Ophys Single Session Analysis

Brennan Lagasse, Advised by Dr. Ethan Meyers

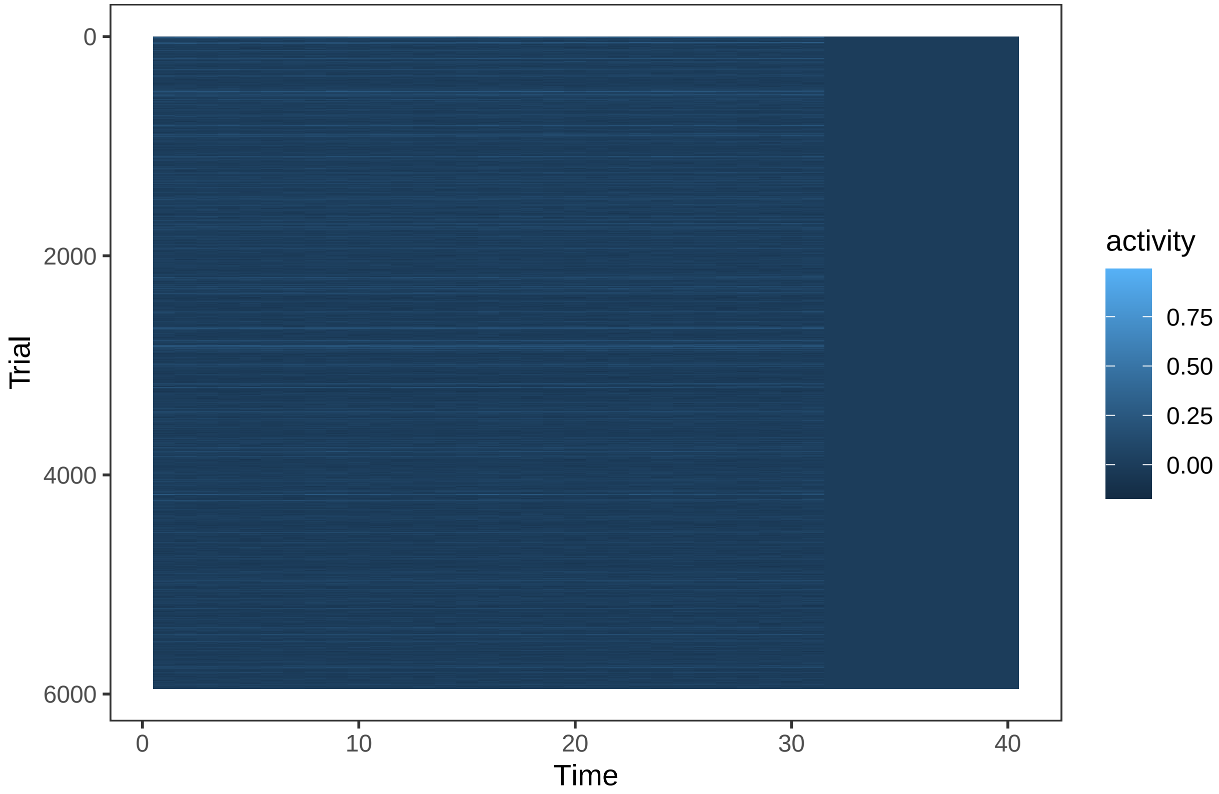
2023-3-26

```r
raster_dir_name = file.path("R_OphysData", "session_496908818", "natural_scenes")
file_name = "517404519.rda"

load(file.path(raster_dir_name, file_name))

test_valid_raster_format(file.path(raster_dir_name, file_name))

plot(raster_data)
```

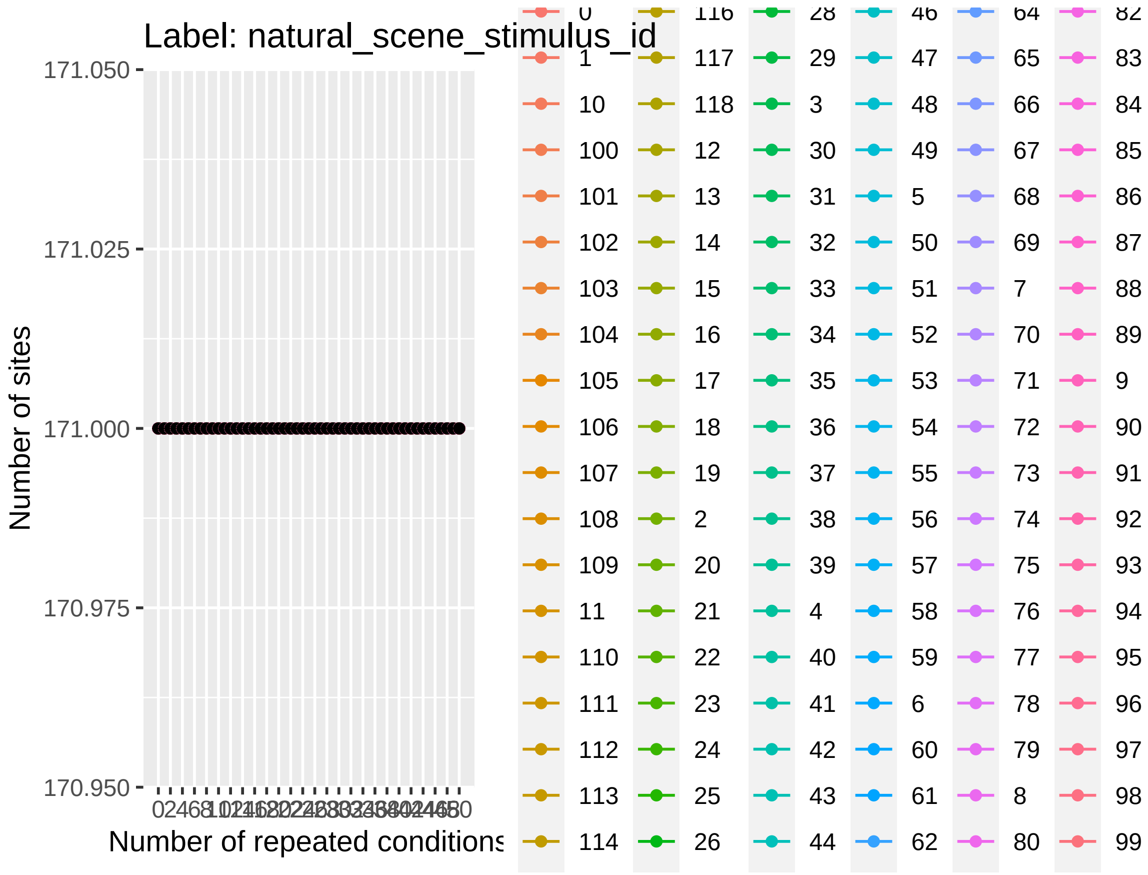


```r
save_dir_name <- file.path("R_OphysData_Binned", "session_496908818", "natural_scenes")

binned_file_name <- create_binned_data(raster_dir_name, file.path(save_dir_name, "ZD"), 2, 1, 1, 31)
```

```
##   |                                                                              |
```

```r
label_rep_info <- get_num_label_repetitions(binned_file_name, "natural_scene_stimulus_id")
plot(label_rep_info)
```



Decoding Analysis

Performing a decoding analysis involves several steps:

1. Creating a datasource (DS) object that generates training and test splits of the data.

```r
# Fewer groups for speed. Up to 50 seems appropriate
variable_to_decode <- "natural_scene_stimulus_id"
num_cv_splits <- 20

ds <- ds_basic(binned_file_name, variable_to_decode, num_cv_splits)
```

```
## Automatically selecting sites_IDs_to_use. Since num_cv_splits = 20 and num_label_repeats_per_cv_spli
```

2. Optionally creating feature-preprocessor (FP) objects that learn parameters from the training data, and preprocess the training and test data.

```r
fps <- list(fp_zscore())
```

3. Creating a classifier (CL) object that learns the relationship between the training data and training labels, and then evaluates the strength of this relationship on the test data.

```
cl <- cl_max_correlation()
```

4. Creating result metric (RM) objects that aggregate the predictions to create result summaries.

```
rms <- list(rm_main_results(), rm_confusion_matrix())
```

5. Running a cross-validator object that using the datasource (DS), the feature-preprocessor (FP) and the classifier (CL) objects to do a cross-validation procedure that estimates the decoding accuracy.

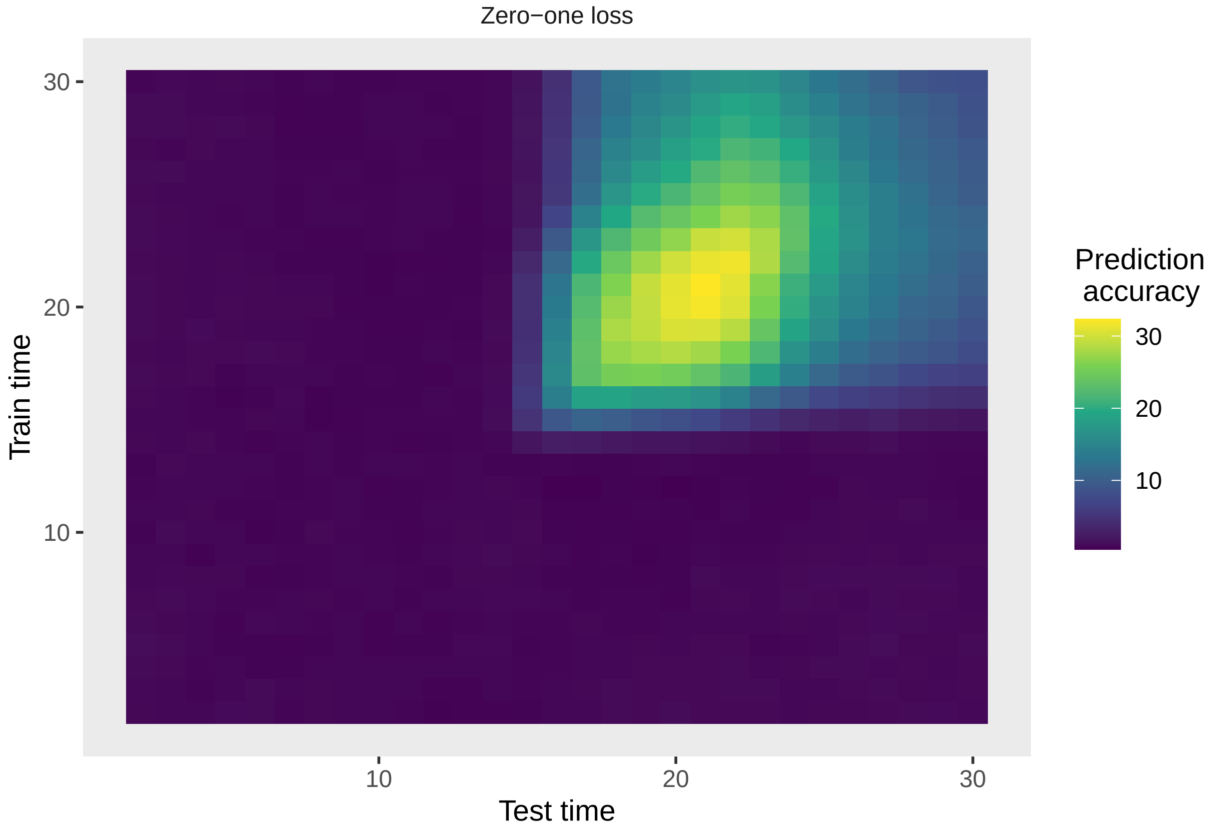
```
# Default parameters for a quick run (resamples increased to 50)

cv <- cv_standard(datasource = ds,
                  classifier = cl,
                  feature_preprocessors = fps,
                  result_metrics = rms,
                  num_resample_runs = 2)
```

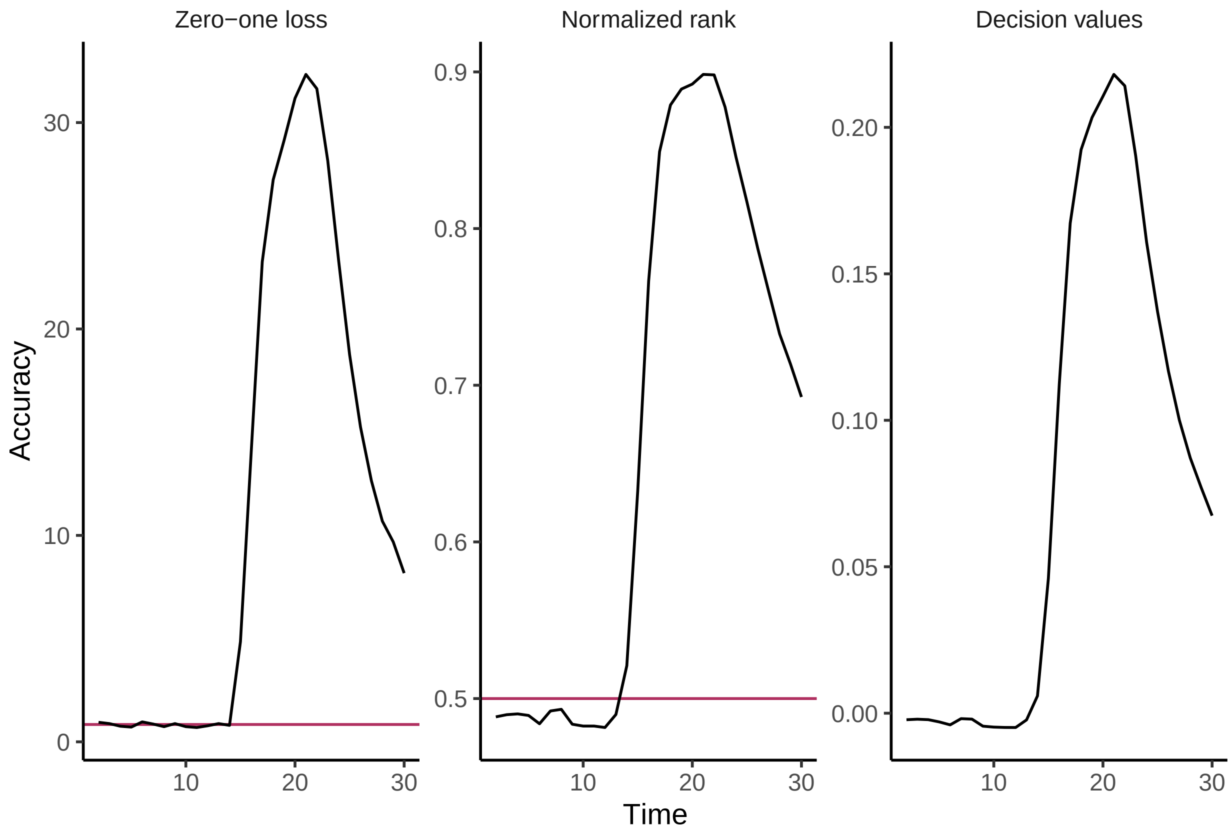
```
DECODING_RESULTS <- run_decoding(cv)
```

```
##   |                                                                              |
```

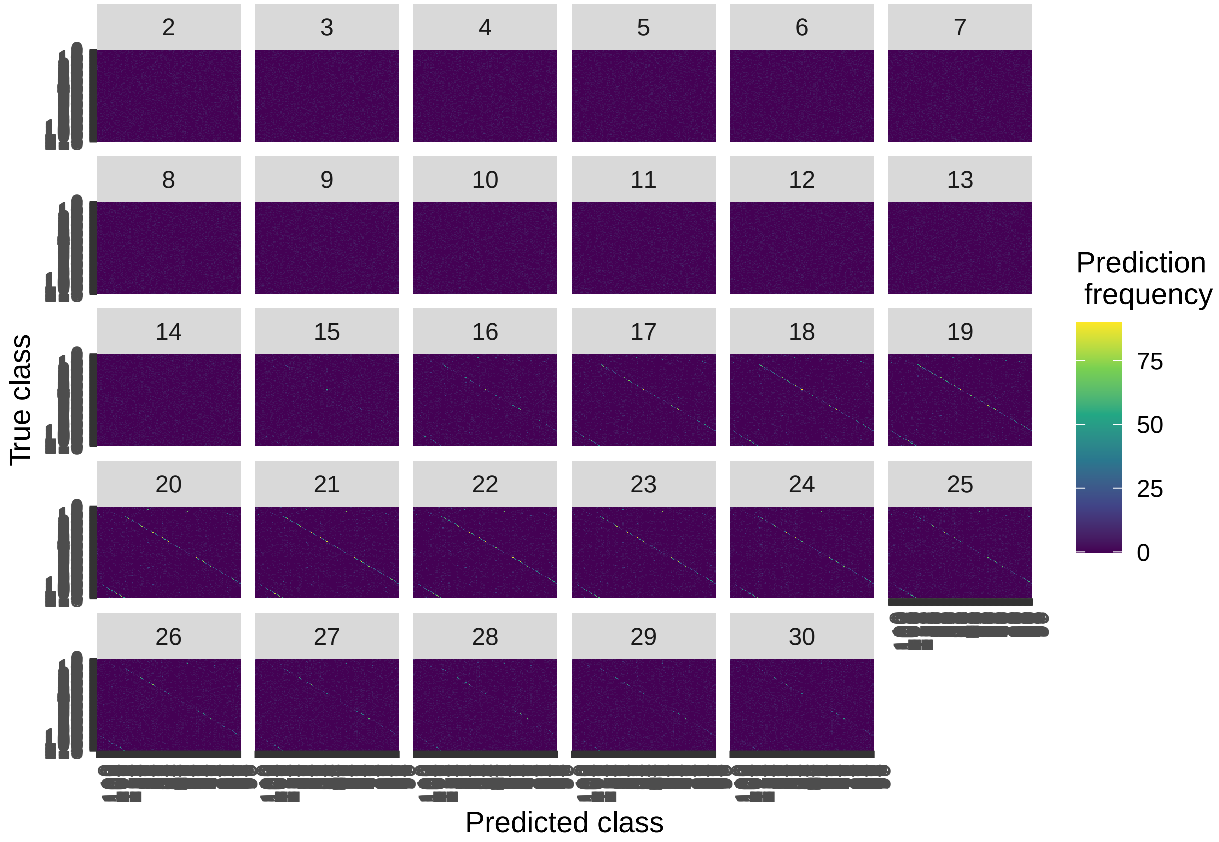
```
plot(DECODING_RESULTS$rm_main_results)
```



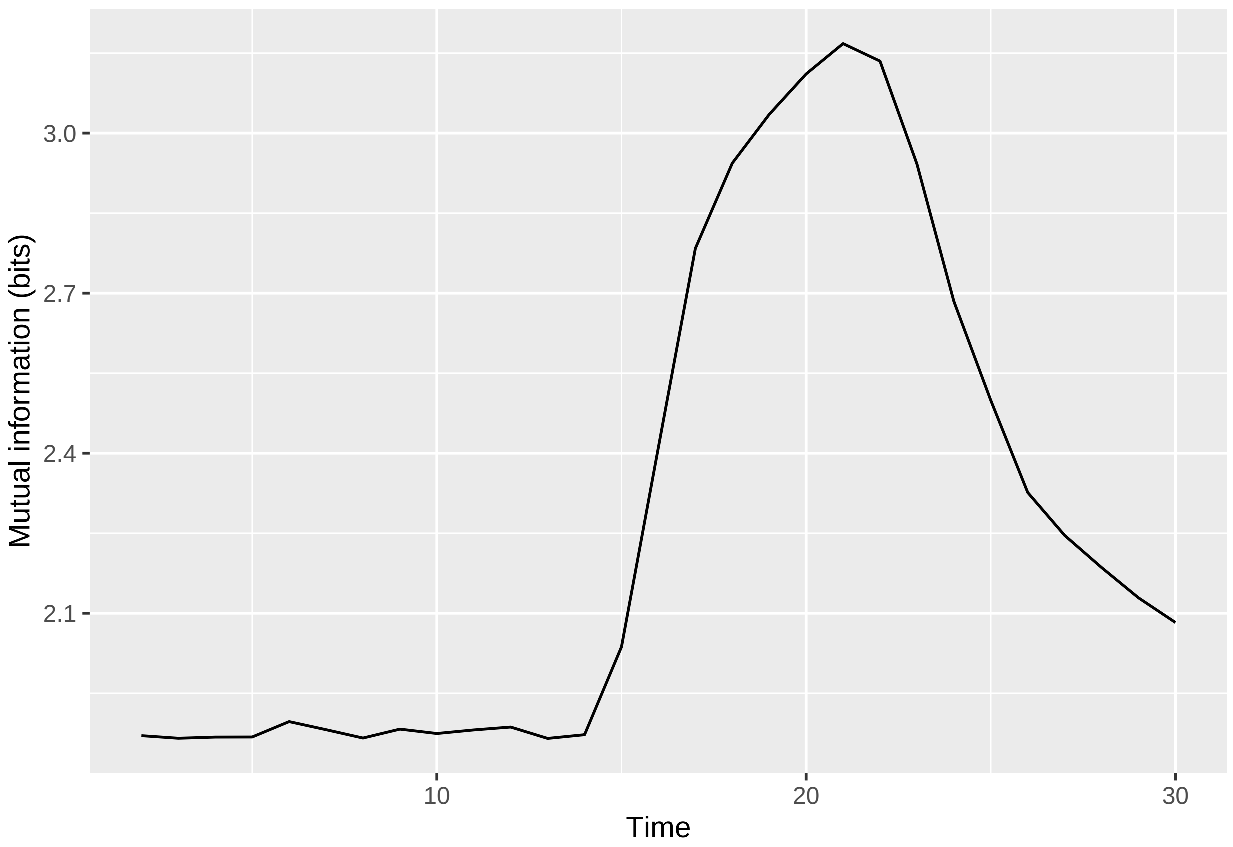
```
plot(DECODING_RESULTS$rm_main_results, results_to_show = 'all', type = 'line')
```



```
plot(DECODING_RESULTS$rm_confusion_matrix)
```



```r
plot(DECODING_RESULTS$rm_confusion_matrix, results_to_show = "mutual_information")
```


Mutual information (bits)
3.0
2.7
2.4
2.1
10
20
30
Time