**Spatial STAPLE Documentation**

1. **Downloading**

* Go to the website: <http://www.nitrc.org/frs/?group_id=462>
* Download “masi-fusion-v1.1”

1. **Installing/Setup Instructions**

* Create the directory where you want the source code to be (e.g. “masi-fusion”).
* Unzip masi-fusion.zip into the created directory.
* The code is written in a combination of MATLAB (frontend) and java (backend) code. Due to the backend being written in java, you have to allocate memory to java. To do this:
  1. Create an empty text file called “java.opts” in the “masi-fusion” directory.
  2. Edit this file to contain the line “-Xmx15000m” (without the quotes).
  3. This will set the maximum memory used by java to 15GB.
     1. Java will not use this much memory unless it is actually needed.
     2. Obviously, you can change this amount by modifying the text file.
* Finally, from the “masi-fusion” directory, start MATLAB and set the PATH by running the “setup\_labeling.m” script.
  1. The output of the script should indicate that there is 15GB (or whatever you set it as) of memory available for java.

1. **Running SpatialSTAPLE**

* For generic information, type “help SpatialSTAPLE” to get basic information on how to run the algorithm.
* I will describe a typical technique to run the algorithm. There are several ways to call Spatial STAPLE, but consider the one that we most commonly use:

[estimate W theta] = SpatialSTAPLE(obs, epsilon, sv\_prior, init\_flag, ...

interp\_type, num\_up, win\_dims, ...

bias, bias\_theta);

* **Inputs**
  + *obs* -> the observation struct
    - This is a structure that we have created to hold the label observations from the raters (or registered atlases).
    - For information on how to construct this see “help create\_obs”, “help add\_obs”, and the example scripts in the “scripts” directory.
    - For an example
      * Let “data” be an X x Y x Z x R, 4D matrix that holds the observations for all R raters, where each observation is a volume of size X x Y x Z.
      * Code to create the observation struct would be:

obs = create\_obs(‘volume’, [X Y Z]);

for i = 1:R

obs = add\_obs(obs, data(:, :, :, i));

end

* + *epsilon* -> the convergence factor
    - We usually use 1e-5 for Spatial STAPLE
  + *sv\_prior* -> an explicit spatially varying prior
    - sv\_prior should be an X x Y x Z x L, 4D matrix, where X, Y, and Z are the dimensions of the volumes, and L is the number of labels.
    - We usually use a majority vote or locally weighted vote estimate to initialize the spatially varying prior.
  + *init\_flag* -> the initialization type
    - We usually use init\_flag = 0, which indicates that the EM algorithm starts with an initial estimate of “W”, instead of “theta”
  + *interp\_type* -> the interpolation type for the local theta estimates
    - interp\_type = 0, indicates nearest neighbor interpolation
    - interp\_type = 1, indicates linear interpolation (recommended)
  + *num\_up* –> a 3 element array indicating the number of times theta is calculated in the X, Y and Z directions.
    - The points where theta is calculated is rectilinearly distributed throughout the volume.
    - This is dependent upon the size of the volume.
    - We usually construct this so that there as at least 50% overlap between windows. (which depends upon “win\_dims”).
  + *win\_dims* -> a 3 element array indicating the size of the windows (regions) for each theta estimate
    - Again, this is dependent upon the size and resolution of the volume.
    - For a standard isotropic brain volume, we usually use something like [10 10 10].
  + *bias ->* the amount of bias applied to the prior estimate of theta (the linkage structure between global and local performance)
    - We usually use N\_w / L, where N\_w is the number of elements in each window and L is the number of labels.
  + *bias\_theta ->* the prior estimate of theta
    - We usually use either a prior theta from majority vote or STAPLE. Obviously, theta should be L x L x R, where L is the number of labels and R is the number of raters.
    - Each column should be normalized to 1.
* **Outputs**
  + estimate – the X x Y x Z estimated segmented volume
  + W – the X x Y x Z x L estimated label probabilities
  + theta – the regional performance level parameters.