

# 脑图像的数据预处理

在“[BrainWeb: Simulated Brain Database使用说明](#)”中已经介绍了如何下载并打开脑数据库，这篇文章将0、1、2、3、8类分割出来，用以后续对图像的处理。

作者：凯鲁嘎吉 - 博客园<http://www.cnblogs.com/kailugaji/>

## 1.准备

从[BrainWeb: Simulated Brain Database](#)网站中下载我们需要的脑图像数据，如t1\_icbm\_normal\_1mm\_pn0\_rf0.rawb文件，表示在T1模态、icmb协议下，切片厚度为1mm，噪声水平为0，灰度不均匀水平为0的正常脑图像。

## 2.程序

### main.m

```
function main(filename,name,num)
% 函数main(filename, num)中的第一个参数filename是欲读取的rawb文件的文件名，第二个参数num就是第多少张。
%例如：main('t1_icbm_normal_1mm_pn0_rf0.rawb','train.txt',90), main('phantom_1.0mm_normal_csf.rawb','train.txt',90)
mark=Mark('phantom_1.0mm_normal_crisp.rawb',num);
read=readrawb(filename, num);
[row,col]=size(read);
for i=1:row %行
    for j=1:col %列
        if mark(i,j)==0
            read_new(i,j)=0;
        else
            read_new(i,j)=read(i,j); %将第0、1、2、3、8类拿出来，其余类为0
        end
    end
end
end
Write_txt(name,read_new); %将数据写入TXT文件
% 旋转90° 并显示出来
read_new=imrotate(read_new, 90);
imshow(uint8(read_new));
end
```

### Mark.m

```

function mark=Mark(filename,num)
%将标签为1、2、3、8类分出来，其余为0，mark取值：0、1、2、3、8
%[mark_new,mark]=Mark('phantom_1.0mm_normal_crisp.rawb',90);
fp=fopen(filename);
temp=fread(fp, 181 * 217 * 181);
image=reshape(temp, 181 * 217, 181);
images=image(:, num);
images=reshape(images, 181, 217);
mark_data=images;
fclose(fp);
%将第0、1、2、3、8类标签所在的坐标点拿出来，其余置0
for i=1:181
    for j=1:217
        if (mark_data(i,j)==1) || (mark_data(i,j)==2) || (mark_data(i,j)==3) || (mark_data(i,j)==8)
            mark(i,j)=mark_data(i,j);
        else
            mark(i,j)=0;
        end
    end
end
end

```

## readrawb.m

```

function g = readrawb(filename, num)
% 函数readrawb(filename, num)中的第一个参数filename是欲读取的rawb文件的文件名，第二个参数num就是第多少张。
fid = fopen(filename);
% 连续读取181*217*181个数据，这时候temp是一个长度为181*217*181的向量。
% 先将rawb中的所有数据传递给temp数组，然后将tempreshape成图片集。
temp = fread(fid, 181 * 217 * 181);
% 所以把它变成了一个181*217行，181列的数组，按照它的代码，这就是181张图片的数据，每一列对应一张图。
% 生成图片集数组。图片集images数组中每一列表示一张图片。
images = reshape(temp, 181 * 217, 181);
% 读取数组中的第num行，得到数组再reshape成图片原来的行数和列数：181*217。
image = images(:, num);
image = reshape(image, 181, 217);
g = image;
fclose(fid);
end

```

## Write\_txt.m

```

function Write_txt(name,read)
%将数据写入txt文件
fp=fopen(name,'w');
[row,col]=size(read);

```

```

for i=1:row    %行
    for j=1:col    %列
        if j==col
            fprintf(fp,'%f\n',read(i,j));    %换行    %f或者%d
        else
            fprintf(fp,'%f\t',read(i,j));    %多个空格tab
        end
    end
end
fclose(fp);

```

## processed\_data.m

```

function processed_data(filename,name,num)
%将1、2、3、8类的数据做归一处理，其余为0
% processed_data('t1_icbm_normal_1mm_pn0_rf0.rawb','train.txt',90)
mark=Mark('phantom_1.0mm_normal_crisp.rawb',num);
read=readrawb(filename, num);
[row,col]=size(read);
for i=1:row    %行
    for j=1:col    %列
        if mark(i,j)==0
            read_new(i,j)=0;
        else
            read_new(i,j)=read(i,j)./255;    %将第0、1、2、3、8类拿出来，其余类为0
        end
    end
end
end
Write_txt(name,read_new);    %将数据写入TXT文件

```

## init\_image.m

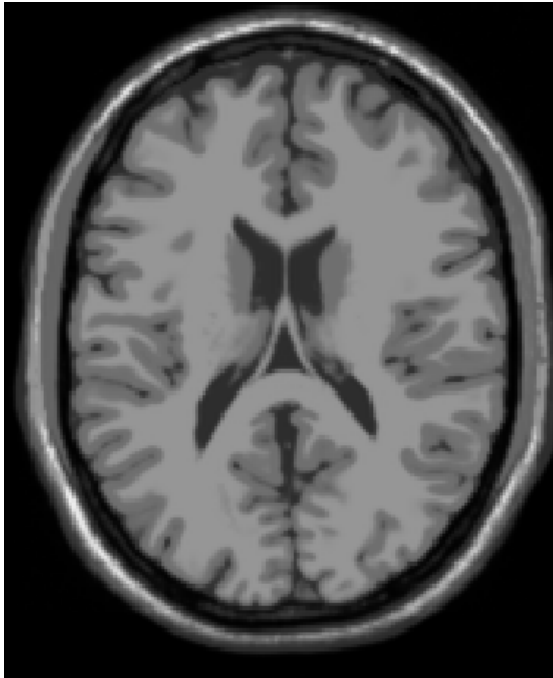
```

function init_image(filename,num)
%function init_image(filename,name,num)
% 函数init_image(filename,num)中的第一个参数filename是欲读取的rawb文件的文件名，第二个参数num就是第多少张。输出为原始图像，未处理
%例如：init_image('t1_icbm_normal_1mm_pn0_rf0.rawb','train.txt',90),  init_image('phantom_1.0mm_normal_csf.rawb','train.txt',90)
read=readrawb(filename, num);
%Write_txt(name,read);    %将数据写入文件
% 旋转90° 并显示出来
read=imrotate(read, 90);
imshow(uint8(read));
end

```

## 3.结果

```
>> init_image('t1_icbm_normal_1mm_pn0_rf0.rawb',90)
```



```
>> main('t1_icbm_normal_1mm_pn0_rf0.rawb','train.txt',90)
```



## 4.注意

init\_image()这个函数输出原图像，main()这个函数将0、1、2、3、8类分离出来（用前四个函数即可），用于后续的研究，processed\_data()这个函数对1、2、3、8类进行归一化，并将结果写入TXT文件。这篇文章仅作为保存我之前所做的内容，今后不会研究脑图像，但我[之前的博客园文章中提到的聚类算法](#)都可以用在脑图像分割中，有兴趣的话可以对聚类算法用在脑图像分割这个领域做进一步研究。

补充：BrainWeb: 20 Anatomical Models of 20 Normal Brains

注意：事先在[https://brainweb.bic.mni.mcgill.ca/brainweb/anatomic\\_normal\\_20.html](https://brainweb.bic.mni.mcgill.ca/brainweb/anatomic_normal_20.html)下载subject04\_crisp\_v.rawb与subject04\_csf\_v.rawb

# BrainWeb: 20 Anatomical Models of 20 Normal Brains

Each anatomical model consists of a set of 3-dimensional "fuzzy" tissue membership volumes, one for each tissue class: Background, Skull, Blood vessels, Connective (region around fat), dura matter and bone marrow.

**Note: Some differences exist between these models classes and our earlier model classes: There is no longer a separate vessels, dura matter, and bone marrow**

The voxel values in these volumes reflects the proportion of that tissue type present in that voxel, in the range [0,1]. In addition, a model is provided which consists of a class label (integer) at each voxel, representing the tissue which contributes the most to that voxel (1=Background, 2=Muscle, 3=Fat, 4=Fat, 5=Muscle, 6=Muscle/Skin, 7=Skull, 8=vessels, 9=around fat, 10 =dura matter, 11=bone marrow).

Subject Number: 04 ▾

Discrete Model: [\[View\]](#) [\[Download\]](#)

## Fuzzy Models:

Background [\[View\]](#) [\[Download\]](#)

CSF [\[View\]](#) [\[Download\]](#)

Grey Matter [\[View\]](#) [\[Download\]](#)

White Matter [\[View\]](#) [\[Download\]](#)

Fat [\[View\]](#) [\[Download\]](#)

Muscle [\[View\]](#) [\[Download\]](#)

Muscle / Skin [\[View\]](#) [\[Download\]](#)

Skull [\[View\]](#) [\[Download\]](#)

Vessels [\[View\]](#) [\[Download\]](#)

Connective [\[View\]](#) [\[Download\]](#)

Dura [\[View\]](#) [\[Download\]](#)

Marrow [\[View\]](#) [\[Download\]](#)

# BrainWeb download: subject04\_crisp

Subject 04 Discrete model

MINC volume info:

image: unsigned byte 0 to 255  
image dimensions: zspace yspace xspace

dimension name	length	step	start
zspace	362	0.5	-72.25
yspace	434	0.5	-126.25
xspace	362	0.5	-90.25

Original MINC file size: 3.4 Mb

---

## File format:

- ☐ MINC  
☒ raw byte (unsigned)  
☐ raw short (12 bit)  
[\(about data formats\)](#)

## Compression:

- ☒ none  
☐ gnuzip  
☐ bzip2

We'd like to know who is downloading these data, so we would appreciate it if you would provide us with the fo

Name:	<input type="text"/>
Institution:	<input type="text"/>
E-mail:	<input type="text"/>

[Back to BrainWeb Home Page](#)

```
function g = readrawb(filename, num)
fid = fopen(filename);
temp = fread(fid, 362 * 434 * 362);
images = reshape(temp, 362, 434, 362);
```

```

image = images(:, :, num);
g = image;
fclose(fid);
end

function init_image(filename, num)
read=readrawb(filename, num);
read=imrotate(read, 90);
imshow(uint8(read));
end

function main(filename, name, num)
mark=Mark('subject04_crisp_v.rawb', num);
read=readrawb(filename, num);
[ row, col ] = size(read);
for i=1:row %行
    for j=1:col %列
        if mark(i, j) == 0
            read_new(i, j) = 0;
        else
            read_new(i, j) = read(i, j);
        end
    end
end
end
Write_txt(name, read_new);
read_new=imrotate(read_new, 90);
imshow(uint8(read_new));
end

function mark=Mark(filename, num)
fp=fopen(filename);
temp=fread(fp, 362 * 434 * 362);
image=reshape(temp, 362, 434, 362);
images=image(:, :, num);
images=reshape(images, 362, 434);
mark_data=images;
fclose(fp);
for i=1:362
    for j=1:434
        if (mark_data(i, j) == 1) || (mark_data(i, j) == 2) || (mark_data(i, j) == 3) || (mark_data(i, j) == 8)
            mark(i, j) = mark_data(i, j);
        else
            mark(i, j) = 0;
        end
    end
end
end
end

```



```
function Write_txt(name,read)
fp=fopen(name,'w');
[row,col]=size(read);
for i=1:row    %行
    for j=1:col    %列
        if j==col
            fprintf(fp,'%f\n',read(i,j));
        else
            fprintf(fp,'%f\t',read(i,j));
        end
    end
end
fclose(fp);
```

```
function processed_data(filename,name,num)
mark=Mark('subject04_crisp_v.rawb',num);
read=readrawb(filename, num);
[row,col]=size(read);
for i=1:row    %行
    for j=1:col    %列
        if mark(i,j)==0
            read_new(i,j)=0;
        else
            read_new(i,j)=read(i,j)./255;
        end
    end
end
end
Write_txt(name,read_new);
```

命令行窗口输入>>main('subject04\_csf\_v.rawb','train.txt',90)

---

