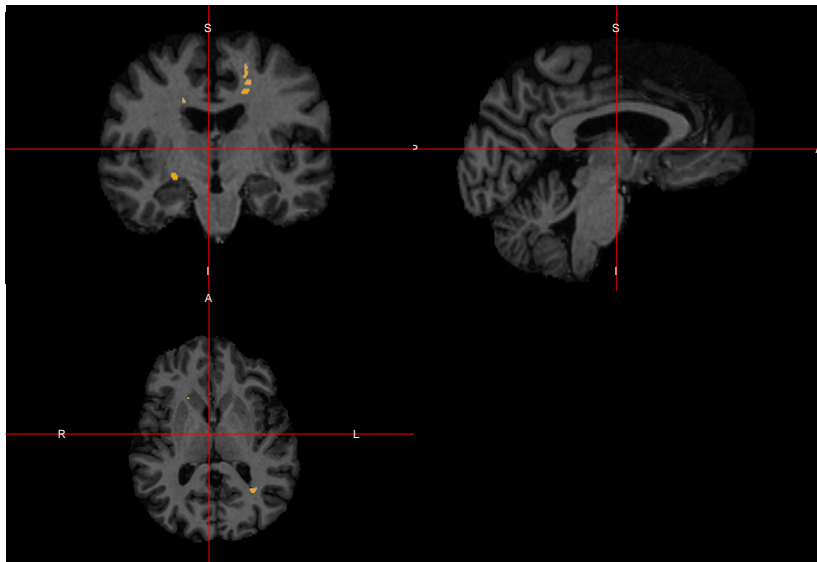# MS Lesion Segmentation

Background

- Obtaining manual lesion segmentations is often resource intensive, so accurate and efficient methods for automatic segmentation are necessary for scalability and research progress.

Goals of this tutorial

- Apply OASIS (Sweeney et al. 2013), an automatic lesion segmentation model, to obtain predicted lesion probability maps.
- Compare the results using the default OASIS settings to those obtained after re-training the model using our data.

Visualization

- Here's the T1 volume for training subject 05 with a 'gold standard' manual lesion segmentation overlayed.

MS Lesion Segmentation with OASIS

- OASIS is Automated Statistical Inference for Segmentation (Sweeney et al. 2013)
- The OASIS algorithm takes FLAIR, T1, T2, and PD images from patients with multiple sclerosis (MS) and produces OASIS probability maps of MS lesion presence, which can be thresholded into a binary lesion segmentation.
- OASIS is based on a logistic regression of the gold standard labels on the images, smoothed versions of the images, and some interaction terms.

Default OASIS Model

- ▶ The OASIS library comes with default parameters that can be used to generate probability maps for new test subjects.
- ▶ Here we apply the function `oasis_predict` with the default model to obtain OASIS probability maps for the test subjects.

```r
library(oasis)
default_predict_ts = function(x){
  res = oasis_predict(
      flair=ts_flairs[[x]], t1=ts_t1s[[x]],
      t2=ts_t2s[[x]], pd=ts_pds[[x]],
      brain_mask=ts_masks[[x]],
      preproc=FALSE, normalize=TRUE,
      model=oasis::oasis_model)
  return(res)
}
default_probs_ts = lapply(1:3, default_predict_ts)
```

Vizualization

▶ Here's the probability map for test subject 01:
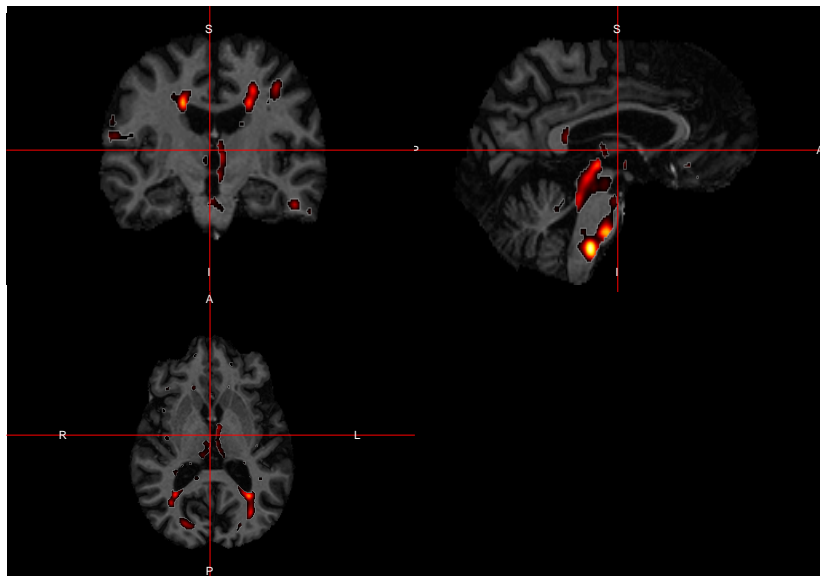
Thresholding

- To get a final estimated segmentation, we must choose a cutoff to binarize the OASIS probability maps.
- The `binary` argument in the `oasis_predict` function is FALSE by default, resulting in the output being the probability map.
- Setting `binary=TRUE` will return the thresholded version, using the input to the `threshold` argument (default $= 0.16$).
- In practice, we might want to use a grid search over thresholds and cross validation to choose the cutoff.

Vizualization

- Here's the binary mask for test subject 01, using the default 0.16 threshold:
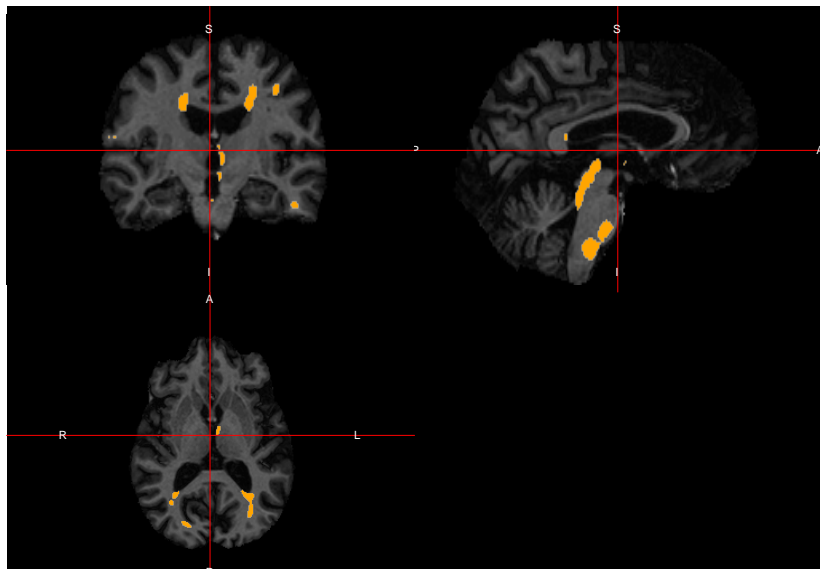
Default OASIS Model

- To evaluate how the default model performs, we need to compare the predictions to a gold standard.
- Let's therefore obtain OASIS probability maps for our training subjects.
- We will use the default threshold to binarize.

```
default_predict_tr = function(x){
  res = oasis_predict(
      flair=tr_flairs[[x]], t1=tr_t1s[[x]],
      t2=tr_t2s[[x]], pd=tr_pds[[x]],
      brain_mask=tr_masks[[x]],
      preproc=FALSE, normalize=TRUE,
      model=oasis::oasis_model, binary=TRUE)
  return(res)
}
default_probs_tr = lapply(1:5, default_predict_tr)
```

Default OASIS Model Results

- Here's the T1 volume for training subject 05 with the OASIS segmentation overlayed.
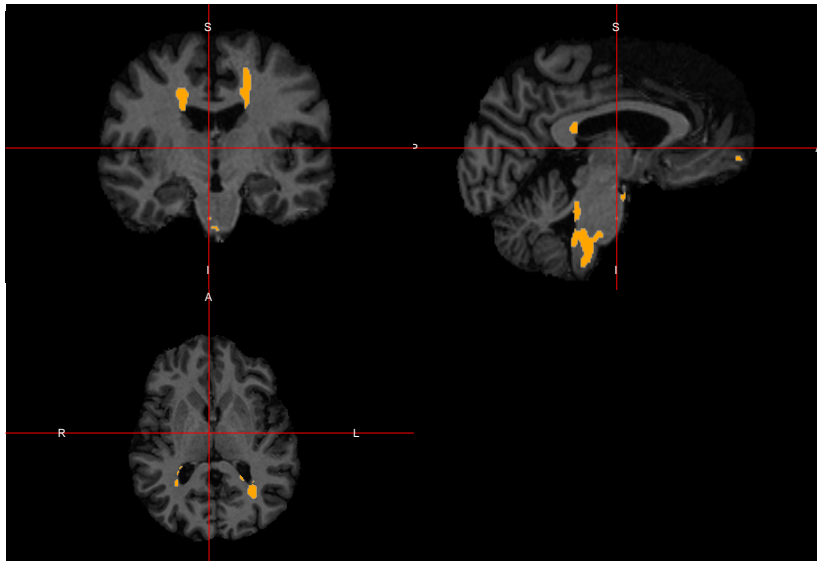
Default OASIS Model Results

- The average dice coeffients (over the two gold standards) for the training subjects are: 0.49, 0.69, 0.43, 0.29, 0.26.

Improving Results

- ► The default model is picking up a lot of false positives in the spinal chord.
- ► We might improve the results by re-training the OASIS model using our five training subjects.
- ► To retrain the model using new data, binary masks of gold standard lesion segmentations are needed and should be in T1 space.

Making OASIS data frames

- ▶ OASIS requires a particular data frame format
- ▶ Includes an option to preprocess your data (`preproc`), which does (1) inhomogeneity correction using `fsl_biascorrect` and (2) rigid coregistration using `flirt` to the T1 space.
- ▶ Includes an option to normalize the intensities of your data using whole-brain normalization (`normalize`)
- ▶ `make_df()` below is a helper function

```
make_df = function(x){
  res = oasis_train_dataframe(
      flair=tr_flairs[[x]], t1=tr_t1s[[x]], t2=tr_t2s[[x]],
      pd=tr_pds[[x]], gold_standard=tr_golds2[[x]],
      brain_mask=tr_masks[[x]],
      preproc=FALSE, normalize=TRUE, return_preproc=FALSE)
  return(res$oasis_dataframe)
}
oasis_dfs = lapply(1:5, make_df)
```

Training OASIS

- The function `oasis_training` takes the data frames we made and fits a logistic regression, where the outcome vector consists of all subjects' voxel-level data (top 85% in intensity)
- The function `do.call` is a useful R function that applies the function named in the first argument to all elements of the list specified in the second argument.

```
model = do.call("oasis_training", oasis_dfs)
```

OASIS model object

- ▶ model is an object of type glm
- ▶ We see the covariates used and assosicated coefficients when we print the model:

```
print(ms.lesion::ms_model)
```

Call: glm(formula = formula, family = binomial, data = tr

Coefficients:
(Intercept)	FLAIR_10	FLAIR	FLAIF
-5.6939	4.1041	1.4076	-2.1
PD_10	PD	PD_20	T2
4.7047	0.1739	-17.3328	9.9
T2	T2_20	T1_10	
0.8376	-19.2016	12.5254	1.2
T1_20	FLAIR_10:FLAIR	FLAIR:FLAIR_20	PD_10
-27.9823	-1.0304	-3.4276	0.2

Trained OASIS Model Results

- The average dice coeffients (over the two gold standards) for the training subjects are: 0.64, 0.71, 0.62, 0.36, 0.36.

References

Sweeney, Elizabeth M, Russell T Shinohara, Navid Shiee, Farrah J
Mateen, Avni A Chudgar, Jennifer L Cuzzocreo, Peter A Calabresi,
Dzung L Pham, Daniel S Reich, and Ciprian M Crainiceanu. 2013.
"OASIS Is Automated Statistical Inference for Segmentation, with
Applications to Multiple Sclerosis Lesion Segmentation in Mri."
NeuroImage: Clinical 2. Elsevier: 402–13.