here we are assuming 'area' is in
# sq kilometres, and we take tauG = 1 (per km^2) which corresponds to assuming that our prior
# guess at the value of the gamma[j]'s is based on 'observing' about
# 1 * (area of study region = 30*20km^2) = 600
# 'prior points' over the study region.

alphaG <- tauG * area      # Note: area of latent grid cells is read in from data file
tauG <- 1

for (j in 1 : J) {
  gamma[j] ~ dgamma(alphaG, tauG)
}

# Priors for beta coefficients:
#
# Assume priors on each beta are of the form beta ~ gamma(a, t).
#
# Here, parameter of the prior are chosen by setting the prior mean to give an equal number of cases
# allocated to each of the 3 'sources' (baseline, NO2 and latent), i.e.
# prior mean = a / t = a * 3 * Xbar / Ybar,
# where Ybar is the overall disease rate in number of cases per unit population,
# and Xbar is the population-weighted mean of covariate X.
#
# We also take shape parameter a = 0.575, since this gives the ratio of the 90th : 10th percentile
# of the prior distn = 100, which is suitably diffuse.

YBAR <- sum(Y[])/sum(pop[])      # mean number of cases per unit population

```

```

NO2BAR <- inprod(NO2[], pop[]) / sum(pop[]) # population-weighted mean NO2

alpha0<- 0.575 # Shape parameter for Intercept (beta0)
tau0 <- 3 * alpha0 / YBAR # Inverse scale parameter for Intercept (beta0)
alpha1<- 0.575 # Shape parameter for effect of NO2 (beta1)
tau1 <- 3 * alpha1 * NO2BAR / YBAR # Inverse scale parameter for effect of NO2 (beta1)
alpha2 <- 0.575 # Shape parameter for Latent coefficient (beta2)
tau2 <- 3 * alpha2 / YBAR # Inverse scale parameter for Latent coefficient (beta2)

beta0 ~ dgamma(alpha0, tau0)
beta1 ~ dgamma(alpha1, tau1)
beta2 ~ dgamma(alpha2, tau2)

# Summary quantities for posterior inference
for(i in 1:l) {
  RATE[i] <- sum(mu[i, ])/pop[i] # Overall disease rate in area i
  LATENT[i] <- beta2 * inprod(k1[i,], gamma[]) # Rate associated with latent spatial factor in area i
}

# Expected rate (number of cases per unit population) attributed to each source
# (see Table 22.2 in Best et al (2000b) )
rate.base <- beta0 # Background (overall baseline) rate per 100 population
rate.no2 <- beta1 * inprod(pop[], NO2[]) / sum(pop[]) # Number of cases per 100 population attributed
# to NO2 = beta1 * population-weighted mean
# of NO2 across the study region.
rate.latent <- inprod(pop[], LATENT[]) / sum(pop[]) # Number of cases per 100 population attributed to
# latent sources = beta2 * population-weighted mean
# value of the latent spatial factor (sum_j k_ij gamma_j)

# Percentage of cases attributed to each source:
Total <- rate.base + rate.no2 + rate.latent
PC.base <- rate.base / Total * 100 # % cases attributed to baseline risk
PC.no2 <- rate.no2 / Total * 100 # % cases attributed to NO2 exposure
PC.latent <- rate.latent / Total * 100 # % cases attributed to latent spatial factors
}

```

Data → [Click here for the data file](#) ←

Initial values → [Click here for the initial values](#) ←

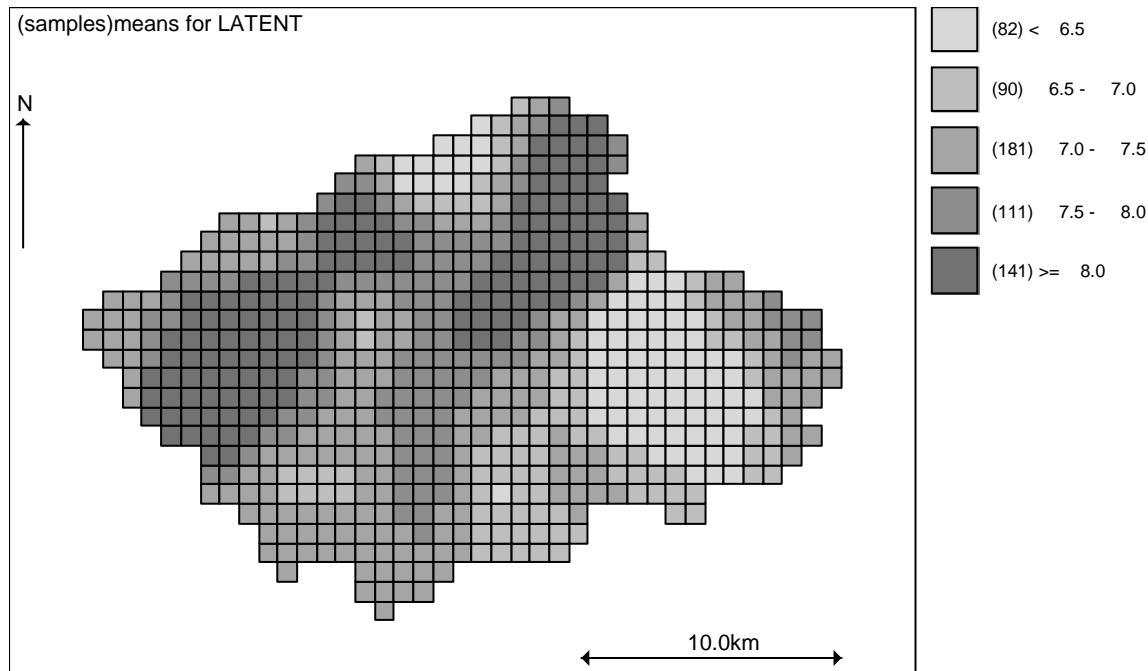
Results

A 10000 iteration burn-in followed by a further 20000 updates gave the following results

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
PC.base	6.674	8.565	0.6041	0.01874	3.805	32.94	10000	20001
PC.latent	74.95	13.99	0.9768	41.11	76.09	97.39	10000	20001
PC.no2	18.38	10.92	0.7131	0.4521	18.08	41.31	10000	20001
rate.base	0.6806	0.8744	0.06162	0.001946	0.3872	3.376	10000	20001
rate.latent	7.641	1.452	0.09948	4.203	7.748	10.01	10000	20001
rate.no2	1.874	1.115	0.07273	0.04625	1.843	4.223	10000	20001

These differ slightly from those in Best et al (2000b) Table 22.2 since the data in this example have been randomly perturbed for confidentiality. However, the picture is broadly similar, with 1.84 (95% interval 0.046 to 4.2) cases per 100 population attributed to NO2 exposure, 7.75 (95% interval 4.2 to 10.0) cases per 100 population attributed to latent spatially varying risk factors and the remaining 0.39 (95% interval 0.0019 to 3.4) cases per 100 population attributed to the (spatially constant) baseline risk.

A map showing the posterior mean rate (per 100 population) of frequent cough attributed to the latent risk factors (cf Fig 22.6b in Best et al, 2000b) in each area in the study region is show below. The relevant map file is available from the GeoBUGS 1.2 map tool and is called **Huddersfield_750m_grid**.



7. Intrinsic multivariate CAR prior for mapping multiple diseases: Oral cavity cancer and lung cancer in West Yorkshire, UK [\[top\]](#)

This example illustrates the use of the multivariate CAR prior distribution for joint mapping of two diseases. The data represent observed and age and sex standardised expected counts of incident cases of oral cavity and lung cancer in each of 126 electoral wards in the West Yorkshire region of England between 1986 and 1991. Some of the counts have been randomly perturbed by a small amount for confidentiality.

Since both outcomes are rare, the mortality counts Y_{ik} for cancer k ($k=1, 2$) in area i ($i=1, 126$) are assumed to follow independent Poisson distributions, conditional on an unknown mean μ_{ik}

$$Y_{ik} \sim \text{Poisson}(\mu_{ik})$$

$$\log \mu_{ik} = \log E_{ik} + \alpha_k + S_{ik}$$

where E_{ik} is the age and sex standardised expected count for cancer k in area i , α_k is an intercept term representing the baseline (log) relative risk of cancer k across the study region, and S_{ik} is the area- and cancer-specific log relative risk of death. For cancer k , we assume that these log relative risks are spatially correlated across areas, and within area i , we assume that the log relative risks for cancer 1 (oral cavity) and cancer 2 (lung) are also correlated due to dependence on shared area-level unmeasured risk factors. We represent these correlation assumptions using an intrinsic bivariate CAR prior for the 2×126 dimensional matrix of S_{ik} values. Technical details of this prior distribution are given in [appendix 1](#). The ***mv.car*** distribution may be used to fit this model. The WinBUGS code is given below:

Model

```
model {
  # Likelihood
  for (i in 1:Nareas) {
```

```

for (k in 1:Ndiseases) {
  Y[i, k] ~ dpois(mu[i, k])
  log(mu[i, k]) <- log(E[i, k]) + alpha[k] + S[k, i] # Note dimension of S is reversed:
                                                    # rows=k, cols=i because mv.car
                                                    # assumes rows represent variables
                                                    # (diseases) and columns represent
                                                    # observations (areas).
}
# The GeoBUGS map tool can only map vectors, so need to create separate vector
# of quantities to be mapped, rather than an array (i.e. RR[i,k] won't work!)
RR1[i] <- exp(alpha[1] + S[1,i]) # area specific relative risk for disease 1 (oral)
RR2[i] <- exp(alpha[2] + S[2,i]) # area specific relative risk for disease 2 (lung)
}

# MV CAR prior for the spatial random effects
S[1:Ndiseases, 1:Nareas] ~ mv.car(adj[], weights[], num[], omega[ , ]) # MVCAR prior
for (i in 1:sumNumNeigh) { weights[i] <- 1 }

# Other priors
for (k in 1:Ndiseases) {
  alpha[k] ~ dflat()
}
omega[1:Ndiseases, 1:Ndiseases] ~ dwish(R[ , ], Ndiseases) # Precision matrix of MVCAR
sigma2[1:Ndiseases, 1:Ndiseases] <- inverse(omega[ , ]) # Covariance matrix of MVCAR
sigma[1] <- sqrt(sigma2[1, 1]) # conditional SD of S[1,] (oral cancer)
sigma[2] <- sqrt(sigma2[2, 2]) # conditional SD of S[2,] (lung cancer)
corr <- sigma2[1, 2] / (sigma[1] * sigma[2]) # within-area conditional correlation
# between oral and lung cancers.
}

```

Data → [Click here for the data file](#) ←

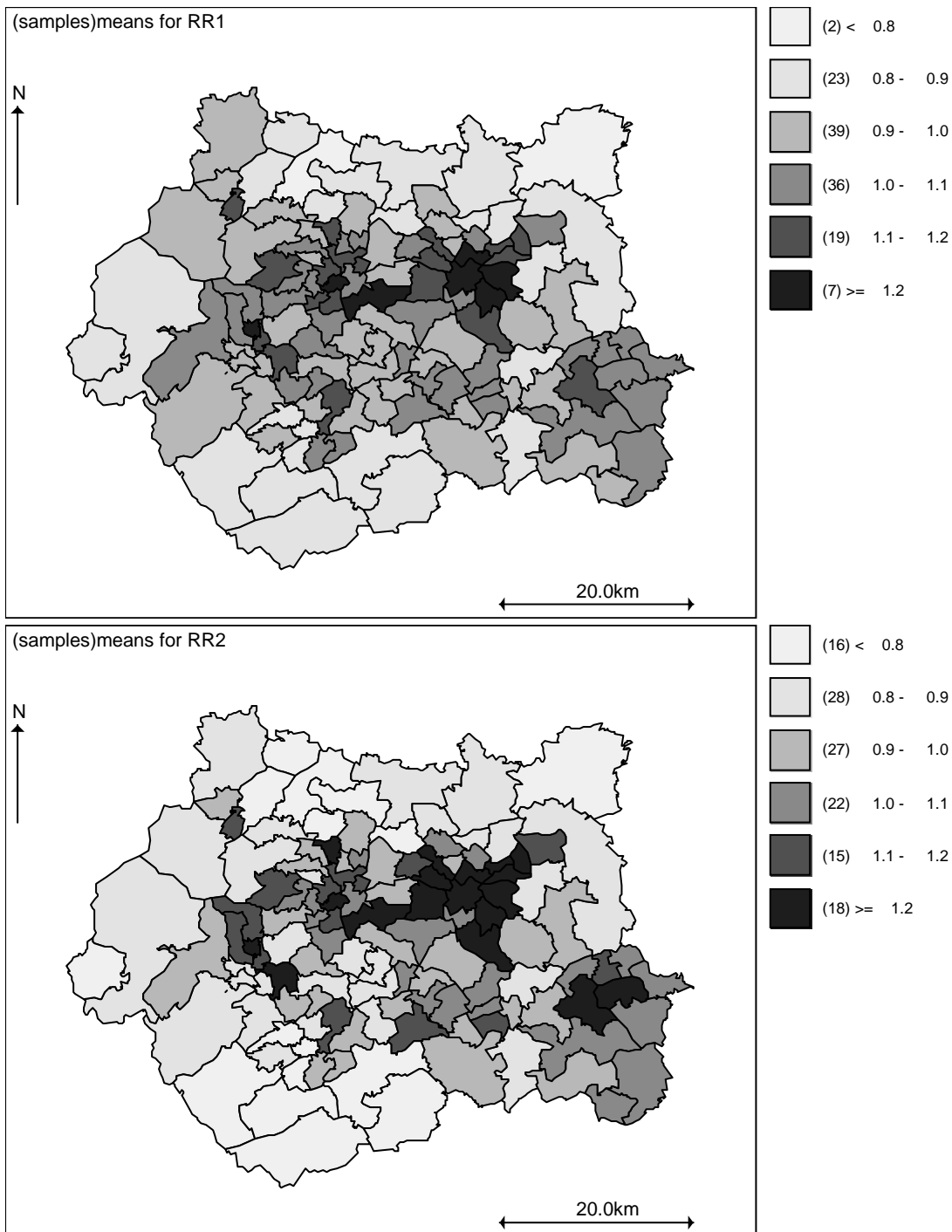
Initial values → [Click here for the initial values](#) ←

Results

A 1000 iteration burn-in followed by a further 49000 updates gave the following results

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
alpha[1]	-0.008137	0.03786	2.714E-4	-0.08334	-0.007713	0.06579	1001	49000
alpha[2]	-0.02277	0.01393	1.28E-4	-0.05064	-0.02281	0.004718	1001	49000
corr	0.8435	0.1258	0.004495	0.5097	0.8807	0.9728	1001	49000
sigma[1]	0.2631	0.07726	0.003216	0.1255	0.2589	0.4288	1001	49000
sigma[2]	0.3662	0.03101	2.766E-4	0.3096	0.3648	0.431	1001	49000

The posterior correlation between risk of oral cavity and lung cancers is 0.84 (95% CI: 0.51, 0.97) suggesting strong shared geographical pattern of risk between the two diseases. Maps of the posterior mean relative risk for each disease are shown below. The relevant map file is available from the GeoBUGS 1.2 map tool and is called **WestYorkshire**.



Multivariate convolution prior

Note that it is also possible to fit a multivariate equivalent of the BYM convolution model, by including an additional unstructured random effect for each area and disease, and assuming independent bivariate normal priors for each pair of unstructured effects within an area:

```

model {
  # Likelihood
  for (i in 1:Nareas) {
    for (k in 1:Ndiseases) {
      Y[i, k] ~ dpois(mu[i, k])
      log(mu[i, k]) <- log(E[i, k]) + alpha[k] + S[k, i] + U[i, k] # Dimension of S reversed as before
    }
  }
  RR1[i] <- exp(alpha[1] + S[1,i]) # area specific relative risk for disease 1 (oral)
  RR2[i] <- exp(alpha[2] + S[2,i]) # area specific relative risk for disease 2 (lung)
}

```

```

}

# MV CAR prior for the spatial random effects
S[1:Ndiseases, 1:Nareas] ~ mv.car(adj[], weights[], num[], omega[ , ]) # MVCAR prior
for (i in 1:sumNumNeigh) { weights[i] <- 1 }

# Bivariate normal prior for unstructured random effects within each area
for (i in 1:Nareas) {
  U[i, 1:Ndiseases] ~ dmnorm(zero[], tau[ , ]) # Unstructured multivariate normal
}

# Other priors
for (k in 1:Ndiseases) {
  alpha[k] ~ dflat()
}
omega[1:Ndiseases, 1:Ndiseases] ~ dwish(R[ , ], Ndiseases) # Precision matrix of MVCAR
sigma2[1:Ndiseases, 1:Ndiseases] <- inverse(omega[ , ]) # Covariance matrix of MVCAR

sigma[1] <- sqrt(sigma2[1, 1]) # conditional SD of S[1,] (oral cancer)
sigma[2] <- sqrt(sigma2[2, 2]) # conditional SD of S[2,] (lung cancer)
corr <- sigma2[1, 2] / (sigma[1] * sigma[2]) # within-area conditional correlation between spatial
# component of variation in oral and lung cancers.

tau[1:Ndiseases, 1:Ndiseases] ~ dwish(Q[ , ], Ndiseases) # Precision matrix of MV Normal
sigma2.U[1:2, 1:2] <- inverse(tau[ , ]) # Covariance matrix of MV Normal

sigma.U[1] <- sqrt(sigma2.U[1, 1])
sigma.U[2] <- sqrt(sigma2.U[2, 2])
corr.U <- sigma2.U[1, 2] / (sigma.U[1] * sigma.U[2]) # within-area correlation between unstructured
# component of variation in oral and lung cancers

# within-area conditional correlation between total random effect (i.e. spatial+unstructured components)
# for oral cancer and for lung cancer
corr.sum <- (sigma2[1, 2] + sigma2.U[1, 2]) /
  (sqrt(sigma2[1, 1] + sigma2.U[1, 1]) * sqrt(sigma2[2, 2] + sigma2.U[2, 2]))
}

```

Data → [Click here for the data file](#) ←

Initial values → [Click here for the initial values](#) ←

Results

A 10000 iteration burn-in followed by a further 50000 updates gave the following results

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
alpha[1]	-0.01189	0.03893	3.608E-4	-0.08872	-0.01172	0.06401	10001	50000
alpha[2]	-0.02289	0.01494	1.798E-4	-0.05267	-0.02281	0.00649	10001	50000
corr	0.8207	0.1939	0.009126	0.2946	0.886	0.9801	10001	50000
corr.U	0.3753	0.4397	0.01982	-0.6377	0.4927	0.9173	10001	50000
corr.sum	0.7521	0.1526	0.006902	0.3648	0.7876	0.944	10001	50000
sigma[1]	0.1999	0.08446	0.004522	0.06658	0.1899	0.3883	10001	50000
sigma[2]	0.3005	0.04725	0.001905	0.2055	0.302	0.3899	10001	50000
sigma.U[1]	0.104	0.04577	0.002039	0.04099	0.09494	0.2111	10001	50000
sigma.U[2]	0.09237	0.02803	0.001276	0.04331	0.09127	0.1488	10001	50000

Again, the posterior correlation between the spatially structured risk components (S) for oral cavity and lung cancers is high (0.82; 95% CI: 0.29, 0.98), although correlation between the unstructured risk components (U) is less strong (0.38; 95% CI: -0.64, 0.92). Since the spatial component of risk dominates, the correlation between the total random effect (U + S) for oral cancer and lung cancer is also high (0.75; 95% CI: 0.36, 0.94),

again suggesting strong shared geographical pattern of risk between the two diseases.

8. Shared component model for mapping multiple diseases: Oral cavity cancer and lung cancer cancer in West Yorkshire, UK [\[top\]](#)

Knorr-Held and Best (2001) analysed data on mortality from oral cavity and oesophageal cancer in Germany using a shared component model. This model is similar in spirit to conventional factor analysis, and partitions the geographical variation in two diseases into a common (shared) component (ϕ), and two disease-specific (residual) components (ψ_1 and ψ_2). Making the rare disease assumption, the likelihood for each disease is assumed to be independent Poisson, conditional on an unknown mean μ_{ik}

$$Y_{ik} \sim \text{Poisson}(\mu_{ik})$$

$$\log \mu_{i1} = \log E_{i1} + \alpha_1 + \phi * \delta + \psi_{i1}$$

$$\log \mu_{i2} = \log E_{i2} + \alpha_2 + \phi / \delta + \psi_{i2}$$

where Y_{ik} and E_{ik} are the observed and age/sex standardised expected counts for cancer k in area i respectively, α_k is an intercept term representing the baseline (log) relative risk of cancer k across the study region, and δ is a scaling factor to allow the risk gradient associated with the shared component to be different for each disease (this is in some sense similar to the factor loadings in conventional factor analysis - see Knorr-Held and Best (2001) for more details). Each of the three components (ϕ , ψ_1 and ψ_2) is assumed to be spatially structured with zero mean; the components are assumed to be independent of each other. Knorr-Held and Best (2001) used a spatial partition model as a prior for each component. In this example, we fit a similar model, but assume an BYM convolution prior for each component. Here we consider the data from [example 7](#) on incidence of oral cavity cancer and lung cancer in 126 electoral wards in the West Yorkshire region of England:

```
model {
  # Likelihood
  for (i in 1:Nareas) {
    for (k in 1:Ndiseases) {
      Y[i,k] ~ dpois(mu[i,k])
      log(mu[i,k]) <- log(E[i,k]) + alpha[k] + eta[i, k]
    }
  }

  for(i in 1:Nareas) {
    # Define log relative risk in terms of disease-specific (psi) and shared (phi) random effects
    eta[i,1] <- phi[i] *delta + psi[1, i] # changed order of k and i index for psi
    eta[i,2] <- phi[i] /delta + psi[2, i] # (needed because car.normal assumes right hand index is areas)
  }

  # Spatial priors (BYM) for the disease-specific random effects
  for (k in 1:Ndiseases) {
    for (i in 1:Nareas) {
      psi[k, i] <- U.sp[k, i] + S.sp[k, i] # convolution prior = sum of unstructured and spatial effects
      U.sp[k, i] ~ dnorm(0, tau.unstr[k]) # unstructured disease-specific random effects
    }
    S.sp[k,1:Nareas] ~ car.normal(adj[], weights[], num[], tau.spatial[k]) # spatial disease-specific effects
  }

  # Spatial priors (BYM) for the shared random effects
  for (i in 1:Nareas) {
    phi[i] <- U.sh[i] + S.sh[i] # convolution prior = sum of unstructured and spatial effects
    U.sh[i] ~ dnorm(0, omega.unstr) # unstructured shared random effects
  }
  S.sh[1:Nareas] ~ car.normal(adj[], weights[], num[], omega.spatial) # spatial shared random effects
}
```

```

for (k in 1:sumNumNeigh) { weights[k] <- 1 }

# Other priors
for (k in 1:Ndiseases) {
  alpha[k] ~ dflat()
  tau.unstr[k] ~ dgamma(0.5, 0.0005)
  tau.spatial[k] ~ dgamma(0.5, 0.0005)
}
omega.unstr ~ dgamma(0.5, 0.0005)
omega.spatial ~ dgamma(0.5, 0.0005)

logdelta ~ dnorm(0, 5.9)      # scaling factor for relative strength of shared component for each disease
delta <- exp(logdelta)      # (prior assumes 95% probability that delta^2 is between 1/5 and 5;
                           # lognormal assumption is invariant to which disease is labelled 1
                           # and which is labelled 2)
RR.ratio <- pow(delta, 2)    # ratio (relative risk of disease 1 associated with shared component) to
                           # (relative risk of disease 2 associated with shared component)
                           # - see Knorr-Held and Best (2001) for further details

# Relative risks and other summary quantities
# The GeoBUGS map tool can only map vectors, so need to create separate vector
# of quantities to be mapped, rather than an array (i.e. totalRR[i,k] won't work!)
for (i in 1:Nareas) {
  SMR1[i] <- Y[i,1]/E[i,1]    # SMR for disease 1 (oral)
  SMR2[i] <- Y[i,2]/E[i,2]    # SMR for disease 2 (lung)

  totalRR1[i] <- exp(eta[i,1]) # overall RR of disease 1 (oral) in area i
  totalRR2[i] <- exp(eta[i,2]) # overall RR of disease 2 (lung) in area i

  specificRR1[i] <- exp(psi[1,i]) # residual RR specific to disease 1 (oral cancer)
  specificRR2[i] <- exp(psi[2,i]) # residual RR specific to disease 2 (lung cancer)

  sharedRR[i] <- exp(phi[i])    # shared component of risk common to both diseases
  logsharedRR1[i] <- phi[i]*delta # Note that this needs to be scaled by delta or 1/delta if the
  logsharedRR2[i] <- phi[i]/delta # absolute magnitude of shared RR for each disease is of interest
}
var.shared[1] <- sd(logsharedRR1[])*sd(logsharedRR1[]) # empirical variance of shared effects
                                                    # (scaled for disease 1)
var.shared[2] <- sd(logsharedRR2[])*sd(logsharedRR2[]) # empirical variance of shared effects
                                                    # (scaled for disease 2)
var.specific[1] <- sd(psi[1,])*sd(psi[1,]) # empirical variance of disease 1 specific effects
var.specific[2] <- sd(psi[2,])*sd(psi[2,]) # empirical variance of disease 2 specific effects

# fraction of total variation in relative risks for each disease that is explained by the shared component
frac.shared[1] <- var.shared[1] / (var.shared[1] + var.specific[1])
frac.shared[2] <- var.shared[2] / (var.shared[2] + var.specific[2])
}

```

Data → [Click here for the data file](#) ←

Initial values → [Click here for the initial values](#) ←

Results

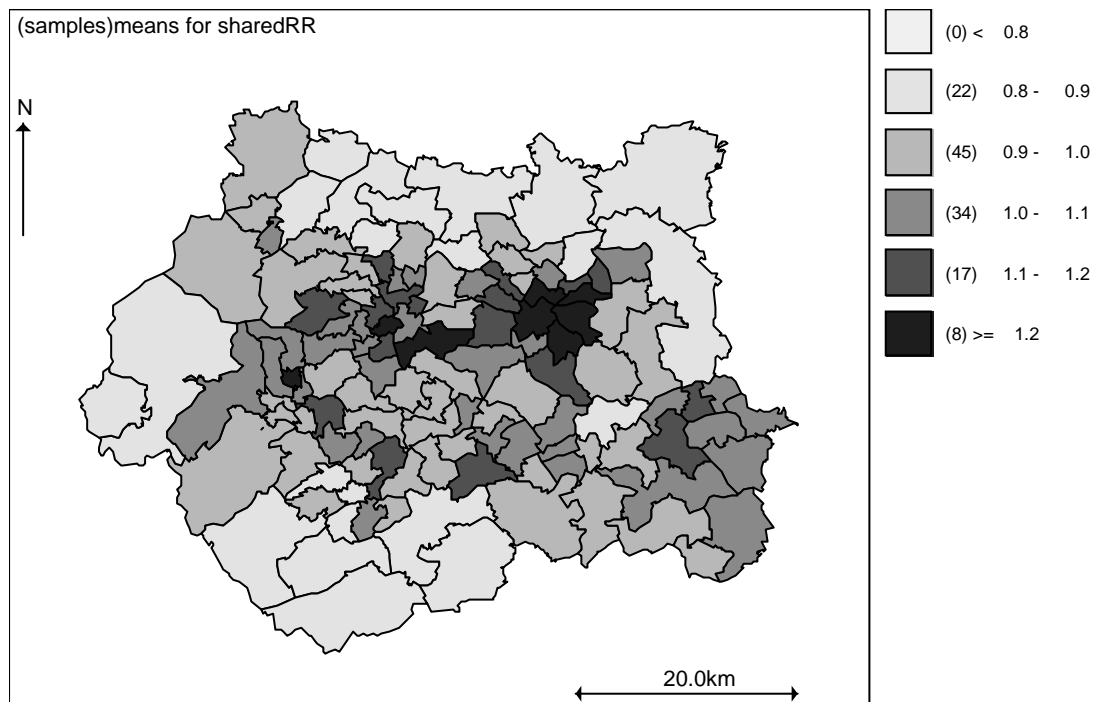
A 10000 iteration burn-in followed by a further 50000 updates gave the following results

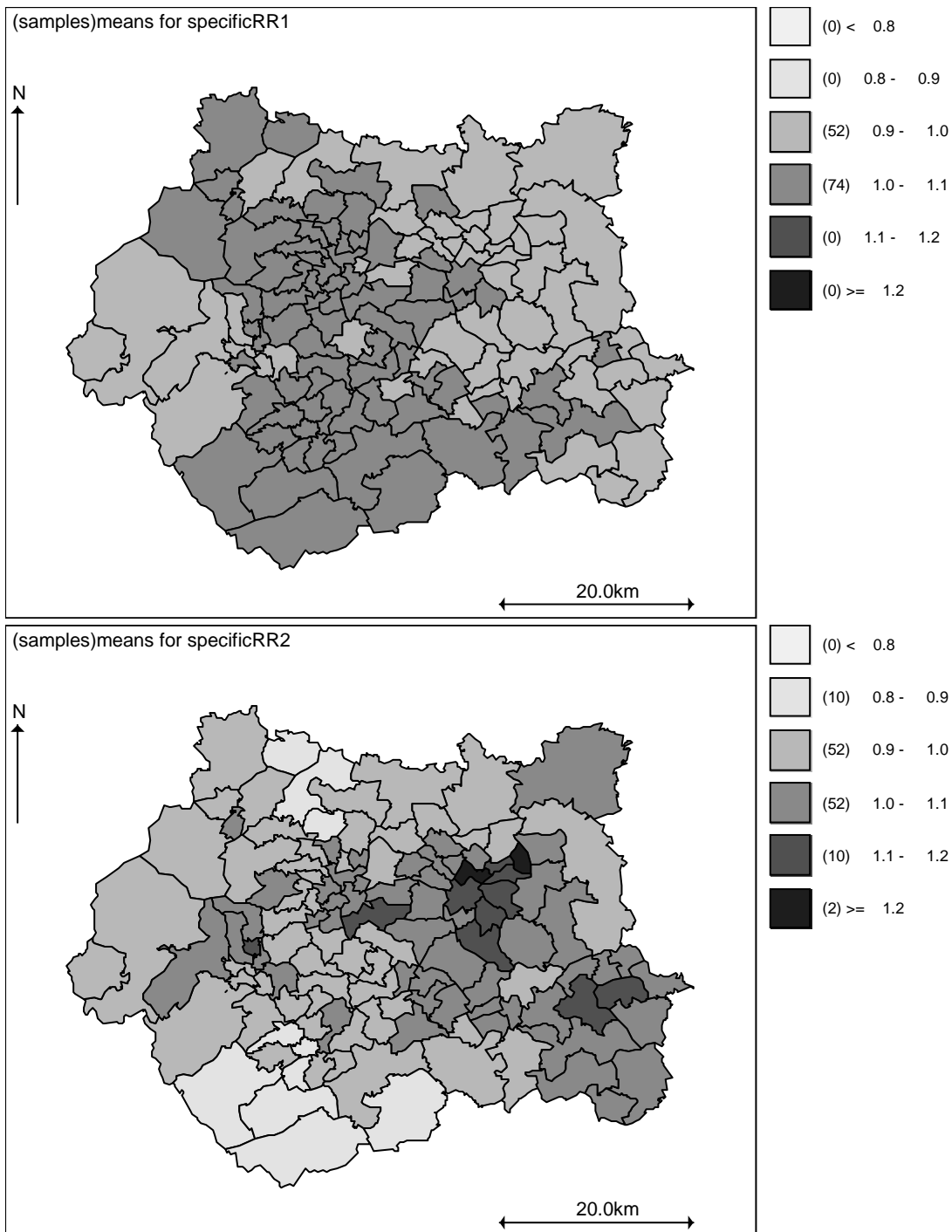
node	mean	sd	MC error	2.5%	median	97.5%	start	sample
RR.ratio	0.9131	0.5656	0.02552	0.2884	0.7828	2.339	10001	50000

delta	0.921	0.2546	0.01243	0.537	0.8848	1.529	10001	50000
frac.shared[1]	0.7551	0.226	0.0116	0.1728	0.841	0.982	10001	50000
frac.shared[2]	0.6383	0.2793	0.01648	0.08775	0.7031	0.9824	10001	50000
var.shared[1]	0.01869	0.01284	5.301E-4	0.001787	0.01617	0.05054	10001	50000
var.shared[2]	0.02687	0.01224	6.946E-4	0.003633	0.02867	0.04581	10001	50000
var.specific[1]	0.006037	0.008325	3.898E-4	4.213E-4	0.002752	0.03051	10001	50000
var.specific[2]	0.01527	0.01213	7.013E-4	7.43E-4	0.01218	0.04023	10001	50000

These indicate that for oral cancer, about 75% of the total between-area variation in risk is captured by the shared component, while for lung cancer about 64% of the total between-area variation in risk is captured by the shared component, although the 95% CI for frac.shared are very wide. RR.ratio is slightly less than 1 (although again with a wide credible interval), indicating that the shared component has a slightly weaker association with risk of oral cancer (disease 1) than with risk of lung cancer (disease 2).

Maps showing the spatial pattern of the shared component and the disease-specific residual components are shown below. The map file for this study region is called **WestYorkshire**, and is available in the maps directory of GeoBUGS 1.2.





9. Random walk priors for temporal smoothing of daily air pollution estimates [\[top\]](#)

Shaddick and Wakefield (2002) consider spatiotemporal modelling of daily ambient air pollution at a number of monitoring sites in London. Here we take a subset of their data on a single pollutant measured at one site for 366 days, and model temporal autocorrelation using a random walk prior.

Conditional on the underlying mean concentration μ_t on day t , the likelihood for the observed pollution concentration Y_t is assumed to be independent Normal i.e.

$$Y_t \sim \text{Normal}(\mu_t, \tau_{\text{err}}) \quad \text{where } 1/\tau_{\text{err}} \text{ is the measurement error variance}$$

$$\mu_t = \beta + \theta_t$$

where β is the overall mean pollution concentration at the site, and θ_t is a (zero mean) random error term representing daily fluctuations about this mean. To reflect the prior belief that these daily fluctuations are correlated, a random walk prior is assumed for $\theta = \{\theta_1, \dots, \theta_{366}\}$ (see equation 7 in Shaddick and Wakefield):

$$\begin{aligned} \theta_t \mid \theta_{-t} &\sim \text{Normal}(\theta_{t+1}, \phi) && \text{for } t = 1 \\ &\sim \text{Normal}((\theta_{t-1} + \theta_{t+1})/2, \phi/2) && \text{for } t = 2, \dots, T-1 \\ &\sim \text{Normal}(\theta_{t-1}, \phi) && \text{for } t = T \end{aligned}$$

where θ_{-t} denotes all elements of θ except the θ_t . This prior may be specified in WinBUGS 1.4 using the **car.normal** distribution, with adjacency vector **adj[]** listing neighbouring time points (i.e. (t-1) and (t+1) are neighbours of time point t), corresponding weight vector **weight[]** set to a sequence of 1's, and vector giving the number of neighbours, **num[]**, set to 2 for all time points except **num[1]** and **num[T]** which are set to 1.

The RW(1) reflects prior beliefs about smoothness of first differences, i.e. sudden jumps between consecutive values of θ are unlikely. Alternatively, we may assume a second order random walk prior RW(2) for θ , which represents prior beliefs that the rate of change (gradient) of θ is smooth:

$$\begin{aligned} \theta_t \mid \theta_{-t} &\sim \text{Normal}(2\theta_{t+1} - \theta_{t+2}, \phi) && \text{for } t = 1 \\ &\sim \text{Normal}((2\theta_{t-1} + 4\theta_{t+1} - \theta_{t+2})/5, \phi/5) && \text{for } t = 2 \\ &\sim \text{Normal}((-\theta_{t-2} + 4\theta_{t-1} + 4\theta_{t+1} - \theta_{t+2})/6, \phi/6) && \text{for } t = 3, \dots, T-2 \\ &\sim \text{Normal}((-\theta_{t-2} + 4\theta_{t-1} + 2\theta_{t+1})/5, \phi/5) && \text{for } t = T-1 \\ &\sim \text{Normal}(-\theta_{t-2} + 2\theta_{t-1}, \phi) && \text{for } t = T \end{aligned}$$

Again this may be specified using the **car.normal** distribution in WinBUGS 1.4 via appropriate specification of the **adj[]**, **weight[]** and **num[]** vectors.

The model code for fitting these two models is given below.

Model

```
model {
# likelihood
for(t in 1:T) {
  y[t] ~ dnorm(mu[t], tau.err)
  mu[t] <- beta + theta[t]
}

# prior for temporal effects
# RW prior for theta[t] - specified using car.normal with neighbours (t-1) and (t+1)
# for theta[2],...,theta[T-1], and neighbours (t+1) for theta[1] and (t-1) for theta[T]

theta[1:T] ~ car.normal(adj[], weights[], num[], tau)
beta ~ dflat()

# Specify weight matrix and adjacency matrix corresponding to RW(1) prior
# (Note - this could be given in the data file instead)

for(t in 1:1) {
  weights[t] <- 1;          adj[t] <- t+1;          num[t] <- 1
}
for(t in 2:(T-1)) {
  weights[2+(t-2)*2] <- 1;  adj[2+(t-2)*2] <- t-1
  weights[3+(t-2)*2] <- 1;  adj[3+(t-2)*2] <- t+1;          num[t] <- 2
}
```

```

}
for(t in T:T) {
  weights[(T-2)*2 + 2] <- -1;      adj[(T-2)*2 + 2] <- t-1;      num[t] <- 1
}

# Alternatively, a weight matrix and adjacency matrix corresponding to RW(2) prior can be specified
# or given in the data file (note, no need to change the prior distribution on theta,
# just the weights/adjacencies)
# for(t in 1:1) {
#   weights[t] <- 2;      adj[t] <- t+1
#   weights[t+1] <- -1;   adj[t+1] <- t+2;      num[t] <- 2
# }
# for(t in 2:2) {
#   weights[t+1] <- 2;   adj[t+1] <- t-1
#   weights[t+2] <- 4;   adj[t+2] <- t+1
#   weights[t+3] <- -1;  adj[t+3] <- t+2;      num[t] <- 3
# }
# for(t in 3:(T-2)) {
#   weights[6+(t-3)*4] <- -1;   adj[6+(t-3)*4] <- t-2
#   weights[7+(t-3)*4] <- 4;   adj[7+(t-3)*4] <- t-1
#   weights[8+(t-3)*4] <- 4;   adj[8+(t-3)*4] <- t+1
#   weights[9+(t-3)*4] <- -1;  adj[9+(t-3)*4] <- t+2;      num[t] <- 4
# }
# for(t in (T-1):(T-1)) {
#   weights[(T-4)*4 + 6] <- 2;   adj[(T-4)*4 + 6] <- t+1
#   weights[(T-4)*4 + 7] <- 4;   adj[(T-4)*4 + 7] <- t-1
#   weights[(T-4)*4 + 8] <- -1;  adj[(T-4)*4 + 8] <- t-2;      num[t] <- 3
# }
# for(t in T:T) {
#   weights[(T-4)*4 + 9] <- 2;   adj[(T-4)*4 + 9] <- t-1
#   weights[(T-4)*4 + 10] <- -1;  adj[(T-4)*4 + 10] <- t-2;      num[t] <- 2
# }

# other priors
tau.err ~ dgamma(0.01, 0.01) # measurement error precision
sigma.err <- 1 / sqrt(tau.err)
sigma2.err <- 1/tau.err

tau ~ dgamma(0.01, 0.01) # random walk precision
sigma <- 1 / sqrt(tau)
sigma2 <- 1/tau

for(t in 1:T) { day[t] <- t } # include this variable to use in time series (model fit) plot
}

```

Data → Click on one of the arrows for data. Note that pollution concentrations were not measured every day. However it is necessary to include days with no measurements as missing values (NA) in the data set, otherwise the temporal neighbourhood structure cannot be specified correctly. ←

Initial values → Click on one of the arrows for initial values ←

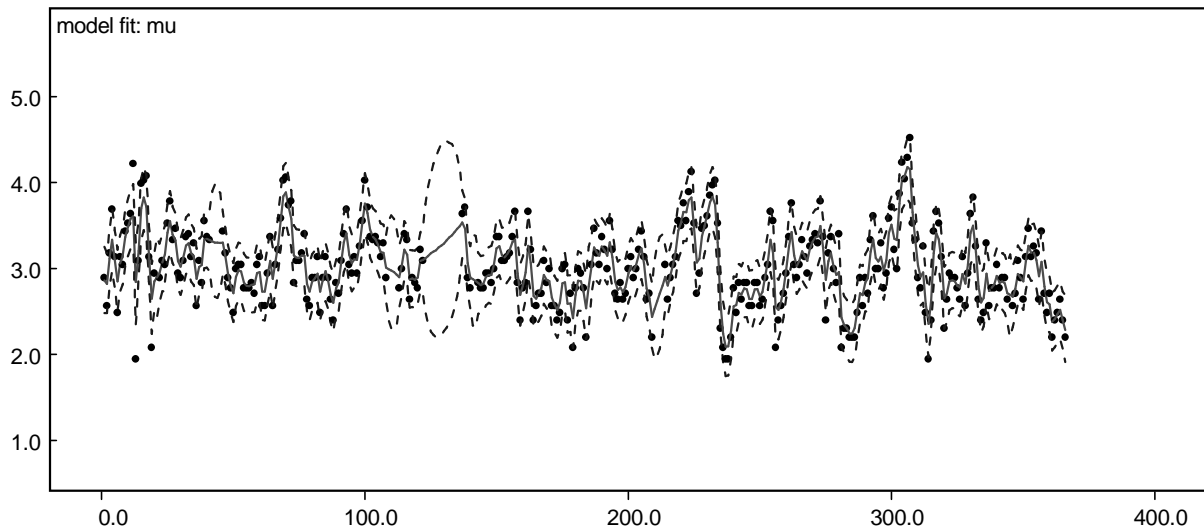
Results

RW(1) prior:

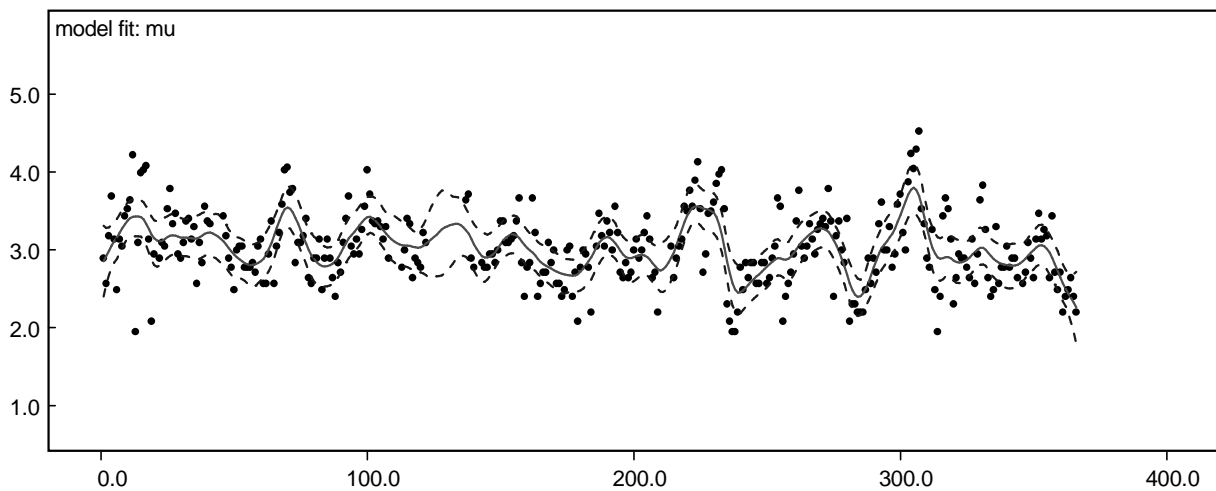
node	mean	sd	MC error	2.5%	median	97.5%	start	sample
beta	3.029	0.0201	9.439E-4	2.99	3.029	3.069	1001	10000
mu[1]	2.867	0.1959	0.00221	2.483	2.863	3.254	1001	10000
mu[2]	2.824	0.1779	0.004028	2.481	2.82	3.179	1001	10000

mu[3]	3.115	0.1703	0.002417	2.783	3.115	3.443	1001	10000
mu[4]	3.334	0.1852	0.005685	2.968	3.336	3.694	1001	10000
mu[5]	3.111	0.1689	0.002165	2.783	3.11	3.446	1001	10000
.....								
.....								
mu[360]	2.584	0.1673	0.002305	2.258	2.583	2.916	1001	10000
mu[361]	2.383	0.1716	0.003417	2.044	2.381	2.72	1001	10000
mu[362]	2.414	0.1678	0.002228	2.084	2.411	2.746	1001	10000
mu[363]	2.47	0.1679	0.00229	2.147	2.47	2.802	1001	10000
mu[364]	2.516	0.1712	0.002963	2.176	2.52	2.847	1001	10000
mu[365]	2.399	0.1706	0.002618	2.063	2.398	2.729	1001	10000
mu[366]	2.288	0.196	0.003037	1.91	2.284	2.684	1001	10000
sigma	0.2709	0.03888	0.002482	0.2006	0.27	0.3485	1001	10000
sigma.err	0.2433	0.03411	0.002153	0.1717	0.2449	0.3063	1001	10000

Plot of posterior median (red line) and posterior 95% intervals (dashed blue lines) for $\mu[t]$ (the true mean daily pollutant concentration), with observed concentrations shown as black dots. (This plot was produced by selecting the *model fit* option from the *Compare* menu (available from the *Inference* menu), with *mu* specified as the *node*, *day* as the *axis* and *y* as *other*). Note that the dashed blue line shows the posterior 95% interval for the estimated mean daily concentration, and is not a predictive interval - hence we would not necessarily expect all of the observed data points to lie within the interval.



Equivalent plot assuming an RW(2) prior. Note the greater amount of smoothing imposed by this prior:



Appendix 1: Technical details of Structured Multivariate Gaussian and Conditional Autoregressive (CAR) models and hyperprior specification. [\[top\]](#)

(Note: for clarity of exposition, we parameterise the Normal distribution in terms of the mean and *variance* in the following discussion. However, *WinBUGS* parameterisation of the CAR and structured MVN models is in terms of the mean and precision as usual.)

Assume we have a set of area-specific spatially correlated Gaussian data or random effects S_i , $i=1, \dots, N$ (where N is the number of areas in the study region). Suppose their joint distribution may be expressed as follows:

$$\mathbf{S} \sim \text{MVN}(\boldsymbol{\mu}, \nu\boldsymbol{\Sigma})$$

where $\mathbf{S} = (S_1, \dots, S_N)$, MVN denotes the N -dimensional multivariate normal distribution, $\boldsymbol{\mu}$ is the $1 \times N$ mean vector, $\nu > 0$ controls the overall variability of the S_i and $\boldsymbol{\Sigma}$ is an $N \times N$ positive definite matrix.

Joint specification [\[top\]](#)

We may assume a parametric form for the elements of the between-area correlation matrix:

$$\Sigma_{ij} = f(d_{ij}; \theta)$$

where d_{ij} = distance between areas i and j . *WinBUGS* 1.4 allows two options for the function $f(\cdot)$ (see Richardson (1992) for further details of these functions):

1. Powered exponential family

$$f(d_{ij}; \phi, \kappa) = \exp[-(\phi d_{ij})^\kappa] \quad \text{where } \phi > 0, \quad \kappa \text{ in } (0, 2]$$

The parameter ϕ controls the rate of decline of correlation with distance:

ϕ large \rightarrow rapid decay
 ϕ small \rightarrow slow decay

One possible strategy for specifying a prior for ϕ is to choose a uniform distribution between ϕ_{\min} and ϕ_{\max} where ϕ_{\min} and ϕ_{\max} are chosen to give a sensible range of values for correlations both at a distance equal to the *maximum* distance between any pair of areas in the study region, and at a distance equal to the *minimum* distance between any pair of areas in the study. For example, if the minimum distance is 1 km (say), and the maximum distance is 20 km (say), then values of $\phi_{\min} = 0.04$ and $\phi_{\max} = 5$ would give a diffuse but plausible prior range of correlations (assuming $\kappa = 1$) between 0.007 and 0.96 at a distance of 1 km, and between 0 and 0.45 at a distance of 20 km. *Note that it is advisable to choose a value of ϕ_{\min} that prevents the correlation at the maximum distance between observations in the study region from being too high, since this can lead to identifiability problems between the overall mean, $\boldsymbol{\mu}$, of the spatial random variables and the correlation parameter, ϕ .*

The parameter κ controls the amount by which spatial variations in the data are smoothed. Large values of κ lead to greater smoothing, with $\kappa = 2$ corresponding to the Gaussian correlation function (although the resulting covariance matrix is nearly singular). It is often difficult to learn much about this parameter, so unless there is a good reason for believing otherwise, it is usually advisable to set $\kappa = 1$ a priori.

2. Disc model

$$f(d_{ij}; \alpha) = \begin{cases} 2 / \pi * \{ \text{COS}^{-1} (d_{ij} / \alpha) - [(d_{ij} / \alpha)(1 - (d_{ij}^2 / \alpha^2))]^{1/2} \} & \text{for } d_{ij} < \alpha \\ = 0 & \text{for } d_{ij} > \alpha \end{cases}$$

The parameter α controls the rate of decline of correlation with distance:

α large \rightarrow slow decay
 α small \rightarrow rapid decay

The disc function leads to an approximately linear decrease in correlation with increasing distance, with correlation declining to zero at a distance equal to α . *Again, it is advisable to choose a prior for α that prevents the correlation at the maximum distance between observations in the study region from being too high, since this can lead to identifiability problems between the overall mean, μ , of the spatial random variables and the correlation parameter, α . An upper prior bound on α equal to a small multiple of the maximum distance in the study region is therefore a sensible 'default' choice. The minimum value allowed a priori for α should also be chosen carefully since values of α less than the minimum distance between observations in the study region will not be identifiable.*

Exploratory analysis using variograms maybe be helpful to help chose an appropriate specification for the correlation function and associated parameters.

Conditional specification [\[top\]](#)

By writing the between-area covariance matrix in the following form:

$$v\Sigma = v(I - \gamma C)^{-1} M$$

where $I = N \times N$ identity matrix

$M = N \times N$ diagonal matrix, with elements M_{ii} proportional to the conditional variance of $S_i | S_j$

$C = N \times N$ weight matrix, with elements C_{ij} reflecting spatial association between areas i and j

$\gamma =$ controls overall strength of spatial dependence

and using standard multivariate normal theory (e.g. Johnson and Kotz, 1972; Besag and Kooperberg, 1995), the joint multivariate Gaussian model can be expressed in the form of a set of conditional distributions

$$S_i | \mathbf{S}_{-i} \sim \text{Normal} \left(\mu_i + \sum_j \gamma C_{ij} (s_j - \mu_j), \phi M_{ii} \right),$$

(here \sum_j denotes summation over $j=1$ to N , not covariance matrix, and \mathbf{S}_{-i} denotes all the elements of \mathbf{S} except S_i)

From a modelling perspective, use of the joint formulation requires specification of the elements of the covariance matrix Σ (see above), while use of the conditional formulation reduces to specification of the matrices C and M and the spatial dependence parameter γ . Various constraints are needed on the values of C , M and γ in order to ensure that Σ is symmetric positive definite:

- * Σ is only symmetric if $C_{ij} M_{jj} = C_{ji} M_{ii}$
- * $\text{Var}(S_i | S_j) = vM_{ii} > 0$ so M_{ii} must be > 0
- * To ensure S is positive definite, γ must lie between γ_{\min} and γ_{\max} where γ_{\min}^{-1} and γ_{\max}^{-1} are the smallest and largest eigenvalues of $M^{-1/2} C M^{1/2}$
- * In practice, we often expect positive spatial dependence, so constrain prior for γ to be between 0 and γ_{\max}
- * $\gamma = 0$ implies no spatial dependence

Intrinsic CAR model [\[top\]](#)

Besag, York and Mollie (1991) propose an intrinsic version of this CAR model in which the covariance matrix Σ is not positive definite. Their model corresponds to choosing $C_{ij} = 1/n_i$ if areas i and j are adjacent and $C_{ij} = 0$ otherwise (with C_{ii} also set to 0), $M_{ii} = 1/n_i$, and setting $\gamma = \gamma_{\max}$ which turns out to always be 1 with this particular choice of C_{ij} and M_{ii} . Here, n_i is the number of areas which are adjacent to area i . Comparison with the equations above shows that this leads to the following model for the conditional distribution of $S_i | \mathbf{S}_{-i}$:

$$S_i | \mathbf{S}_{-i} \sim \text{Normal} (\bar{S}_i, v/n_i)$$

where $S_{\cdot i} = \sum_{j \in \delta_i} S_j / n_i$ and δ_i denotes the set of labels of the "neighbours" of area i . Hence S_i has a normal distribution with conditional mean given by the average of the neighbouring S_j 's and conditional variance inversely proportional to the number of neighbours n_i . Note that an equivalent specification is take *unnormalised weights* $W_{ij} = 1$ if areas i and j are adjacent and $W_{ij} = 0$ otherwise, and set $C_{ij} = W_{ij} / W_{i+}$ where $W_{i+} = \sum_j W_{ij}$. The **car.normal** and (**car.11**) distributions in *GeoBUGS* requires the user to specify *unnormalised weights*.

Since the CAR model defined above is improper (the overall mean of the S_i is not defined), it can only be used as a prior distribution for spatially distributed random effects, and not as a likelihood for data. It is often convenient to assume that such random effects have zero mean. Besag and Kooperberg (1995) show that constraining the random effects to sum to zero and specifying a separate intercept term with a location invariant Uniform(-inf, +inf) prior is equivalent to the unconstrained parameterisation with no separate intercept. WinBUGS 1.4 includes a distribution called **dflat()** which corresponds to an improper (flat) prior on the whole real line. This prior must always be used for the intercept term in a model including CAR random effects.

We must also specify prior distributions for the overall variance parameter v . As usual in WinBUGS, the **car.normal** (and **car.11**) distributions are parameterised in terms of the precision $\tau = 1/v$. Care is needed when choosing a prior for τ since the posterior variance of the random effects can be quite sensitive. One option is a gamma distribution with shape and inverse scale parameters both equal to 0.01. This has a mean of $0.01/0.01 = 1$ and a large variance of $0.01/(0.01)^2 = 100$; however, this tends to place most of the prior mass away from zero (on the scale of the random effects standard deviation), and so in situations when the true spatial dependence between areas is negligible (i.e. standard deviation close to zero) this may induce artefactual spatial structure in the posterior. Kelsall and Wakefield (1999) suggest an alternative gamma(0.5, 0.0005) prior for the precision parameter of the spatial random effects in a CAR model. This expresses the prior belief that the random effects standard deviation is centered around 0.05 with a 1% prior probability of being smaller than 0.01 or larger than 2.5.

Proper CAR model [\[top\]](#)

If γ is constrained to lie in the interval $(\gamma_{\min}, \gamma_{\max})$ where γ_{\min}^{-1} and γ_{\max}^{-1} are the smallest and largest eigenvalues of $\mathbf{M}^{-1/2} \mathbf{C} \mathbf{M}^{1/2}$ as defined above, rather than being fixed to its maximum as in the intrinsic CAR, then the resulting distribution is proper provided the constraints on \mathbf{C} and \mathbf{M} are still satisfied. The choice $C_{ij} = 1/n_i$ if areas i and j are adjacent and $C_{ij} = 0$ otherwise (with C_{ii} also set to 0) and $M_{ii} = 1/n_i$, where n_i is the number of areas which are adjacent to area i is still valid for the proper CAR. In the context of disease mapping Cressie and Chan (1989) and Stern and Cressie (1999) choose an alternative parameterisation:

$$M_{ii} = 1/E_i \quad (\text{the inverse of the expected count or population size in area } i)$$

$$C_{ij} = (E_j / E_i)^{1/2} \text{ for neighbouring areas } i, j \text{ and } 0 \text{ otherwise}$$

Multivariate intrinsic CAR model [\[top\]](#)

Assume we have a multivariate p -dimensional vector of spatially correlated Gaussian data or random effects in each area, $\mathbf{S}_i = (S_{1i}, S_{2i}, \dots, S_{pi})'$, $i=1, \dots, N$. The CAR models discussed above extend naturally to a multivariate setting by replacing the univariate Gaussian conditional distribution for $S_i | S_j$ with a multivariate conditional distribution. In the case of the intrinsic CAR prior with 0-1 adjacency weights as used by Besag, York and Mollie (1991), and taking the case of $p=2$ for simplicity, this gives:

$$\mathbf{S}_i | (\mathbf{S}_{1(-i)}, \mathbf{S}_{2(-i)}) \sim \text{Bivariate Normal} (\mathbf{S}_{\cdot i}, \mathbf{V} / n_i),$$

(here $(\mathbf{S}_{1(-i)}, \mathbf{S}_{2(-i)})$ denotes the elements of the $2 \times N$ matrix \mathbf{S} excluding the i th area (column))

where $\mathbf{S}_{\cdot i} = (S_{\cdot i1}, S_{\cdot i2})$ with $S_{\cdot ip} = \sum_{j \in \delta_i} S_{jp} / n_i$ and, as in the univariate case, δ_i and n_i denote the set of labels of the "neighbours" of area i and the number of neighbours, respectively. \mathbf{V} is a 2×2 covariance matrix with diagonal elements v_{11} and v_{22} representing the conditional variances of S_1 and S_2

respectively, and off-diagonal element v_{12} representing the (conditional) within-area covariance between S_1 and S_2 . In general \mathbf{S}_i has a p-dimensional multivariate normal distribution with pth element of the conditional mean vector given by the average of the neighbouring S_{jp} 's and conditional covariance matrix inversely proportional to the number of neighbours n_i , with pth diagonal element representing the conditional variance of the pth component of \mathbf{S} and off-diagonal elements representing the conditional covariances between each pair of the p elements of \mathbf{S} .

Gelfand and Vounatsou (2003) discuss a multivariate generalisation of the proper version of CAR distribution, but only the improper intrinsic multivariate CAR is currently implemented in WinBUGS.

Appendix 2: Technical details of the Poisson-gamma Spatial Moving Average convolution model . [\[top\]](#)

Spatial moving average models are a flexible class of models that have been used to describe continuous spatial processes, particularly in geostatistical applications. Such models are constructed by integrating a simple two-dimensional random noise process (for example, a grid of iid Gaussian random variables) with a smoothing kernel that is a function of distance and, possibly, location. The kernel can be thought of as a device to 'smear out' the random noise process in two-dimensional space to give a smooth surface.

Spatial moving average models have been developed primarily for continuous spatial processes and are currently not implementable in WinBUGS 1.4. However, Ickstadt and Wolpert (1998) and Best et al (2000b) proposed a discrete version of a gamma moving average process, for use in identity-link Poisson regression models. Suppose we have a set of area-specific spatially correlated Poisson count data (or random effects) S_i , $i=1, \dots, N$ (where N is the number of areas in the study region). Ickstadt and Wolpert (1998) and Best et al (2000b) model spatial dependence at the level of the Poisson mean, and assume that the counts S_i are conditionally independent given this mean:

$$S_i \sim \text{Poisson}(\lambda_i)$$

The model for each λ_i is constructed by specifying an arbitrary grid of latent iid gamma random variables γ_j ($j=1, \dots, J$ where J is the total number of grid cells defining the latent process) covering the study region. These are then convolved with a kernel matrix whose elements, k_{ij} , represent the relative contribution of the latent variable in grid cell j to the Poisson mean in area i :

$$\lambda_i = \sum_j \gamma_j * k_{ij}$$

Best et al (2000b) assume an isotropic, stationary Gaussian kernel function (although other kernel forms are easily accommodated, such as an adjacency-based kernel - see [example 5](#)):

$$k_{ij} = \tau / (2\pi\rho^2) \exp(-d_{ij}^2 / 2\rho^2)$$

where τ can be thought of as a scale factor for the λ 's, d_{ij} is the distance between the centroid of area i and the centroid of latent grid cell j , and ρ is the spatial range parameter governing how rapidly the influence of the latent gamma random variables on the area-specific Poisson means declines with distance.

One interpretation of Poisson-gamma moving average model is to view the gamma random variables as representing the location and magnitude of unmeasured risk factors, and the area-specific Poisson means λ_i as representing the cumulative effect of these risk factors in each area, weighted by their distance from the area according to the kernel 'weights' k_{ij} .

The model may be extended to include observed covariates and an offset adjustment (e.g. to account for different populations in different areas). Observed covariates are included as additive terms in the linear predictor λ_i as in a standard identity link Poisson regression:

$$\lambda_i = \sum_k \beta_k X_{ik} + \sum_j \gamma_j * k_{ij}$$

where X_{ik} is the value of the k th observed covariate in area i . Note that due to the identity link, all covariates and regression coefficients must take positive values, yielding an interpretation as additive excess risk factors. The latent terms $\theta_i = \sum_j \gamma_j * k_{ij}$ may then be thought of as a spatial (non-negative) random effect for area i . Adjustment for an offset - say the population living in each area, N_i - is easily achieved by setting

$$\lambda_i = \rho_i N_i$$

$$\rho_i = \sum_k \beta_k X_{ik} + \sum_j \gamma_j * k_{ij}$$

Note that Best et al (2004) discuss how to standardise the offset N_i for different age and sex distributions within each area in an epidemiological application of these models.

Estimation of the **pois.conv** distribution is carried out in WinBUGS 1.4 using a data augmentation scheme to exploit the Poisson-gamma conjugacy of the full conditionals for the γ_j parameters (see Ickstadt and Wolpert (1998) and Best et al (2000b) for details). It is also a good idea to assume gamma prior distributions for any other uncertain parameters in the linear predictor λ_i (e.g. the β_k parameters above) to exploit this conjugacy, although it is possible to assume any suitable prior distribution with positive support for these other parameters. Hyperprior specification is discussed in detail by Ickstadt and Wolpert (1998) and Best et al (2000b). Note however that the prior shape (a) and precision (b) parameters of the latent gamma variables should be chosen such that γ_j has prior mean proportional to the area of the j th latent grid cell. This makes the model spatially extensible in the sense that any partition of the latent gamma random variables will lead to identical probability distributions for the kernel-weighted sums $\sum_j \gamma_j * k_{ij}$.

References [\[top\]](#)

- Besag, J., York, J. and Mollie, A. (1991). Bayesian image restoration, with two applications in spatial statistics. *Annals of the Institute of Statistical Mathematics*, **43**, 1--59, (With discussion).
- Besag, J. and Kooperberg, C.L. (1995). On conditional and intrinsic autoregressions. *Biometrika*, **82**, 733--746.
- Best, N.G., Richardson, S. and Thomson, A. (2004). A comparison of Bayesian spatial models for disease mapping. *Statistical Methods in Medical Research* (to appear).
- Best, N.G., Ickstadt, K. and Wolpert, R.L. (2000a). Spatial Poisson regression for health and exposure data measured at disparate resolutions. *Journal of the American Statistical Association*, **95**, 1076-1088.
- Best, N.G., Ickstadt, K., Wolpert, R.L. and Briggs, D.J. (2000b). Combining models of health and exposure data: the SAVIAH study. In *Spatial Epidemiology: Methods and Applications*. P. Elliott, J.C. Wakefield, N.G. Best and D.J. Briggs (eds), Oxford: Oxford University Press, p. 393-414.
- Clayton, D.G and Kaldor, J. (1987). Empirical Bayes estimates of age-standardized relative risks for use in disease mapping. *Biometrics*, **43**, 671--681.
- Cressie, N.A. and Chan, N.H. (1989). Spatial modeling of regional variables. *Journal of the American Statistical Association*, **84**, 393--401.
- Diggle, P.J., Tawn, J.A. and Moyeed, R.A. (1998). Model-based geostatistics. *Applied Statistics*, **47**, 299-350.
- Fahrmeir, L. and Lang, S. (2001). Bayesian inference for generalized additive mixed models based on Markov random field priors. *Applied Statistics*, **50**, 201-220.
- Gelfand, A. and Vounatsou, P. (2003). Proper multivariate conditional autoregressive models for spatial data analysis. *Biostatistics*, **4**, 11-25.

- Ickstadt, K. and Wolpert, R.L. (1998). Multiresolution assessment of forest inhomogeneity. In *Case Studies in Bayesian Statistics, Volume 3*. Lecture Notes in Statistics, **121**. C. Gatsonis, J.S. Hodges, R.E. Kass, R. McCulloch, P. Rossi and N.D. Singpurwalla (eds), New York: Springer-Verlag, p. 371-386.
- Johnson, N.L. and Kotz, S. (1972). *Distributions in Statistics: continuous multivariate*. Wiley: New York.
- Kelsall, J.E. and Wakefield, J.C. (1999). Discussion of "Bayesian models for spatially correlated disease and exposure data", by Best *et al*. In *Bayesian Statistics 6*. J.M. Bernardo, J.O. Berger, A.P. Dawid and A.F.M. Smith (eds), Oxford: Oxford University Press, p. 151.
- Knorr-Held L. and Best N.G. (2001). A shared component model for joint and selective clustering of two diseases. *Journal of the Royal Statistical Society, Series A*.
- Mollie, A. (1996). Bayesian mapping of disease. In *Markov Chain Monte Carlo in Practice*. W.R. Gilks, S. Richardson and D.J. Spiegelhalter (eds.), New York: Chapman & Hall, pp. 359--379.
- Richardson, S. (1992). Statistical methods for geographical correlation studies. In *Geographical and Environmental Epidemiology*. P. Elliott, J. Cuzick, D. English and R. Stern (eds.), Oxford: Oxford University Press, p. 181-204.
- Shaddick, G. and Wakefield, J. (2002). Modelling daily multivariate pollutant data at multiple sites. *Applied Statistics*, **51**, 351-372.
- Stern, H.S. and Cressie, N.A. (1999). Inference for extremes in disease mapping. In *Disease mapping and risk assessment for public health*. A. Lawson, A. Biggeri, D. Bohning, E. Lesaffre, J-F. Viel and R. Bertollini (eds.), Chichester: Wiley, p. 63--84.
- Wakefield, J.C., Best, N.G., and Waller, L.A. (2000). Bayesian Approaches to Disease Mapping. In *Spatial Epidemiology: Methods and Applications*. P. Elliott, J.C. Wakefield, N.G. Best and D.J. Briggs (eds), Oxford: Oxford University Press, p. 104-127.
- Wolpert, R.L. and Ickstadt, K. (1998). Poisson/Gamma random field models for spatial statistics. *Biometrika*, **85**, 251-267.