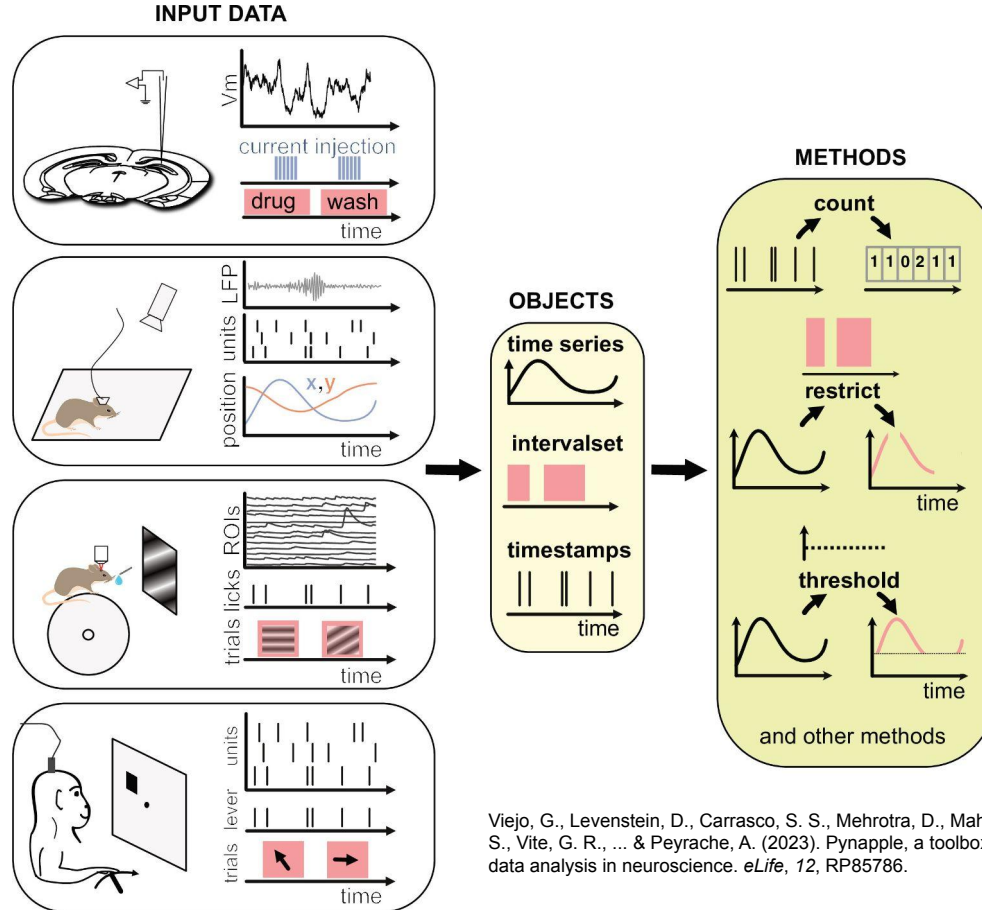


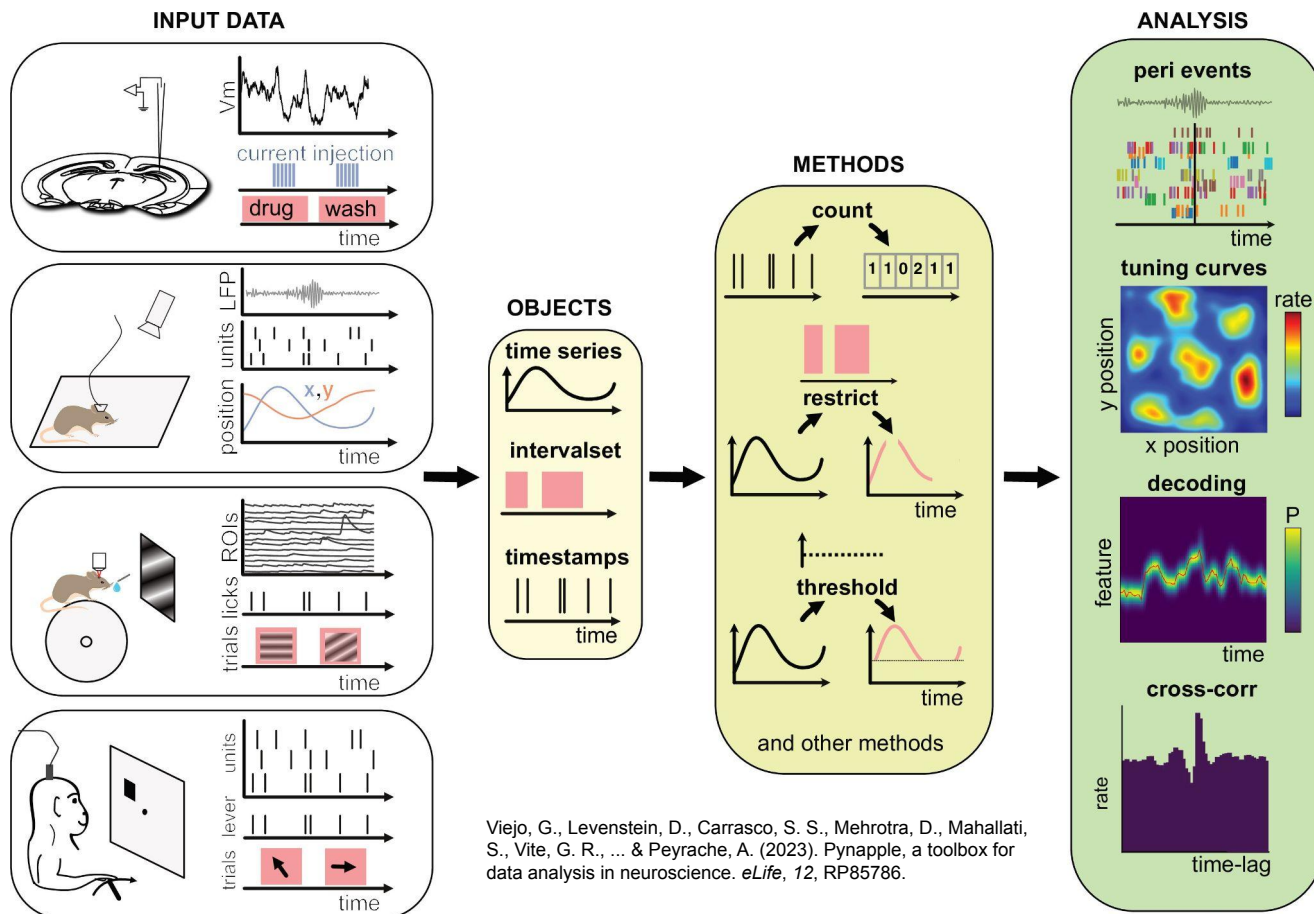
Advanced methods in Pynapple: Standard analysis in systems neuroscience

SFN Workshop, November 2025

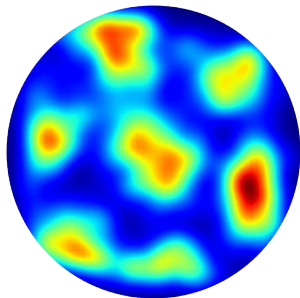
Sarah Jo Venditto



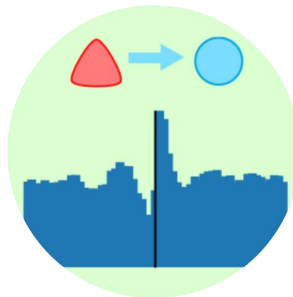
Viejo, G., Levenstein, D., Carrasco, S. S., Mehrotra, D., Mahallati, S., Vite, G. R., ... & Peyrache, A. (2023). Pynapple, a toolbox for data analysis in neuroscience. *eLife*, 12, RP85786.



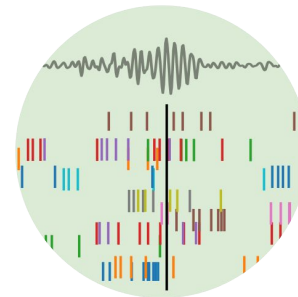
Tuning curves



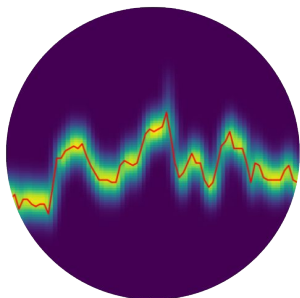
Correlation



Perievent



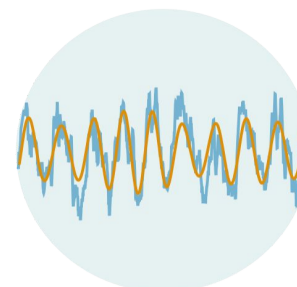
Decoding

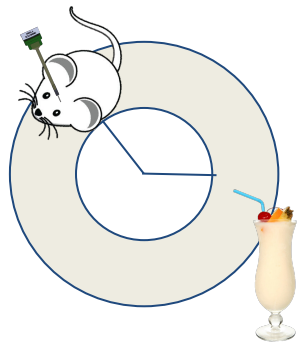


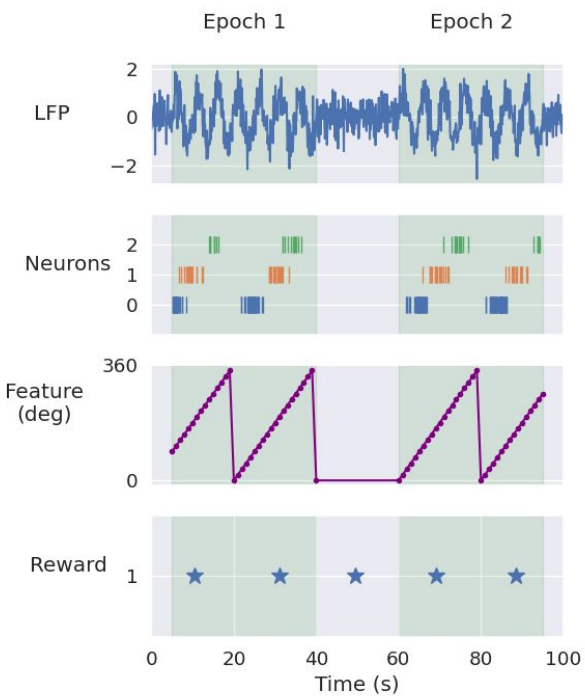
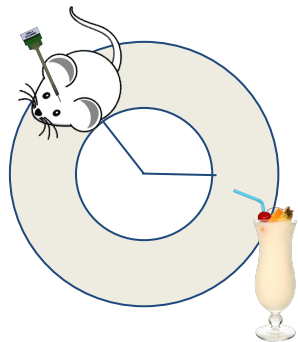
Wavelets

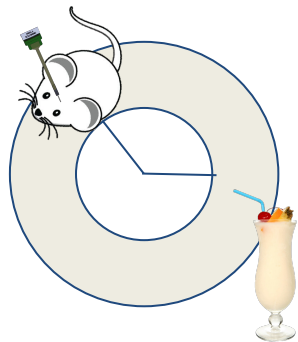


Filtering









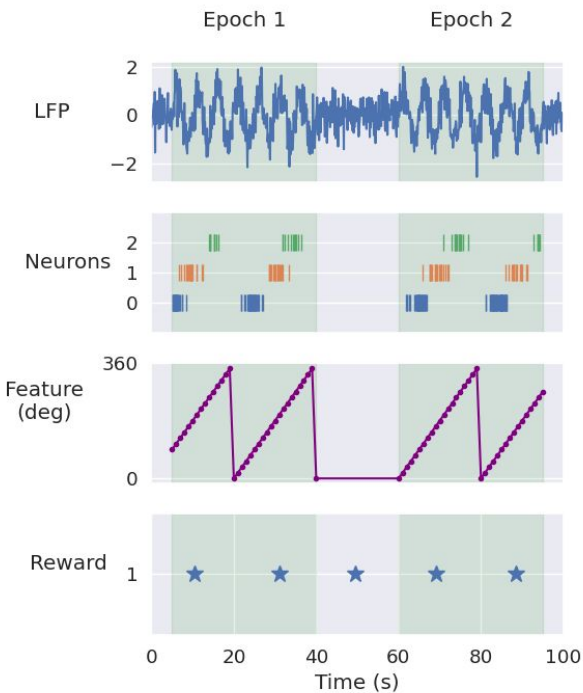
```
In [48]: ep
Out[48]:
      start  end
0         5.0 40.0
1        60.0 95.0
```

```
In [11]: lfp
Out[11]:
Time (s)
-----
0.0      -0.655732
0.1      -0.175004
0.2       0.55584
0.3       0.347774
0.4       0.0922895
...
99.5     -0.333844
99.6      0.145915
99.7     -0.377362
99.8     -0.466354
99.9      0.418279
dtype: float64, shape: (1000,)
```

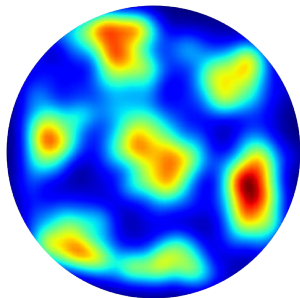
```
In [49]: spikes
Out[49]:
      Index  Freq. (Hz)
-----
0          0         2.07
1          1         1.08
2          2         0.66
```

```
In [50]: feature
Out[50]:
Time (s)
0.0      0.0
1.0     18.0
2.0     36.0
3.0     54.0
4.0     72.0
...
95.0    270.0
96.0    288.0
97.0    306.0
98.0    324.0
99.0    342.0
Length: 100, dtype: float64
```

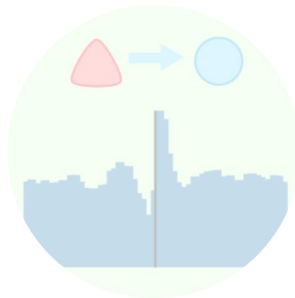
```
In [5]: reward
Out[5]:
Time (s)
10.841068764
31.582233204
51.683610725
71.239549326
91.661298984
shape: 5
```



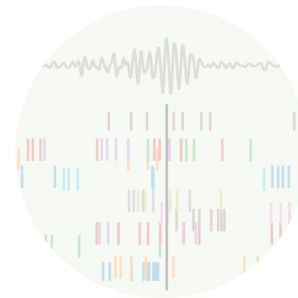
Tuning curves



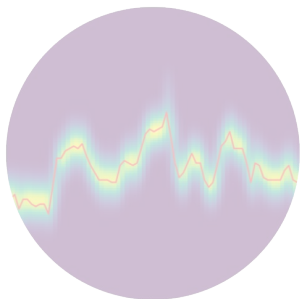
Correlation



Perievent



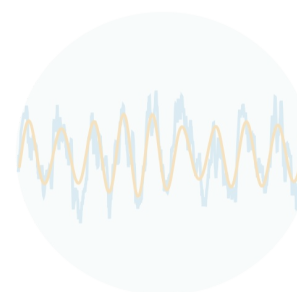
Decoding

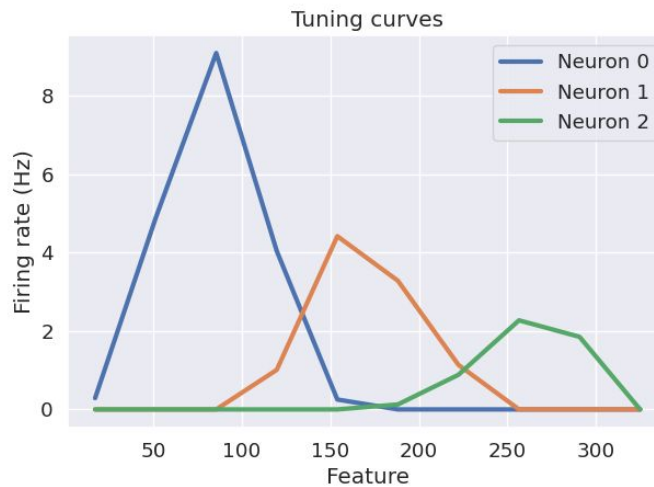
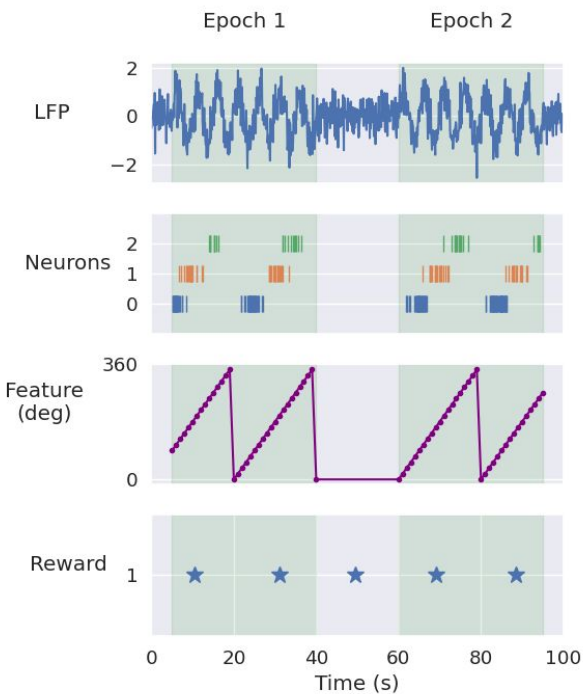
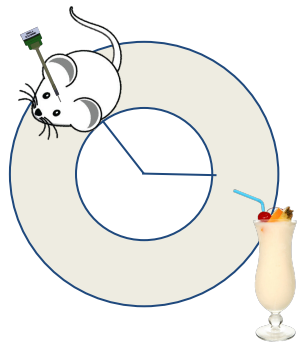


Wavelets



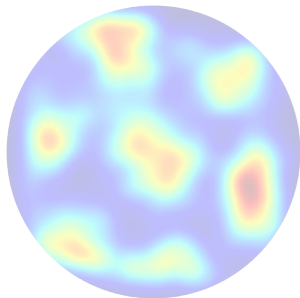
Filtering



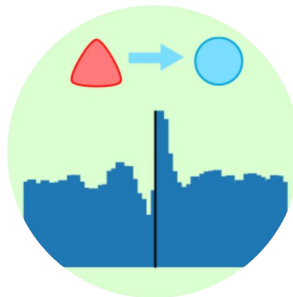


```
tuning_curves = nap.compute_tuning_curves(
    data=spikes,
    features=feature,
    nb_bins=120,
    epochs=ep,
    range=(0, 360),
    feature_names=["feature"]
)
```

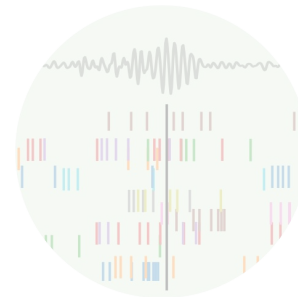
Tuning curves



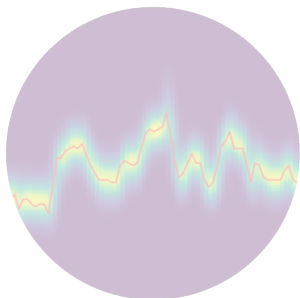
Correlation



Perievent



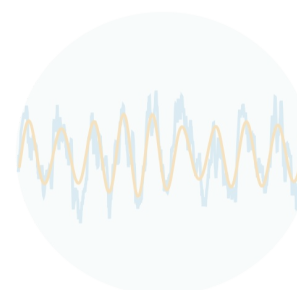
Decoding

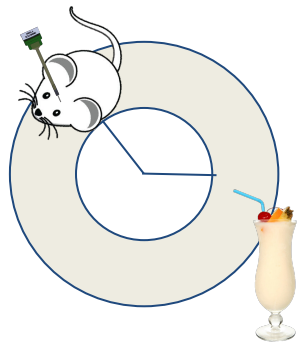


Wavelets



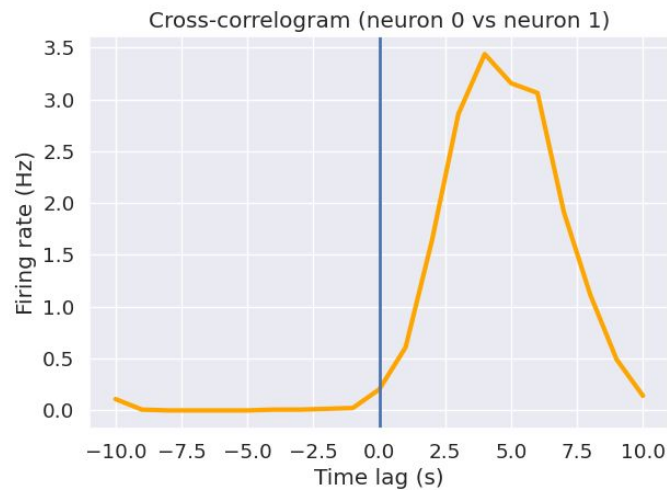
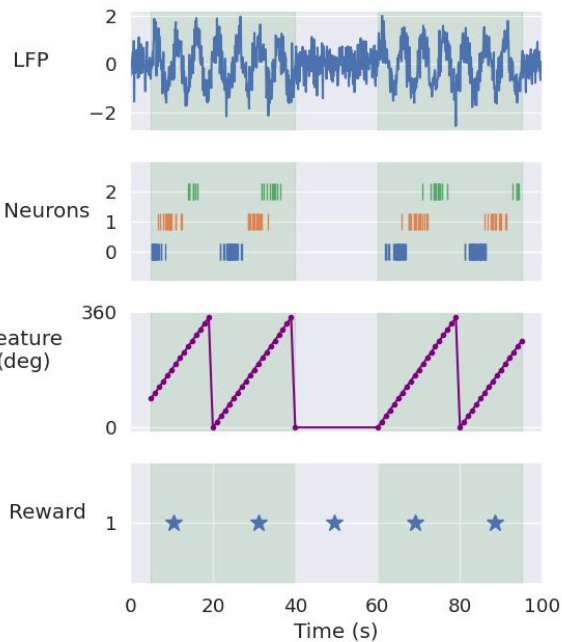
Filtering





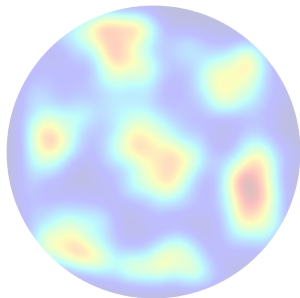
Epoch 1

Epoch 2

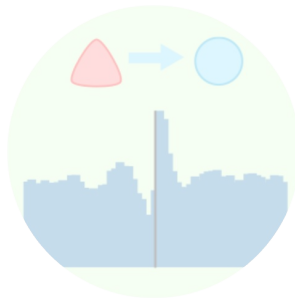


```
cross_corr = nap.compute_crosscorrelogram(
    spikes,
    binsize = 1,
    windowsize = 10,
    ep = ep,
    norm = False)
```

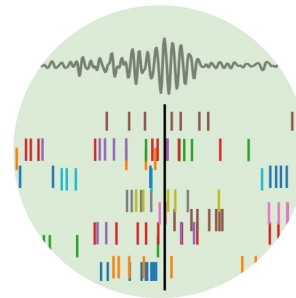
Tuning curves



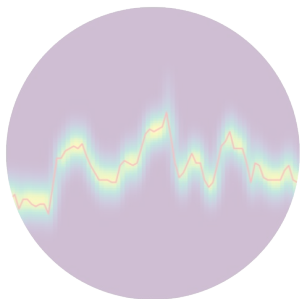
Correlation



Perievent



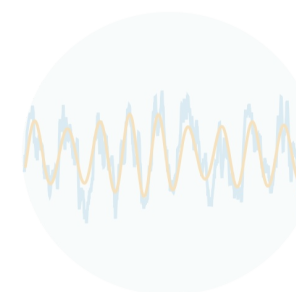
Decoding

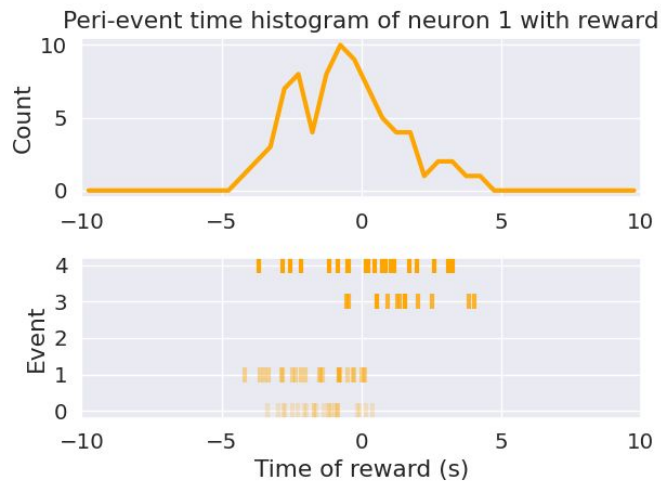
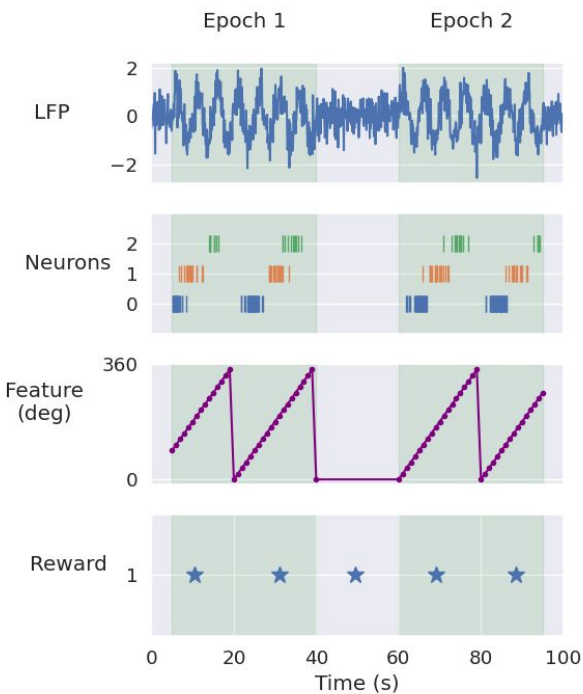
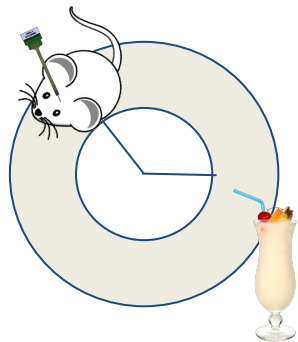


Wavelets



Filtering



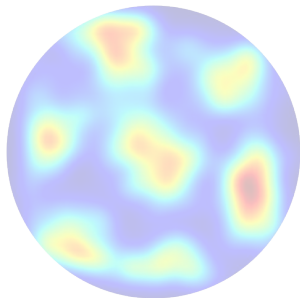


```
In [20]: perievent
Out[20]:
```

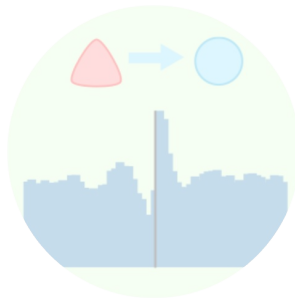
Index	Freq. (Hz)	ref_times
0	1.15	10.79
1	1.15	31.1445
2	nan	49.1539
3	0.65	67.7832
4	1	89.4536

```
perievent = nap.compute_perievent(
    spikes,
    tref = reward,
    minmax=(-10, 10))
```

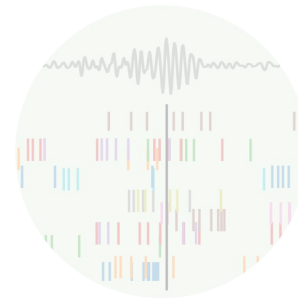
Tuning curves



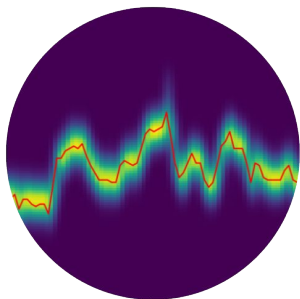
Correlation



Perievent



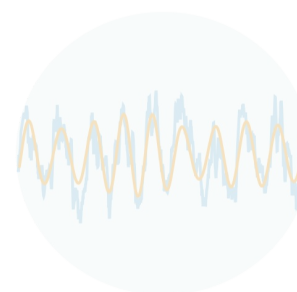
Decoding

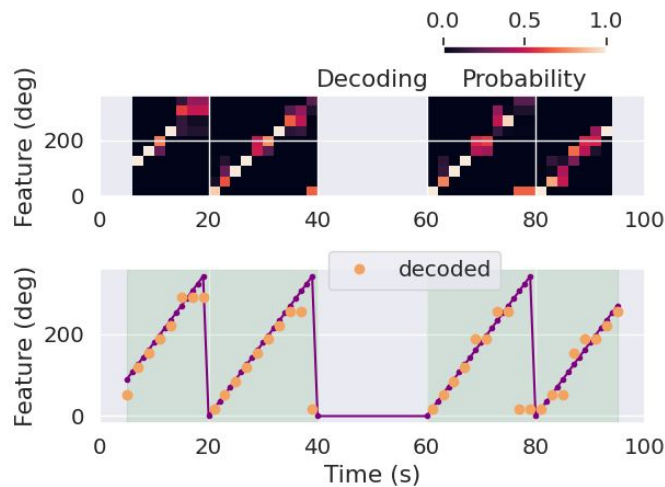
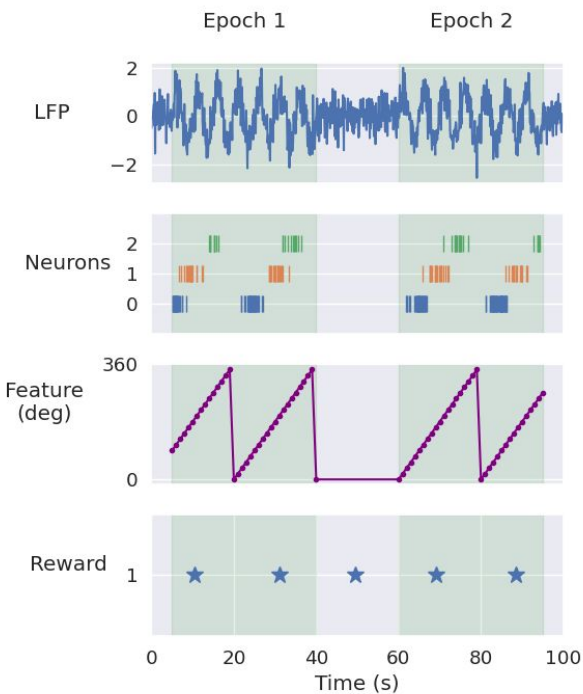
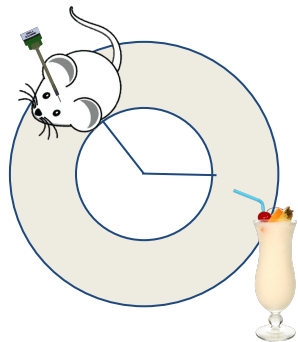


Wavelets

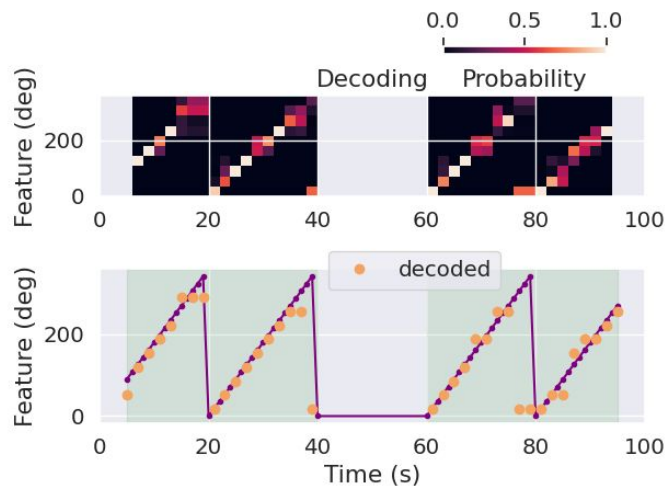
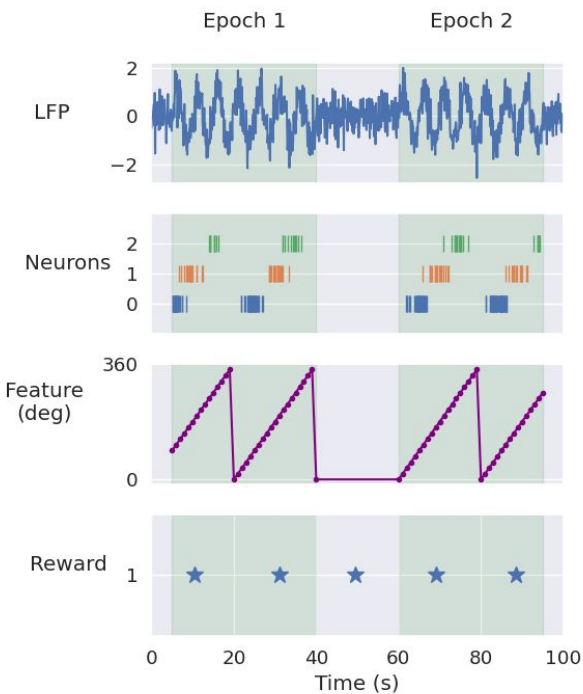
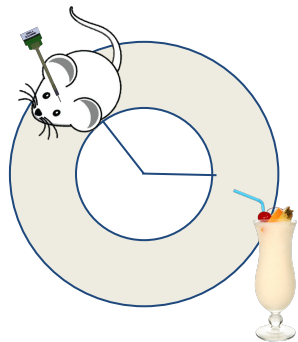


Filtering



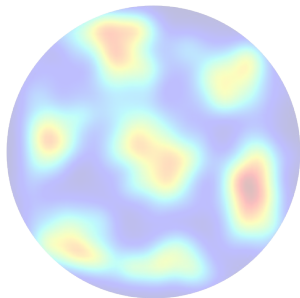


```
decoded, proba = nap.decode_bayes(
    tuning_curves,
    data,
    epochs=ep,
    bin_size=2 # seconds
)
```

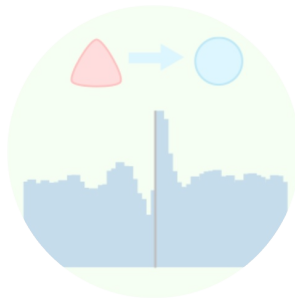


```
decoded, dist = nap.decode_template(
    tuning_curves,
    data,
    epochs=ep,
    bin_size=2,
    metric="correlation")
```

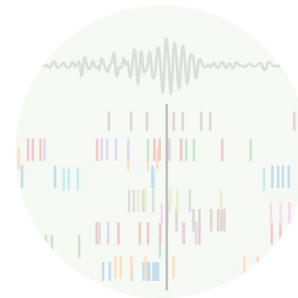

Tuning curves



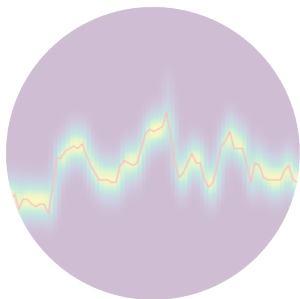
Correlation



Perievent



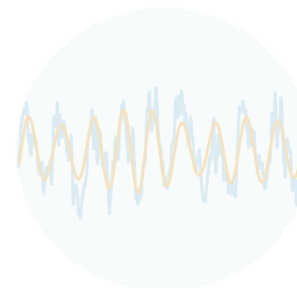
Decoding

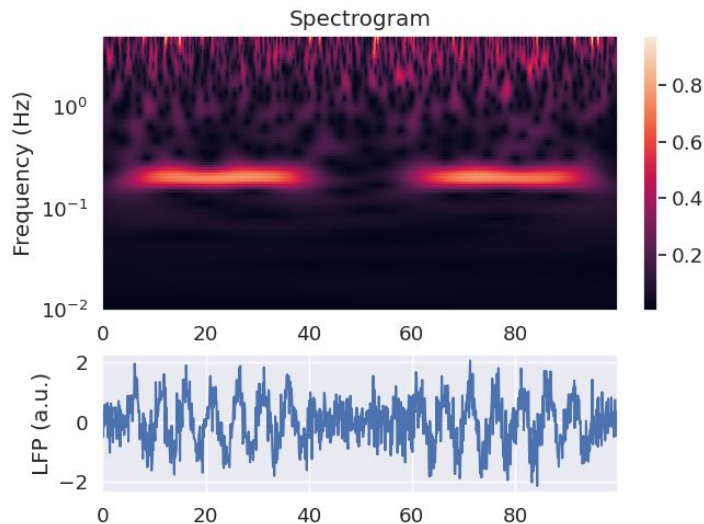
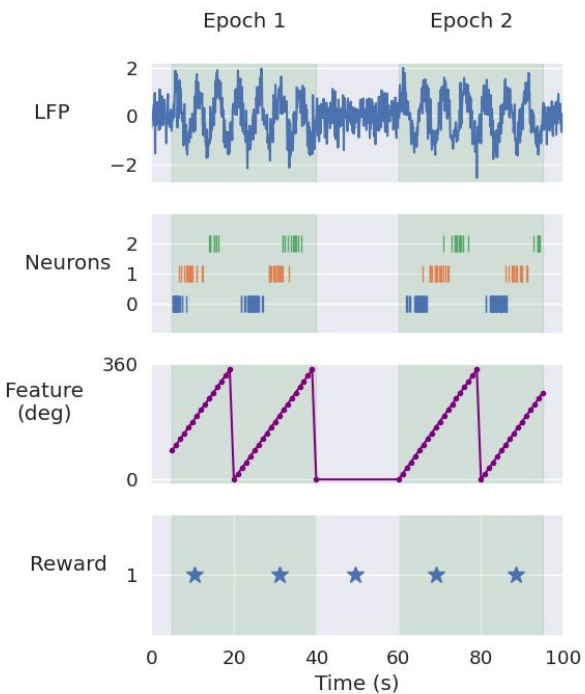
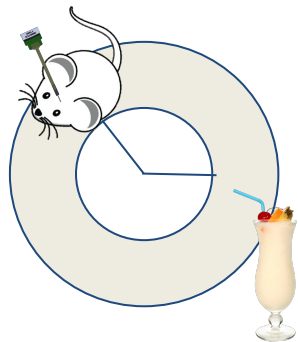


Wavelets



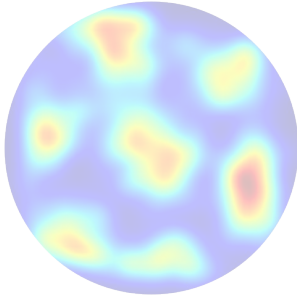
Filtering



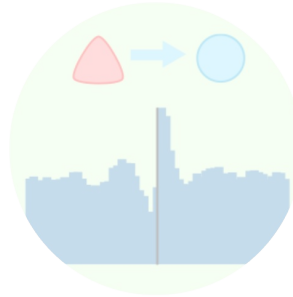


```
cwt = nap.compute_wavelet_transform(
    sig = lfp,
    freqs = [0.01, 0.1, 1, 10],
    fs = 10.0)
```

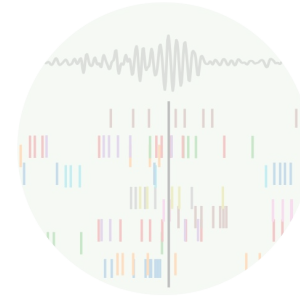
Tuning curves



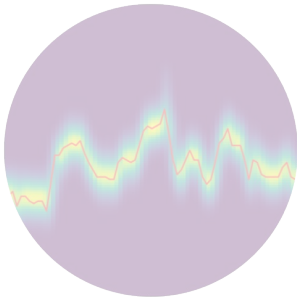
Correlation



Perievent



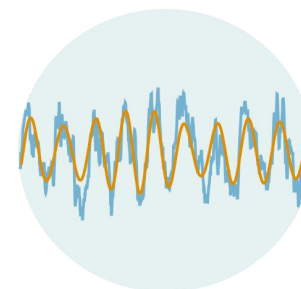
Decoding

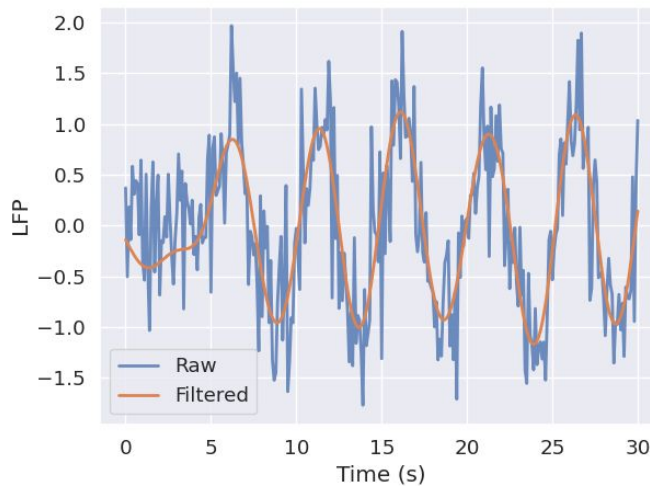
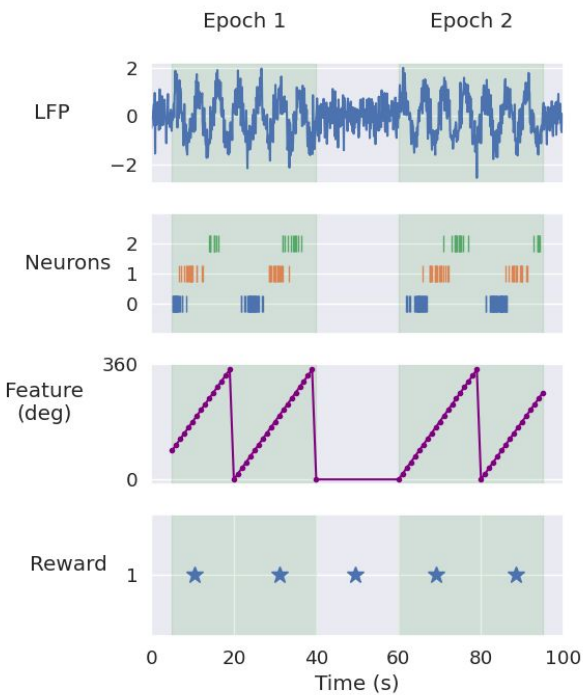
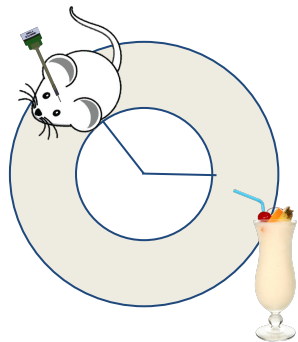


Wavelets

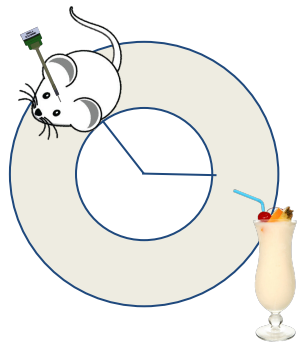


Filtering





```
flfp = nap.apply_bandpass_filter(
    data= lfp,
    cutoff = (0.05, 0.3),
    fs=10.0)
```



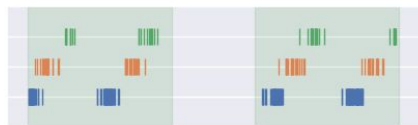
Epoch 1

Epoch 2

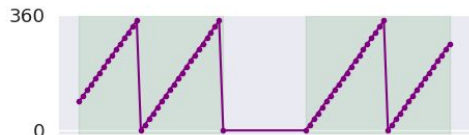
LFP



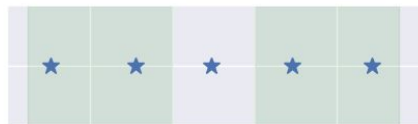
Neurons



Feature (deg)

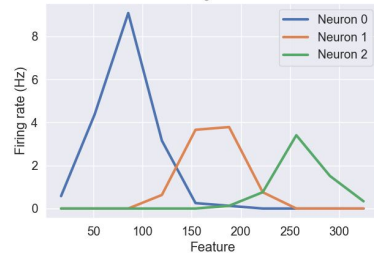


Reward

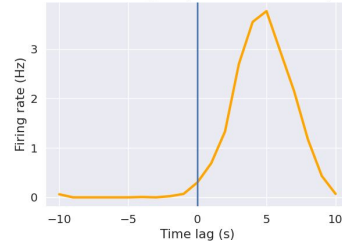


Time (s)

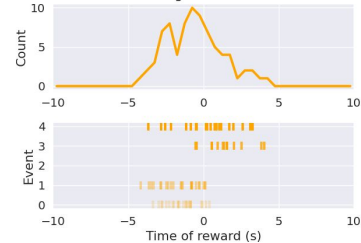
Tuning curves



Cross-correlogram (neuron 0 vs neuron 1)

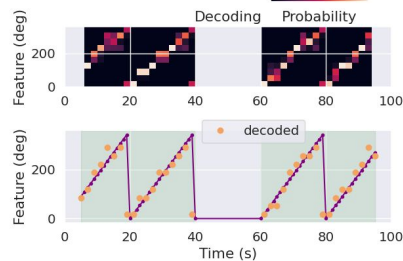


Peri-event time histogram of neuron 1 with reward

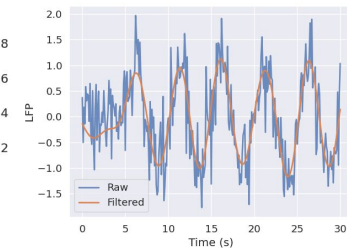
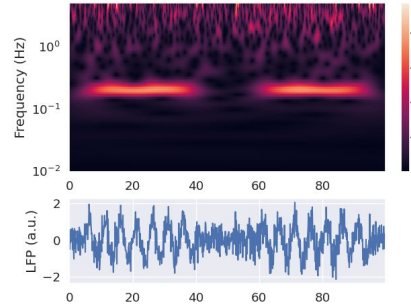


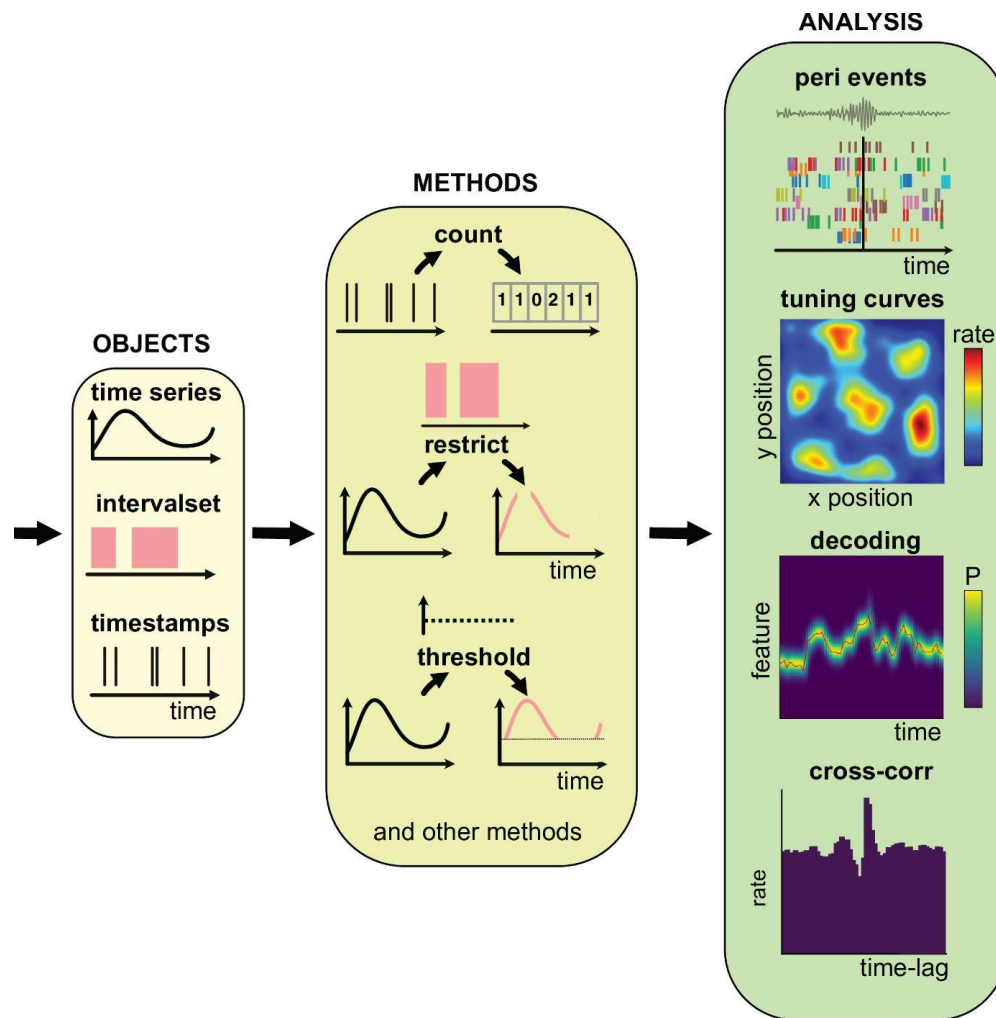
0.0 0.5 1.0

Decoding Probability



Spectrogram





Additional functions

- **Fourier transform and power spectral density**

```
nap.compute_fft      nap.compute_power_spectral_density  
nap.compute_mean_power_spectral_density
```

- **Creating trial-based tensors (and linear trial time warping)**

```
nap.build_tensor      nap.warp_tensor
```

- **Randomization of timestamps**

```
nap.shift_timestamps      nap.shuffle_ts_intervals  
nap.jitter_timestamps      nap.resample_timestamps
```

- **Interspike interval distributions**

```
nap.compute_isi_distribution
```

- **LFP oscillatory event detections**

```
nap.filtering.detect_oscillatory_events (in the future will be nap.detect_oscillatory_events)
```