Simultaneous Inference for FMRI Data

B. Douglas Ward
Biophysics Research Institute
Medical College of Wisconsin
email: ward@mcw.edu

June 19, 2000

1 Program AlphaSim

1.1 Purpose

Program AlphaSim provides a means of estimating the overall significance level (the probability of a false detection) for an entire *AFNI* 3D functional image. This is accomplished by Monte Carlo simulation of the process of image generation, spatial correlation of voxels, voxel intensity thresholding, masking, and cluster identification. Based upon the combination of individual voxel probability thresholding and minimum cluster size thresholding, the probability of a false positive detection per image is determined from the frequency count of cluster sizes.

Program AlphaSim can also estimate the statistical power (the probability of a true detection) for a user specified region of true activation. The probability of a true positive detection per image is similarly determined from Monte Carlo simulation.

A word of caution: The applicability of simulation results is limited by the validity of the assumptions involved. In particular, the modelling of spatial correlation of voxels assumes that the underlying population of voxel intensities has a normal distribution. The sensitivity of the results to departures from normality is not well known at this time.

1.2 Theory

In order to achieve a specified level of statistical significance, allowance must be made when multiple statistical comparisons are required. This is typically accomplished by using the Bonferroni method: when making N simultaneous inferences, an overall significance level of α is achieved by using individual significance levels of $\frac{\alpha}{N}$. For a typical *AFNI* 3D functional image, with (say) $N = 16 \times 64 \times 64 = 65536$ voxels, the Bonferroni correction is quite severe, causing loss of power to detect areas of true activation.

Several recent papers (see [1], [2]) have addressed this problem. Instead of using the individual voxel probability threshold alone in achieving the desired overall significance level, these papers use probability thresholding in combination with minimum cluster size thresholding. The underlying principle is that true regions of activation will tend to occur over contiguous voxels, whereas noise has much less of a tendency to form clusters of activated voxels. Therefore, the presence of clustering can be used as one criterion to distinguish between signal and noise. By using a combination of probability thresholding and cluster thresholding, the power of the statistical test is greatly enhanced.

Although tables of false detection probability vs. cluster size have been published, these tables have several limitations. First, the tables are for 2D images, and so cannot be applied directly to 3D images. Second, even if tables were published for 3D images, it is doubtful that such tables could cover every possible combination of experimental parameters. Therefore, the approach used here is to provide the researcher with the software tools necessary to investigate the trade-off between probability threshold and cluster threshold to achieve the desired significance level, using those parameters which apply for the user's particular experimental conditions.

Schematic Representation of Program AlphaSim

$$
\begin{array}{rcc}
nx, ny, nz[, ax, ay, az, zsep] & \longrightarrow & \boxed{\text{Random Image Generator}} \\
 & & \downarrow \\
 & & \text{Uncorrelated-Voxel Image} \\
 & & \downarrow \\
FWHM_x, FWHM_y, FWHM_z & \longrightarrow & \boxed{\text{Gaussian Filter}} \\
 & & \downarrow \\
 & & \text{Correlated-Voxel Image} \\
 & & \downarrow \\
p_{thr} & \longrightarrow & \boxed{\text{Scaling and Thresholding}} \\
 & & \downarrow \\
 & & \text{Binary Image} \\
 & & \downarrow \\
mask & \longrightarrow & \boxed{\text{Apply Mask}} \\
 & & \downarrow \\
 & & \text{Masked Binary Image} \\
 & & \downarrow \\
dx, dy, dz, rmm & \longrightarrow & \boxed{\text{Cluster Identification}} \\
 & & \downarrow \\
 & & \text{Frequency Table}
\end{array}
$$

Program AlphaSim is a Monte Carlo simulation. By iteration of the process of random image generation, Gaussian filtering (to simulate voxel correlation), thresholding, image masking, and tabulation of cluster size frequencies, the program generates an estimate of the overall significance level achieved for various combinations of probability threshold and cluster size threshold. The program is represented by the above schematic. The different modules of the program are described below.

1.2.1 Random Image Generator

Program AlphaSim generates a series of random images, each having spatially uncorrelated voxels, by filling the specified volume ($nx \times ny \times nz$ voxels) with independent normal $n(0, 1)$ random numbers. Note that 2-dimensional images can be simulated by setting $nz = 1$.

For power calculations, the procedure is similar, except that voxels in the true activation region $ax \times ay \times az$ (which region is placed at the center of the image), are independent normal $n(zsep, 1)$ random numbers, where $zsep$ is the separation, in z-scores, between the mean of the noise distribution and the mean of the signal distribution.

1.2.2 Gaussian Filter

The effect of voxel correlation can be simulated by convolving the random image with a Gaussian function. Since convolution in the space domain is equivalent to multiplication in the frequency domain, the filtering is accomplished by the computationally efficient method of taking the 3D fast Fourier transform of the random image, multiplying this transform by the transform of the Gaussian function, and inverse fast Fourier transforming the result.

The width of the Gaussian function along each of the three principal axes is specified by the numbers $FWHM_x$, $FWHM_y$, and $FWHM_z$ ("FWHM" for "Filter Width Half Maximum").

Gaussian filtering is performed in exactly the same way for statistical power calculations as for statistical significance calculations.

Note that the Gaussian filtering function is essentially the same as that obtained using the -1blur command in program 3dmerge.

1.2.3 Scaling and Thresholding

The image, after Gaussian filtering, must be scaled to provide the specified individual voxel probability threshold p_{thr}. This is done by determining, from the sample mean and sample standard deviation, the value $z = z_{thr}$ such that approximately $p_{thr} \times nx \times ny \times nz$ of the voxels have intensity greater than z_{thr}.

For power calculations, the region of true activation is first extracted from the random image. The scaling is then accomplished in a similar fashion, except that z_{thr} is now determined by examining only those voxels which lie inside this true activation region. Note that the combination of user inputs p_{thr} and z_{sep} determine the nominal individual voxel power threshold $power_{thr}$. This follows since $z_{sep} = z_{1-\alpha} + z_{1-\beta}$, where $z_{1-\alpha}$ is the normal distribution upper tail $z-$value yielding probability p_{thr}, and $z_{1-\beta}$ is the normal distribution upper tail $z-$value yielding probability $power_{thr}$. Then, for the region of activation, the value $z = z_{thr}$ is chosen so that approximately $power_{thr} \times ax \times ay \times az$ of the voxels have intensity greater than z_{thr}. The sequence of calculations is as follows:

$$
\begin{aligned}
z_{1-a} &\longleftarrow p_{thr} \\
z_{1-\beta} &\longleftarrow z_{sep} - z_{1-\alpha} \\
power_{thr} &\longleftarrow z_{1-\beta} \\
z_{thr} &\longleftarrow power_{thr}
\end{aligned}
$$

Once z_{thr} has been set, then all voxels inside the entire volume (for statistical significance calculations) or inside the true activation region (for statistical power calculations) are compared against z_{thr}. The thresholding of the data is then accomplished by setting those voxels with intensity greater than z_{thr} to 1 (these are the"activated" voxels); voxels with intensity less than z_{thr} are set to 0. The result of the thresholding operation is a "binary" image, i.e., an image containing voxels having intensities of either "0" or "1".

1.2.4 Apply Mask

The numbers nx, ny, and nz specify the dimensions of the entire dataset. However, the user may be interested in only a subset of this dataset. For example, the user might choose to mask out voxels which lie outside the brain, prior to clustering. Or, the user might exclude all voxels except those in specific areas of interest. By reducing the number of voxels available for cluster formation, it should be possible to relax the cluster size threshold required to achieve a given level of statistical significance. Therefore, program AlphaSim allows the user to enter a mask dataset, in order to identify the subset of voxels which is available for clustering. Note: The mask option applies only for calculation of statistical significance, not for calculation of statistical power.

1.2.5 Cluster Identification

The next step is to identify which activated voxels belong to clusters. Here, "activated" refers to a voxel whose magnitude is 1. Cluster identification is performed by the same software used by programs 3dmerge and 3dclust. As in those programs, two voxels whose centers are less than (or equal to) rmm mm. apart are in the same cluster. Every activated voxel is a member of one, and only one, cluster. Once all clusters have been found, the size (in voxels) of each cluster is recorded in a frequency table.

For statistical power calculations, only clusters within the true activation region are counted.

1.3 Results

1.3.1 Significance Calculations

Program AlphaSim was used to calculate the probability of obtaining a false positive (in an entire 3D image) when the individual voxel probability threshold is used in combination with a cluster size threshold, assuming spatially uncorrelated voxels. For this example we are using a 3D image of size $16 \times 64 \times 64$ voxels, where each voxel has dimensions $7.00 \times 3.75 \times 3.75$ mm. The resulting plot of false detection probability (alpha) vs. cluster size threshold is shown in Figure 1 for three different voxel probability thresholds (p_{thr} = 0.001, 0.01, 0.02). Observe that the overall α level can be made arbitrarily small by increasing the cluster size threshold.

A similar analysis was performed for the case of spatially correlated voxels. Here we assume that the spatial correlation between voxels can be modeled by convolution with a Gaussian filter of width FWHM = 4.0 mm. The results are shown in Figure 2. Observe that the effect of spatial correlation of the voxels is to shift the curves to the right. This is as expected, since a tendency for noise to form clusters requires the use of larger cluster size thresholds to provide the same protection against false positives.

Each of the curves in Figures 1 and 2 represents the output from one 1000 iteration Monte Carlo simulation. By repeating the analysis for many different values of p_{thr}, one can determine, for each cluster size threshold, the value of p_{thr} which yields a specified significance level. This was done for a significance level of $\alpha = 0.05$ for three Gaussian filter widths $FWHM = 0$, 4, and 8 mm., and the results are plotted in Figure 3. Note that the vertical scale is the base 10 logarithm of p_{thr}. This plot indicates that, if one is willing to accept a minimum cluster size, it is possible to obtain an orders-of-magnitude improvement in p_{thr} over the value of p_{thr} required if cluster size thresholding is not used (in the plot, "no cluster size threshold" is equivalent to a cluster size threshold of 1).

1.3.2 Power Calculations

The statistical power is calculated based upon an assumed area of true activation. "Power" is here defined as the probability of a true detection, i.e., the probability that at least one cluster of voxels in the region of true activation is detected, i.e., has the specified size. It is also necessary to specify the z-score separation ($zsep$) between the mean of the noise distribution and the mean of the signal distribution. Throughout this section, we will take $zsep = 2.0$.

Figure 4 shows the power as a function of cluster size threshold for spatially uncorrelated voxels, holding alpha constant at $\alpha = 0.05$. Note that this implicitly defines the voxel probability threshold (see Figure 3). The three curves shown are for true activation regions of $2 \times 2 \times 2$, $3 \times 3 \times 3$, and $4 \times 4 \times 4$ voxels.

The statistical power calculations for spatially correlated voxels are similar. After the random volume is generated (having an "island" of true activation), the Gaussian filter is applied to the entire volume. The region of true activation is then extracted; this region is then "normalized", and clusters are then identified. The results for a Gaussian filter FWHM = 4.0 mm are presented in Figure 5.

1.3.3 Validation of Results

A number of assumptions and approximations are involved in calculating the statistical significance and statistical power for FMRI data by Monte Carlo simulation. Therefore, some means of verifying the accuracy of the results is required. Here, we will compare the AlphaSim results with some previously published results for 2-dimensional datasets. Since the same software is used in AlphaSim for both 2- and 3-dimensional datasets, it is reasonable to conclude that software which is valid for 2-dimensional data will be valid for 3-dimensional data as well.

As mentioned previously, the Forman paper does not report "alpha" values directly. Instead, the tables in that paper list the probability of a false detection per voxel. These values do not seem to be directly useful. However, for purposes of comparison, the AlphaSim output has been augmented to report these numbers (which we will refer to, somewhat misleadingly, as either "alpha/voxel" or "p/voxel"). Moreover, the "alpha" parameter used in the Forman paper is what we refer to here as the individual voxel probability threshold (p_{thr}).

The calculations for alpha/voxel were performed for a 128×128 pixel image. The results are presented in Figure 6 for the case of spatially uncorrelated voxels, for three different individual voxel probability thresholds (p_{thr} = .005, .025, and .05).

Figure 7 presents the corresponding results for spatially correlated voxels. To simulate spatial correlation, a Gaussian filter of width = 0.6 voxels (1 sigma) was used.

The Forman paper also presents a couple of graphs of statistical "power" vs. cluster size threshold. To conform with the present notation, the "power" in the Forman paper will be referred to here as "power/voxel". Power/voxel was calculated for different regions of true neural activation. A comparison of the results is shown in Figure 8 for uncorrelated voxels, using 4×4, 6×6, and 2×18 voxel activation regions.

Finally, Figure 9 presents the corresponding results for the case of spatially correlated voxels (Gaussian filter width = 0.6 voxels (1 sigma). In this figure we see more divergence in the results than in the previous figures.

1.4 Usage

The command line format for program AlphaSim is as follows:

AlphaSim \
-nx *n1* **-ny** *n2* **-nz** *n3* **-dx** *d1* **-dy** *d2* **-dz** *d3* \
[**-mask** *mset*] \
[**-fwhm** *s*] [**-fwhmx** *sx* **-fwhmy** *sy* **-fwhmz** *sz*] \
[**-gfw** *s*] [**-gfwx** *sx* **-gfwy** *sy* **-gfwz** *sz*] \
[**-power** **-ax** *n1* **-ay** *n2* **-az** *n3* **-zsep** *z*] \
-rmm *r* **-pthr** *p* **-iter** *n* [**-quiet**] [**-out** *filename*]

The different command line options are explained below.

1.5 Options

-nx *n1*, **-ny** *n2*, **-nz** *n3*

These commands indicate the dimensions of the image, in voxels, along the x, y, and z axes, respectively. Here, $n1$, $n2$, and $n3$ must be positive integers. For a 2-dimensional image, set $n3 = 1$.

-dx $d1$**, -dy** $d2$**, -dz** $d3$

These commands indicate the dimensions, of a single voxel, along the x, y, and z axes. Here, $d1$, $d2$, and $d3$ are the voxel dimensions (in mm.), and must be positive real numbers. For a 2-dimensional image, set $dz = 1.0$.

-mask *mset*

The optional -mask command instructs the program to use the $0th$ sub-brick of dataset *mset* (a sub-brick selector is allowed) as a mask to indicate which voxels to include in the cluster formation. The default is to use all voxels. This option applies only to calculation of statistical significance, not to calculation of statistical power. Note: The -mask command *replaces* the -nx, -ny, -nz, -dx, -dy, and -dz commands.

-fwhm s

The -fwhm command sets the width of the Gaussian filter which is convolved with the randomly generated image to simulate spatial correlation of voxel intensities. The number s (in mm.), which must be a nonnegative real number, is equal to the Filter Width Half Maximum (FWHM) of the Gaussian function which is applied to the data. The default value is $s = 0$.

-fwhmx sx**, -fwhmy** sy**, -fwhmz** sz

The -fwhm command applies the *same* Gaussian filter width to each axis of the image. Alternatively, one may specify independently, for each axis, the width of the Gaussian filter (in FWHM) to be applied along that axis of the image. Here, the numbers sx, sy, and sz specify the FWHM for the Gaussian filter along the x-, y-, and z-axes, respectively. Note: one should use *either* the -fwhm command, *or* all three -fwhmx, -fwhmy, and -fwhmz commands.

-sigma s

The -sigma command is an alternative to the -fwhm command, which sets the width of the Gaussian filter. The number s (in mm.), which must be a nonnegative real number, is equal to one standard deviation (or 1 sigma) for the Gaussian function which is applied to the data. The default value is $s = 0$.

-sigmax sx**, -sigmay** sy**, -sigmaz** sz

The -sigma command applies the *same* Gaussian filter width to each axis of the image. Alternatively, one may specify independently, for each axis, the width of the Gaussian filter (in sigmas) to be applied along that axis of the image. Here, the numbers sx, sy, and sz specify the 1-sigma value for the Gaussian filter along the x-, y-, and z-axes, respectively. Note: one should use *either* the -sigma command, *or* all three -sigmax, -sigmay, and -sigmaz commands.

-power

The -power command tells program AlphaSim to calculate the probability of detection of a user specified activation region. The program then reports the power for different cluster size thresholds. When the -power command is used, then the -ax, -ay, -az, and -zsep commands must also be present.

-ax $n1$**, -ay** $n2$**, -az** $n3$

The -ax, -ay, and -az commands indicate the dimensions of the true activation region, in voxels, along the x, y, and z axes, respectively. Here, $n1$, $n2$, and $n3$ must be positive integers. For a 2-dimensional image, set $n3 = 1$. Note that the -ax, -ay, and -az commands are used if and only if the -power command is used.

-zsep z

The -zsep command is used to specify the separation, in standard z-scores, between the mean of the signal distribution and the mean of the noise distribution. The real number z must be positive. Note that the -zsep command is used if and only if the -power command is used.

-rmm r

The -rmm command is used to set the cluster connectivity radius r (in mm.). That is, two activated voxels whose centers are less than r mm. apart are assumed to be members of the same cluster of associated voxels.

-pthr p

The -pthr command is used to assign the individual voxel probability threshold p. Obviously, it is necessary that p satisfy: $0 < p < 1$.

-iter n

The -iter command sets the number of Monte Carlo iterations to be performed by the program. The positive integer n should be large enough to give statistically significant results. A typical value would be $n = 1000$.

-quiet

The -quiet command suppresses output to the screen.

-out *filename*

The -out command is used to write the table of alpha vs. cluster size probabilities to a file with the given filename.

1.6 Examples

Example 1. Uncorrelated Voxels

A researcher is studying an *AFNI* 3D dataset to determine which voxels correspond to truly activated regions of the brain. The image consists of 17 slices of 64×64 voxels, each voxel measuring 3.75×3.75×7.0 mm. The researcher wishes to achieve a significance level of $\alpha = 0.05$, i.e., there should be only 0.05 probability of classifying an "inactive" voxel as "active". Since there are a total of 64×64×17= 69632 voxels, the Bonferroni correction implies using an individual voxel probability threshold of

$$p_{thr} = \frac{0.05}{69632} = 7.2 \times 10^{-7}.$$

Using such a low probability threshold does indeed provide protection against false positives, but it also destroys the power of the test, i.e., it is very difficult to detect true positives.

As an alternative, the researcher decides to use a combination of probability thresholding and minimum cluster size thresholding. Suppose that the researcher knows from experience that there is essentially zero spatial correlation of noise between voxels. Also, suppose that the researcher wishes to use a per voxel probability threshold of at least 0.005, in order to have sufficient power to detect true activated regions. An appropriate command line for execution of program AlphaSim is as follows:

```
AlphaSim -nx 64 -ny 64 -nz 17 -dx 3.75 -dy 3.75 -dz 7.0 \
-iter 1000 -rmm 7.1 -pthr 0.005 -fwhm 0.0 -out Example1.out
```

The -nx, -ny, and -nz commands tell the program how many voxels to use in the simulations, and the -dx, -dy, and -dz commands specify the voxel dimensions. The -iter command instructs the program to perform 1000 Monte Carlo simulations. The -rmm command specifies that active voxels whose centers are less than 7.1 mm apart are to be considered as belonging to the same cluster. The -pthr command sets the individual voxel detection probability threshold at 0.005. Since it is assumed that the voxels are spatially uncorrelated, the -fwhm command sets the Gaussian filter width to zero. Finally, the program output is sent to file "Example1.out". The program output is reproduced below.

Output from Program AlphaSim for Example 1.

Program:	AlphaSim
Author:	B. Douglas Ward
Initial Release:	18 June 1997
Latest Revision:	15 June 2000

Data set dimensions:

nx =	64	ny =	64	nz =	17	(voxels)
dx =	3.75	dy =	3.75	dz =	7.00	(mm)

Gaussian filter widths:

sigmax = 0.00	FWHMx = 0.00
sigmay = 0.00	FWHMy = 0.00
sigmaz = 0.00	FWHMz = 0.00

Cluster connection radius: rmm = 7.10

Threshold probability: pthr = 5.000000e-03

Number of Monte Carlo iterations = 1000

Cl Size	Frequency	Cum Prop	p/Voxel	Max Freq	Alpha
1	331660	0.975890	0.00500276	0	1.000000
2	7901	0.999138	0.00023972	750	1.000000
3	282	0.999968	0.00001278	239	0.250000
4	11	1.000000	0.00000063	11	0.011000

■

The program output, in addition to listing the various command line inputs, contains a table with 6 columns of numbers. The first column lists different cluster sizes (in voxels). The second column lists the frequency of occurrence for clusters of the size given in the first column. For example, in the 1000 random images that were produced by the program, clusters of exactly 3 voxels occurred 282 times. The third column lists the cumulative proportion of clusters having the size given in column 1. For example, 0.999968 of all clusters consist of 3 or fewer voxels. Column 4 lists the probability, per voxel, of a false positive detection, when the cluster threshold is set to the size given in column 1.

Column 5 lists the number of times, out of the 1000 random images, that the maximum size of a cluster in the image was as given by column 1. For example, in 11 of the 1000 images, the largest cluster in the image contained exactly 4 voxels. Therefore, if the cluster size listed in column 1 is used as the minimum cluster size threshold, then column 5 estimates the probability of a false detection occurring in the entire image.

Therefore, we see that if the researcher uses an individual voxel detection probability threshold of $p_{thr} = 0.005$ and a minimum cluster size threshold of $CS_{thr} = 4$ voxels, this is equivalent to an overall significance level of $\alpha = 0.011$. Thus, the desired significance level has been achieved through a combination of probability threshold and cluster size threshold. The individual voxel probability threshold of $p_{thr} = 0.005$ is much larger than the threshold of $p_{thr} = 7.2 \times 10^{-7}$ required by the Bonferroni procedure using probability thresholding alone. This means that the power of the test to detect true activated regions is greatly enhanced. However, this comes at the cost of not being able to detect regions consisting of 3 or fewer voxels (in this particular case).

By execution of Program AlphaSim using different values for p_{thr}, one can find various combinations of p_{thr} vs. CS_{thr} which yield the desired overall significance level α. In this way, one can get a sense of the trade-off between p_{thr} and CS_{thr}.

Example 2. Brain Mask

In the previous example, program AlphaSim estimated the false detection probabilities for voxels throughout the dataset. However, if we restrict consideration to just those voxels inside the brain, this should relax the cluster size threshold requirement. Therefore, suppose that a mask has been created for voxels inside the brain (this does not have to be very accurate; a simple intensity threshold should suffice), and this mask has been stored in dataset brain+orig. An appropriate command line for execution of program AlphaSim is as follows:

```
AlphaSim -mask brain+orig \
-iter 1000 -rmm 7.1 -pthr 0.005 -fwhm 0.0 -out Example2.out
```

Note that the -mask command replaces the -nx, -ny, -nz, -dx, -dy, and -dz commands, as these values are read directly from the mask dataset. The program output is reproduced below.

Output from Program AlphaSim for Example 2.

Program:	AlphaSim
Author:	B. Douglas Ward
Initial Release:	18 June 1997
Latest Revision:	15 June 2000

Data set dimensions:

nx =	64	ny =	64	nz =	17	(voxels)
dx =	3.75	dy =	3.75	dz =	7.00	(mm)

Mask filename = brain+orig
Voxels in mask = 11302

Gaussian filter widths:

sigmax = 0.00	FWHMx = 0.00
sigmay = 0.00	FWHMy = 0.00
sigmaz = 0.00	FWHMz = 0.00

Cluster connection radius: rmm = 7.10

Threshold probability: pthr = 5.000000e-03

Number of Monte Carlo iterations = 1000

Cl Size	Frequency	Cum Prop	p/Voxel	Max Freq	Alpha
1	54100	0.976305	0.00502362	284	1.000000
2	1266	0.999152	0.00023686	669	0.716000
3	43	0.999928	0.00001283	43	0.047000
4	4	1.000000	0.00000142	4	0.004000

■

By using the brain mask, the number of voxels under consideration has been reduced from 69632 to 11302. This allows the cluster size threshold to be reduced to $CS_{thr} = 3$ voxels, while maintaining $\alpha < 0.05$.

Example 3. ROI Mask

The user can further restrict the mask to just specific regions of interest. For example, suppose that dataset ROI+orig contains 1's for several small ROI's, and 0's elsewhere. An appropriate command line for execution of program AlphaSim is as follows:

```
AlphaSim -mask ROI+orig \
-iter 1000 -rmm 7.1 -pthr 0.005 -fwhm 0.0 -out Example3.out
```

The program output is reproduced below.

Output from Program AlphaSim for Example 3.

Program:	AlphaSim
Author:	B. Douglas Ward
Initial Release:	18 June 1997
Latest Revision:	15 June 2000

Data set dimensions:

nx =	64	ny =	64	nz =	17	(voxels)
dx =	3.75	dy =	3.75	dz =	7.00	(mm)

Mask filename = ROI+orig
Voxels in mask = 370

Gaussian filter widths:

sigmax = 0.00	FWHMx = 0.00
sigmay = 0.00	FWHMy = 0.00
sigmaz = 0.00	FWHMz = 0.00

Cluster connection radius: rmm = 7.10

Threshold probability: pthr = 5.000000e-03

Number of Monte Carlo iterations = 1000

Cl Size	Frequency	Cum Prop	p/Voxel	Max Freq	Alpha
1	1826	0.984897	0.00509459	831	0.859000
2	25	0.998382	0.00015946	25	0.028000
3	3	1.000000	0.00002432	3	0.003000

■

By using the ROI mask, the number of voxels under consideration has been reduced to 370. This allows the cluster size threshold to be reduced to $CS_{thr} = 2$ voxels, while maintaining $\alpha < 0.05$.

Example 4. Correlated Voxels

We next consider an example which is similar to Example 2, except that now we allow for spatial correlation of noise between voxels. Specifically, assume that spatial correlation between voxels can be represented by Gaussian filtering with FWHM filter width of 8.0 mm. along each of the x-, y-, and z-axes. Since spatial correlation increases the clustering of noise voxels, we will compensate by lowering the probability threshold to $p_{thr} = 0.001$. The necessary command line format is as follows:

```
AlphaSim -mask brain+orig \
-iter 1000 -rmm 7.1 -pthr 0.001 -fwhm 8.0 -out Example4.out
```

When executed, the program sends output to file "Example4.out", which is reproduced below:

Output from Program AlphaSim for Example 4.

Program:	AlphaSim
Author:	B. Douglas Ward
Initial Release:	18 June 1997
Latest Revision:	15 June 2000

Data set dimensions:

nx =	64	ny =	64	nz =	17	(voxels)
dx =	3.75	dy =	3.75	dz =	7.00	(mm)

Mask filename = brain+orig
Voxels in mask = 11302

Gaussian filter widths:

sigmax = 3.40	FWHMx = 8.00
sigmay = 3.40	FWHMy = 8.00
sigmaz = 3.40	FWHMz = 8.00

Cluster connection radius: rmm = 7.10

Threshold probability: pthr = 1.000000e-03

Number of Monte Carlo iterations = 1000

Cl Size	Frequency	Cum Prop	p/Voxel	Max Freq	Alpha
1	4932	0.640603	0.00109874	63	0.999000
2	1692	0.860371	0.00066236	265	0.936000
3	587	0.936615	0.00036294	279	0.671000
4	274	0.972204	0.00020713	203	0.392000
5	124	0.988310	0.00011016	104	0.189000
6	37	0.993116	0.00005530	35	0.085000
7	32	0.997272	0.00003566	29	0.050000
8	12	0.998831	0.00001584	12	0.021000
9	7	0.999740	0.00000734	7	0.009000
10	2	1.000000	0.00000177	2	0.002000

■

The effect of Gaussian filtering is to increase the prevalence of clustering of voxels. This increases the reported alpha values for each cluster size threshold. In particular, to achieve $\alpha = 0.05$, it is necessary to go to a cluster size threshold of $CS_{thr} = 7$ voxels (in combination with the individual voxel probability threshold of $p_{thr} = 0.001$). Recall that $CS_{thr} = 3$ sufficed in Example 2 (uncorrelated voxels) with $p_{thr} = 0.005$. If the cluster size threshold is not satisfactory (i.e., if the user wishes to be able to detect regions of activation smaller than 7 voxels), then it will be necessary to further reduce p_{thr}.

2 Program 3dFWHM

2.1 Purpose

Program 3dFWHM provides a means of estimating the spatial correlation of voxels in an *AFNI* 3D dataset. In order to use program AlphaSim, it is necessary to know the degree of voxel spatial correlation, so that this can be accounted for in the simulation. Spatial correlation is modeled by applying a Gaussian filter to the random image data. The extent of spatial correlation is specified by entering the width of the Gaussian filter corresponding to each axis. The numbers $FWHM_x$, $FWHM_y$, and $FWHM_z$ (FWHM, for "Full Width Half Maximum") are estimated by program 3dFWHM for a user specified input dataset.

2.2 Theory

Forman *et al.* [1] derive the following equation for estimating the Gaussian filter width s:

$$s = \sqrt{-\frac{1}{4 \times \ln\left(1 - \frac{S_\delta^2}{2S^2}\right)}}$$

where

s = width (standard deviation) of Gaussian filter in voxels;

S_δ^2 = variance of the difference between each voxel and its nearest neighbors over the entire image;

S^2 = variance of voxel intensities over the entire image.

It is assumed in [1] that the (two-dimensional) image is isotropic. For *AFNI* 3-dimensional datasets, it is better not to make this assumption. Therefore, a separate filter width is calculated for each of the principal axes.

Using a procedure similar to that described in [2], Program 3dFWHM first estimates the variances of the differences between voxels along each of the 3 principal axes. At each voxel location (i, j, k) we have:

$$\begin{aligned}
u_x(i,j,k) &= u(i+1,j,k) - u(i,j,k), \\
u_y(i,j,k) &= u(i,j+1,k) - u(i,j,k), \\
u_z(i,j,k) &= u(i,j,k+1) - u(i,j,k).
\end{aligned}$$

The variances are then estimated by:

$$
\begin{aligned}
V &= \frac{1}{N-1}\sum_{i,j,k}\left[u(i,j,k)-\bar{u}\right]^2, \\
V_x &= \frac{1}{N_x-1}\sum_{i,j,k}\left[u_x(i,j,k)-\bar{u}_x\right]^2, \\
V_y &= \frac{1}{N_y-1}\sum_{i,j,k}\left[u_y(i,j,k)-\bar{u}_y\right]^2, \\
V_z &= \frac{1}{N_z-1}\sum_{i,j,k}\left[u_z(i,j,k)-\bar{u}_z\right]^2.
\end{aligned}
$$

The Gaussian filter widths are then given by:

$$
\begin{aligned}
FWHM_x &= 2\sqrt{2\ln(2)} \times s_x = 2\sqrt{2\ln(2)} \times \sqrt{-\frac{1}{4\times\ln\left(1-\frac{V_x}{2V}\right)}} \times dx, \\
FWHM_y &= 2\sqrt{2\ln(2)} \times s_y = 2\sqrt{2\ln(2)} \times \sqrt{-\frac{1}{4\times\ln\left(1-\frac{V_y}{2V}\right)}} \times dy, \\
FWHM_z &= 2\sqrt{2\ln(2)} \times s_z = 2\sqrt{2\ln(2)} \times \sqrt{-\frac{1}{4\times\ln\left(1-\frac{V_z}{2V}\right)}} \times dz.
\end{aligned}
$$

where dx, dy, and dz are the voxel dimensions.

2.3 Usage

The command line format for program 3dFWHM is as follows:

3dFWHM -dset filename [-quiet] [-out filename]

2.4 Options

-dset filename

The -dset command specifies that "filename" is the name of the *AFNI* 3D dataset to be analyzed.

-quiet

The -quiet command suppresses output to the screen.

-out filename

The -out command sends the results to file "filename".

2.5 Examples

Example 1. Suppose that the user wishes to determine the equivalent Gaussian filter width for simulating the spatial correlation between voxels in dataset r1:time@1+orig.BRIK (and .HEAD). The appropriate command line to use is:

```
3dFWHM -dset /home/ward/test/AlphaSim/verbal/r1:time@1+orig
```

The program responds with:

```
Program 3dFWHM
Last revision: 20 February 1997

 var = 0.000003
 varxx = 0.000005   varyy = 0.000004   varzz = 0.000005
 sx =1.288034       sy = 1.477418      sz = 2.576037

Gaussian filter widths:
 sigmax = 1.29   FWHMx = 3.03
 sigmay = 1.48   FWHMy = 3.48
 sigmaz = 2.58   FWHMz = 6.07
```

3 References

[1] S. D. Forman, J. D. Cohen, M. Fitzgerald, W. F. Eddy, M. A. Mintun, D. C. Noll, Improved Assessment of Significant Activation in Functional Magnetic Resonance Imaging (fMRI): Use of a Cluster-Size Threshold. 636-647 (1995).

[2] J. Xiong, J-H Gao, J. L. Lancaster, P. T. Fox, Clustered Pixels Analysis for Functional MRI Activation Studies of the Human Brain, *Human Brain Mapping* 3: 287-301 (1995).