

1 (Work in progress)

1.1 Overview

Project summary; 'Elephant-Nest-Integration', 'Modular Science', 'SIMS-LS'. Figure 1.1 shows an overview, with a detailed description of each component below.

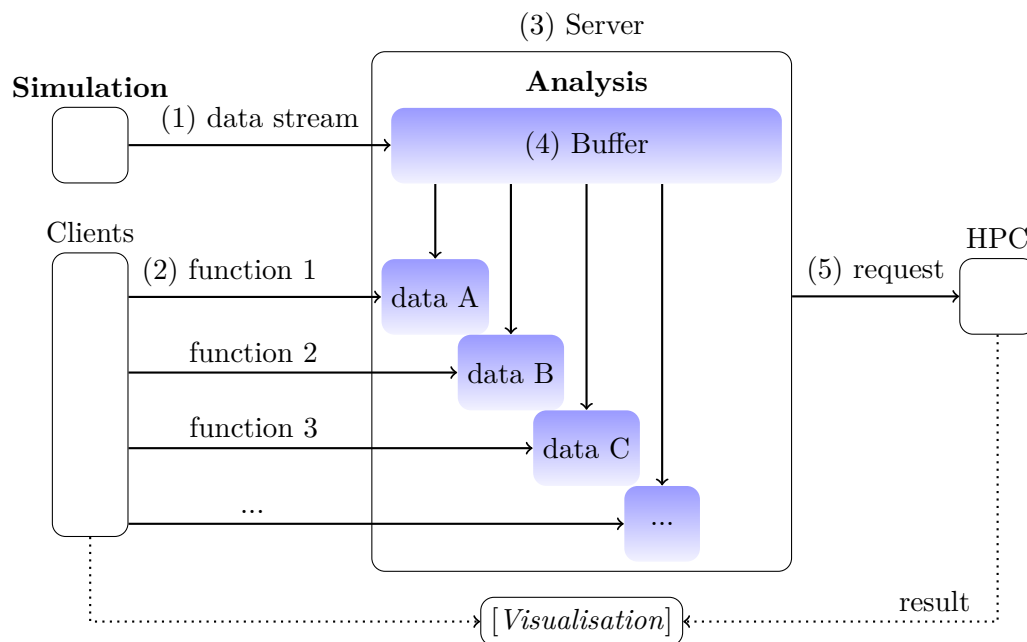


Figure 1.1: Project overview. The Server (3) provides different Elephant functions for analysis. Clients (2) can request those. Data is then taken from the simulation output stream (1), stored in a buffer (4), further processed depending on the function and forwarded to HPC resources (5).

1.2 Details

(1) Input data stream from simulation:

- Protocol -> MUSIC (?)
- Input/Output:
NEST <-> Elephant (, NEST <-> TVB, TVB <-> Elephant)
- Handle input
- Handle data loss

(2) Multiple Clients:

- Multithreading (in Python?) -> multiprocessing
- Protocol

(3) Server:

- Server architecture
- Running where? -> (for now) running on SC (e.g. Jureca)
- Logging

(4) Buffer:

- Buffer-size -> (for now) expect buffer to handle all data, buffer large enough for complete simulation output
- Extract relevant data for each Elephant analysis provided.
- Input conversion -> Simulation output to Elephant input (NEO Framework)

(5) HPC:

- Connect to SC resources -> (for now) server running directly on SC (e.g. Jureca)

2 Elephant functions

List of Elephant functions.

- Statistics:
 - ISI - Inter Spike Intervals
 - Mean Firing Rate
 - CV - Coefficient of Variation (also: LV,CV2)
 - Instantaneous Firing Rate
 - Time Histogram
 - Complexity PDF
- ASSET - Analysis of Sequence of Synchronous Events in massively parallel spikeTrains
- SPADE - spatio-temporal Spike PAttern Detection and Evaluation
- (*UE - Unitary Events*)

ERRORS?:

- *ASSET:*
 - line 376/377: `signal.view(pq.Quantities)[elements_to_keep]`
 - line 631: `mask[~(mask >= -np.inf)] = False`
- *Statistics, IFR:*
 - line 754: `sskernel['optw']` with `'bootstrap = TRUE'`...not needed?
- *SPADE: FIM instead of fast_fca:*
 - line 558ff, `fpgrowth` with `FIM`: keyword argument `min/max` (now) `zmin/zmax` (new version of `fim`?)
 - line 689: `zmax < zmin` (depends on input)
 - line 702: `int obj error: concepts[1][0] -> concepts[1] is int not tuple` (depends on input)
 - line 667: `index out of bounds`

2.1 Required modules and imports

- Neo Framework
imported within elephant
- Numpy
imported within elephant

- Quantities
» *import quantities as pq*
- Elephant
» *import elephant.spike_train_generation as stg*
» *import elephant.asset as asset*
» *import elephant.statistics as stat*
» *import elephant.spade as spade*

2.2 Required input parameters

- Global parameters
 - List of <neo.SpikeTrain> objects
-> list of neurons
 - Firing rate
-> Quantities datatype
» *rate = 15 * pq.Hz*
 - Length of the Signal/SpikeTrain
-> Quantities datatype
» *T = 1 * pq.s*
 - Binsize or sampling period
-> Quantities datatype
*binsize = 5 * pq.ms*
- Additional parameters per module:
 - Statistics:
 - * Kernel for Instantaneous Firing Rate
 - ASSET:
 - * neo.SpikeTrain objects need additional t_stop attribute if signal length > 1s (standard is 1s).
 - * Build the intersection matrix 'imat' and probability matrix 'pmat':
 - Number of surrogates for the bootstrapping method
 - Window size for spike train dithering
 - * Joint probability matrix 'jmat', using a suitable filter:
 - Filter shape (length, width)
 - Number of largest neighbors
 - * Create from 'pmat' and 'jmat' a masked version of 'imat':
 - alpha1
 - alpha2
 - * Cluster significant elements of 'imat' into diagonal structures:
 - epsilon
 - minimum size
 - stretch

- SPADE:
 - * Window length
 - * Some optional parameters

NOTE: convert input (numpy array) from simulation to Neo.SpikeTrain and get the needed global parameters

2.3 Profiling

2.3.1 Statistics

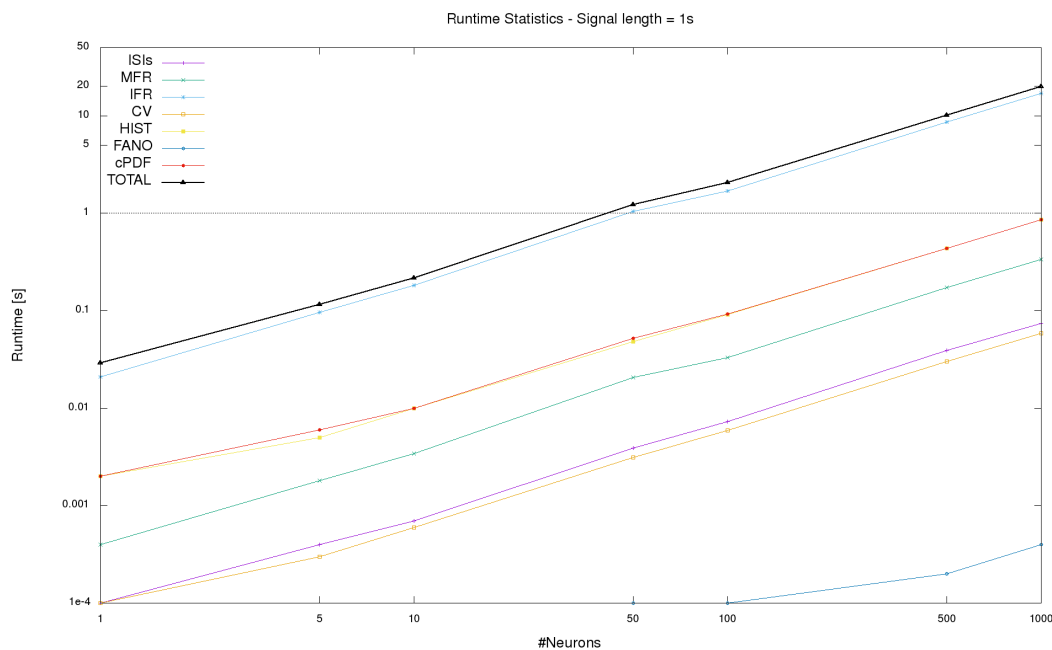


Figure 2.1: Statistics runtime with Signal length of 1 second, rate of 20 Hz and binsize of 5 ms. **Runtime < Signal length, except for IFR.**

Number of Neurons: 1000
Length of Signal: 1000ms

%	:	name	:	(count)	:	total time spend
=====						
0	:	start	:	1	:	0.0
4	:	DATA	:	1000	:	0.877937078476
0	:	ISIs	:	1000	:	0.0740790367126
2	:	MFR	:	1000	:	0.33157491684
0	:	IFR_start	:	1000	:	0.00199174880981
3	:	IFR_sskernel_density	:	1000	:	0.682367086411

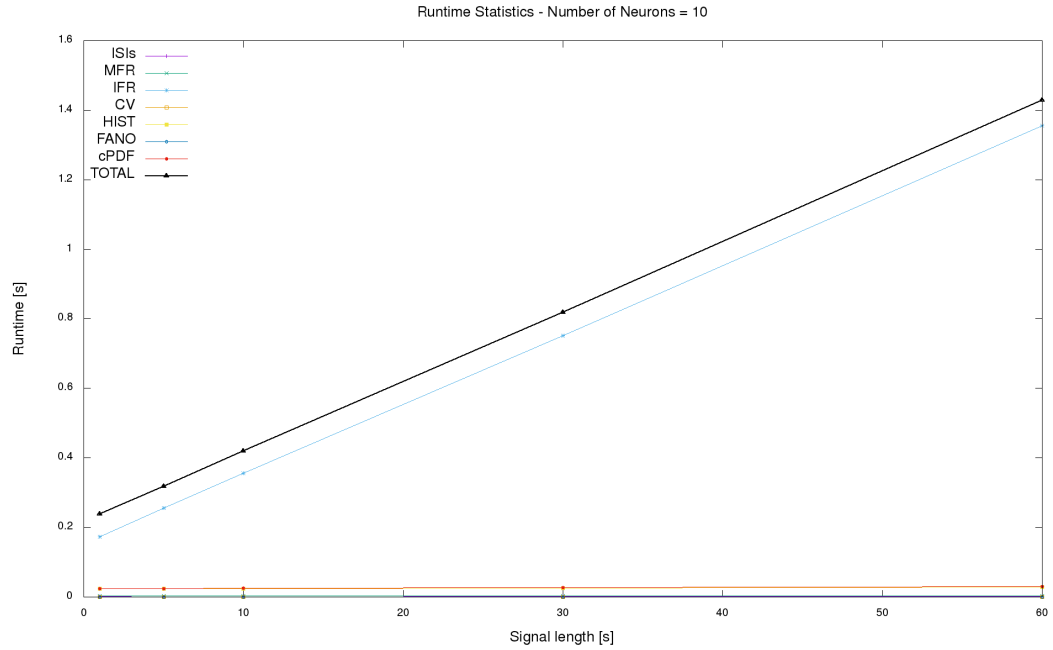


Figure 2.2: Statistics runtime with 10 neurons, rate of 20 Hz and binsize of 5 ms.

```

25 : IFR_sskernel_optw : 1000 : 5.00267696381
14 :   IFR_rescale     : 1000 : 2.86129784584
11 :   IFR_slice       : 1000 : 2.28599452972
14 :   IFR_fft         : 1000 : 2.75114107132
17 :   IFR_rest        : 1000 : 3.52695941925
 0 :   IFR_end        : 1000 : 0.00867342948914
 0 :   CV             : 1000 : 0.06112408638
 4 :   HIST           :    1 : 0.858650922775
 0 :   FANO           :    1 : 0.000443935394287
 4 :   cPDF           :    1 : 0.858818054199
 0 :   end            :    1 : 0.00200796127319
Total time : 20.1857380867

```

Summary:

Once per neuron:

- DATA: Data will be taken from simulation. No runtime needed for SpikeTrain creation...
- MFR (mean firing rate): 2% of total Statistics runtime.
Neurons: $O(n)$
Signal length: $O(1)$
- IFR (instantaneous firing rate): 85% of total Statistics runtime.

Neurons: $O(n)$
 Signal length: $O(n)$

- CV, ISIs: almost no runtime.
 Neurons: $O(n)$
 Signal length: $O(1)$

Once for all neurons:

- HIST: $\sim O(n)$, FANO: almost no runtime, cPDF: $\sim O(n)$

2.3.2 ASSET

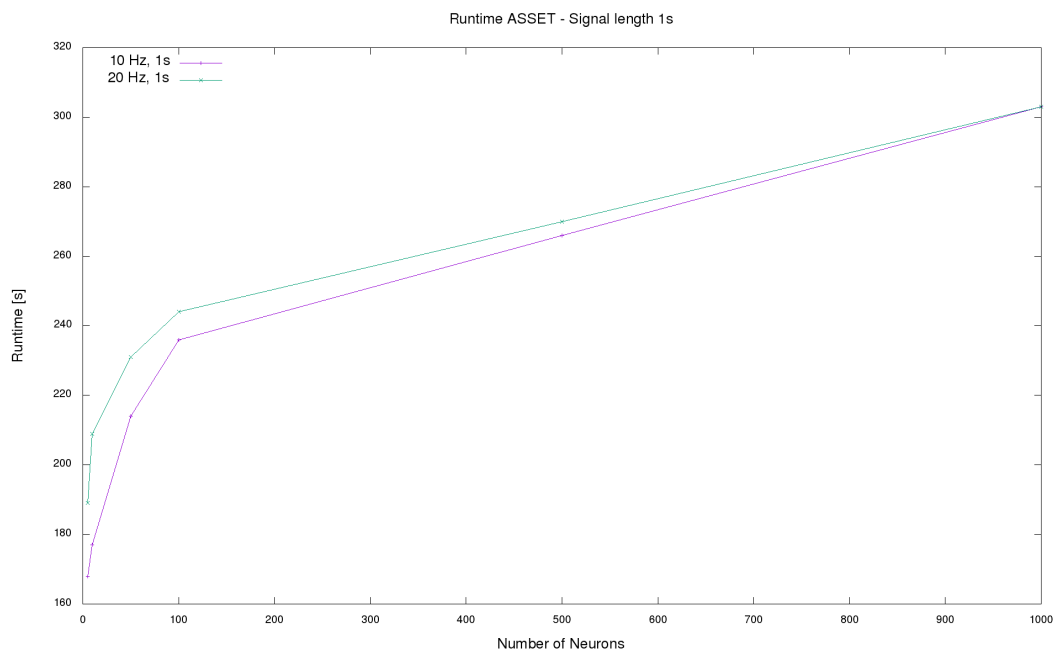


Figure 2.3: ASSET runtime with Signal length of 1 second and binsize of 5 ms.

Runtime \gg Signal length

	Signal length: 1s		Num. neurons: 10
#Neurons	Runtime [s]	Sig.len [s]	Runtime [s]
5	189		
10	209	1	209
100	244	2	719
1000	303	3	1542

Number of Neurons: 5
Length of Signal: 1000ms

```
% :      name      : (count) : total time spend
=====
0 :      start      :      1 : 0.0
0 : generate data    :      1 : 0.005774974823
0 : intersec mat     :      1 : 0.0325310230255
0 : prob mat mc      :      1 : 0.363873958588
0 : jmat_pmat_neighb :      1 : 0.0803229808807
0 : jmat_jsf_uni_init :      1 : 0.00139403343201
16 : jmat_jsf_uni_diff : 1395360 : 30.1089406013
7 : jmat_jsf_uni_reshape : 31878 : 12.6044211388
7 : jmat_jsf_uni_sum : 31878 : 14.2393944263
56 : jmat_jsf_uni_log : 31878 : 105.956119537
8 : jmat_jsf_uni_exp : 31878 : 14.9054570198
7 : jmat_jsf_uni_step : 1395360 : 12.4127993584
0 : jmat_jsf_uni     :      1 : 5.96046447754e-06
0 : jmat             :      1 : 0.000458002090454
0 : mask             :      1 : 0.000148057937622
0 : cmatrix          :      1 : 0.0209028720856
0 : extract_sse      :      1 : 6.31809234619e-05
0 : end              :      1 : 5.79357147217e-05
Total time : 190.732665062
```

Summary:

Once for all neurons:

- 100% of runtime for Joint Probability Matrix (jmat)
- Almost no runtime for:
 - Intersection Matrix (imat)
 - Probability Matrix, MonteCarlo (pmat)
 - Mask (from pmat and jmat)
 - Cluster Matrix
 - Extraxt SSE (synchronous spike events)

2.3.3 SPADE

Few Neurons: Runtime < Signal length, $\sim O(n)$

Increasing Neurons: Runtime > Signal length, $\sim O(n^2)$

Increasing Firing Rate: Runtime > Signal length (>90% fast_fca, $\sim O(n^2)$)

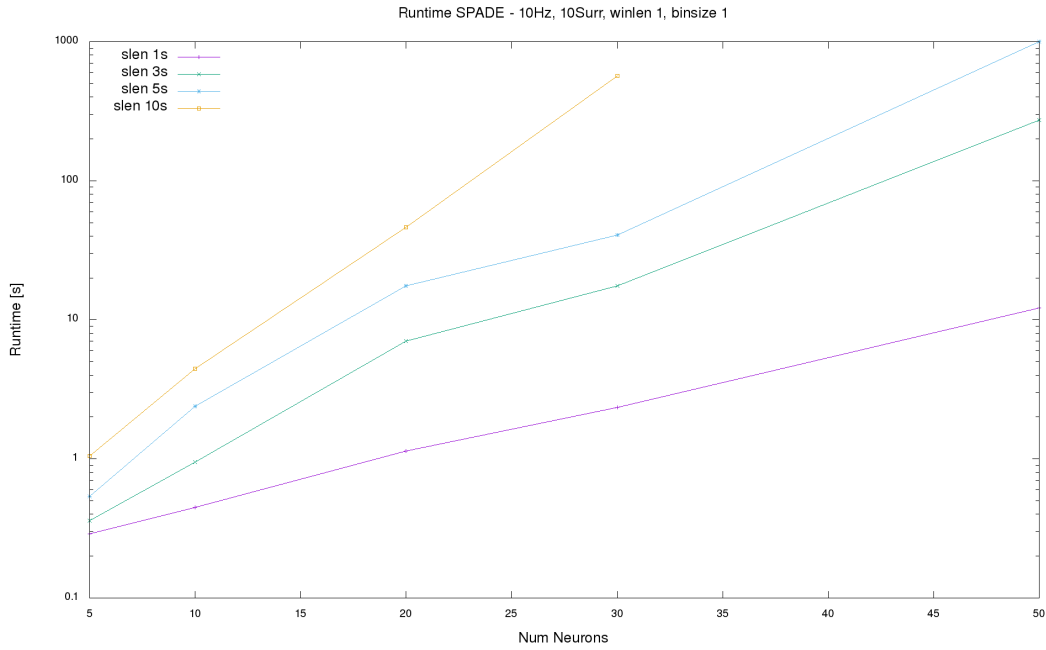


Figure 2.4: SPADE runtime with a rate of 10Hz, window length of 1, binsize of 1 ms and 10 surrogates.

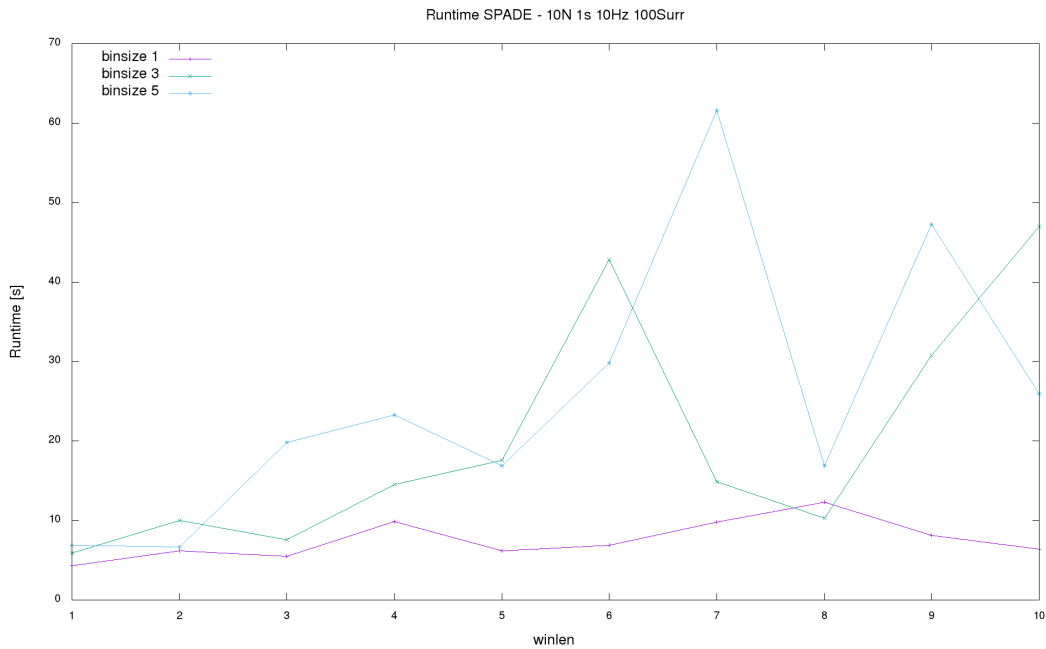


Figure 2.5: SPADE runtime for diff winlen/binsize. With 10 Neurons, 1s signal length, 10Hz and 100 surrogates.

10 ELEPHANT FUNCTIONS

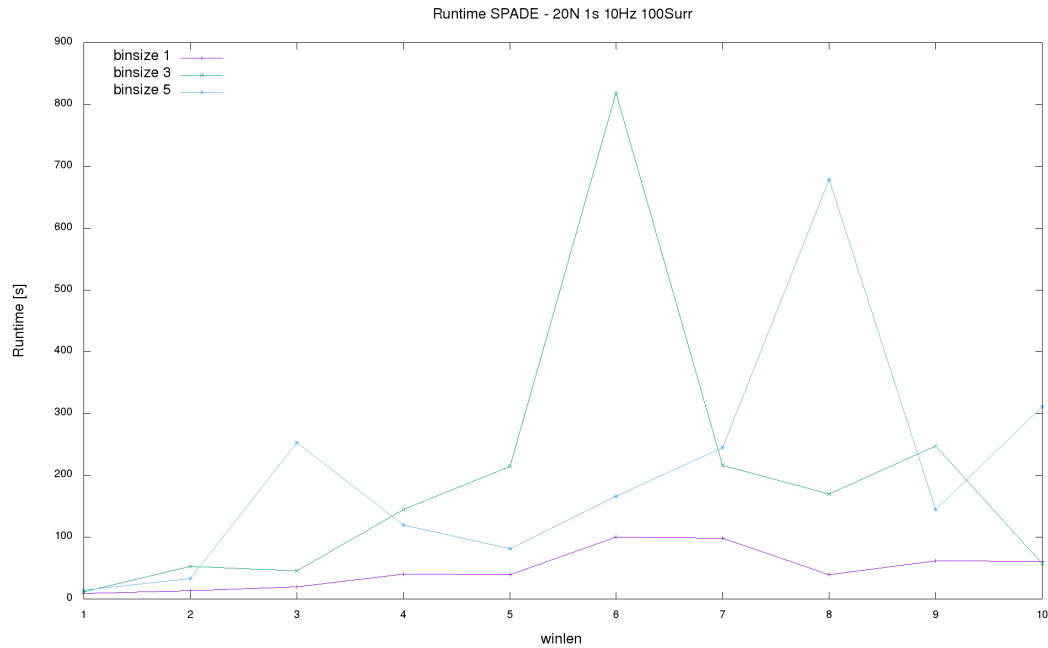


Figure 2.6: SPADE runtime for diff winlen/binsize. With 20 Neurons, 1s signal length, 10Hz and 100 surrogates.

10 Neurons, 10 seconds, 10 Hz

```
% :      name      : (count) : total time spend
=====
0 :      import      :      1 : 0.0
0 :      data-gen      :      1 : 0.0178639888763
0 :      data mining    :      1 : 0.000244140625
2 : pvalue: surr_init :     10 : 0.110061883926
6 : pvalue: con_mining_binning :    11 : 0.257157325745
29 : pvalue: con_mining_context :    11 : 1.34415078163
0 : pvalue: con_mining_max_spike :    11 : 0.00502347946167
0 : pvalue: con_mining_max_occ :    11 : 0.00521540641785
62 : pvalue: con_mining_fast_fca :    11 : 2.81482839584
0 : pvalue: step       :     10 : 0.00246357917786
0 :      spade end      :      1 : 0.000770807266235
Total time : 4.55782604218
```

10 Neurons, 10 seconds, 20 Hz

```
% :      name      : (count) : total time spend
=====
0 :      data-gen      :      1 : 0.023374080658
1 : pvalue: con_mining_binning :    11 : 0.258544206619
3 : pvalue: con_mining_context :    11 : 0.566279649734
0 : pvalue: con_mining_max_spike :    11 : 0.00187826156616
```

```

0 : pvalue: con_mining_max_occ : 11 : 0.00199890136719
1 : fast_fca_lattice_init : 11 : 0.24431180954
3 : fast_fca_lattice__compUpNeigh_init : 4616 : 0.638545274734
32 : fast_fca_lattice__compUpNeigh_intent_update : 4813666 : 6.31500482559
0 : fast_fca_lattice__compUpNeigh_else1 : 850 : 0.0161757469177
0 : fast_fca_lattice__compUpNeigh_else2 : 7213 : 0.0132088661194
25 : fast_fca_lattice__compUpNeigh_if1 : 4812816 : 4.93453645706
24 : fast_fca_lattice__compUpNeigh_if2 : 4806453 : 4.67416810989
2 : fast_fca_lattice__compUpNeigh_for2 : 7213 : 0.329411268234
1 : fast_fca_lattice_compUpNeigh : 4616 : 0.167640447617
6 : fast_fca_lattice_end : 11 : 1.15846991539
0 : pvalue: con_mining_fast_fca : 11 : 0.0549900531769
0 : data mining : 1 : 0.000248908996582
1 : pvalue: surr_init : 10 : 0.112371206284
0 : pvalue: step : 10 : 0.00310611724854
0 : pvalue spectrum : 1 : 3.69548797607e-05
0 : spade end : 1 : 0.000790119171143
Total time : 19.5150949955

```

2.3.4 Unitary Events (UE)

Parameters:

- spiketrains (number of trials, number of neurons, timestamps)
- binsize
- winsize (size of window)
- winstep (window step)
- pattern hash

Test for runtime: (**TODO: finfish tests**)

- number of trials $\sim O(?)$
- number of neurons $\sim O(?)$
- timestamps (firing rate, signal length) $\sim O(?)$
- winsize, winstep $\sim O(?)$
- hash?

Example1:

number of trials = 36

number of neurons: 2

firing rate = 10 Hz, signal length = 3s

winsize = 100ms, winstep = 10ms

12 ELEPHANT FUNCTIONS

```
% :      name      : (count) : total time spend
=====

0 :      start      :      1   : 0.0
12 :    data_init    :      1   : 0.250292062759
0 :      init        :      1   : 0.00165700912476
9 : binnedSpikeTrain_bool :    1   : 0.182360887527
0 :   init_indices   :      1   : 2.09808349609e-05
79 : _for_num_trials : 10476   : 1.64006233215
0 :   for_win_pos    :    291   : 0.000309705734253
0 :      UE_end       :      1   : 6.38961791992e-05
0 :      end          :      1   : 0.000160217285156
Total time : 2.0749270916
```

Example2:

number of trials = 36
number of neurons: 64
firing rate = 10 Hz, signal length = 3s
winsize = 100ms, winstep = 10ms

```
% :      name      : (count) : total time spend
=====

0 :      start      :      1   : 0.0
52 :    data_init    :      1   : 6.45083498955
0 :      init        :      1   : 0.00162696838379
31 : binnedSpikeTrain_bool :    1   : 3.87675404549
0 :   init_indices   :      1   : 2.8133392334e-05
17 : _for_num_trials : 10476   : 2.19099497795
0 :   for_win_pos    :    291   : 0.000298976898193
0 :      UE_end       :      1   : 0.000151872634888
0 :      end          :      1   : 0.00414204597473
Total time : 12.5248320103
```