

## 动脉自旋标记灌注磁共振成像信号处理工 具包(ASLtbx)

(2012-05-24)

<https://cfn.upenn.edu/~zewang/ASLtbx.php>

王 泽

宾夕法尼亚大学佩雷尔曼医学院精神病学系和放射系

[zewang@mail.med.upenn.edu](mailto:zewang@mail.med.upenn.edu)

[redhatw@gmail.com](mailto:redhatw@gmail.com)

中文版首稿由杜妮翻译而成

## CONTENT

<b>1. 简介</b> .....	<b>3</b>
1.1 理论和背景 .....	3
1.2 软件及硬件要求.....	4
1.3 图像格式.....	4
1.4 使用协议.....	4
<b>2. 安装 ASLTBX</b> .....	<b>4</b>
2.1 下载 ASLTBX .....	4
2.2 解压缩以及安装 ASLTBX .....	6
2.3 MATLAB 中添加 ASLTBX 搜索路径.....	6
2.4 样本数据.....	6
2.4.1 数据结构及参数.....	6
2.4.2 功能性刺激.....	6
2.4.3 使用脚本文件批处理数据.....	6
<b>3. 使用 SPM GUI 对 ASL 数据进行预处理</b> .....	<b>7</b>
<b>3.1 图像复位</b> .....	<b>7</b>
3.1.1 SPM GUI 中设置或选择原始图像 .....	7
3.1.1.1 复位 .....	9
3.1.1.2 确定 AC.....	9
3.1.1.3 重设图像原点.....	9
3.2 头动校正 (新) .....	9
3.3 功能项与结构项配准.....	9
3.4 平滑.....	9
<b>4. 使用脚本文件处理 ASL 数据</b> .....	<b>9</b>
4.1 流程文件和数据设置文件.....	9
4.2 复位以及设置原点.....	10
4.3 头动校正.....	10
4.4 配准.....	10
4.5 平滑.....	10
4.6 去除颅外体素, 生成 MASK 文件.....	10
4.7 计算 CBF.....	10
4.8 单被试广义线性模型.....	11
4.9 组分析.....	12
<b>5. ASLTBX 计算 CBF</b> .....	<b>13</b>
5.1 GUI 模式.....	13
5.2 命令行.....	16
5.3 ASL 数据采集参数设置.....	23
<b>6 致谢</b> .....	<b>24</b>

## 1. 简介

本手册是一个关于动脉自旋标记灌注磁共振成像数据处理工具箱 (ASLtbx) 的简短的使用指南<sup>1</sup>。该工具箱是基于 MATLAB 和 SPM 来处理 ASL 数据, 包括脉冲 ASL 数据, 连续 ASL 数据以及伪连续 ASL 数据的工具包<sup>2</sup>。所有学术用户都可以免费使用, 在 <https://cfn.upenn.edu/~zewang/> 可以下载获得 (包含 GPL 许可证)。每一个改进的版本都包含了原始的 GPL 许可证以及头文件。同样可以下载得到的还有样本数据, 包括静息态 ASL 数据和用户自定义的功能 ASL 数据。没有宾夕法尼亚大学的正式许可, ASLtbx 以及样本数据都严禁商用。基于本数据包做成的产品, 我们 (包括作者和宾夕法尼亚大学, 下同) 不承担任何责任。网站上提供的样本数据, 不提供图像的参考或标准, 血流量的测量以及任何方面的结果。而那些使用本数据处理工具包得到的结果以及对数据的解释我们也不承担任何责任。我们只提供原始的代码以及数据。

使用 ASLtbx 之前, 用户需要掌握 fMRI 和 ASL-fMRI 的基本概念, 并且具备 Matlab 的基本编程技能。如果使用了该工具包, 请引用这个工具包和相关的文献。

针对那些仅对脑血流量测定感兴趣的, 第三章和第四章可以跳过。

### 1.1 理论和背景

动脉自旋标记(ASL)灌注磁共振成像是一种通过对动脉血作磁标记作为内源性示踪剂来测量脑血流量的无创技术[1, 2]。ASL 的方法机制与 PET 和 SPET 扫描是类似的[3]。基于血流的标记方法, 可以大致将 ASL 技术分为脉冲 ASL (PASL) [4-7] 和连续 ASL (CASL) [1, 2, 8-10]。CASL 的优势在于当生理噪声比较强时, 它的信噪比比较高[11], 而 PASL 的优势在于它容易实现。目前 ASL 标注最先进的技术是伪连续 ASL (pCASL) [10]。ASL 中, 灌注信号理论上是由成对的标记对照图像的对比确定的。

ASLtbx 包括了处理 ASL 数据的处理步骤。最早的版本发表在下面的文章中:

- 1) Ze Wang, Geoffrey Aguirre, Hengyi Rao, JiongJiong Wang, Anna R. Childress, John A. Detre, Empirical ASL data analysis using an ASL data processing toolbox: ASLtbx, Magnetic Resonance Imaging, 2008, 26(2):261-9.

后面的加强版本:

- 2) William T. Hu\*, Ze Wang\*, Virginia M.-Y. Lee, John Q. Trojanowski, John Detre, Murray Grossman, Distinct Cerebral Perfusion Patterns in FTLN and AD, Neurology, 2010 Sep 7;75(10):881-8. (contributed equally).
- 3) Ze Wang, Improving Cerebral Blood Flow Quantification for Arterial Spin Labeled Perfusion MRI by Removing Residual Motion Artifacts and Global Signal Fluctuations Magnetic Resonance Imaging, Magnetic Resonance Imaging 30(10):1409-15, 2012.

最近的一个摘要, 关注各种量化方法的用户可能会感兴趣:

- 4) Yufen Chen, Ze Wang, John A. Detre, Impact of equilibrium magnetization of blood on ASL

---

<sup>1</sup> 本手册第一版由李正军博士在 2009 年起草完成。

<sup>2</sup> 所有的参考文献都可以在 pubmed 数据库以及代码中找到。下一版本会将参考文献整理为列表。

quantification, ISMRM 2011 annual meeting, page: 300. (paper in submission)

ASLtbx 以 Matlab m 脚本文件的形式执行。它在一定程度上是基于 SPM 的。SPM 是 Matlab 中关于大脑成像可视化以及分析的软件包，可以在英国伦敦大学学院免费下载(<http://www.fil.ion.ucl.ac.uk/spm/>)。目前测量脑血流量的函数为: asl\_perf\_subtract。尽管我们用 C++将其变成了一个独立的部件，但还是使用了 SPM 中的输入输出函数。

## 1.2 软件及硬件要求

ASLtbx 需要在 MATLAB (The MathWork, Inc., Natick, MA) 5.3 或以上的版本下运行。由于需要 SPM 中的一些函数，因此 SPM (SPM5 支持运行，SPM8 和 SPM12 比较好，2013 年起不再支持 SPM9 和 SPM2) 也需要安装并添加搜索路径。具体操作请登录 SPM 的网站。考虑到数据的大小，内存需要在 256MB 以上。上述为运行 ASLtbx 所有的硬件要求。

## 1.3 图像格式

当前版本支持 3D 和 4D NIfTI 图像格式以及 3D Analyze 图像格式。

## 1.4 使用协议

关于 ASLtbx，宾夕法尼亚大学保留所有权利。ASLtbx 对于学术用户来说都是免费使用的。用户可以根据通用公共许可证 (GUN, 免费软件基金会发布) 的条款对 ASLtbx 进行修改，无论是版本 3 还是之后的版本。商业以及工业用户需要使用 ASLtbx，请联系宾大以及作者。

希望 ASLtbx 对您有所帮助，但我们不提供任何担保，不提供商品性能保证以及特定用途的适用性。详细信息请参考通用公共许可证。

下载 ASLtbx 时，会有一份通用公共许可证 (已经附录在文件头中)，如果没有请参考 <<http://www.gnu.org/licenses/>>.

## 2. 安装 ASLtbx

### 2.1 下载 ASLtbx

ASLtbx 可以在作者的网站上下载 (<https://cfn.upenn.edu/~zewang>, 图 2.1 所示)。下载下来是一个名为“ASLtbx.zip”的压缩文件。样本数据可以按照下面阐述的步骤在相同的网站上下载得到。请占用您 1 分钟左右的时间填写好下面的表格(图 2.2)，所填写的信息都会被保密。我们将只会在软件有较大更新时给您自动发送邮件。**请不要在注册框中填写虚假信息。**



图 2.1 下载页面示意图

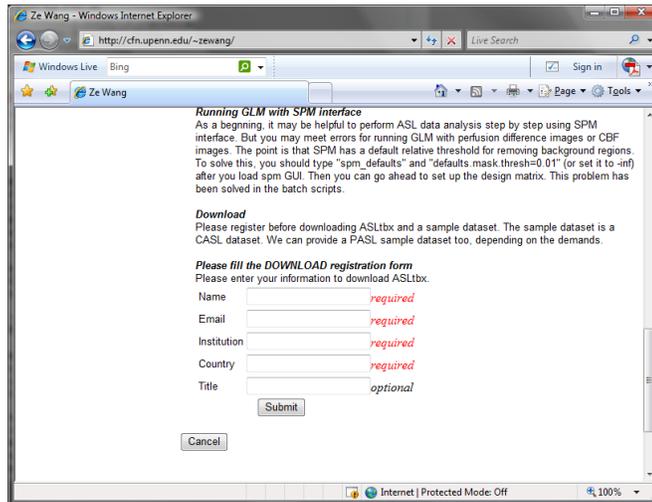


图 2.2 注册示意图

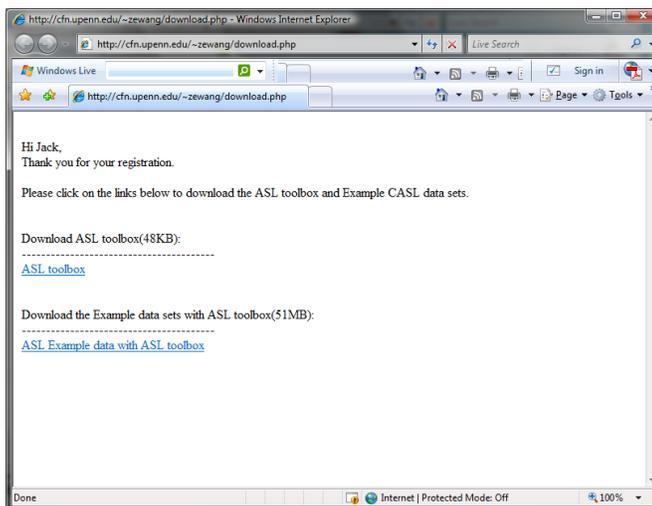


图 2.3 下载软件几样本页面示意图

## 2.2 解压缩以及安装 ASLtbx

将压缩文件复制到某一路径下，如：“Z:\ASL”。解压缩并将解压缩之后的所有.m 文件复制到某一路径下，如：“Z:\ASL\ASLtbx”。

## 2.3 Matlab 中添加 ASLtbx 搜索路径

假设工具包安装在 Z:\ASL\ASLtbx，在 Matlab 命令窗口中，通过输入：

```
addpath Z:\ASL\ASLtbx
```

来添加 ASL 搜索路径。

## 2.4 样本数据

我们提供了 3 个样本数据：一个是 CASL fMRI 数据，一个是 PASL fMRI 数据，最后一个是 pCASL fMRI 数据。每个数据都有预设脚本显示采集参数和处理设置。

下面的说明都是针对 CASL fMRI 数据的。分析大脑状态的 ASL 数据，比如长视频刺激条件与中性条件的对比，也可以使用相似的处理步骤。处理静息态 ASL 数据相对来说需要预处理的步骤少一些。更多细节请参考相关的脚本文件和数据。

### 2.4.1 数据结构及参数

样本数据可在下载 ASLtbx 工具包的页面下载得到。首先，将数据解压缩到一个新的路径下（如：“Z:\ASL\ASL\_Example\_data”）然后将解压缩的内容复制到新的工作路径下。CASL fMRI 样本数据文件包括四个子文件夹：“batch\_scripts”，“sub1”，“sub2”和“sub3”。第一个文件夹中是一些针对该样本数据定义的脚本文件，剩余的三个文件夹中为三个被试的样本数据。每个被试的文件夹中都有一个名为“anat\_anlz”的文件夹，里面是结构图像；还有一个名为“func\_anlz”的文件夹，里面为功能图像。

结构图像为 3D MPRAGE 序列扫描获得，扫描参数为：FOV=250 mm, TR/TE=1620/3 ms, 192×256 matrix, 160 slices, Slice thickness = 1 mm。功能图像为 CASL 图像序列，磁场强度为 3T。扫描参数为：FOV=22 cm, 64×64×12 matrix, Bandwidth=3 kHz/pixel, Flip angle=90°, TR=3 s, TE=17 ms, Slice thickness=6 mm, Inter-slice space=1.5 mm, Labeling time=1.6 s, Post label delay time=800 ms。每个被试包含 72 个标签-对照图像对。

### 2.4.2 功能性刺激

功能性试验为组块设计，包含两个交叉条件。条件开启时，会呈现一段视觉刺激，具体为一段频率为 8Hz 的黑白棋转换，呈现时间为 72 秒。被试需要根据提示学习并做出反应，在视觉刺激时做手指敲击任务。条件关闭时，只呈现黑屏，屏幕中央为十字。

### 2.4.3 使用脚本文件批处理数据

要分析数据时，可以简单的打开 Matlab，将工作目录切换到“batch\_scripts”文件夹，然后在 Matlab 的命令窗口输入“batch\_run”回车即可。详细的介绍参见第 4 节。

注：如果是自己的数据，建议将它们整理成与上述一致，至少数据结构一致。或者也可以修改对应的脚本文件。

### 3. 使用 SPM GUI 对 ASL 数据进行预处理

ASLtbx 可以处理那些相同方向，相同提速大小的图像。因此，建议在计算测量脑血流量之前对 ASL 数据进行预处理。

#### 3.1 图像复位

(如果 SPM 的版本为 SPM8 及以上版本，则不用做这一步处理。值得注意的是，最新的 SPM8 是将图像复位，原点作为中心之后再做图像配准，并不需要手动设置在 AC-PC 线上，这一点很方便。因此，如果你想省略掉这一繁琐的步骤的话，就需要将你的 SPM8 升级一下。不过，复位原始图像还是必须的。FSL 或者 Afni 则完全不需要这一步。)

SPM 中是将原点设置在 AC-PC 线上。虽然 SPM8 中并不需要这一步骤，但设置原点在处理 ASL 数据时是必须的，因为我们注意到当设置原点为中心时会有显著的差异 (spm5 中发现这一点)。考虑到同一个数据在发布之前可能已经处理过很多次，所以把 AC-PC 线的坐标记录下来是很方便的。这样这个数据可以在任意时间都可以重新处理，并且如果处理流程不变的话，会和之前得到相同的结果。下面的处理步骤针对不同类型的图像都适用 (包括结构项，功能项和 DTI 图像等)。

##### 3.1.1 SPM GUI 中设置或选择原始图像

点击“Display”按钮打开结构项或者功能项的第一幅图像。

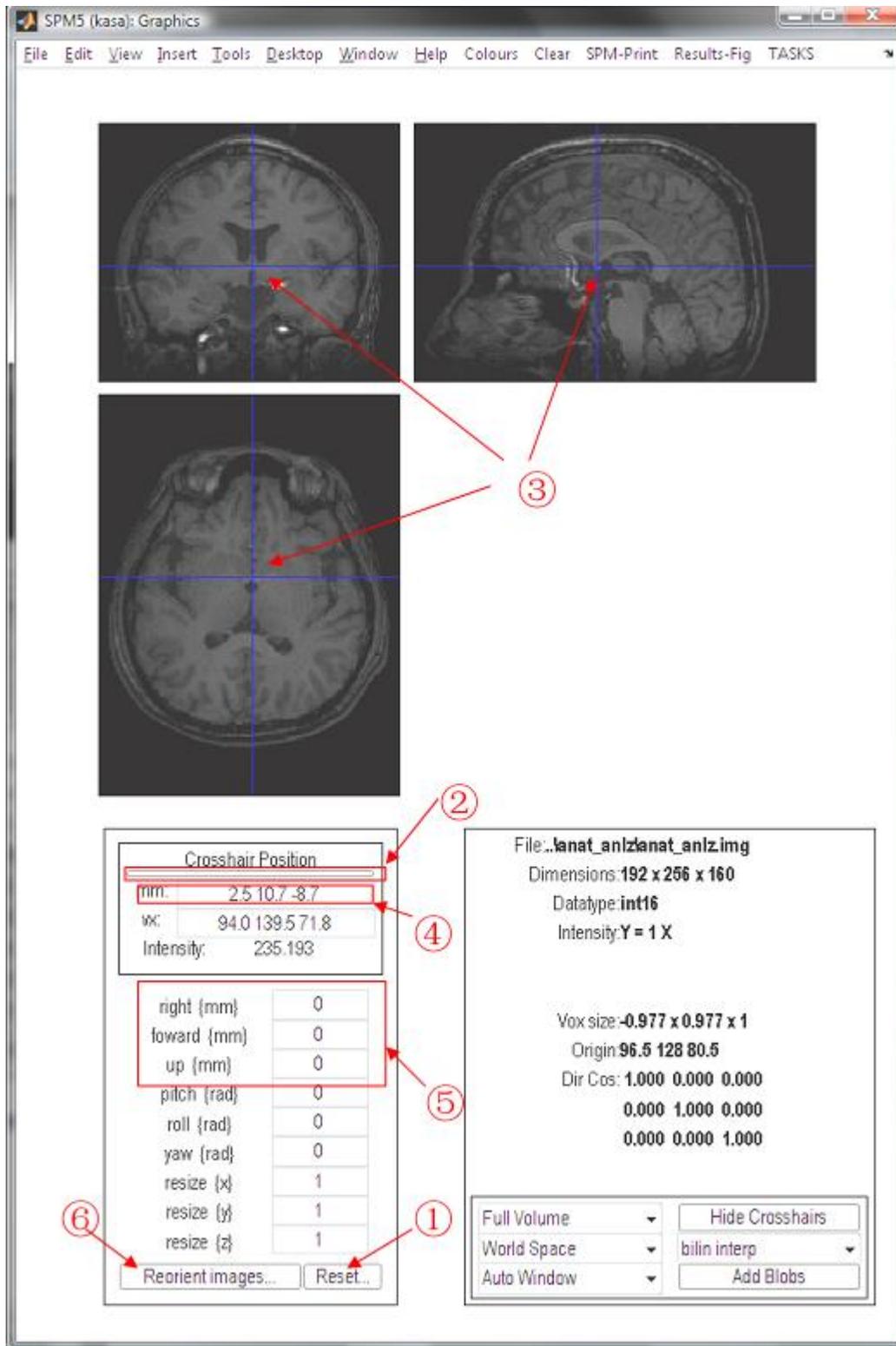


图 3.1 使用 SPM GUI 图像复位

### 3.1.1.1 复位

点击“Reset..”按钮（图 3.1 ①），选择要处理的图像，包括结构项和功能项。这一步包括记录图像的体素大小以及将原始图像设置为中心并设置所有的旋转为 0。

### 3.1.1.2 确定 AC

打开一幅复位过的图像。点击水平横条（图 3.1 ②），通过观察三幅图的交叉点来核实原点是否在中央。如果没有，重复 3.1.1.1 步骤。

在三幅图上点击直到交叉点在 AC 位置。（图 3.1 ③）。

有关确定 AC 的详细解释见：<http://imaging.mrc-cbu.cam.ac.uk/imaging/FindingCommissures>。

当前图像的中心（复位之后的原点）与 AC（交叉点位置）之间的位移会在“mm”（图 3.1 ④）这一栏显示出来。

### 3.1.1.3 重设图像原点

当你把光标移动到所需的原点位置时，与当前图像原点（此例中为图像的中心）之间的位移就会显示出来。这个时候重新设置图像的原点位置，只需要输入 x, y, z 坐标值（“mm”这一栏显示）的相反值到“right”，“forward”，“up”（图 3.1 ⑤）对应的栏中即可。

然后点击“Reorient images...”按钮（图 3.1 ⑥），选择需要新的原点的图像。这一操作在 SPM5 以上版本中，会修改图像头文件中的仿射变换矩阵。

## 3.2 头动校正 (新)

ASL 数据的头动校正，控制组和标签组数据需要分开来做[12]。这样就需要将两个组额外配准一下。我们将 SPM 中提供的头动校正函数进行了修改，避免了将 zig-zagged 自旋标记模式作为额外头动的情况[13]。具体细节请参阅脚本文件 `batch_realign.m`。

## 3.3 功能项与结构项配准

ASL 图像需要与结构图像配准之后，才能标准化到 MNI 空间（或者其他模板空间，标准空间）进行组分析。目标图像和源图像分别选择 T1 图像和 mean ASL 图像。T2 加权结构图像也可以使用。

## 3.4 平滑

在计算 CBF(脑血流量)之前，需要对 ASL 图像进行平滑操作来减少噪声。**计算完 CBF 之后可以再进行一次平滑操作，空间标准化之后如果脑内结构变化较大也可进行二次空间标准化。** 用户可以利用 SPM GUI 来做平滑，也可以用工具包中的脚本文件 `batch_smooth.m`。

## 4. 使用脚本文件处理 ASL 数据

利用 SPM GUI 也可以处理 ASL 数据，但是我们建议用脚本文件来做处理。使用脚本文件来处理样本数据，用户可以在预设置脚本文件的时候对整个数据的处理流程有所掌握。对于任意数据来说，用户都可运行脚本文件“`batch_run.m`”，它包含了所有的处理步骤。

### 4.1 流程文件和数据设置文件

文件“`batch_run.m`”中定义了 ASL 数据的处理流程，一般来说包括了：头动校正，配准，平滑，计算 CBF，标准化，模型分析，组分析等。不同的研究（例如静息态研究）需要的处理步骤不同。所有的数据参数，数据路径，以及分析的一些设置都存储在一个全局结构体“`par.m`”（`par` 意为 `parameter`，参数）中。

如果需要检验参数文件，在 Matlab 的命令窗口敲“`par`”，然后全局结构体 `PAR` 的各个值都可以通过敲对应的名称来检查（例如，敲“`PAR.subjects{:}`”会出现当前数据中所有被试的列表）。

## 4.2 复位以及设置原点

这一部分包括了三个文件。文件“`batch_reset_orientation`”将每个图像的原点设置为中心，并将所有的旋转调回 0。这会直接修改每个图像的头文件；“`batch_generateorigintable`”会产生一个表格存储图像中心与 AC-PC 线之间的相对坐标。这一步之后，你需要打开文件“`origintable.m`”填写在设置新的所需原点时得到的坐标（见图 3.1 坐标 2.5 10.7 -8.7 记录为 `PAR.subs(x).t1org = [2.5 10.7 -8.7]`）。然后，“`batch_setorigin`”会利用之前在 `origintable.m` 中记录的位移找到新的所需的原点。这会直接修改每幅图像的头文件。（注：复位的脚本文件已经更新，旧版本在处理矢状位图像上可能会有问题。）

假如之前没有运行过“`batch_reset_orientation`”的话，是不允许运行“`batch_setorigin`”的。很多用户对于这三步（复位，生成表格，填写坐标）的处理有点混淆。然而，在找到新的解决方法之前，只能这样来操作。用户可以用样本数据来多次进行这三步操作并观察图像，以达到对这几个步骤更深的理解。

## 4.3 头动校正

上述设置完成之后，运行脚本文件“`batch_realign.m`”，将 ASL 图像校正到参考图像。参考图像可以设置为第一幅图像或者平均图像。关于如何修改设置，请阅读代码的具体内容。重采样之后，会生成以“`r`”开头的图片文件（`r+`文件名）和一个平均图像（以“`mean`”开头）。

## 4.4 配准

脚本文件“`batch_coreg.m`”是将头动校正过后的 ASL 图像与每个被试的结构图像进行配准。配准过后，每个功能图像头文件中的转换矩阵会被改变。

## 4.5 平滑

脚本文件“`batch_smooth.m`”利用高斯核平滑对那些已经头动校正以及配准过的功能图像进行平滑操作以减少噪声。新生成的图像会以“`sr`”打头。

## 4.6 去除颅外体素，生成 mask 文件

步骤 4.5 操作完成后，会生成一个功能图像的平均图像，脚本文件“`batch_create_mask.m`”基于该平均图像会生成一个名为“`mask_perf_cbf.img`”的 mask 文件并保存在每个被试的功能文件夹中。这个 mask 文件是为了去除颅外体素，用户也可自己定义。

## 4.7 计算 CBF

脚本文件“`batch_perf_subtract.m`”是调用“`asl_perf_subtract`”来计算 CBF（“`asl_perf_subtract`”的详细解释见第五节）。用户自己的设置，CBF 图像序列，变量灌注信号序列以及平均 CBF 图像（“`meanCBF_*.img`”）都会生成并保存在每个被试的文件夹中。下图为样本数据中被试 2 的 CBF 平均图像。

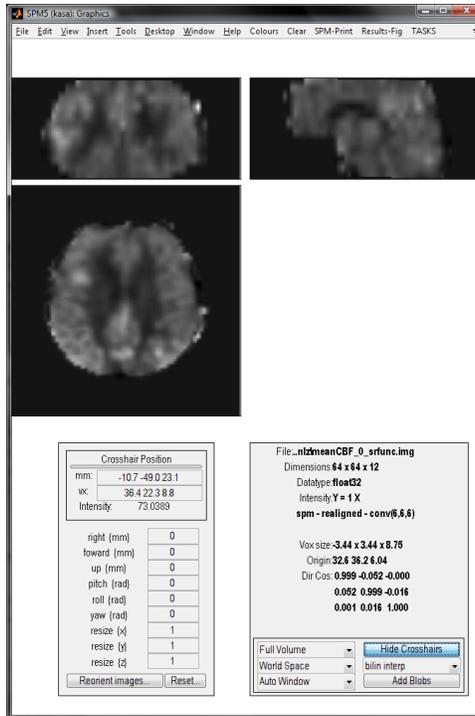


图 4.1 被试 2 的平均 CBF 示意图

#### 4.8 单被试广义线性模型

这一步是评估一下个人层面的影响，之后可以加入到整个组来做组分析，用来模拟两阶段随机效应模型，这个模型是为了血氧水平依赖功能磁共振成像技术分析而设计的。大脑状态分析例如服药前后或治疗前后或者是长视频条件与中视频条件的对比，用户也可以用这个两阶段模型。或者也可以像 PET 图像分析一样，将平均 CBF 图像加入到组方差分析中。理论上来说，这两种方法是等价的，但是利用前一种方法还可以得到每个被试的统计结果，这一点对接下来的实验计划可能很重要。缺点是利用第一种方法就必须将数据整理好顺序以便提取显著差异区域的 CBF。

脚本文件“`batch_model.m`”通过在减影图像上运用广义线性模型得到脑灌注的 tap-rest 效果。如果用户想用纯矩阵窗函数作为一般线性模型的参考函数，可以通过将 73 行 “`SPM.xBF.name='Fourier set'`” 取消注释，74 行 “`SPM.xBF.name='hrf'`” 取消注释的方式来完成设置。

脚本文件 “`batch_contrasts.m`” 运行之后会生成两个对照条件下的对照图。

运行之后生成的结果图以及 `SPM.mat` 文件，保存在每个被试文件夹下的一个名为“`glm_cbf`”的文件夹中（在“`par.m`”中定义）。下图为被试 2 的广义线性模型分析结果示意图。

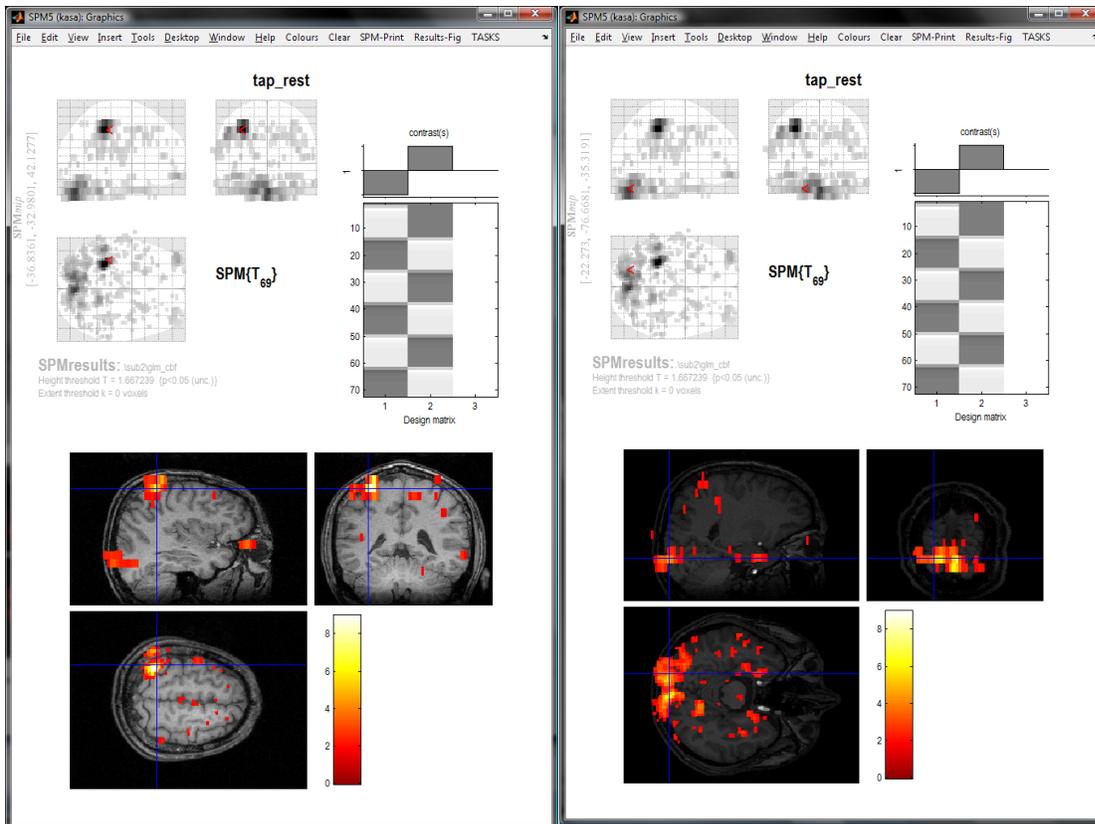


图 4.2 被试 2 的广义线性模型分析结果示意图。视觉刺激和右手敲击任务的激活图，左图为左运动皮质的激活图，右图为视觉皮质的激活图。

## 4.9 组分析

这一部分分 3 或 4 步完成。

脚本文件“[batch\\_2nd\\_cp\\_confiles.m](#)”将每个被试的结果图复制到文件夹“ASL\_Example\_data”下名为“group\_anasmallPerf\_sinc\tap\_rest”的文件夹中。

脚本文件“[batch\\_usegment\\_spm5.m](#)”利用 SPM 中统一的分割方法来将每个被试的组分析结果图标准化到 MNI 空间。标准化过后的文件名以“w”打头。

脚本文件“[batch\\_smooth\\_wconfiles.m](#)”针对由于脑内不同带来的变动问题，对标准化后的图像做平滑操作，生成文件“swcon\*.img”。

脚本文件“[batch\\_2nd\\_glm.m](#)”对标准化、平滑过后的图像进行第二次组分析，得到实验设计的组效果。结果会保存在“ASL\_Example\_data”下名为“group\_anasmallPerf\_sinc\tap\_rest\RFX”的文件夹中。下图为 CASL 手指敲击数据的组分析结果示意图。

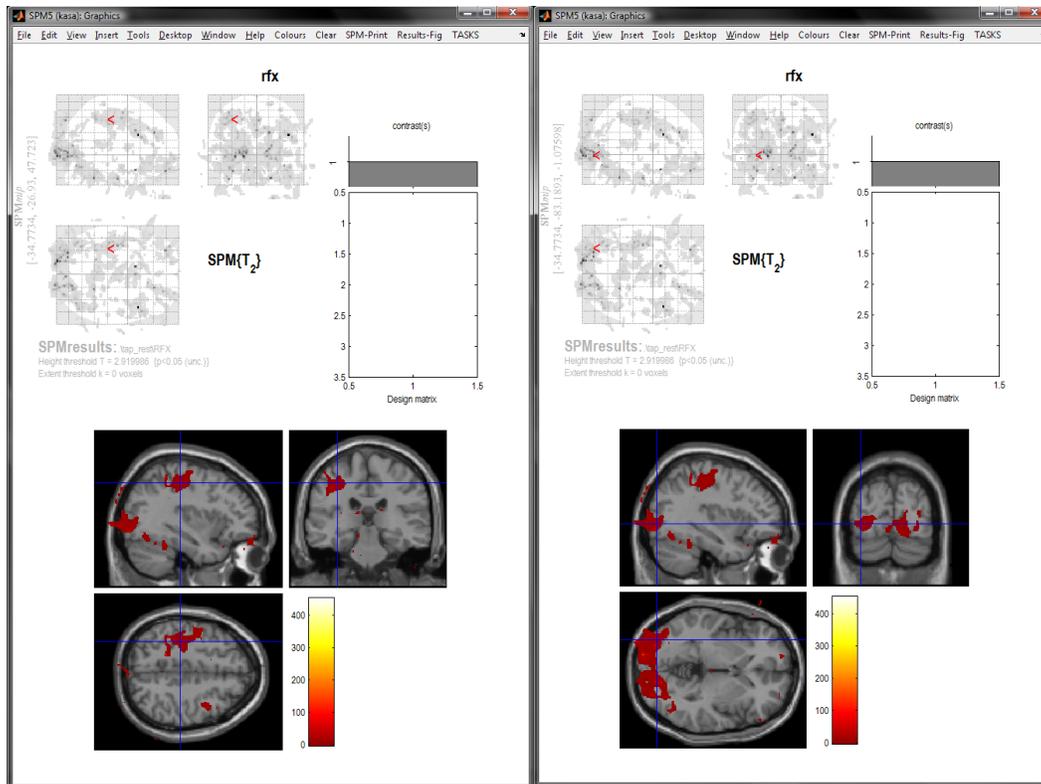


图 4.3 CASL 手指敲击数据组分析结果示意图

## 5. ASLtbx 计算 CBF

ASLtbx 中 CBF 的计算有 GUI 和运行脚本文件两种方式。计算 CBF 的模型在 [asl\\_perf\\_subtract.m](#) 详细的描述。与引用的文献中相比，在公式方面可能会有不同的地方，这主要是因为单位不同或者是有一些小的改进造成的。请引用编码中提到的文献，以及引用的文献。有关公式的问题，我（作者）很乐意跟大家通过邮件交流。前提是提问题之前这些公式以及引用的文献都详细阅读过，不然每天都会收到各种各样的有关 ASLtbx 的邮件，处理不过来。

开始之前，确保 SPM 和 ASLtbx 都已经在 Matlab 中设置过了搜索路径，详细的设置参考 1.2 节以及 2.3 节。

### 5.1 GUI 模式

在 Matlab 命令窗口，敲：

[asl\\_perf\\_subtract](#)

回车会弹出对话框窗口（如图 5.1），要求选择 ASL 图像。图像需要是预处理过的，设置的原点以及体素大小一致。目前最新的版本已经允许选择 4D Nifti 图像。

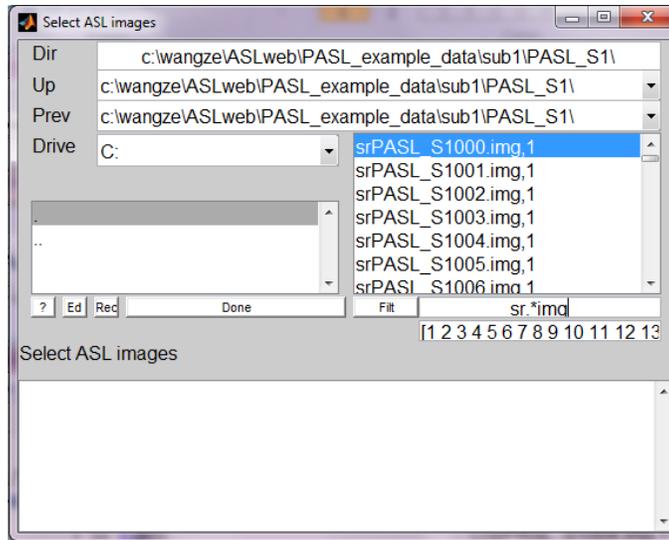


图 5.1 GUI 计算 CBF 弹出选择文件窗口示意图



图 5.2 定义 ASL 图像参数示意图

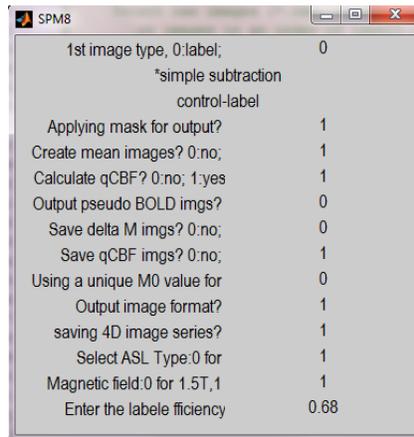


图 5.3 计算 CBF 参数设置示意图

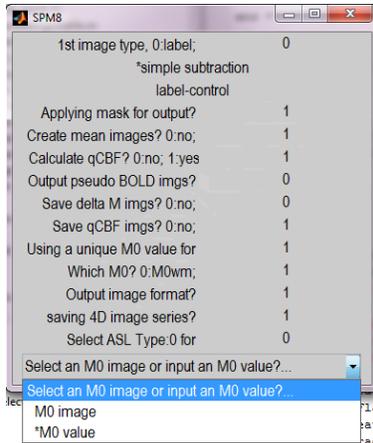


图 5.4 M0i 值选项具体内容示意图

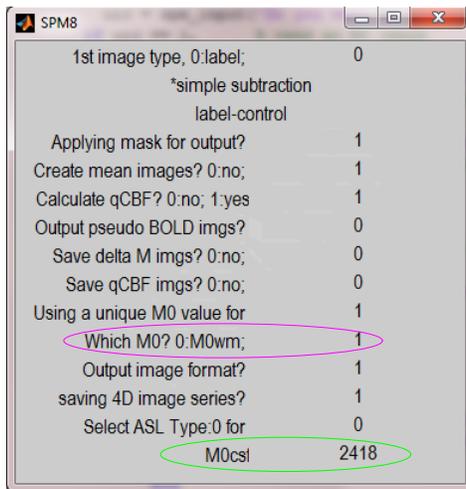


图 5.5 使用指定 M0csf 计算 CBF 的参数设置示意图

选择好源文件之后，点“Done”（如图 5.1），之后会有第二个对话框窗口弹出来（如图 5.2），需要输入 ASL 图像采集参数以及计算 CBF 需要的设置。

如果需要用指定的 M0b (M0 blood)来计算 CBF，那么问题 “Using a unique M0 value for all voxels?” 选 1。然后会继续有个问题，问是否直接指定(M0wm or M0csf) 值 (见图 5.4 底部)或者运行脚本文件从 M0 图像中得到 M0i。

如果你已经提取了 M0i (M0wm or M0csf) ，并且想利用它来计算 CBF 的话，问题 “Using a unique M0 value for all voxels?” 选择 1(见图 5.5 中紫色椭圆)，问题 “Select an M0 image or input an M0 value?”选择“M0 value” (见图 5.4)，然后就会弹出对话框，输入你的 M0i 值即可 (见图 5.5 绿色椭圆)。

选择使用指定的 M0b 来计算 CBF，并且问题“Select an M0 image or input an M0 value” 选择“M0 image” 时，会弹出对话框(见图 5.6 底部)，问是否利用提取 M0i (M0csf 或者 M0wm，取决于图 5.5 紫色椭圆中的选项) 的图像来提取 ROI，否则选择 ROI 模型文件。如果选择 “yes”，就会有个弹出窗口显示选择的 M0 图像的某一层，

而 Matlab 的命令行可以选择 M0 图像的层数。层数范围为 1 到 M0 图像的最大层数。如果发现了很好提取 ROI 的图像层，可以先在 Matlab 命令行输入 0 终止选择层数进程，然后打开提取 ROI 的弹出窗口。例如，图 5.7 中，ROI 应该提取在脑室，双击完成 ROI 的提取。

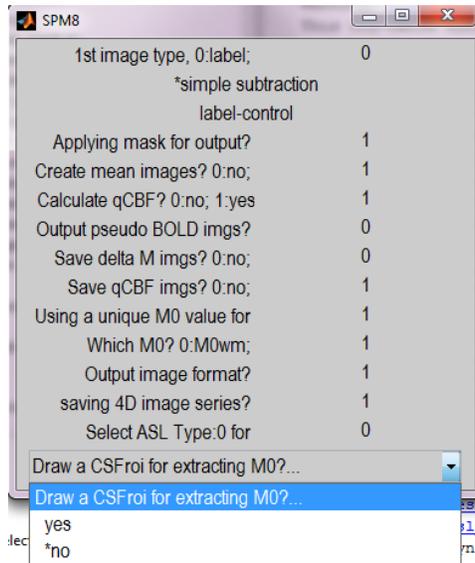


图 5.6 提取 ROI 参数设置

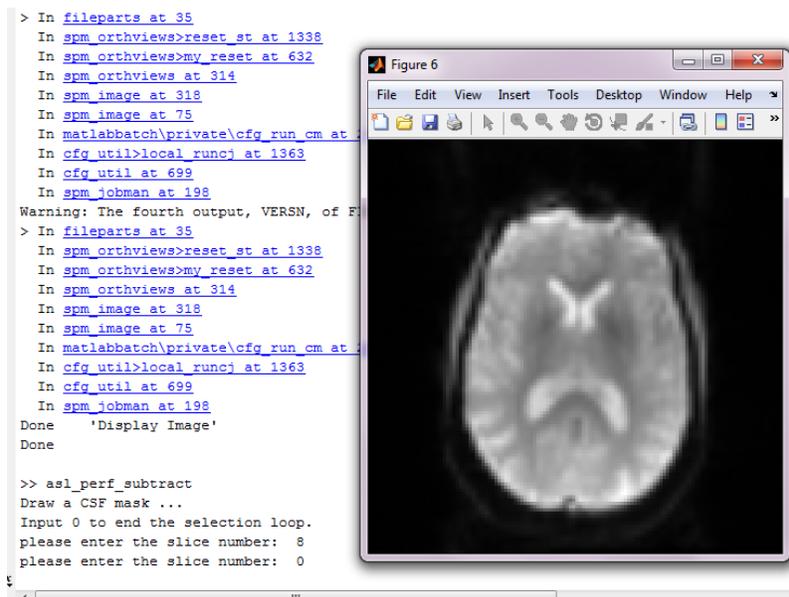


图 5.7 选择提取 ROI 的图像层数

当所有参数都设置完成之后，脚本文件“asl\_perf\_subtract”会开始计算所有指定的输出，例如灌注差异信号，平均血流信号，脑血流量，pseudo BOLD，平均灌注差异，mean BOLD，平均脑血流量等。这些输出会保存在源图像文件夹中。除此之外，还会生成一个文本文件，里面存储着总体灌注差异和脑血流量信号（详见下一节）。

## 5.2 命令行

如果知道如何使用所有的参数，那么可以不用使用 GUI 模式，直接调用主函数“asl\_perf\_subtract”即可。函数原型的定义如下：

[perfnm, glcbf] = asl\_perf\_subtract(Filename, FirstimageType, SubtractionType, SubtractionOrder, Flag, Timeshift, AslType, labeff, MagType, Labeltime, Delaytime, Slicetime, TE, M0img, M0seg, maskimg, M0csf, M0wm)

下面的说明在文件“asl\_perf\_subtract.m”开始可以找到，或者在 Matlab 命令窗口直接敲“help asl\_perf\_subtract”也可以看到。

参数:

Filename: 输入图像序列列表。3D 和 4D NifTI 格式的图像以及 3D Analyze 格式的图像都可以。这个列表可以通过 SPM 中选择文件函数得到。例如：下面的代码可以得到包含所有文件名中包含 srfunc\_anlz 的 .img 格式的图像，并赋值给变量“Filename”:

```
Filename=spm_select('ExtFPList','Z:\ASL\ASL_Example_data\sub1\func_anlz', ['^sr' 'func_anlz' '.*\.img$']);
```

FirstimageType – 整数变量，用来表征第一幅图像的类型。

– 0: 标签; 1: 对照; 宾夕法尼亚大学功能神经影像中心发布的序列中，第一幅图像为标签。

SubtractionType – 整数变量，用来表示差减法法的类别。

– 0: 简单差减法; 1: 围绕减法; 2: sinc 减法

对于对照-标签这样的序列来说，原始的图像为: (C1, L1, C2, L2, C3...), 简单差减法为: (C1-L1, C2-L2...) 围绕减法为: ((C1+C2)/2-L1, (C2+C3)/2-L2,...), sinc 减法为 (C1.5-L1, C2.5-L2...)。如果原始图像为: (L1, C1, L2, C2...), 简单差减法为: (C1-L1, C2-L2...); 围绕减法为: (C1-(L1+L2)/2, C2-(L2+L3)/2...), sinc 减法为: (C1-L1.5, C2-L2.5...)。对于标签-对照序列来说也一样。

SubtractionOrder – 整数变量，用来表示减法取向。1: 对照-标签; 0: 标签-对照。

注：确定减法取向的最好的标准为在灰质中检查 CBF 的值。如果大多数体素的 CBF 值为负数，这时就需要切换减法取向。一般来说，CASL 数据选择对照-标签，而基于 PASL 数据的 FAIR 选择标签-对照。如果应用了背景抑制，减法取向可以略过。

Flag – Flag 向量为: [MaskFlag, MeanFlag, CBFFlag, BOLDFlag, OutPerfFlag, OutCBFFlag, QuantFlag]

MaskFlag – 整数变量，表示灌注图像是否模板化。模板化是为了去除背景噪声以及没有灌注的区域。0 没有模板化; 1 模板化。

MeanFlag – 整数变量，用来表示是否生成各类平均文件。0: 不保存平均文件; 1: 生成平均文件。

CBFFlag – 1: 计算 CBF 值; 0: 不计算 CBF 值。

BOLDFlag – 1 和 0 表示是否提取 pseudo BOLD 文件。

OutPerfFlag – 是否保存灌注差异文件? 1: 是; 0: 否。

OutCBFFlag – 是否将 CBF 信号写入磁盘?

QuantFlag – 是否对全脑使用指定 M0 值? 1: 是; 0: 否。想要更好的理解这一点，可以阅读一下 ISMRM (国际磁共振大会) 的一篇摘要: Chen et al., Impact of equilibrium magnetization of blood on ASL quantification, Proc of ISMRM 2011 #300.

ImgFormatFlag – 图像保存的格式 Analyze 格式 (0) 或者 Nifti 格式 (1)。

4DFlag – 输出图像保存格式为 3D 或者 4D Nifti 格式(1)

M0wmcsfFlag – QuantFlag 为 1 的情况下，可以选择利用 M0csf (M0wmcsfFlag=1) 或者 M0wm (M0wmcsfFlag=0) 来估计 M0b 并计算 CBF 值。

Labeff – 标记效率，PASL 为 0.95, CASL 为 0.68, pCASL 为 0.85。可以通过模拟或者体内测量来确定。

MagType – 磁场强度指标，1 为 3T, 0 为 1.5T。

Timeshift – 一个介于 0 到 1 之间的数值，用来向前或向后移动标签图像，只在 sinc 插值时有效。

AslType – PASL 为 0，CASL 或 pCASL 为 1。

Labeltime – 动脉旋转时间标签 (秒)。

Delaytime – 标签旋转到图像层的延迟时间，QUIPSS 中这个设置为 TI2 (第二个间隔)。

Slicetime – 扫一层的时间，可以通过 (最小 TR-标签时间-标签延迟) / 层数 得到。最小 TR 可以通过挡住自旋标记部分并设置标签序列中的延迟为 0 得到。扫描时间包括激发时间 (包括所有层的梯度选择时间，调整梯度时间)，脂肪或其他饱和时间，层数据采集时间 (相位编码数/逐像素带宽，一般来说相位编码数与图像 y 维维数是相同的，这是因为相位编码通常为沿 y 方向二维成像。例如对于 64\*64 的图像矩阵来说，相位编码数即为 64。) 为了确定层数据采集时间，我们添加了更多详细的相关数据。

M0img – 通过短 TE 和长 TR 得到的 M0 图像。

M0seg – 计算 PASL 图像的 CBF 所需的分割白质 M0 图像。

maskimg – 预定义的 mask。背景抑制数据的话，请指定 mask 或者修改代码中的阈值。

输出：

perfnun – 灌注图像的数量。

示例：在命令行使用 ASL 工具包：

假如我们现在已经选好了预处理过的 ASL 图像序列并把它们的名字存储为变量“imgs”，M0 image 命名为“M0img”，脑脊液 mask 命名为“M0csfroi”，白质 mask 命名为“M0wmroi”，M0csf 值以及 M0wm 值分别存储为“m0csf”和“m0wm”。下面的命令行解释了利用“asl\_perf\_subtract”文件，根据不同的输入设置选择不同的计算 CBF 的方法：

```
1. asl_perf_subtract(imgs,  
0, % 序列中第一幅 ASL 图像为标签图像  
0, % 差减法类型：简单差减  
1, % 减法取向：对照一标签  
[1 1 1 0 0 1 0], % MaskFlag: 1, 剔除模板外的体素。  
... MeanFlag: 1, 会输出平均图像(mean PERF, mean CBF, and mean BOLD)  
... CBFFlag: 1, 会计算CBF值 (如果要做减法，设置为0)  
... BOLDFlag: 0, 不存储pseudo bold  
... OutPerfFlag: 0, 不存储灌注差异图像序列  
... OutCBFFlag: 1, 存储CBF图像序列  
... QuantFlag: 0, 使用M0 校准  
... ImgFormatFlag: 未设置，默认为0 (使用Analyze格式)  
... D4Flag: 未设置，默认为0，即不存储为4D格式  
... M0wmcsfFlag: 未设置，禁用。  
0.5, % sinc 插值的时间迁移。由于前面设置的为简单减法，所以此处设置无用。  
1, % CASL  
0.68, % 标记效率  
1, % 3T  
1.6, % 标签时间
```

```

1, % 时间延迟
45, % slice timing for 2D acquisition. 3D ASL 应设置为 0
17, % TE, 仅 QuantFlag 为 1 时有效。
[], % M0 图像, CASL 数据不使用额外的 M0 图像, 使用 M0 对照。
[], % M0 感兴趣区 mask, 此处不用
[], % 全脑 mask。如果给定, 则用来去除脑外体素。
[], % M0csf, 此处无用
[] % M0wm,此处无用。
);

```

假定 ASL 数据为 CASL, 这行命令会运行一个基于简单减法的量化计算。

```

2.    asl_perf_subtract(imgs,
0, % 序列中第一幅 ASL 图像为标签图像
0, % 差减法类型: 简单差减
1, % 减法取向: 对照—标签
[1 1 1 0 0 1 1 1 1], % MaskFlag: 1, 剔除模板外的体素
... MeanFlag: 1, 会输出平均图像(mean PERF, mean CBF, and mean BOLD)
... CBFFlag: 1, 会计算CBF值 (如果要做减法, 设置为0)
... BOLDFlag: 0, 不存储pseudo bold
... OutPerfFlag: 0, 不存储灌注差异图像序列
... OutCBFFlag: 1, 存储CBF图像序列
... QuantFlag: 1, 使用指定M0 校准
... ImgFormatFlag: 1, 结果保存为 Nifti格式
... D4Flag: 1, 图像存储为 4D格式
... M0wmcsfFlag: 未设置, 默认为调为1(即使用M0csf).
0.5, % sinc 插值的时间迁移。由于前面设置的为简单减法, 所以此处设置无用。
1, % CASL
0.68, %标记效率
1, % 3T
1.6, % 标签时间
1, % 时间延迟
45, % slice timing for 2D acquisition. 3D ASL 应设置为 0
17, % TE, 仅 QuantFlag 为 1 时有效。
[], % M0 图像, CASL 数据不使用额外的 M0 图像, 使用 M0 对照。
[], % M0 感兴趣区 mask, 此处不用
[], % 全脑 mask。如果给定, 则用来去除脑外体素。
m0csf, % M0csf,
[] % M0wm, 此处无用
);

```

这行命令会对 CASL 数据运行一个基于 M0csf 的简单减法的量化计算。

```

3.    asl_perf_subtract(imgs,
0, % 序列中第一幅 ASL 图像为标签图像
0, % 差减法类型: 简单差减
1, % 减法取向: 对照—标签
[1 1 1 0 0 1 1 1 1], % MaskFlag: 1, 剔除模板外的体素
... MeanFlag: 1, 会输出平均图像(mean PERF, mean CBF, and mean BOLD)
... CBFFlag: 1, 会计算CBF值 (如果要做减法, 设置为0)
... BOLDFlag: 0, 不存储pseudo bold
... OutPerfFlag: 0, 不存储灌注差异图像序列
... OutCBFFlag: 1, 存储CBF图像序列
... QuantFlag: 1, 使用指定M0 校准
... ImgFormatFlag: 1, 结果保存为 Nifti格式
... D4Flag: 1, 图像存储为 4D格式
... M0wmcsfFlag: 未设置, 默认为调为1(即使用M0csf).
0.5, % sinc 插值的时间迁移。由于前面设置的为简单减法, 所以此处设置无用。
0, % PASL
0.9, % 标记效率
1, % 3T
1.6, % 标记时间, 对 PASL 无用。
1, % 时间延迟
45, % slice timing for 2D acquisition. 3D ASL 应设置为 0
17, % TE, 仅 QuantFlag 为 1 时有效。
[], % M0 图像, CASL 数据不使用额外的 M0 图像, 使用 M0 对照。
[], % M0 感兴趣区 mask, 此处不用
[], % 全脑 mask。如果给定, 则用来去除脑外体素。
m0csf, % M0csf,
[] % M0wm, 此处无用
);

```

这行命令会对 PASL 数据运行一个基于 M0csf 的简单减法的量化计算。

```

4.    asl_perf_subtract(imgs,
0, % 序列中第一幅 ASL 图像为标签图像
0, % 差减法类型: 简单差减
1, % 减法取向: 对照—标签
[1 1 1 0 0 1 1 1 1 0], % MaskFlag: 1, 剔除模板外的体素
... MeanFlag: 1, 会输出平均图像(mean PERF, mean CBF, and mean BOLD)
... CBFFlag: 1, 会计算CBF值 (如果要做减法, 设置为0)
... BOLDFlag: 0, 不存储pseudo bold
... OutPerfFlag: 0, 不存储灌注差异图像序列

```

```

... OutCBFFlag: 1, 存储CBF图像序列
... QuantFlag: 1, 使用指定M0 校准
... ImgFormatFlag: 1, 结果保存为 Nifti格式
... D4Flag: 1, 图像存储为 4D格式
... M0wmcsfFlag: 0, 使用M0wm.
0.5, % sinc 插值的时间迁移。由于前面设置的为简单减法，所以此处设置无用。
0, % PASL
0.9, % 标记效率
1, % 3T
1.6, % 标记时间，对 PASL 无用。
1, % 时间延迟
45, % slice timing for 2D acquisition. 3D ASL 应设置为 0
17, % TE, 仅 QuantFlag 为 1 时有效。
[], % M0 图像, CASL 数据不使用额外的 M0 图像，使用 M0 对照。
[], % M0 感兴趣区 mask, 此处不用
[], % 全脑 mask。如果给定，则用来去除脑外体素。
m0csf, % M0csf, 此处无用
m0wm % M0wm, 此处无用
);

```

这行命令会对 PASL 数据运行一个基于 M0wm 的简单减法的量化计算。

```

5. asl_perf_subtract(imgs,
0, % 序列中第一幅 ASL 图像为标签图像
0, % 差减法类型: 简单差减
1, % 减法取向: 对照—标签
[1 1 1 0 0 1 1 1 1 0], % MaskFlag: 1, 剔除模板外的体素
... MeanFlag: 1, 会输出平均图像(mean PERF, mean CBF, and mean BOLD)
... CBFFlag: 1, 会计算CBF值 (如果要做减法, 设置为0)
... BOLDFlag: 0, 不存储pseudo bold
... OutPerfFlag: 0, 不存储灌注差异图像序列
... OutCBFFlag: 1, 存储CBF图像序列
... QuantFlag: 1, 使用指定M0 校准
... ImgFormatFlag: 1, 结果保存为 Nifti格式
... D4Flag: 1, 图像存储为 4D格式
... M0wmcsfFlag: 0, 使用M0wm.
0.5, % sinc 插值的时间迁移。由于前面设置的为简单减法，所以此处设置无用。
0, % PASL
0.9, % 标记效率
1, % 3T
1.6, % 标记时间，对 PASL 无用。

```

```

1, % 时间延迟
45, % slice timing for 2D acquisition. 3D ASL 应设置为 0
17, % TE, 仅 QuantFlag 为 1 时有效。
M0img, % M0 image,
M0wmroi, M0 感兴趣区 mask, 此处无用
[], % 全脑 mask。如果给定, 则用来去除脑外体素。
m0csf, % M0csf, 此处无用
m0wm % M0wm, 此处无用
);

```

这行命令会对 PASL 数据运行一个基于 M0wm 的简单减法的 CBF 量化计算。注: 虽然提供了 M0img 和 M0wmroi, 但是 M0wm 会直接使用最后提供的 m0wm。对于 M0csf 也一样。

```

6. asl_perf_subtract(imgs,
0, % 序列中第一幅 ASL 图像为标签图像
0, % 差减法类型: 简单差减
1, % 减法取向: 对照—标签
[1 1 1 0 0 1 1 1 1 0], % MaskFlag: 1, 剔除模板外的体素
... MeanFlag: 1, 会输出平均图像(mean PERF, mean CBF, and mean BOLD)
... CBFFlag: 1, 会计算CBF值 (如果要做减法, 设置为0)
... BOLDFlag: 0, 不存储pseudo bold
... OutPerfFlag: 0, 不存储灌注差异图像序列
... OutCBFFlag: 1, 存储CBF图像序列
... QuantFlag: 1, 使用指定M0 校准
... ImgFormatFlag: 1, 结果保存为 Nifti格式
... D4Flag: 1, 图像存储为 4D格式
... M0wmcsfFlag: 0, 使用M0wm.
0.5, % sinc 插值的时间迁移。由于前面设置的为简单减法, 所以此处设置无用。
0, % PASL
0.9, % 标记效率
1, % 3T
1.6, % 标记时间, 对 PASL 无用。
1, % 时间延迟
45, % slice timing for 2D acquisition. 3D ASL 应设置为 0
17, % TE, 仅 QuantFlag 为 1 时有效。
M0img, % M0 image,
M0wmroi, M0 感兴趣区 mask, 此处无用
[], % 全脑 mask。如果给定, 则用来去除脑外体素。
[], % M0csf, 此处无用
[], % M0wm, 此处无用
);

```

这行命令会对 PASL 数据运行一个基于 M0wm 的简单减法的 CBF 量化计算。M0wm 通过提供的 WM mask 中的 M0 图像来得到。

输出：perfnm – 有效对照/标签图像对的数量。

CBF values – 你会发现有消息提示例如“**The mean CBF is 56.81 (58.10)**”。第二个值（小括号中的值）是来自全脑（由 mask 定义或者设置阈值为平均对照图像最大强度\*0.2 得到）；第一个值也是来自全脑，但已经应用了全脑 mask，剔除了异常值。

输出图像：

写入磁盘的图像有：灌注图像，血流图像以及 CBF 图像（对应的参数设置为输出）。

全脑灌注差异信号和 CBF 值会存储在一个文本文档中。

所有的输出文件都会保存在与输入文件相同的文件夹中。

### 5.3 ASL 数据采集参数设置

不同的 ASL 序列数据采集的参数设置也不同，因此联系供应商或者序列提供者很有必要。下面的图文是针对西门子平台而言。对于 GE 的序列产品用户来说，只会得到一个 M0 图像和一个减法灌注差异图像。

**Slice timing for 2D**: 在 ASL 扫描台打开扫描方案，就会在屏幕的右下角看到一个与图 5.8 所示相似的界面。点击“TR”（如图 5.8 蓝色箭头所示），界面底端会显示一个绿色的横条。横条左下角显示的数字(如图 5.8 紫色箭头所示) 为最小 TR。层数可以也在这个页面找到，左面第二项。层扫描时间可以通过下面计算得到：

(最小 TR – 标记时间 – 标记延迟)/(层数)

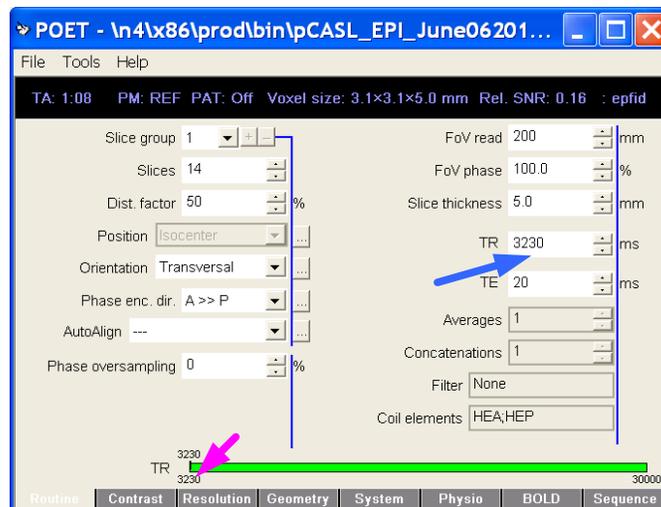


图 5.8 确定最小 TR 示意图

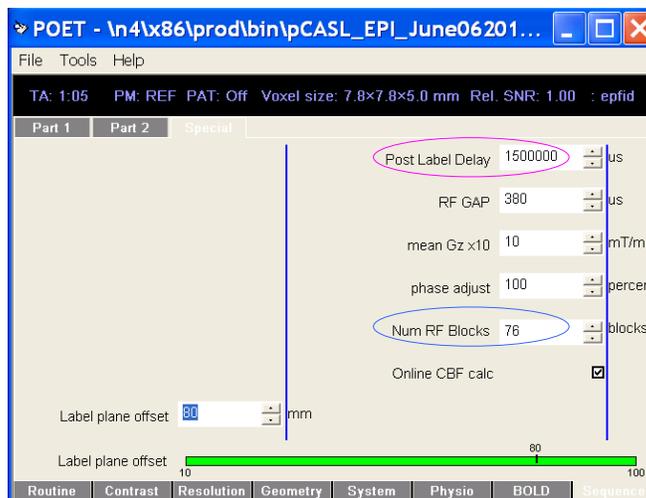


Fig. 5.9. The pCASL parameter tab.

举例来说，本节显示的方案中，层扫描时间为： $(3230-1406-1500)/14=23.14$  ms. 注：这只是举例，并不是标准方案。对于西门子 PICORE-based PASL 序列来说，层扫描时间为： $(\text{最小 TR} - 50\text{ms} - \text{TI2})/(\text{层数})$ ，其中，50 ms 为预饱和和脉冲时间。

#### 标记时间和标记延迟：

**情况一)** 宾夕法尼亚 pCASL 以及由 Danny JJ Wang 开发的旧版本的 CASL 序列中，标记时间由“Num RF Blocks” (如图 5.9 蓝色椭圆所示)定义得到。每一个 RF 模块持续时间为 18500 us，因此图 5.9 中所示的 76 个模块的标记时间为  $18500\text{us} \times 76 = 1.406$  sec. 标记延迟可以直接从上图所示的紫色椭圆中读取。

**情况二)** 西门子 PASL (PICORE)序列中，可以直接找到 "inversion time 1" 和 "inversion time 2", 即 Eric Wong 开发的 Q2tips 技术中的 TI1 和 TI2. 标记延迟时间输入到文件 asl\_perf\_subtract 中, ASLtbx 中 ASL CBF 的计算需要的为 TI2. 在 Q2tips 中, asl\_perf\_subtract 中的“labeling time”习惯将 TI1(单位为秒)作为输入 (注: TI1 默认为 0.7 秒, 如果 labeling time >0.5 且 <1, 则设置 TI1=labeling time)。

**情况三)** GE 的序列产品(由 David Alsop 开发)中，标记时间以及标记延迟时间在研究模式下都可以找到。标记延迟时间在扫描方案中可以直接找到。

**情况四)** 宾夕法尼亚大学提供的 3D FSE background suppressed spiral pCASL 序列。其标记时间可以参考情况一获取。注意所有三维 ASL 序列图像在计算血流时，slice time 都需要设为 0。

**影响层扫描时间以及最小 TR 的参数:** 如果你想减少读出时间以减少非共振产生的畸变，有下面几种方法：1. 使用平行成像；2. 使用局部傅里叶采集；3. 增加带宽，减少回波间隔。其中第三种方法不建议用户使用，因为会造成严重的重影，除非你可以离线重建。磁场强度为 3T 的典型的贷款范围是 2232 到 3004 Hz/pixel。无论采取了上述哪种方法，别忘了最后将 TE 调整到最小值。

## 6 致谢

开发以及维护 ASLtbx 纯粹为自愿，也没有经费支持。我要感谢我们实验室的李正军博士。正军帮助建立了网站的主要框架，修改了脚本文件以及撰写了本手册的第一个版本。计算 CBF 代码的最早版本翻译自一段 IDL 代码，这段代码是由宾夕法尼亚大学的 Danny JJ Wang, Geoffrey K. Aguirre 和 David C. Alsop 编写的。

1. Detre, J.A., et al., *Perfusion imaging*. Magnetic Resonance in Medicine, 1992. **23**: p. 37-45.
2. Williams, D.S., et al., *Magnetic resonance imaging of perfusion using spin inversion of arterial water*. Proceedings of the National Academy of Sciences, 1992. **89**(1): p. 212-216.
3. Detre, J.A., et al., *Tissue specific perfusion imaging using arterial spin labeling*. NMR Biomed, 1994. **7**(1-2): p. 75-82.
4. Edelman, R.R., et al., *Qualitative mapping of cerebral blood flow and functional localization with echo-planar MR imaging and signal targeting with alternating radio frequency*. Radiology, 1994. **192**(2): p. 513-20.
5. Kim, S.G., *Quantification of relative cerebral blood flow change by flow-sensitive alternating inversion recovery (FAIR) technique: application to functional mapping*. Magn Reson Med, 1995. **34**(3): p. 293-301.
6. Wong, E.C., R.B. Buxton, and L.R. Frank, *Implementation of quantitative perfusion imaging techniques for functional brain mapping using pulsed arterial spin labeling*. NMR Biomed, 1997. **10**(4-5): p. 237-49.
7. Golay, X., et al., *Transfer insensitive labeling technique (TILT): application to multislice functional perfusion imaging*. Journal of magnetic resonance imaging : JMRI, 1999. **9**(3): p. 454-61.
8. Alsop, D.C. and J.A. Detre, *Reduced transit-time sensitivity in noninvasive magnetic resonance imaging of human cerebral blood flow*. J Cereb Blood Flow Metab, 1996. **16**: p. 1236-1249.
9. Alsop, D.C. and J.A. Detre, *Multisection Cerebral Blood Flow MR Imaging with Continuous Arterial Spin Labeling*. Radiology, 1998. **208**: p. 410-416.
10. Dai, W., et al., *Continuous flow-driven inversion for arterial spin labeling using pulsed radio frequency and gradient fields*. Magn Reson Med, 2008. **60**(6): p. 1488-97.
11. Wu, W.C., et al., *Physiological modulations in arterial spin labeling perfusion magnetic resonance imaging*. IEEE Trans Med Imaging, 2009. **28**(5): p. 703-9.
12. Wang, Z., et al., *Empirical optimization of ASL data analysis using an ASL data processing toolbox: ASLtbx*. Magn Reson Imaging, 2008. **26**(2): p. 261-269, PMC2268990.
13. Wang, Z., *Improving Cerebral Blood Flow Quantification for Arterial Spin Labeled Perfusion MRI by Removing Residual Motion Artifacts and Global Signal Fluctuations*. Magnetic Resonance Imaging, 2012(in press).