Defining ROIs

You can use the built-in ROI standard objects, or you can specify you own. Recognized sizes are 1, 7, 19, 27.

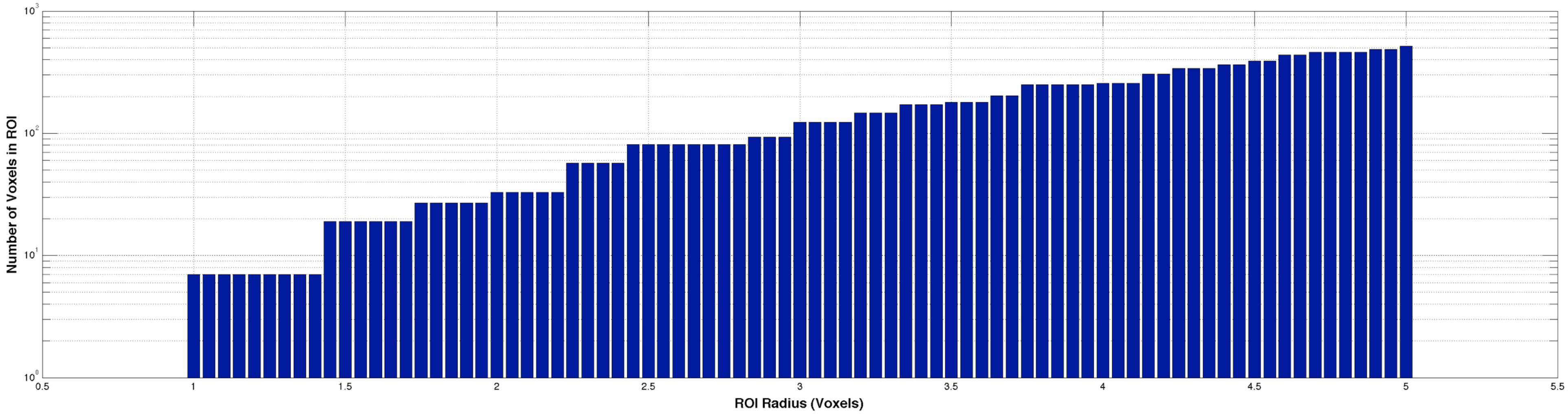
```
parameters.mni.
   coodinates          = [x_0 y_0 z_0;
                           x_1 y_1 z_1;
                           ...
                           x_n y_n z_n];
   size                = #;
                    Or
   size.XROI           = [-1  0 -1];
       .YROI           = [ 0  0  0];
       .ZROI           = [ 0  0  0];
```

So specify you own use:

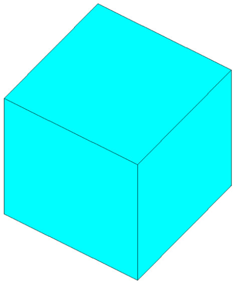
Or, have the code build the arrays

```
XYZ = SOM_MakeSphereROI(radius);
parameters.mni.rois.size.XROI=XYZ(1,:);
parameters.mni.rois.size.YROI=XYZ(2,:);
parameters.mni.rois.size.ZROI=XYZ(3,:);
```

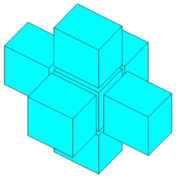
Obviously, some range of radii will give same ROI definition.



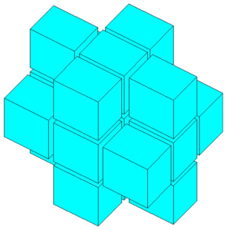
n Voxels : 1, Radius = 0.000000



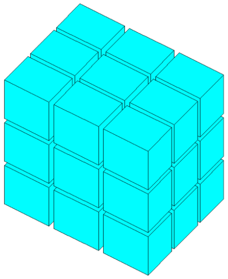
n Voxels : 7, Radius = 1.000000



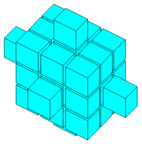
n Voxels : 19, Radius = 1.450000



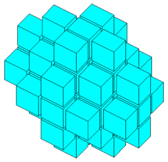
n Voxels : 27, Radius = 1.750000



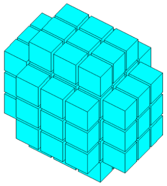
n Voxels : 33, Radius = 2.000000



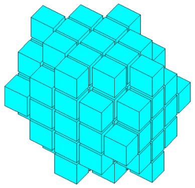
n Voxels : 57, Radius = 2.250000



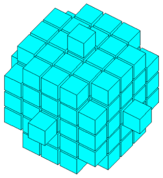
n Voxels : 81, Radius = 2.450000



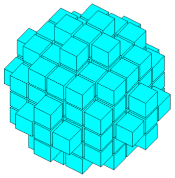
n Voxels : 93, Radius = 2.850000



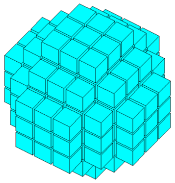
n Voxels : 123, Radius = 3.000000



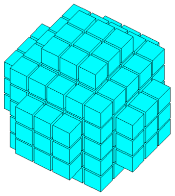
n Voxels : 147, Radius = 3.200000



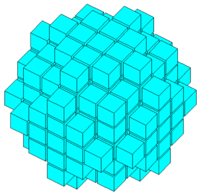
n Voxels : 171, Radius = 3.350000



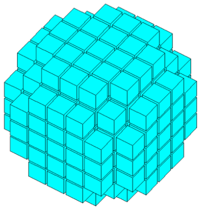
n Voxels : 179, Radius = 3.500000



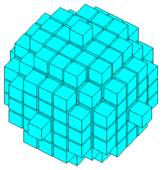
n Voxels : 203, Radius = 3.650000



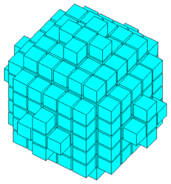
n Voxels : 251, Radius = 3.750000



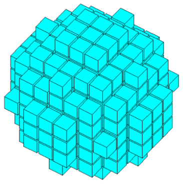
n Voxels : 257, Radius = 4.000000



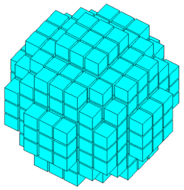
n Voxels : 305, Radius = 4.150000



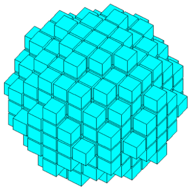
n Voxels : 341, Radius = 4.250000



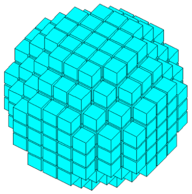
n Voxels : 365, Radius = 4.400000



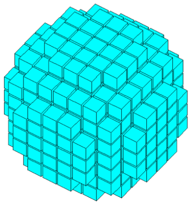
n Voxels : 389, Radius = 4.500000



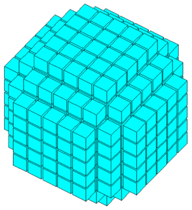
n Voxels : 437, Radius = 4.600000



n Voxels : 461, Radius = 4.700000



n Voxels : 485, Radius = 4.900000



n Voxels : 515, Radius = 5.000000