# TPM++说明

TPM++是 SPM12 中Tissue Probability Map的c++重写版本，使用Eigen进行矩阵运算替代Matlab框架，使用SimpleITK作为图像库，图像格式和坐标空间均和ITK 一致。TPM++基于GPL v2协议。

TPM++计算结果与SPM12相同，计算性能比spm12 快N倍，其中N为CPU物理核心数量。默认情况下TPM++不使用超线程，超线程已经无法提供能高性能。

TPM++ 提供了一个命令行程序，计算Wrap并进行图像分割：

命令如下：TPM++.exe -in image.nii -tpm D:/TPM.nii -out D:/\_out.nii -json D:/tpminput.json -AffineOnly 0

参数说明：

-in: 输入图像路径

-tpm:TPM模板路径

-out:输出文件路径，实际输出文件将在此路径基础上添加后缀：

\_FieldInverse：逆向形变场

\_FieldForward：正向形变场

\_BiasCorrected：变差矫正后

\_BiasField：偏差场

\_TissNative：分割结果（原始空间下）

\_TissImported:分割结果（Dartel空间下）

\_TissModulate:分割结果（导入空间下，调制后）

\_TissUnModulate:分割结果（导入空间下，调制前）

-AffineOnly: 设置为1 时仅仅计算仿射变换，输出仿射矩阵。

-json: 记录了TPM运算的所有参数的json文件TpmInput.json文件是一个采样数为3，分割为6个组织类型的基本参数案例。

json文件案例：

{

// "各个组织是否生成相关TPM的设定"

    "m\_tissues": [

        {

            "m\_ngaus": 1,

            "m\_NativeTissue\_NativeSpace": true,

            "m\_NativeTissue\_DartelImported": true,

            "m\_WarpTissue\_Modulated": true,

            "m\_WarpTissue\_UnModulated": true

        },

        {

            "m\_ngaus": 1,

            "m\_NativeTissue\_NativeSpace": true,

            "m\_NativeTissue\_DartelImported": true,

            "m\_WarpTissue\_Modulated": true,

            "m\_WarpTissue\_UnModulated": true

        },

        {

            "m\_ngaus": 2,

            "m\_NativeTissue\_NativeSpace": true,

            "m\_NativeTissue\_DartelImported": true,

            "m\_WarpTissue\_Modulated": true,

            "m\_WarpTissue\_UnModulated": true

        },

        {

            "m\_ngaus": 3,

            "m\_NativeTissue\_NativeSpace": true,

            "m\_NativeTissue\_DartelImported": true,

            "m\_WarpTissue\_Modulated": true,

            "m\_WarpTissue\_UnModulated": true

        },

        {

            "m\_ngaus": 4,

            "m\_NativeTissue\_NativeSpace": true,

            "m\_NativeTissue\_DartelImported": true,

            "m\_WarpTissue\_Modulated": true,

            "m\_WarpTissue\_UnModulated": true

        },

        {

            "m\_ngaus": 2,

            "m\_NativeTissue\_NativeSpace": true,

            "m\_NativeTissue\_DartelImported": true,

            "m\_WarpTissue\_Modulated": true,

            "m\_WarpTissue\_UnModulated": true

        }

    ],

    "m\_Channels": [//每个输入文件（通道）的设定

        {

            "m\_Bisareg": 0.001000,

            "m\_Biasfwhm": 60.000000,

            "m\_biascorrect": true,

            "m\_biasfield": true

        }

    ],

    "m\_nsample": 3,//采样数

    "m\_fwhm": 0.000000,//模糊

    "m\_reg": [//规则化

        0.000000,

        0.001000,

        0.500000,

        0.050000,

        0.200000

    ],

    "m\_lkp": [//高斯函数查找表,sTpmInputs(sTissues &tss, sChannels &Channels) 构造函数可以计算该表

        1.000000,

        2.000000,

        3.000000,

        3.000000,

        4.000000,

        4.000000,

        4.000000,

        5.000000,

        5.000000,

        5.000000,

        5.000000,

        6.000000,

        6.000000

    ],

"m\_ptype": 0,

/\*先验类型：

enum PriorsType

{

    PriorsType\_mni = 0,

    PriorsType\_imni,

    PriorsType\_rigid,

    PriorsType\_subject,

    PriorsType\_eastern,

    PriorsType\_none = 0xffffffff

};\*/

    "m\_dFieldInverse": true,

    "m\_dFieldForward": true,

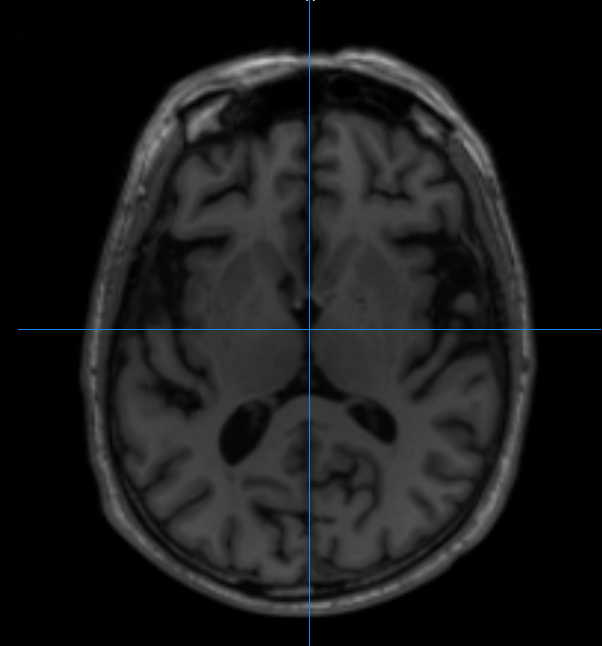
    "m\_Mrf": false,

    "m\_Cleanup": 1

}

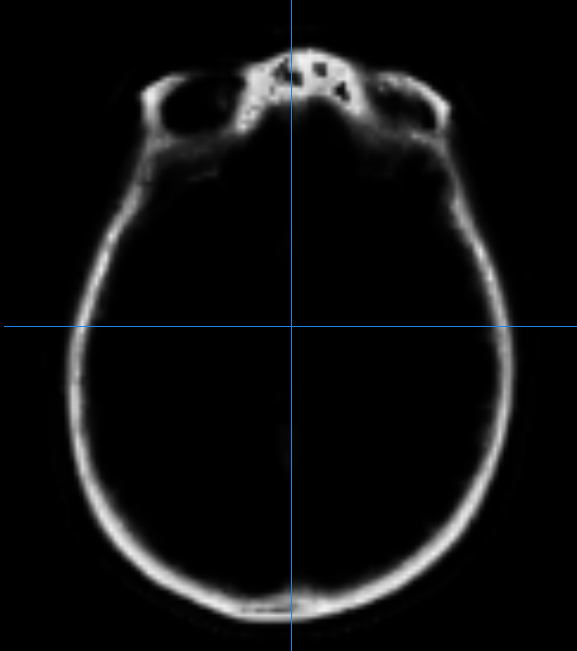
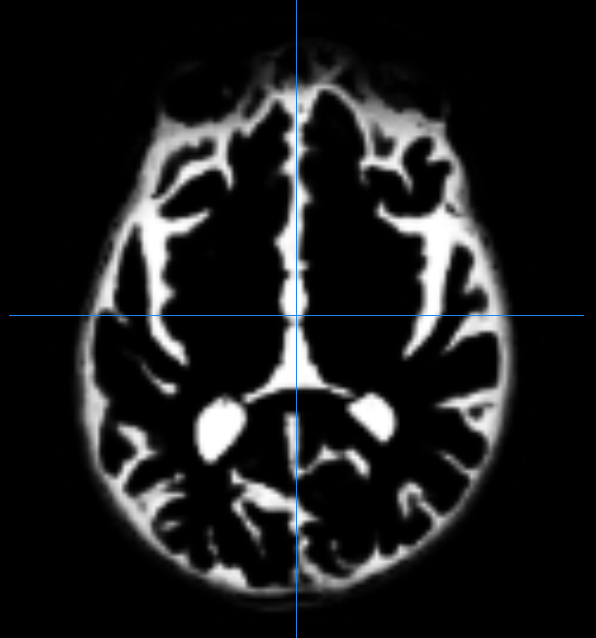
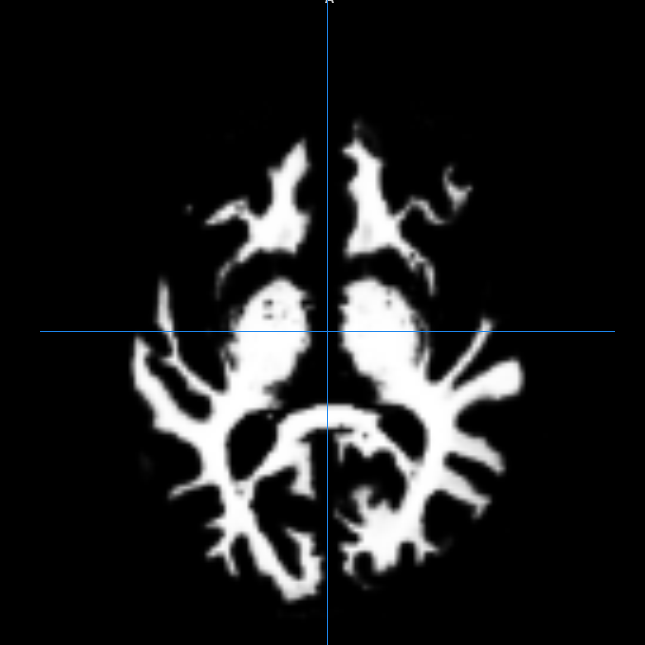
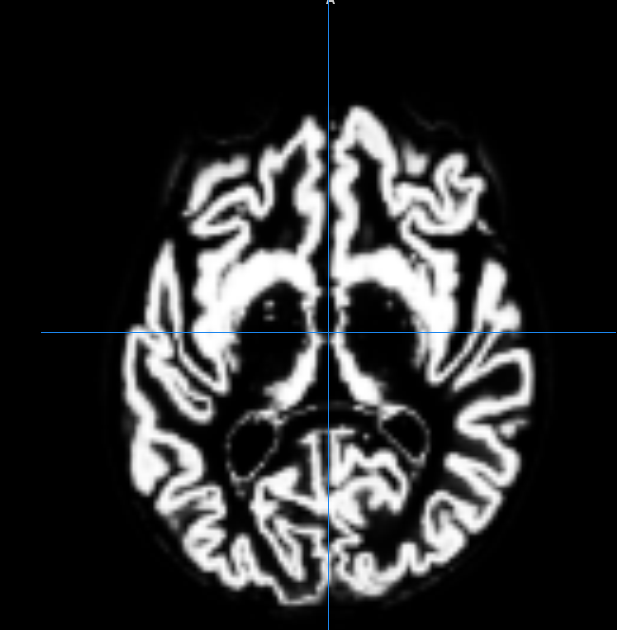
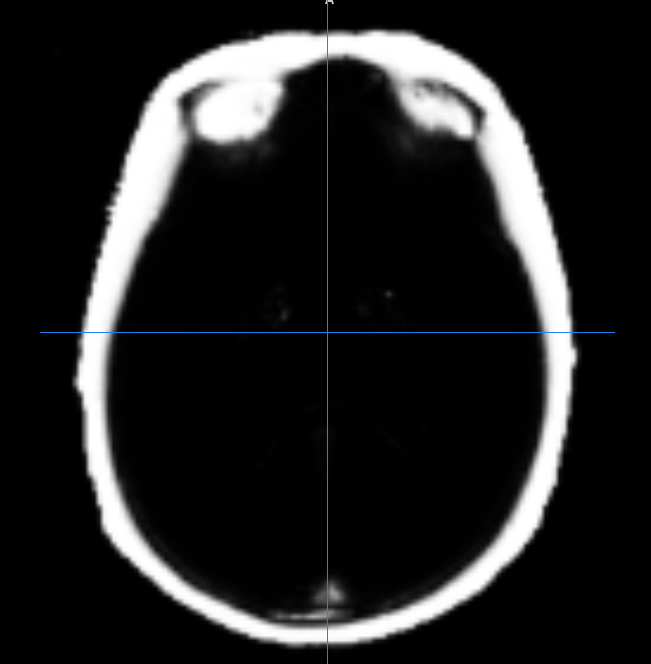
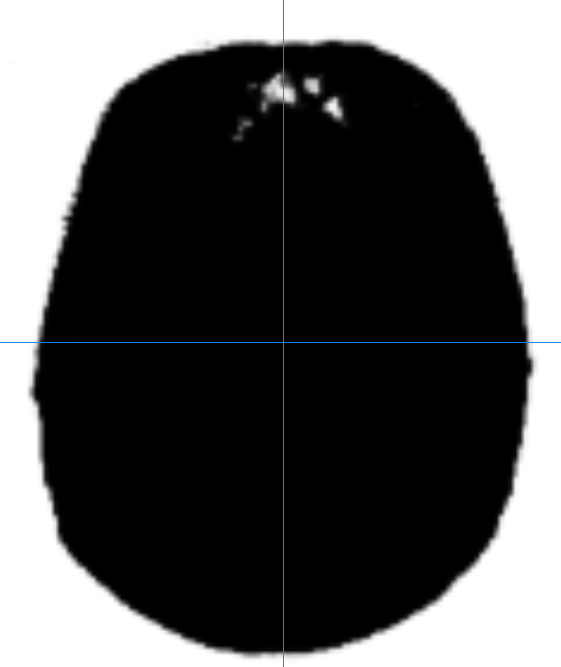
测试案例：

输入：某核磁脑影像文件，SPM12提供的TPM.nii文件

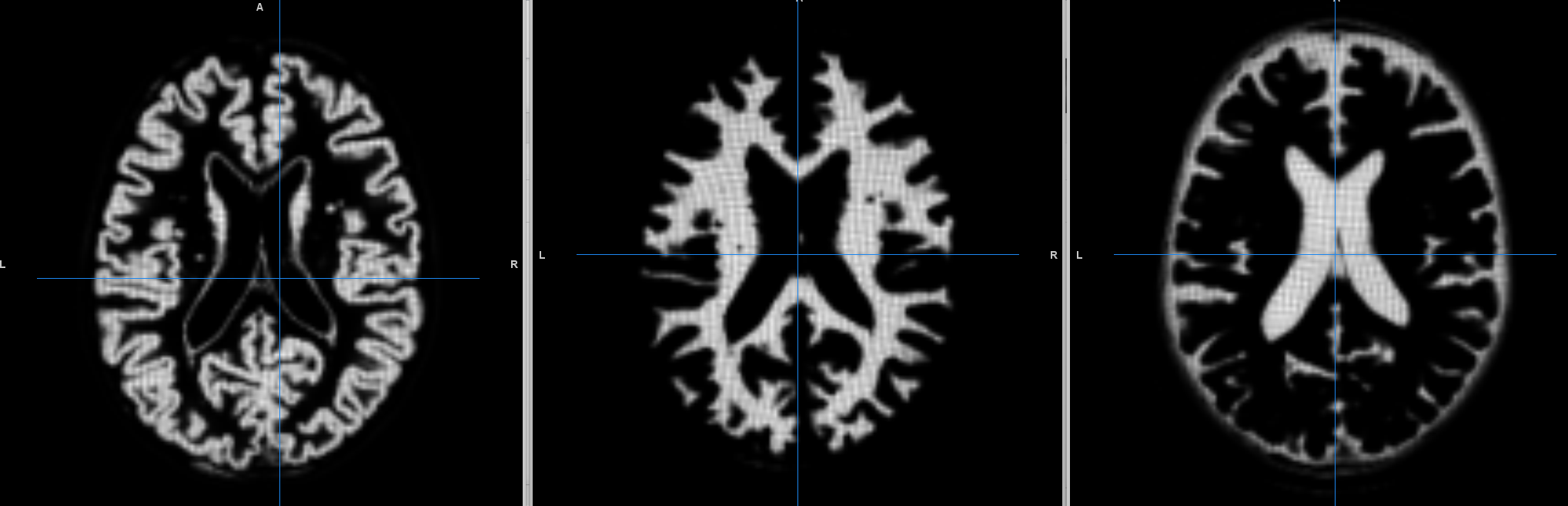


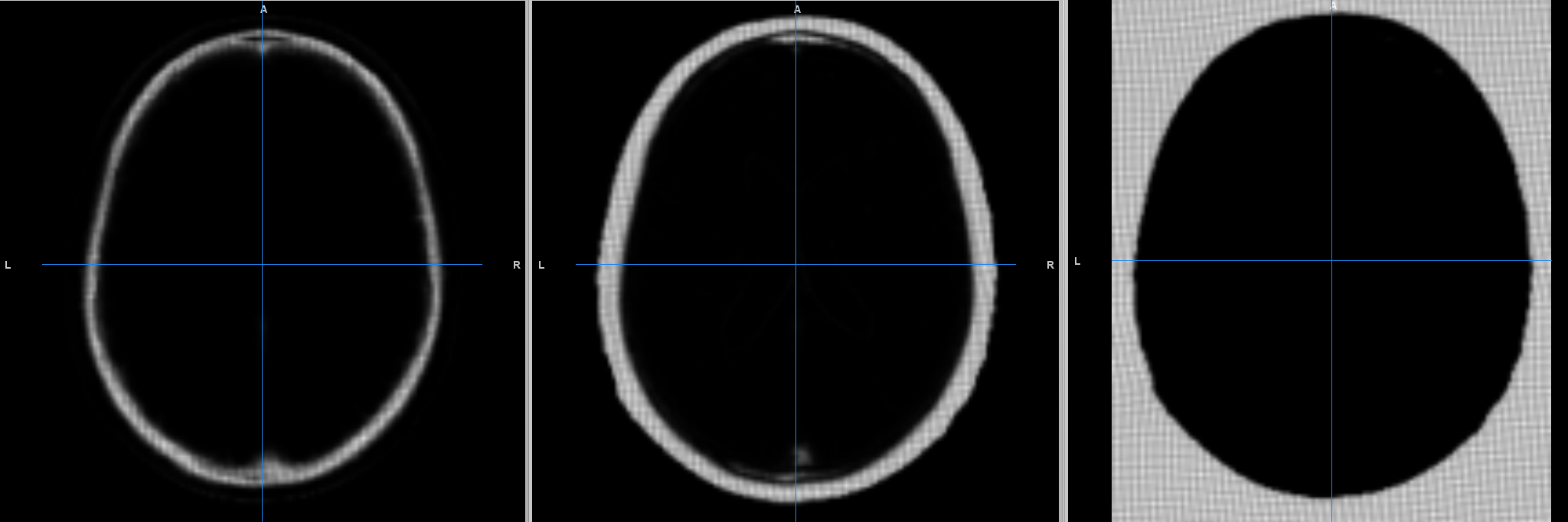
输出：

TissImported 0-5

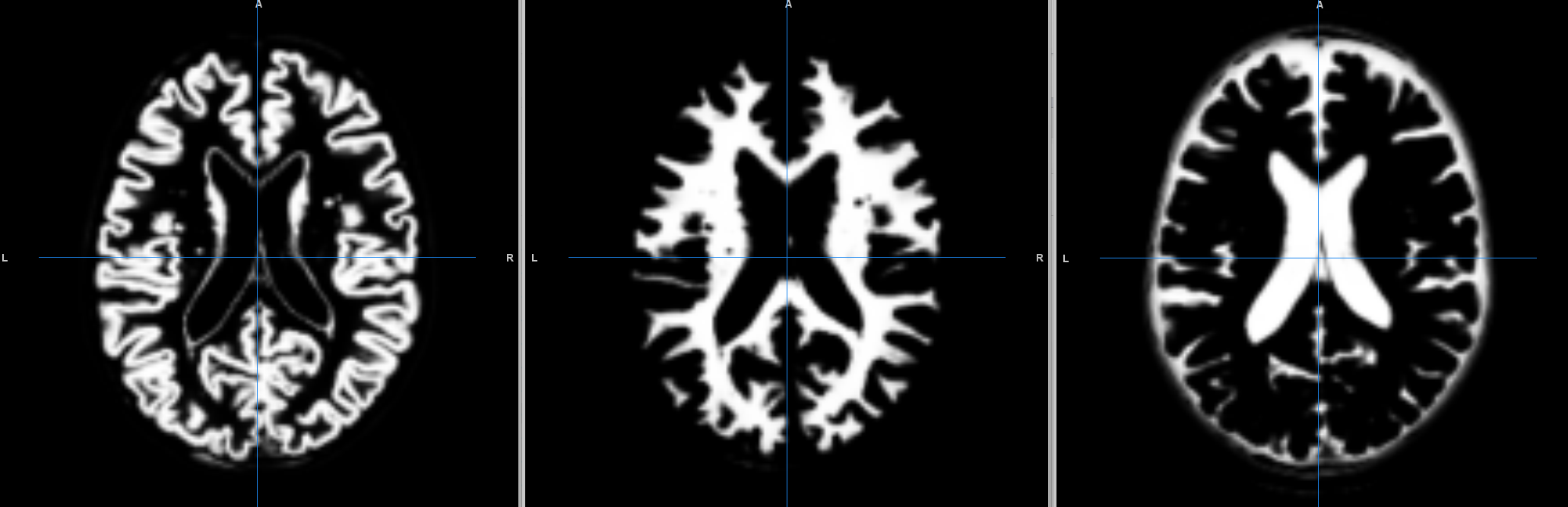
  

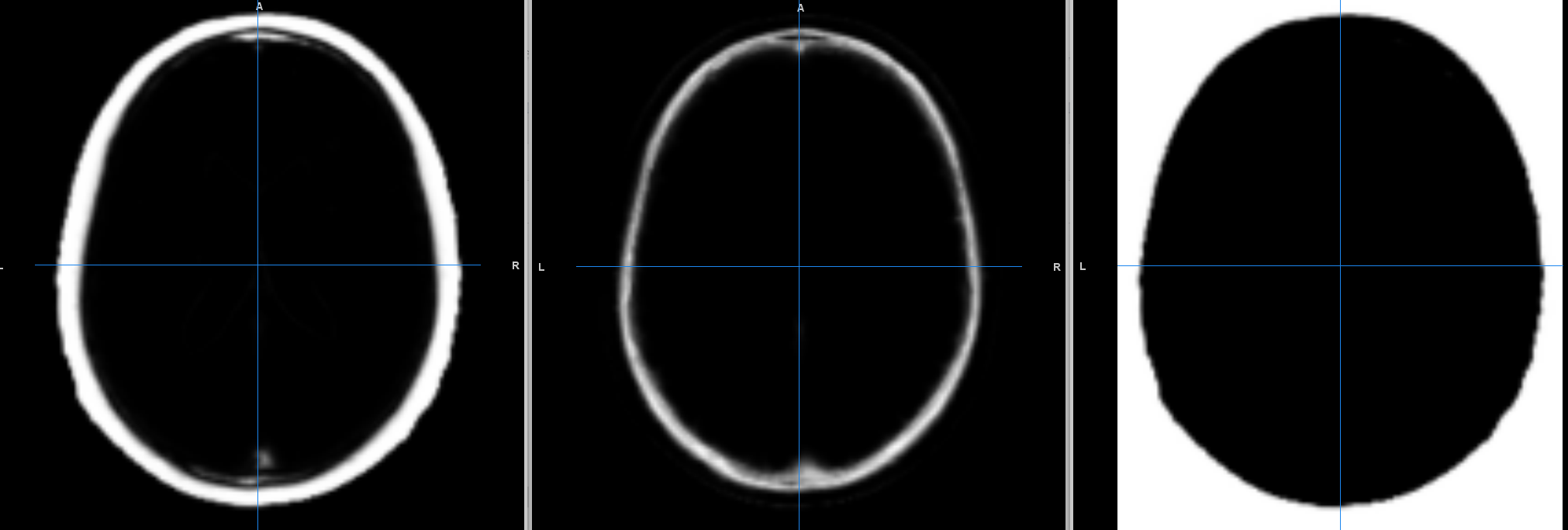
TissUnModulate 0-5



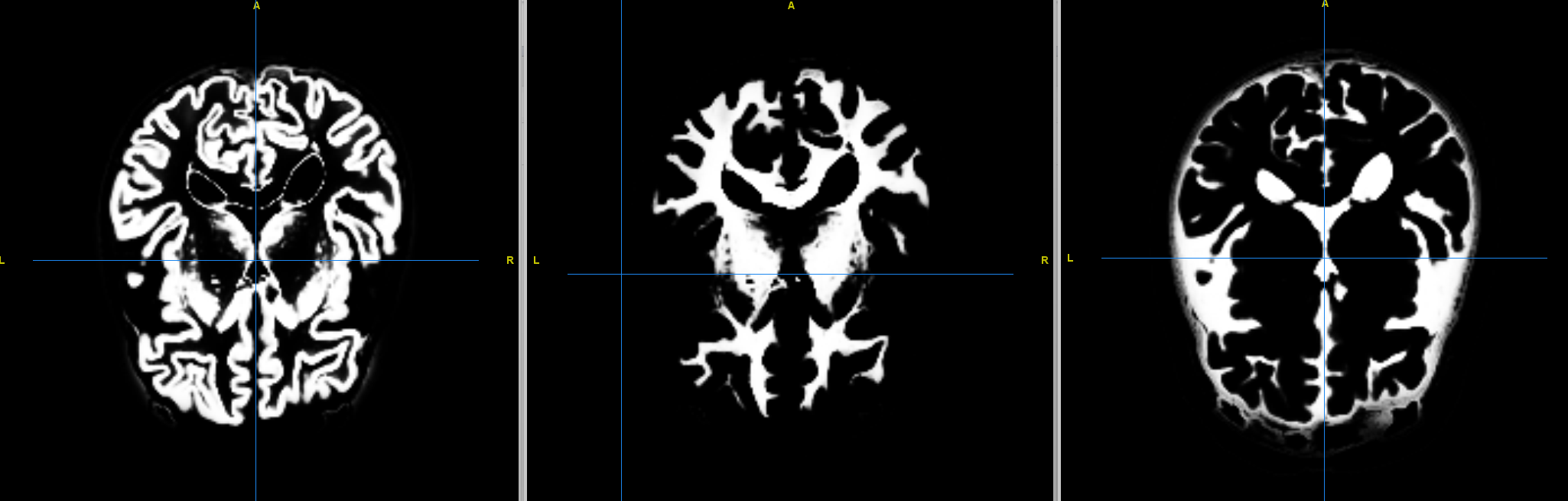


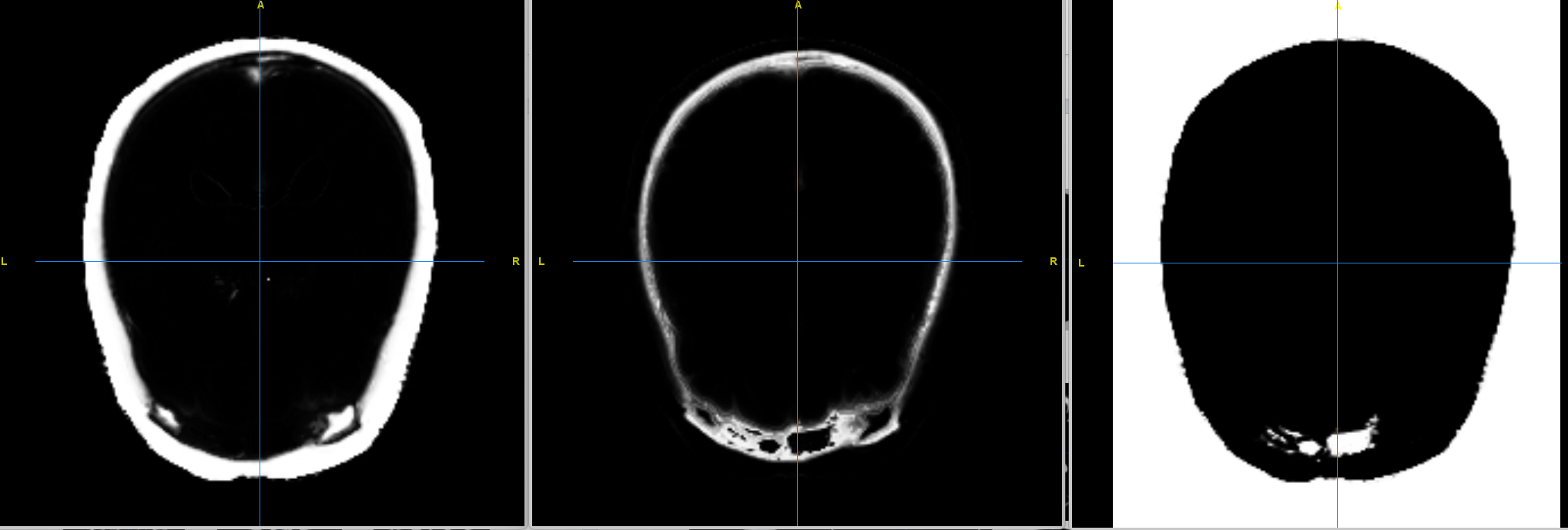
TissModulate 0-5



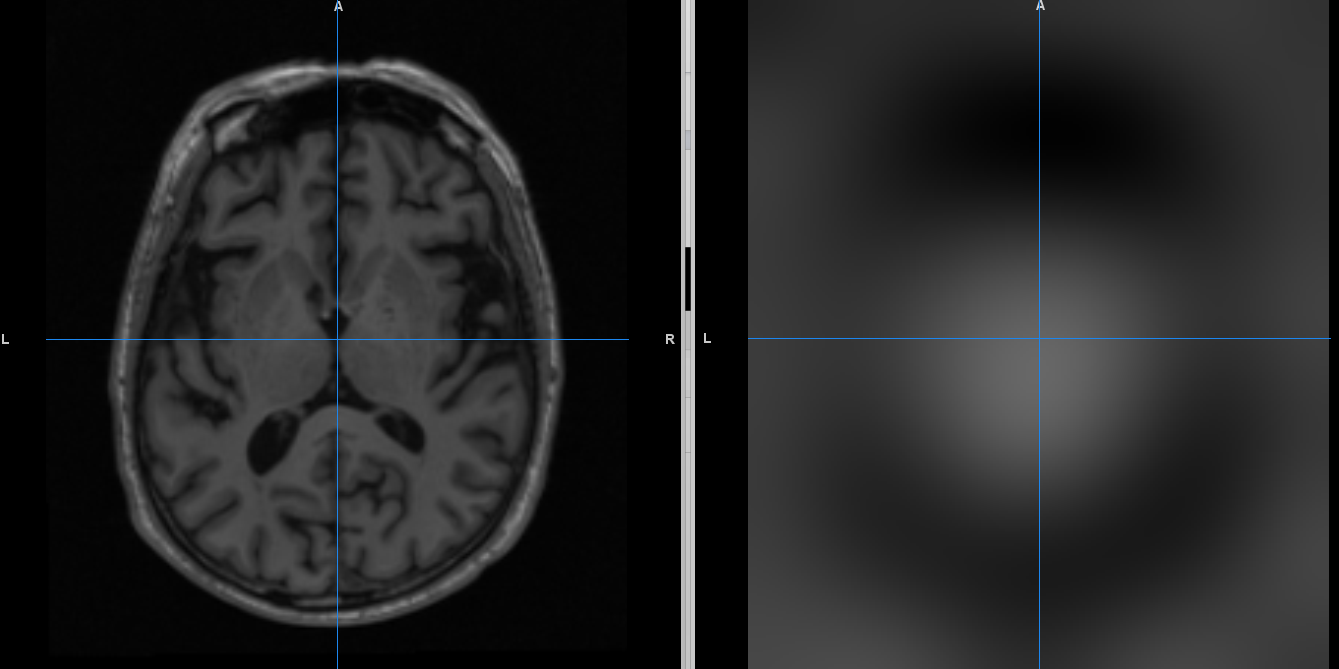


Tissnative0-5

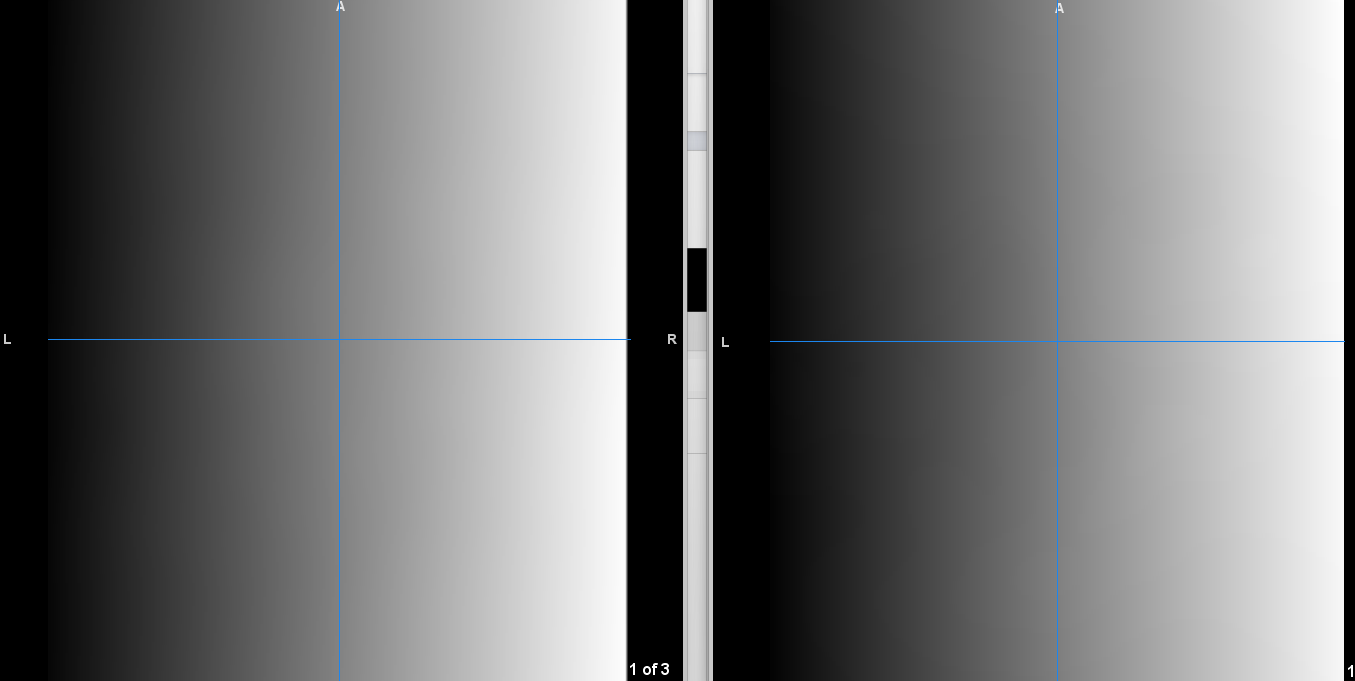




BiasCorrected BiasField



FiledForwared FieldInverse



文件夹说明:

Eigen-3.4.0: Eigen依赖包

ITK-prefix:ITK头文件

Sitk:SITK头文件

Libjson:jsonlibjson头文件

SPMSRC:SPM原生C代码

Struct2x：Struct2x序列化库

TPM++ Read Me

TPM++ is a C++ rewrite of the Tissue Probability Map in SPM12. It uses Eigen for matrix operations to replace the Matlab framework and uses SimpleITK as the image library. The image format and coordinate space are consistent with ITK. TPM++ is based on the GPL v2 license. TPM++ produces the same results as SPM12, but its computational performance is N times faster than SPM12, where N is the number of physical CPU cores.

By default, TPM++ does not use hyper-threading, as hyper-threading can no longer provide high performance.

TPM++ provides a command-line program to calculate Wrap and perform image segmentation: The command is as follows:

***TPM++.exe -in image.nii -tpm D:/TPM.nii -out D:/\_out.nii -json D:/tpminput.json -AffineOnly 0***

Parameter description:

-in: Input image path

-tpm: TPM template path

-out: Output file path, the actual output file will add a suffix to this path:

\_FieldInverse: Inverse deformation field

\_FieldForward: Forward deformation field

\_BiasCorrected: After bias correction

\_BiasField: Bias field

\_TissNative: Segmentation result (in original space)

\_TissImported: Segmentation result (in Dartel space)

\_TissModulate: Segmentation result (in imported space, after modulation)

\_TissUnModulate: Segmentation result (in imported space, before modulation)

-AffineOnly: //When set to 1, only the affine transformation is calculated and the affine matrix is output.

-json: The json file that records all parameters of TPM operation.

The TpmInput.json file is a basic parameter case with a sample number of 3 and divided into 6 tissue types.

Jsons sample：

{

// "Settings for whether to generate related TPM for each tissue"

    "m\_tissues": [

        {

            "m\_ngaus": 1,

            "m\_NativeTissue\_NativeSpace": true,

            "m\_NativeTissue\_DartelImported": true,

            "m\_WarpTissue\_Modulated": true,

            "m\_WarpTissue\_UnModulated": true

        },

        {

            "m\_ngaus": 1,

            "m\_NativeTissue\_NativeSpace": true,

            "m\_NativeTissue\_DartelImported": true,

            "m\_WarpTissue\_Modulated": true,

            "m\_WarpTissue\_UnModulated": true

        },

        {

            "m\_ngaus": 2,

            "m\_NativeTissue\_NativeSpace": true,

            "m\_NativeTissue\_DartelImported": true,

            "m\_WarpTissue\_Modulated": true,

            "m\_WarpTissue\_UnModulated": true

        },

        {

            "m\_ngaus": 3,

            "m\_NativeTissue\_NativeSpace": true,

            "m\_NativeTissue\_DartelImported": true,

            "m\_WarpTissue\_Modulated": true,

            "m\_WarpTissue\_UnModulated": true

        },

        {

            "m\_ngaus": 4,

            "m\_NativeTissue\_NativeSpace": true,

            "m\_NativeTissue\_DartelImported": true,

            "m\_WarpTissue\_Modulated": true,

            "m\_WarpTissue\_UnModulated": true

        },

        {

            "m\_ngaus": 2,

            "m\_NativeTissue\_NativeSpace": true,

            "m\_NativeTissue\_DartelImported": true,

            "m\_WarpTissue\_Modulated": true,

            "m\_WarpTissue\_UnModulated": true

        }

    ],

    "m\_Channels": [// Settings for each input file (channel)

        {

            "m\_Bisareg": 0.001000,

            "m\_Biasfwhm": 60.000000,

            "m\_biascorrect": true,

            "m\_biasfield": true

        }

    ],

    "m\_nsample": 3,//sample count

    "m\_fwhm": 0.000000,//smooth

    "m\_reg": [//Regularization

        0.000000,

        0.001000,

        0.500000,

        0.050000,

        0.200000

    ],

"m\_lkp": [//Gaussian function lookup table, sTpmInputs(sTissues &tss, sChannels &Channels) constructor can calculate this table

  1.000000,

        2.000000,

        3.000000,

        3.000000,

        4.000000,

        4.000000,

        4.000000,

        5.000000,

        5.000000,

        5.000000,

        5.000000,

        6.000000,

        6.000000

    ],

"m\_ptype": 0,

/\*Prior type：

enum PriorsType

{

    PriorsType\_mni = 0,

    PriorsType\_imni,

    PriorsType\_rigid,

    PriorsType\_subject,

    PriorsType\_eastern,

    PriorsType\_none = 0xffffffff

};\*/

    "m\_dFieldInverse": true,

    "m\_dFieldForward": true,

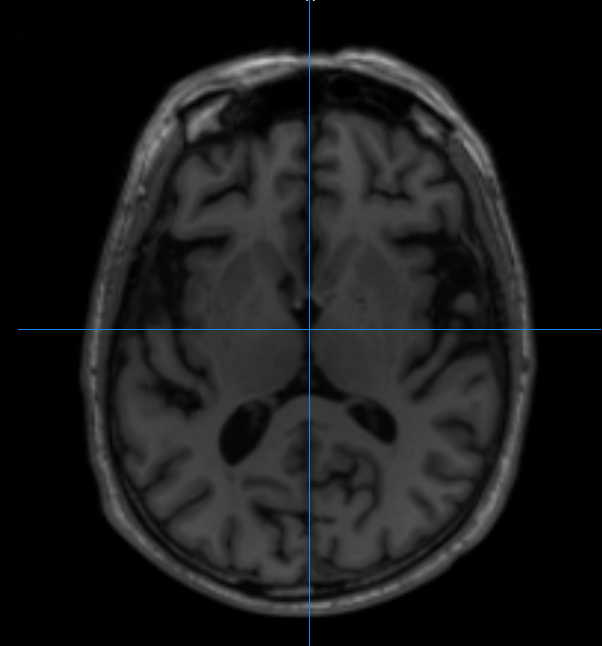
    "m\_Mrf": false,

    "m\_Cleanup": 1

}

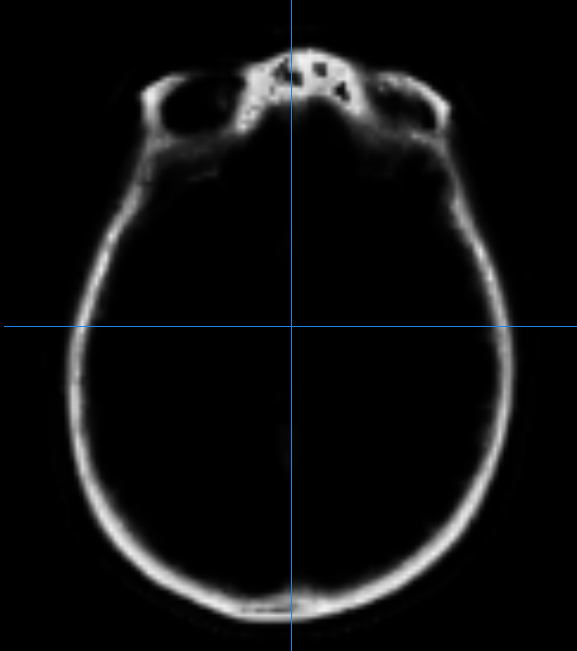
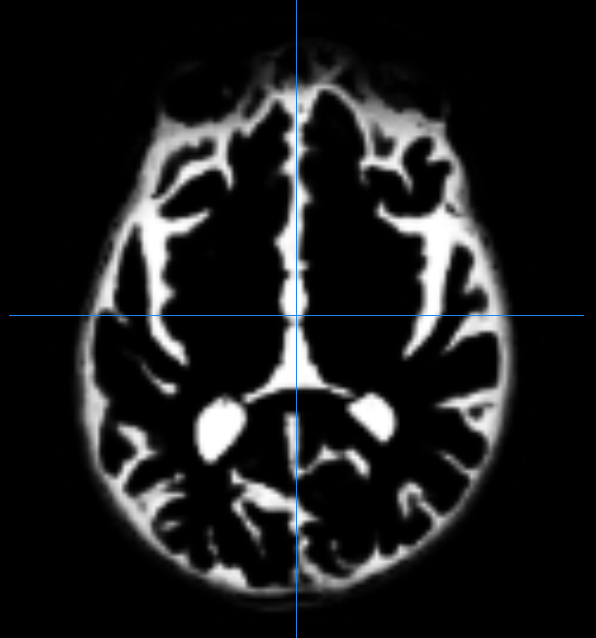
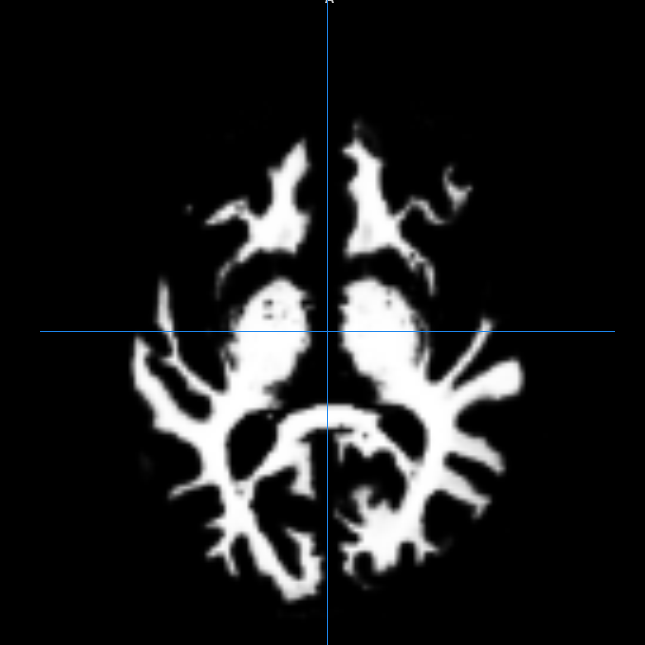
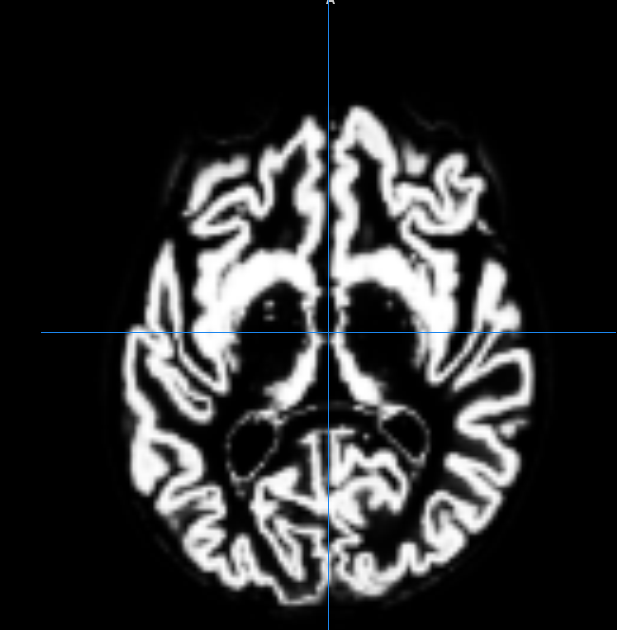
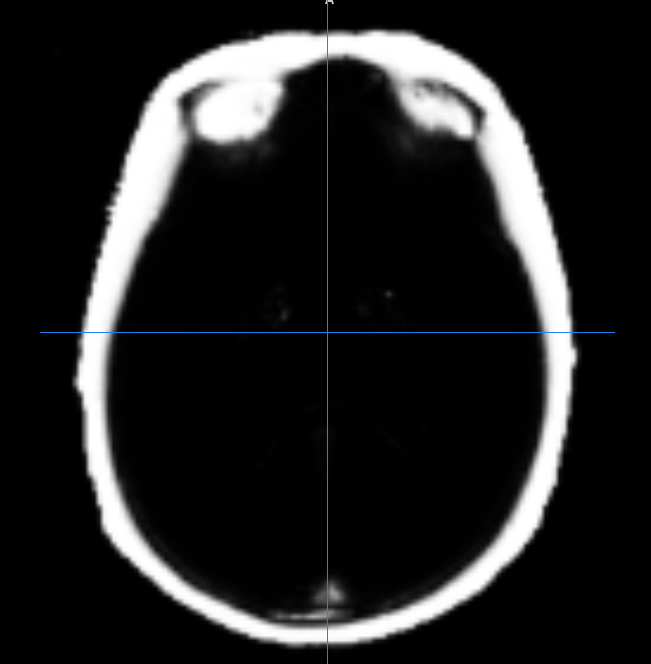
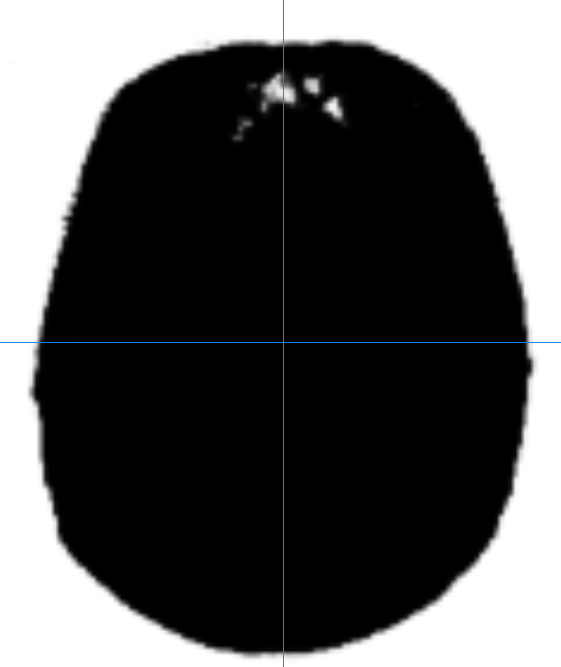
Test case：

Input: A MRI brain image file，TPM.nii file provided by SPM12

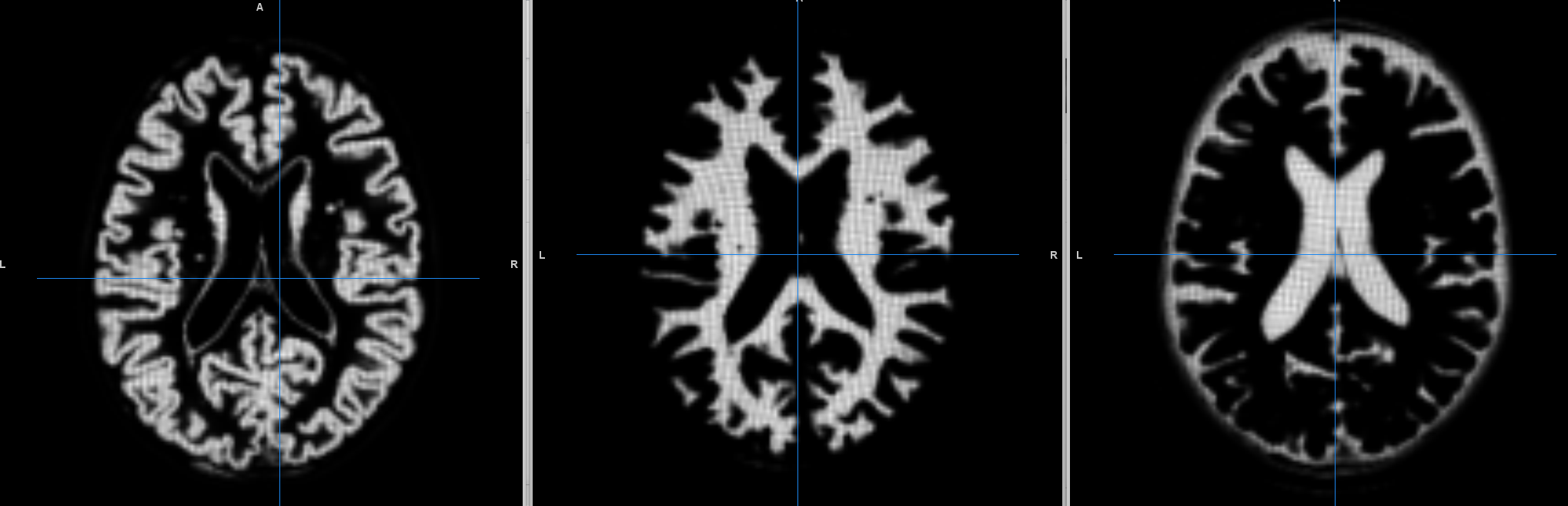


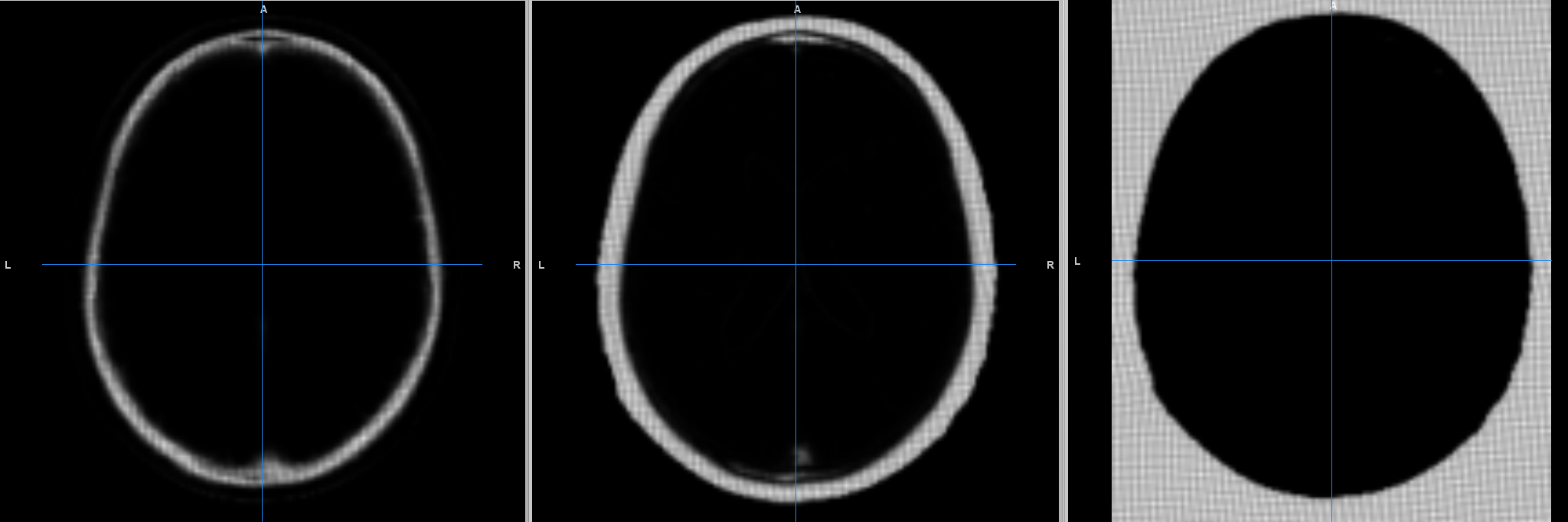
输出：

TissImported 0-5

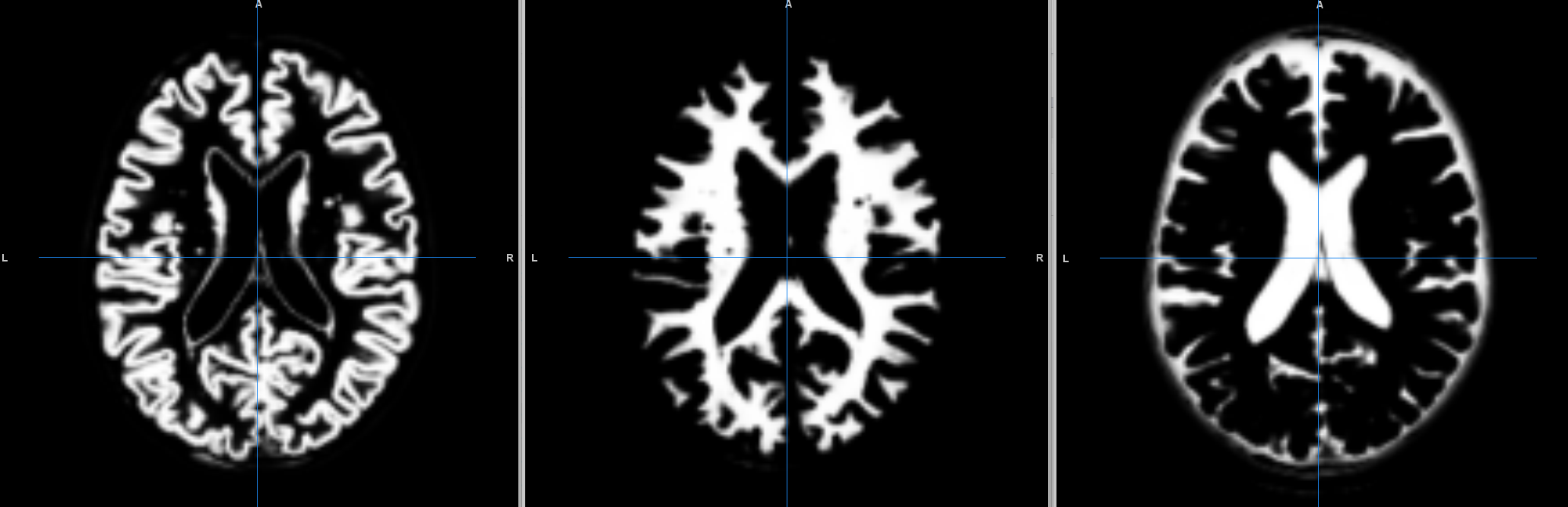
  

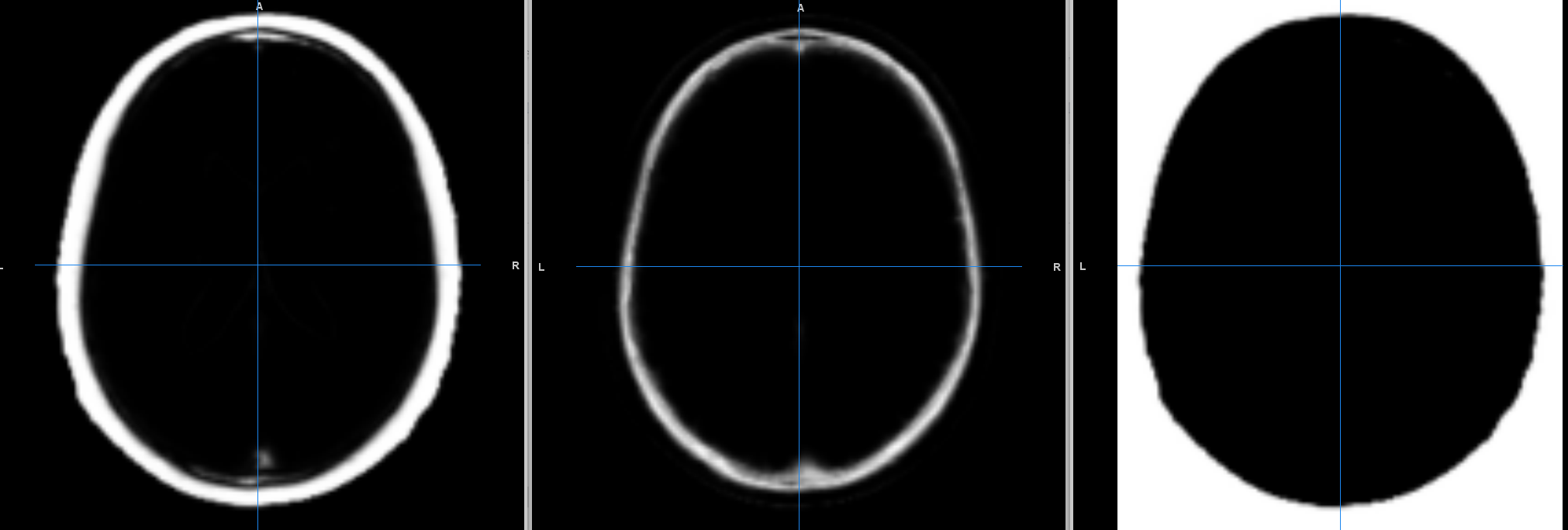
TissUnModulate 0-5



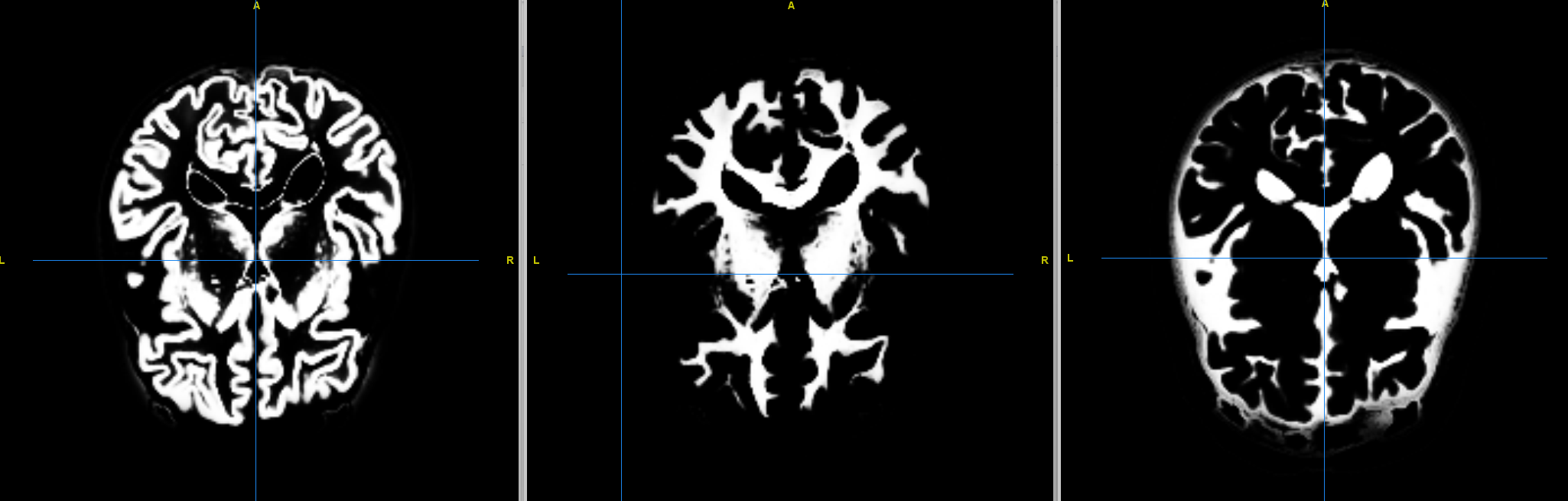


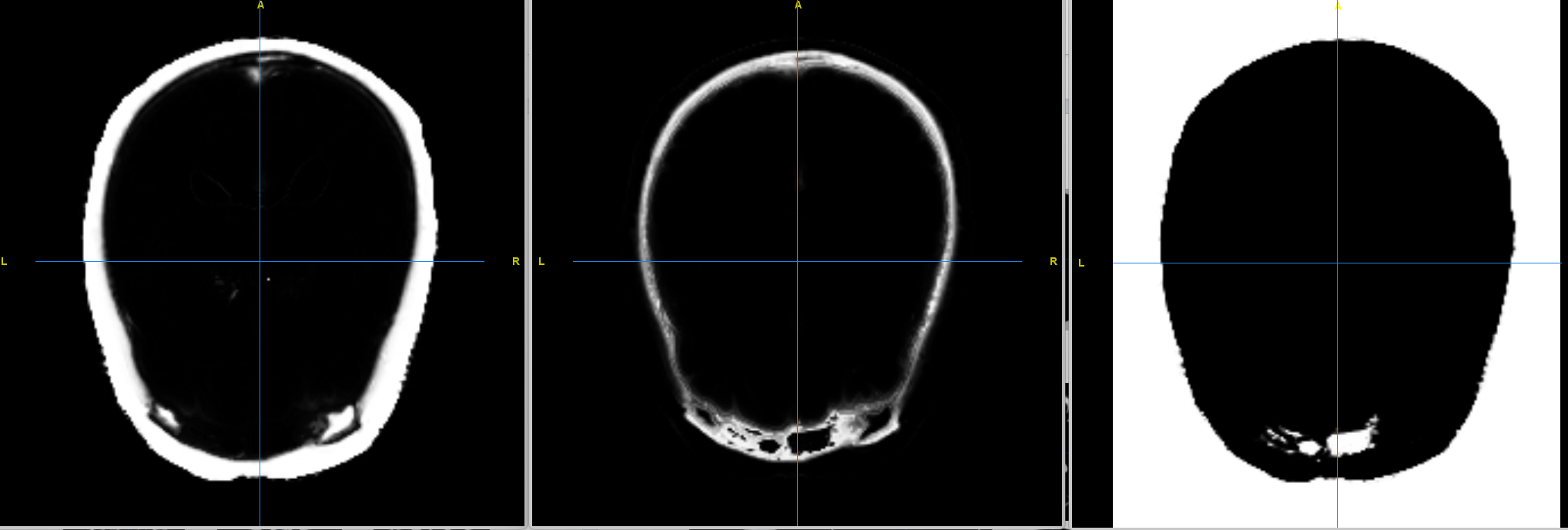
TissModulate 0-5



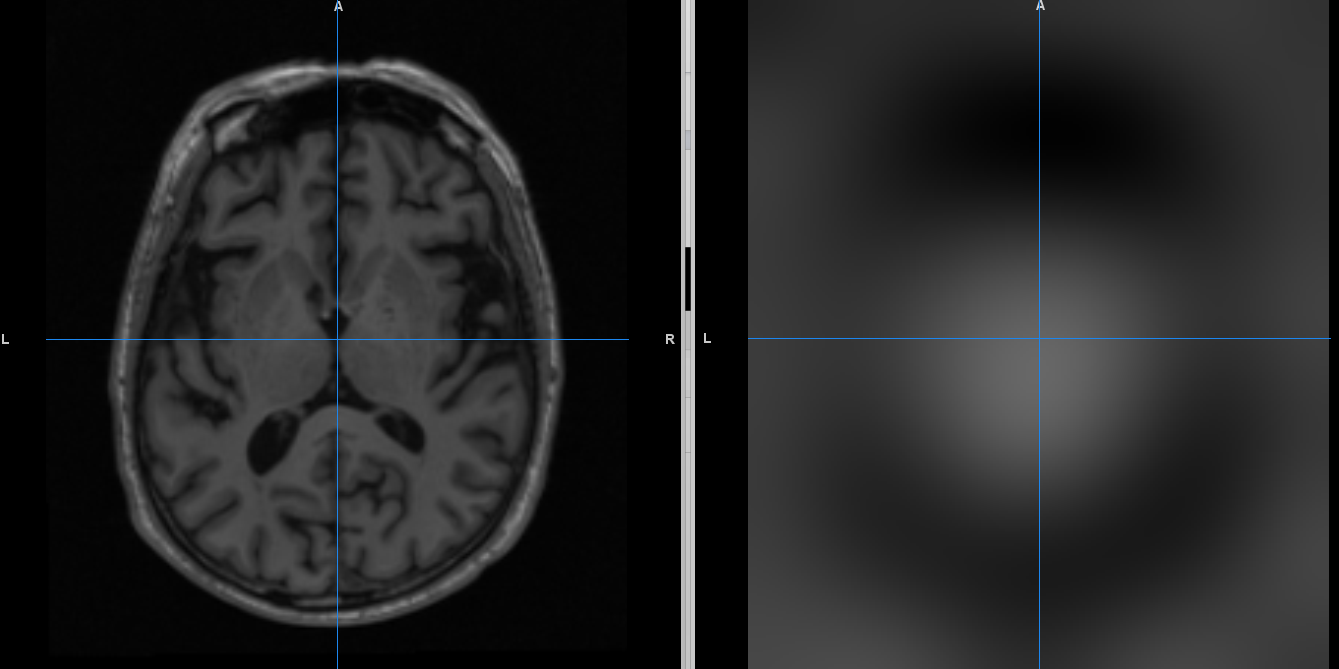


Tissnative0-5

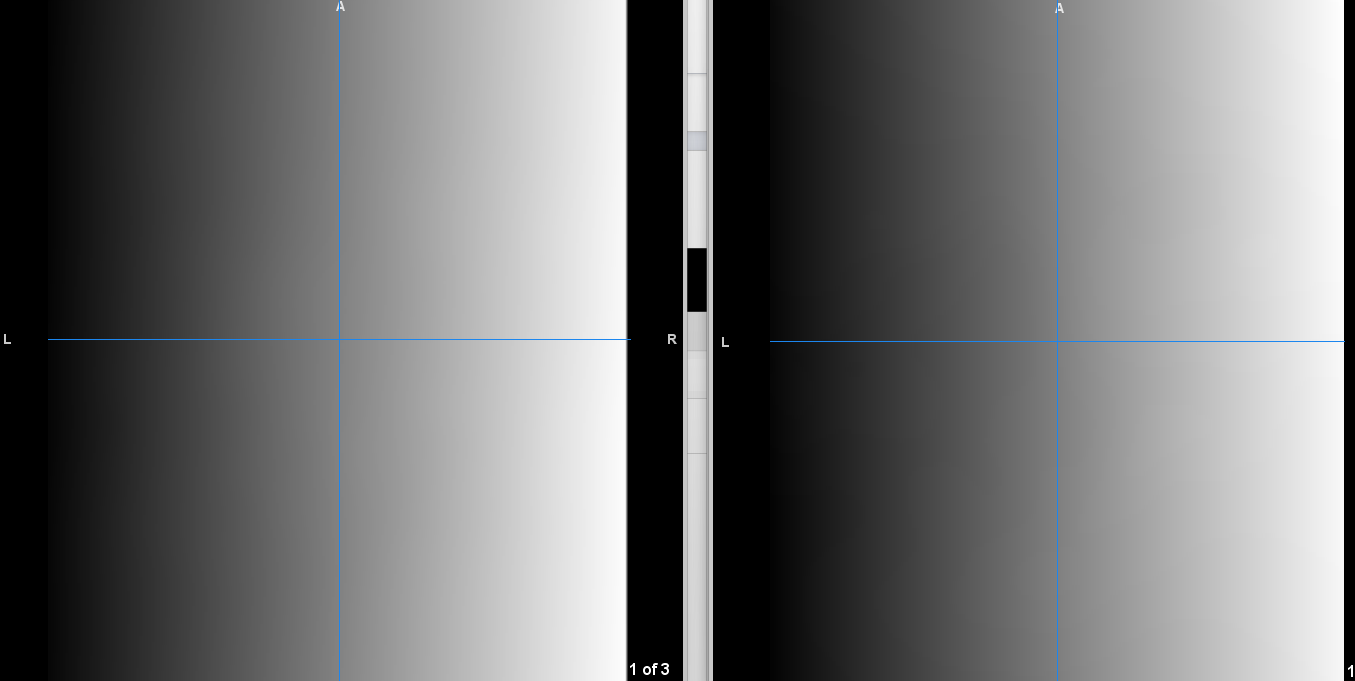




BiasCorrected BiasField



FiledForwared FieldInverse



Folder description:

Eigen-3.4.0: Eigen dependency package

ITK-prefix: ITK header files

Sitk: SITK header files

Libjson: jsonlibjson header files

SPMSRC: SPM native C code

Struct2x: Struct2x serialization library