

blech _ clust

The secrets within
a.k.a.

Some of the lesser known but important features (according to Abu)

As of 1/15/26

Docs + Migration guide

The screenshot shows a GitHub repository page for 'blech_clust'. The top navigation bar includes links for Code, Issues (197), Pull requests (68), Discussions, Actions, Projects (1), Wiki, Security, Insights, and Settings. The main content area displays the 'Code' tab, showing the 'master' branch. A message indicates the branch is 1852 commits ahead of and 4 commits behind 'narendramukherjee/blech_clust:master'. Below this are buttons for Contribute and Sync fork. The commit list starts with a merge pull request from 'abuzarmahmood' (fbfdcd - 2 days ago). The next few commits are related to documentation and workflow configuration. A red arrow points to a commit from 'docs' (2 days ago) titled 'docs: remove blog links from migration guide'. The right sidebar contains an 'About' section describing the toolkit as a Python and R-based toolkit for clustering and sorting electrophysiology data. It mentions its origin in the Katz Lab at Brandeis University and its purpose for spike sorting, EMG analysis, and quality assessment. The repository URL is listed as katzlabbrandeis.github.io/blech_clust/. The footer features a blue bar with the repository name 'blech_clust Documentation' and links to Home, Getting Started, Workflow Diagrams, Tutorials, and API Reference.

Getting Started
Installation
Quick Start

Migration Guide
Overview
Removed Features
File Mapping
QA Improvements

Migration Guide from Original blech_clust

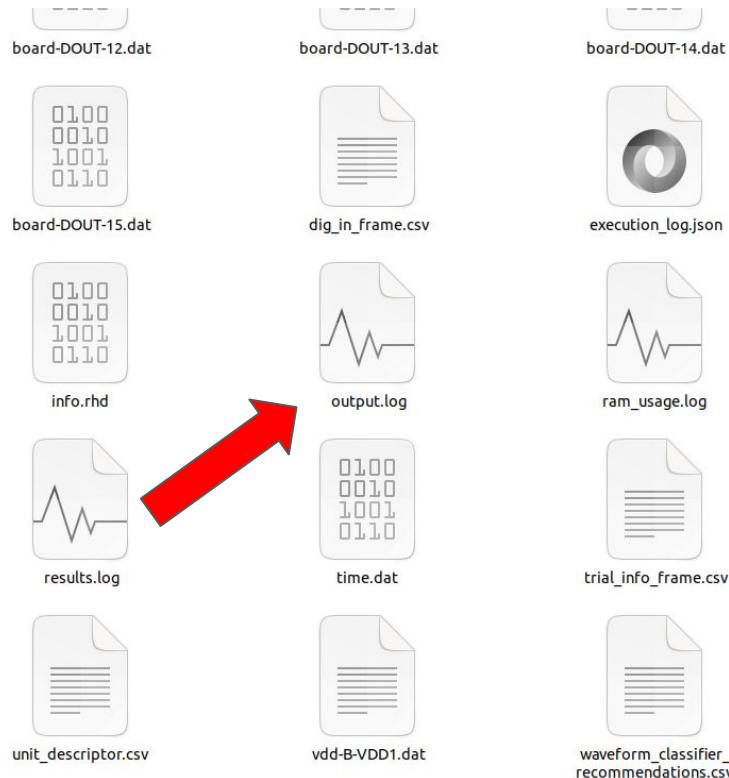
This guide documents the changes between the original blech_clust and the current katzlabbrandeis fork.

Overview of Changes

The katzlabbrandeis fork represents a significant modernization of the original codebase:

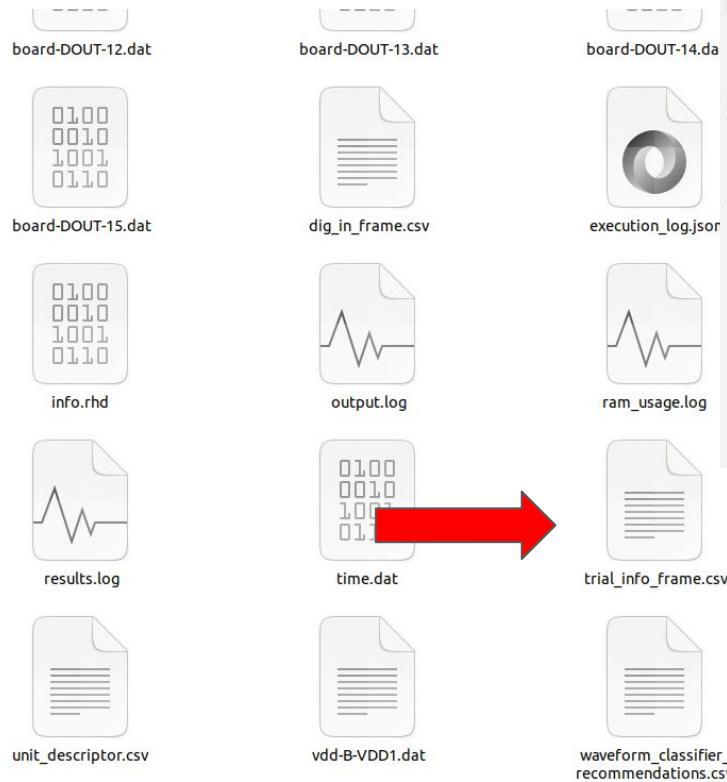
Table of contents
Overview of Changes
Migration Guide Sections
Quick Reference: New Features
Installation and Environment
Testing Infrastructure
Metadata and Parameter Recording
Common Average Reference Improvements

Basic debugging



```
Open Save
4.0 TB Volume /media/storage/abu_resorted/bla_gc/AM11_4Tastes_191030_114043_copy
=====
1 =====
2 Attempting blech_exp_info.py, started at 2026-01-13 16:23:47
3 Git branch: master
4 Git commit: 454f07821b5ca186cd6fdc87430a58ca3727d544
5 =====
6 =====
7 Completed blech_exp_info.py, ended at 2026-01-13 16:23:47
8 =====
9 =====
10 Attempting blech_common_avg_reference.py, started at 2026-01-13 16:26:05
11 Git branch: master
12 Git commit: 454f07821b5ca186cd6fdc87430a58ca3727d544
13 =====
14 Processing : /media/storage/abu_resorted/bla_gc/AM11_4Tastes_191030_114043_copy/
15 Number of groups : 11
16 bla-04 :: 9 channels :: 
17 ['A_00' 'A_01' 'A_24' 'A_26' 'A_27' 'A_28' 'A_29' 'A_30' 'A_31']
18
19 bla-05 :: 5 channels :: 
20 ['A_02' 'A_03' 'A_04' 'A_05' 'A_07']
21
22 bla-00 :: 1 channels :: 
23 ['A_06']
24
25 gc-05 :: 5 channels :: 
26 ['A_08' 'A_10' 'A_14' 'A_16' 'A_20']
27
28 gc-09 :: 1 channels :: 
29 ['A_09']
30
31 gc-06 :: 1 channels :: 
32 ['A_11']
33
34 gc-08 :: 1 channels :: 
35 ['A_12']
36
37 gc-03 :: 6 channels :: 
38 ['A_13' 'A_18' 'A_19' 'A_21' 'A_22' 'A_23']
```

Easy access to trial info

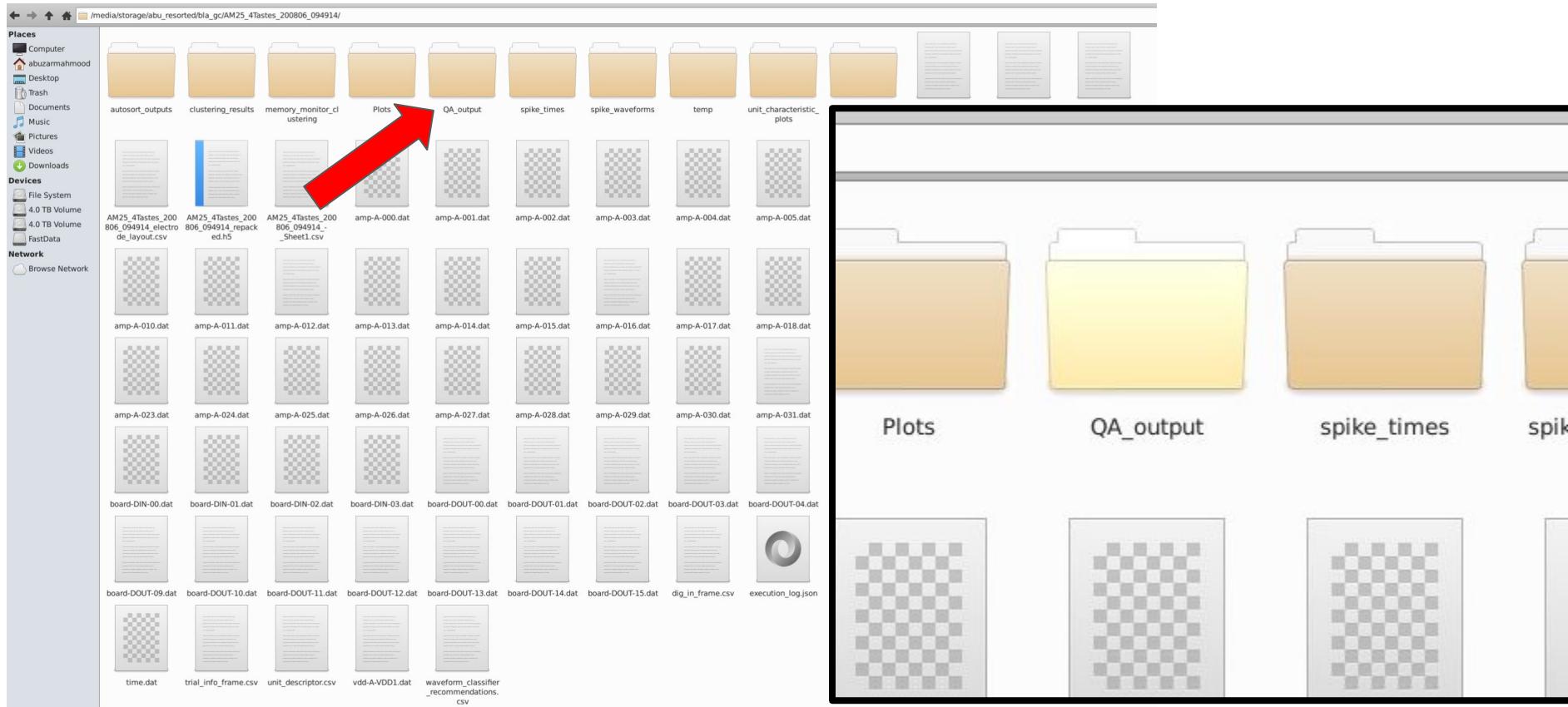


The screenshot shows a spreadsheet application window titled 'ONLYOFFICE *trial_info_fra...'. The table contains 17 rows of data, with the first row serving as the header. The columns are labeled A through F, and the header row includes column labels such as 'dig_in_num', 'dig_in_name', 'taste', 'start_taste', 'end_taste', 'abs_trial_num', and 'taste'. The data rows represent various trials, with some entries being blank or containing specific codes like 'nacl' or 'qhcl'.

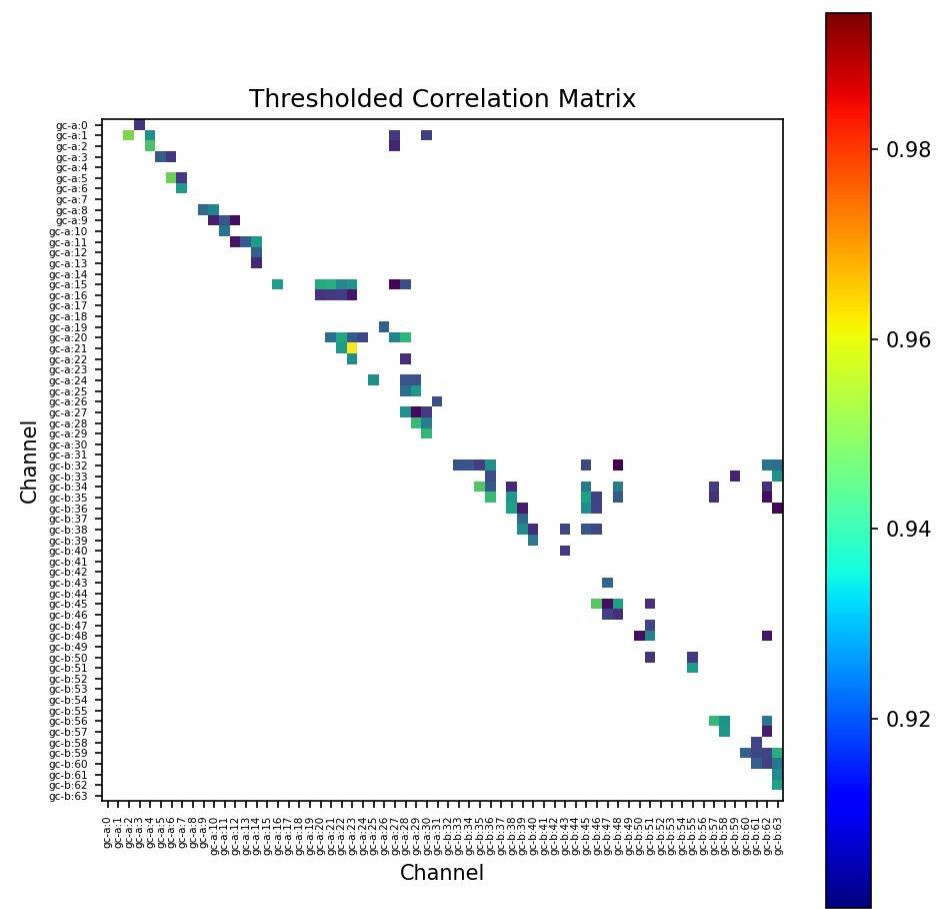
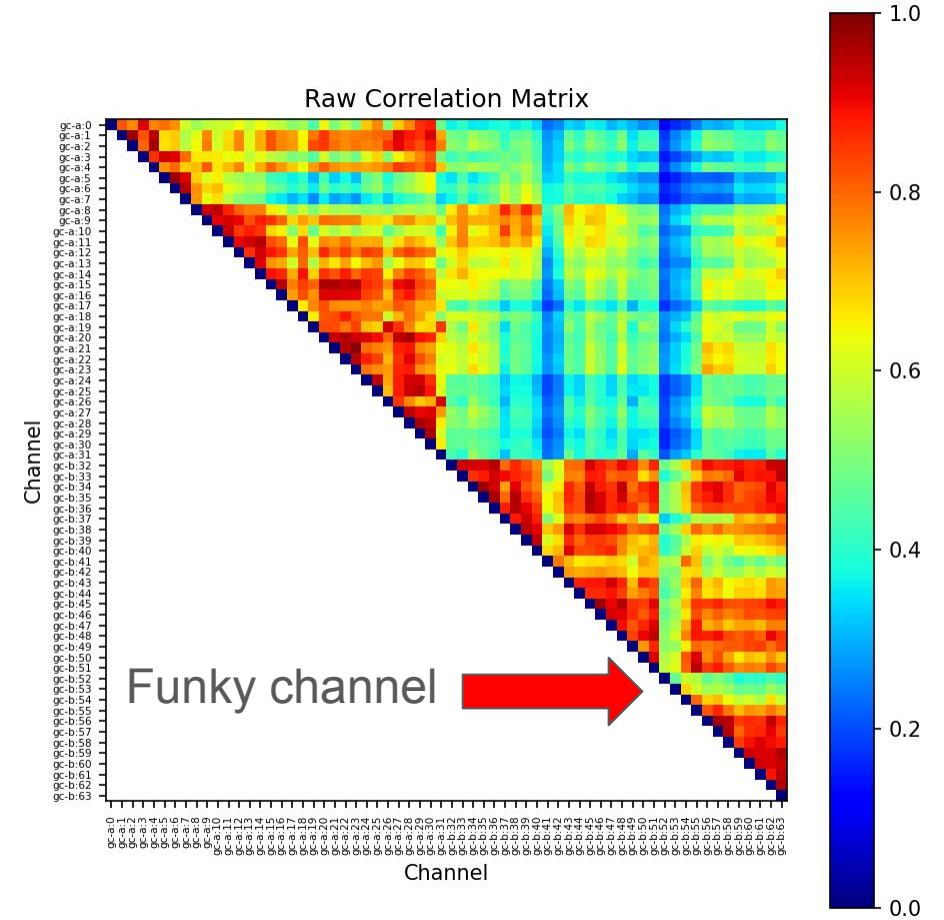
	A	B	C	D	E	F
1	dig_in_num	dig_in_name	taste	start_taste	end_taste	abs_trial_num
2	2	board-DIN-02	ca	1529342	1534149	0
3	1	board-DIN-01	suc	2134755	2140163	1
4	2	board-DIN-02	ca	2740485	2745292	2
5	3	board-DIN-03	qhcl	3345897	3350704	3
6	1	board-DIN-01	suc	3951309	3956717	4
7	3	board-DIN-03	qhcl	4557322	4562129	5
8	2	board-DIN-02	ca	5162734	5167540	6
9	0	board-DIN-00	nacl	5768145	5770699	7
10	0	board-DIN-00	nacl	6370774	6373328	8
11	3	board-DIN-03	qhcl	6973933	6978739	9
12	2	board-DIN-02	ca	7579344	7584150	10
13	0	board-DIN-00	nacl	8184755	8187309	11
14	2	board-DIN-02	ca	8787609	8792416	12
15	2	board-DIN-02	ca	9392661	9397467	13
16	2	board-DIN-02	ca	9998058	1E+07	14
17	0	board-DIN-00	nacl	1.1E+07	1.1E+07	15

Dig_in_num_taste,
Dig_in_name_taste,
Taste,start_taste,
End_taste,
Abs_trial_num,
Taste_rel_trial_num,
Dig_in_num_laser,
Dig_in_name_laser,
Laser,
Start_laser,
End_laser,
Laser_duration,
Laser_lag,
Start_taste_ms,
End_taste_ms,
Start_laser_ms,
End_laser_ms,
Laser_duration_ms,
laser_lag_ms

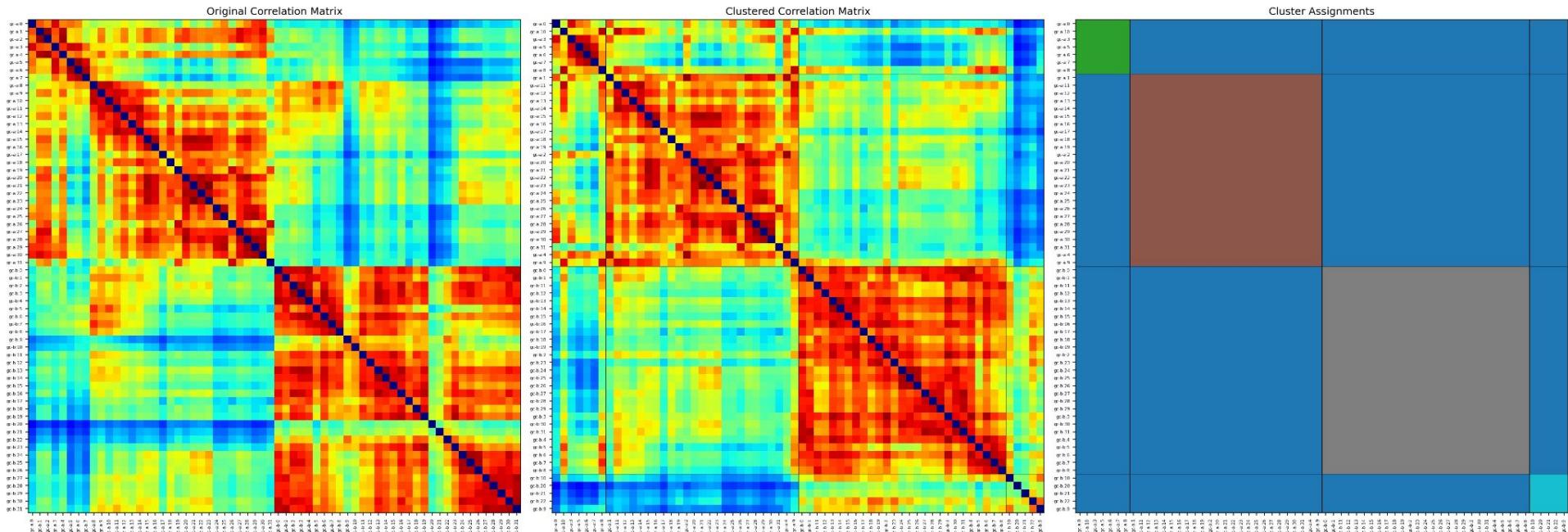
Lots of things in QA_outputs dir



Common Average Reference outputs

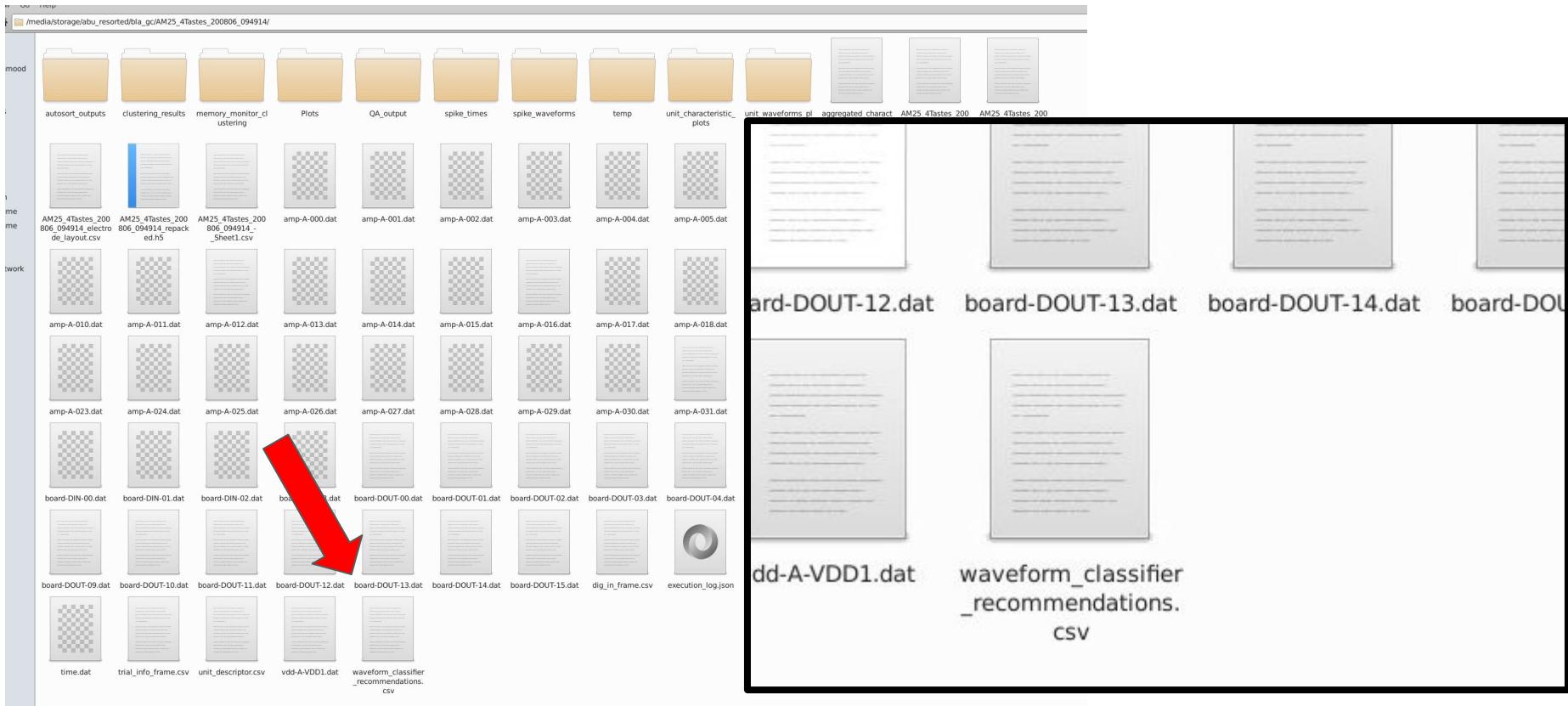


Automated sub-clustering of CAR groups

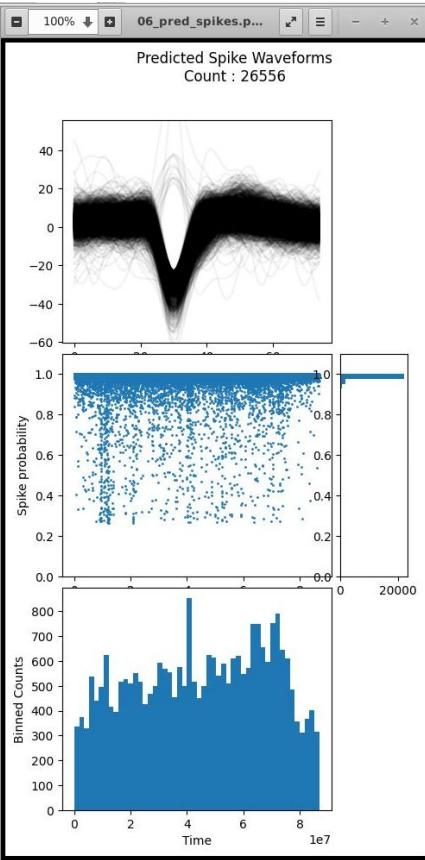


of clusters determined using BIC on K-Means

Classifier recommendations



Classifier recommendations



WPS Office waveform_classifier_recommendations +

Menu Insert Page Layout

Cut Copy Format Painter

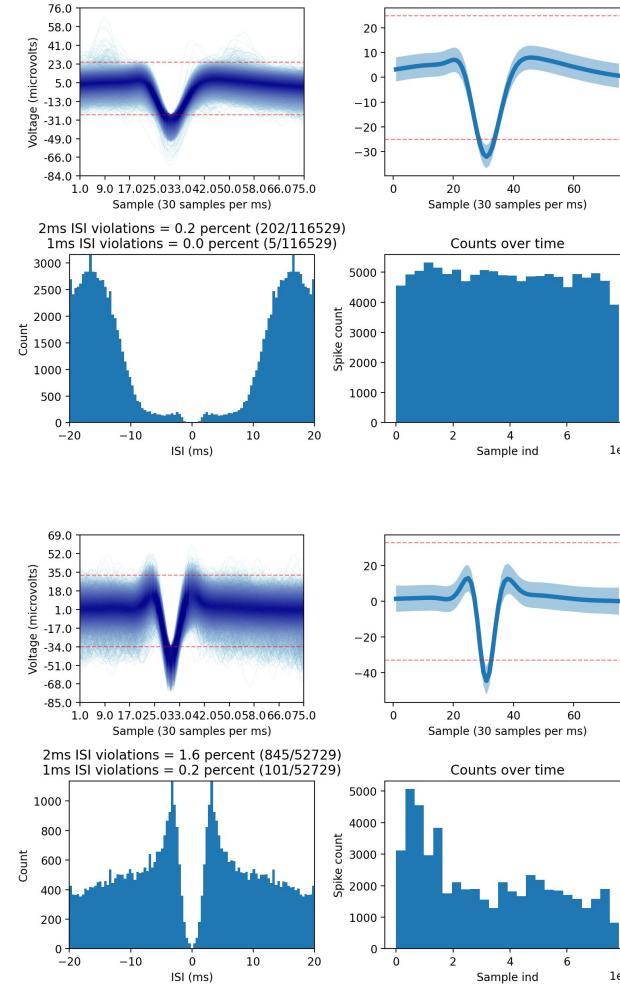
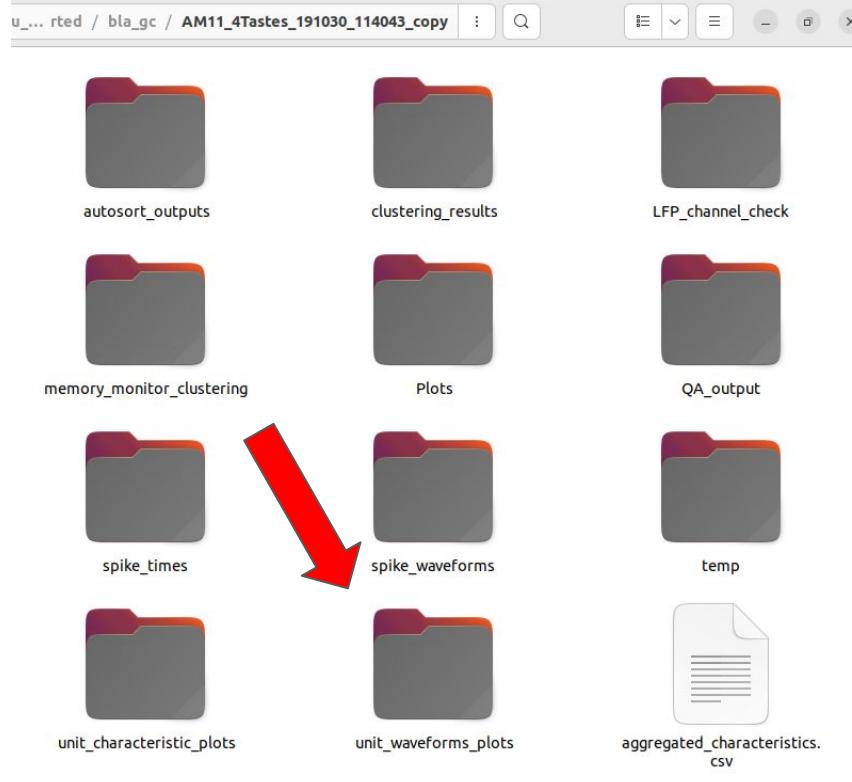
Calibri 11 A A- Font icons

Paste Clipboard icons

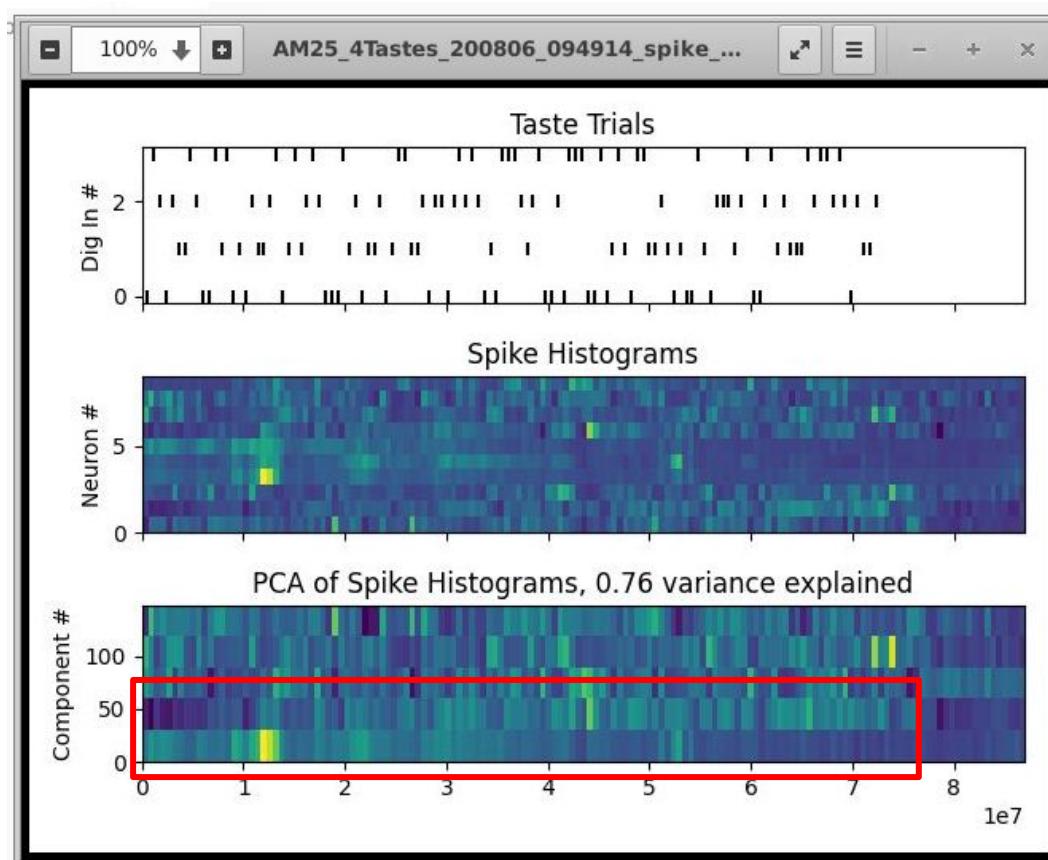
A	B	C	D	E	F
electrode	count	mean_prob	percentile_90	percentile_95	99%
1	1	2110	0.878	0.377	0.997
3	3	30760	0.965	0.821	0.999
6	6	26556	0.971	0.288	0.987
7	7	7363	0.66	0.286	0.979
8	8	2316	0.635	0.313	0.995
9	9	4432	0.789	0.359	0.996
10	10	14113	0.845	0.375	1
11	11	12135	0.908	0.301	0.995
12	12	3780	0.739	0.29	0.98
13	13	2243	0.647	0.292	0.985
14	14	2198	0.654	0.291	0.981
16	16	28631	0.96	0.792	0.999
19	19	5459	0.648	0.288	0.98
21	21	5257	0.654	0.322	0.996
22	22	2158	0.83	0.298	0.99
23	23	3946	0.687	0.303	0.996
25	25	16421	0.967	0.294	0.985
26	26	2245	0.727	0.29	0.968
27	27	5631	0.624	0.276	0.998
28	28	9271	0.955	0.412	0.999
29	29	5088	0.659	0.36	0.996
30	30	2583	0.911	0.294	0.985
31	31	2219	0.843	0.36	0.999
32					

All about unit stability

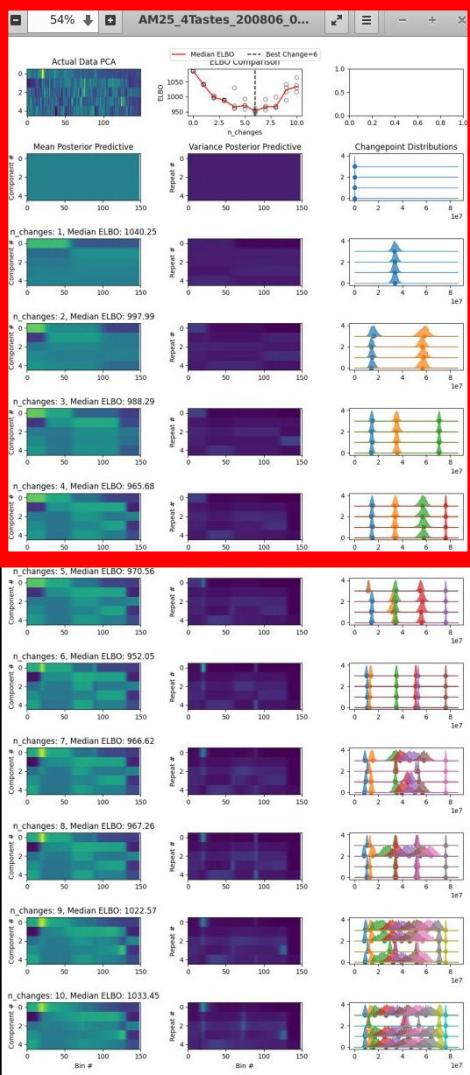
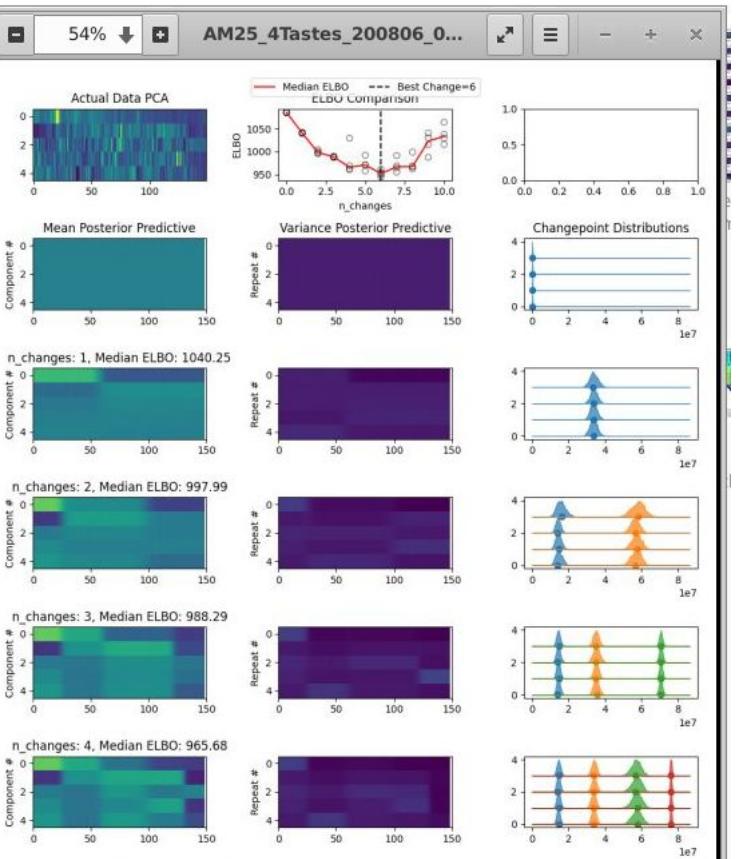
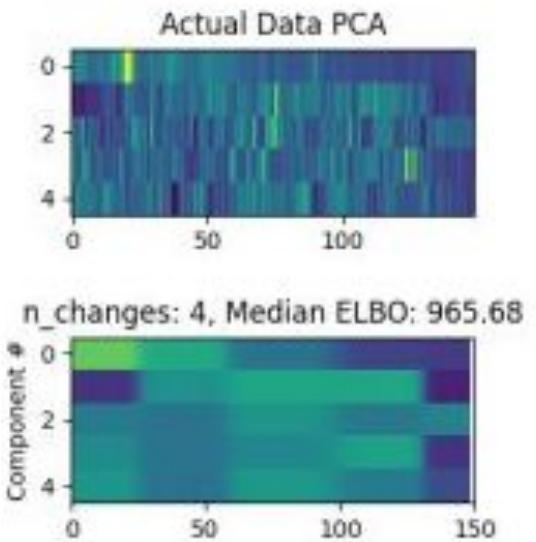
Sorted units spike-counts



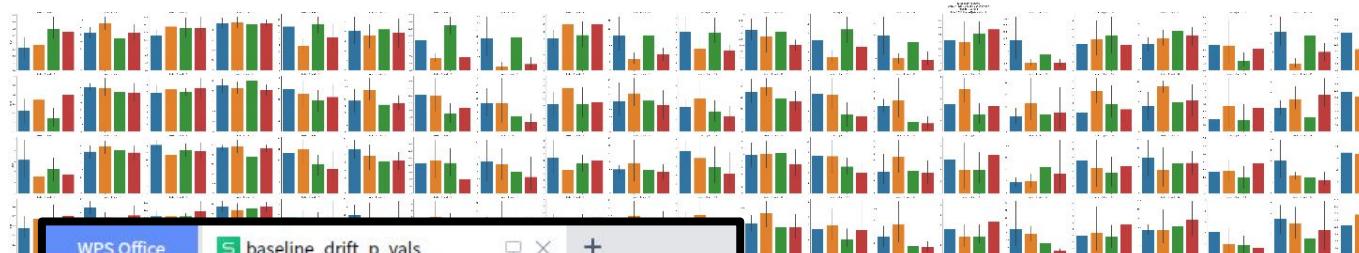
Bulk firing variability over session



Bulk firing variability over session



Changes in baseline rates



WPS Office S baseline_drift_p_vals

Home Insert Page

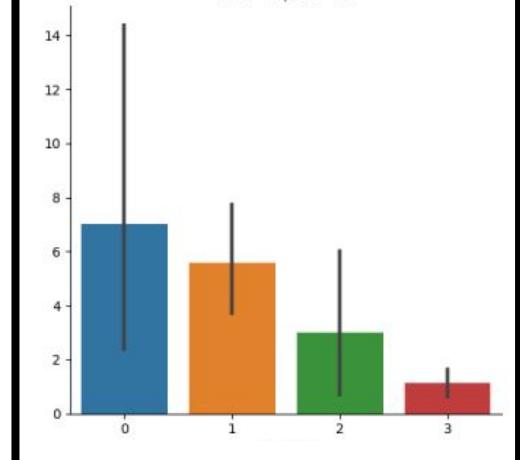
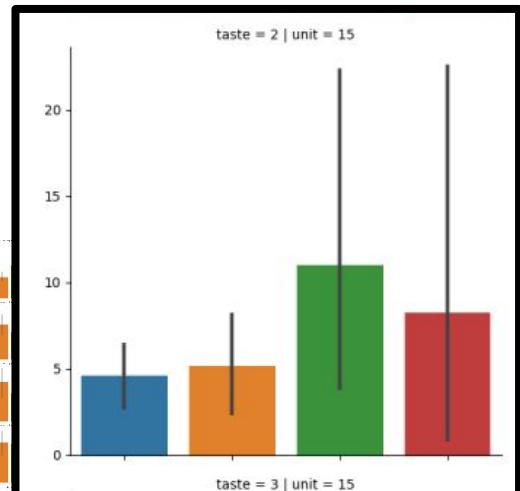
Cut Copy Format Painter

Calibri 11 A+ A-

B I U H A ◊

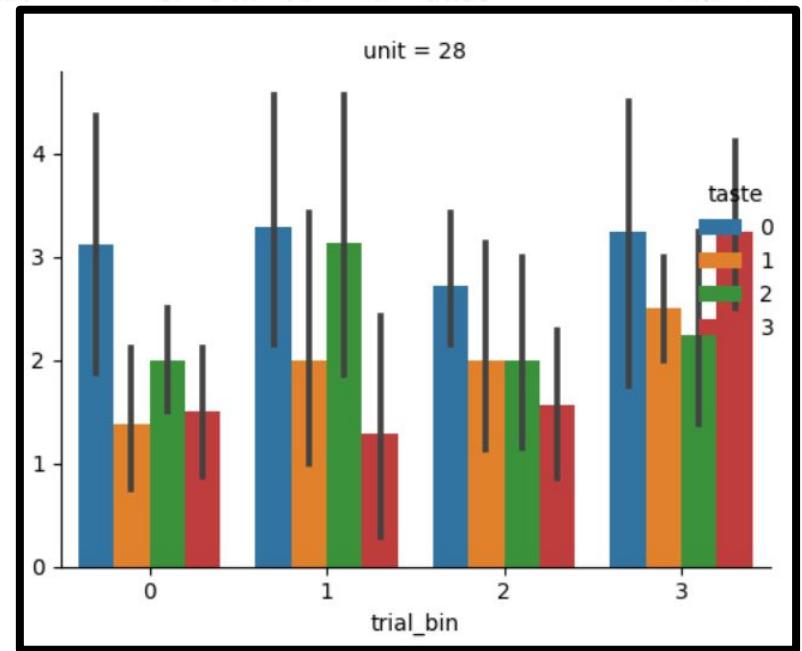
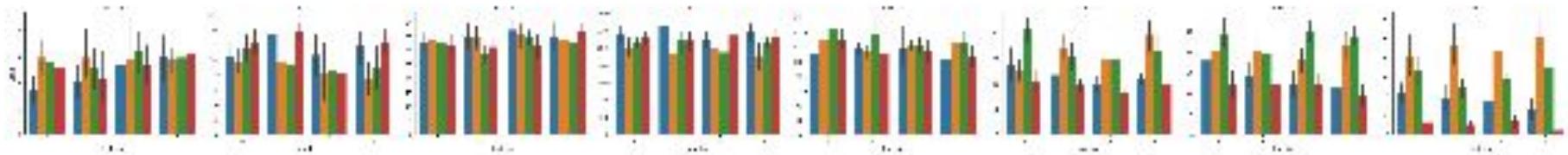
A1 fx

	A	B	C	D	E	F
1	trial_bin	taste	trial_bin * Residual	unit		
2	0	0.67601	0.03115	0.06798		0
3	1	0.10324	0.77822	0.4861		1
4	2	0.68838	0.49648	0.61009		2
5	3	0.84285	0.76545	0.08746		3
6	4	0.22501	0.49345	0.39878		4
7	5	0.23565	0.27651	0.71571		5
8	6	0.0747	0.87517	0.00599		6
9	7	0.03895	0.732	0.26052		7
10	8	0.64293	0.86977	0.36059		8
11	9	0.50257	0.16761	0.2267		9

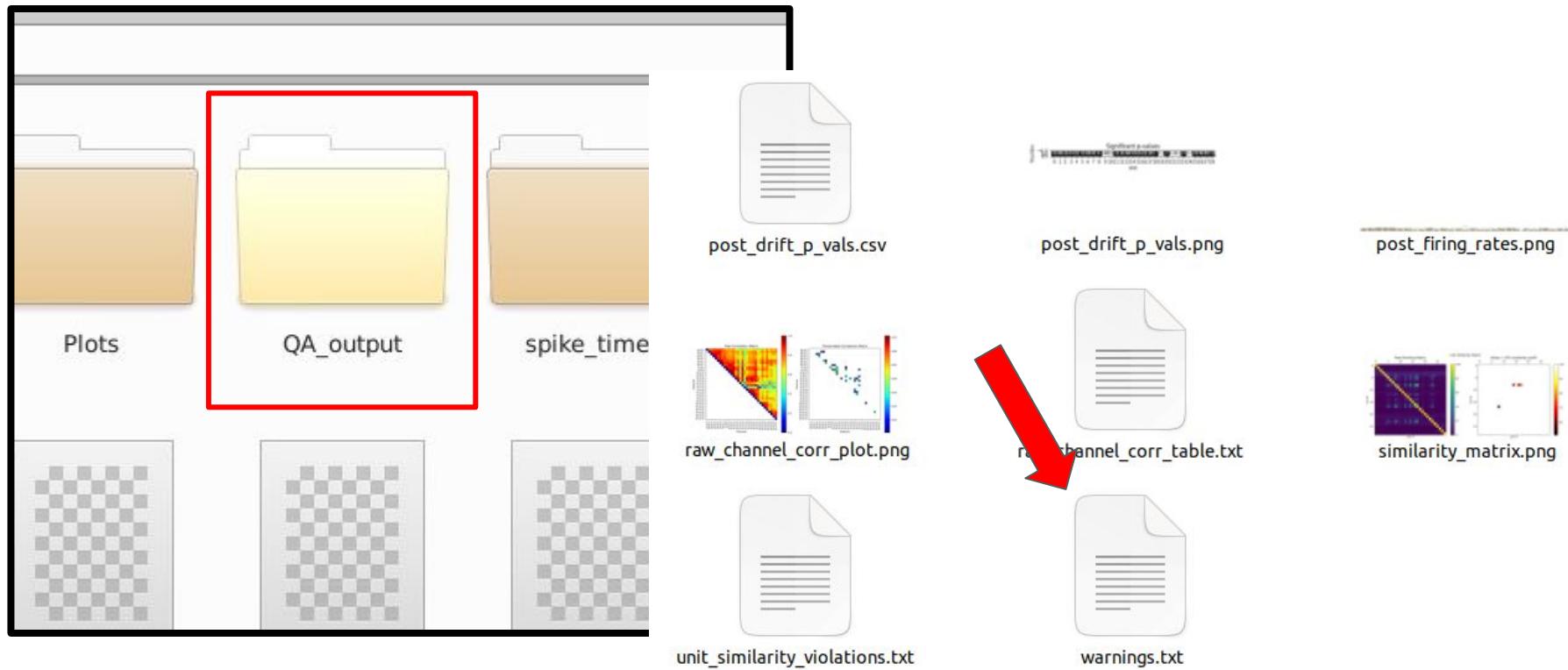


Trial Bin

Changes in baseline rates



Warnings file



Ephys_data submodule

- Data handling and processing
- Visualizing
 - Rasters
 - Firing rate grid plots

```
from utils.ephys_data.ephys_data import ephys_data

# Initialize with data directory

data = ephys_data(data_dir='/path/to/data')
```

Ephys_data submodule

Loading spikes	<code>data.get_spikes()</code>
Extracting LFPs, calculating STFT	<code>data.get_lfps()</code> <code>data.get_stft()</code>
Calculating firing rates + unit properties	<code>data.get_firing_rates()</code> <code>data.calc_palatability()</code>
Separating data by region	<code>data.get_region_units()</code> <code>region_spikes = data.return_region_spikes('region_name')</code> <code>region_firing = data.get_region_firing('region_name')</code> <code>region_lfps, region_names = data.return_region_lfps()</code>
Separating data by laser condition	<code>data.separate_laser_data()</code> <code># Access separated data</code> <code>on_spikes = data.on_spikes</code> <code>off_spikes = data.off_spikes</code> <code>on_firing = data.on_firing</code> <code>off_firing = data.off_firing</code>
Loading experimental + unit metadata	<code>data.get_info_dict()</code> <code>data.get_trial_info_frame()</code> <code>data.get_unit_descriptors()</code>

What to do if you have noise

1. Cry 😭
2. Try out these folk remedies

Reduce spike detection threshold

- <data_dir>/<session_name>.params
- Set “waveform_threshold” to 3 or 4

```
1 {  
2     "max_parallel_cpu": 16,  
3     "waveform_threshold": 5|,   
4     "use_rolling_threshold": true,  
5     "rolling_threshold_window": 5.0,  
6     "rolling_threshold_step": 5.0,  
7     "voltage_cutoff": 10000,  
8     "max_breach_rate": 1,  
9     "max_secs_above_cutoff": 60,  
10    "max_mean_breach_rate_persec": 100,  
11    "wf_amplitude_sd_cutoff": 3,  
12    "bandpass_lower_cutoff": 300,  
13    "bandpass_upper_cutoff": 3000,  
14    "spike_snapshot_before": 1
```

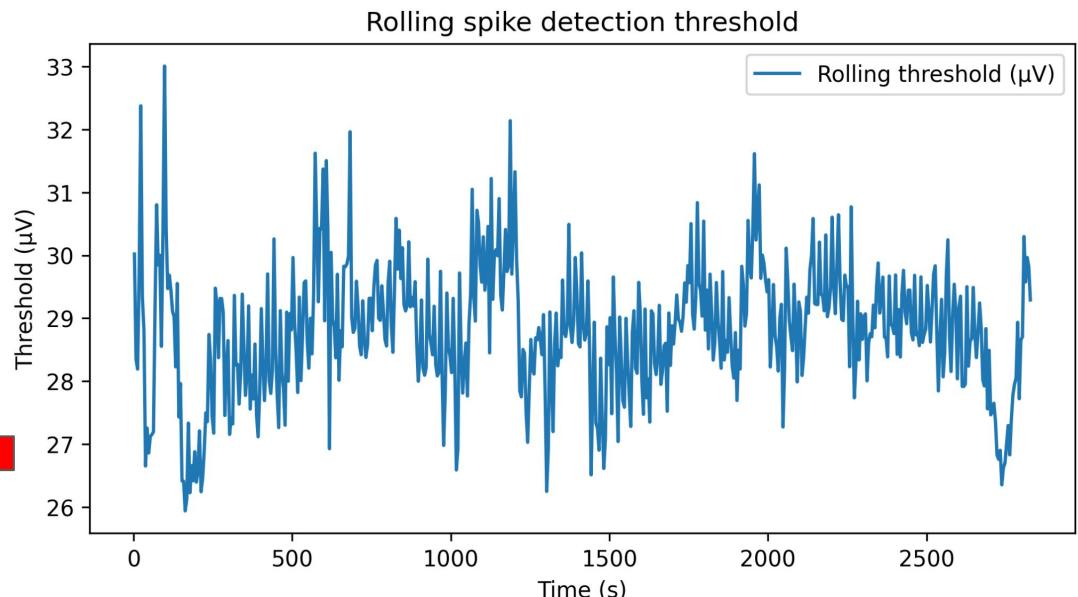
Adjust classifier settings

- blech_clust/params/waveform_classifier_params.json
- Set “throw_out_noise” to true
- Can override classifier threshold by setting “override” to true
 - This may increase false positive waveforms sorted

```
File: waveform_classifier_params.json
1  {
2      "use_neuRecommend": true,
3      "use_classifier": true,
4      "throw_out_noise": false,
5      "min_suggestion_count": 2000,
6      "classifier_threshold_override": {
7          "override": false,
8          "threshold": 0.8
9      }
10 }
```

Use rolling-window threshold (experimental)

```
1 {
2     "max_parallel_cpu": 16,
3     "waveform_threshold": 5|,
4     "use_rolling_threshold": true,
5     "rolling_threshold_window": 5.0,
6     "rolling_threshold_step": 5.0,
7     "voltage_cutoff": 10000,
8     "max_breach_rate": 1,
9     "max_secs_above_cutoff": 60,
10    "max_mean_breach_rate_persec": 100,
11    "wf_amplitude_sd_cutoff": 3,
12    "bandpass_lower_cutoff": 300,
13    "bandpass_upper_cutoff": 3000,
14    "spike_snapshot_before": 1
```



Housekeeping

- Please upload your code to katzlabbrandeis github
 - You can also have your repository forked (copied) to katzlabbrandeis
 - It DOESN'T have to be clean code...the repository can be kept private