Electrophys Feature Extraction Library (eFEL)

Ruben Moor, Werner Van Geit

June 30, 2015

Contents

About this document

The following list is the result of the implementation of the feature concept. It serves as a reference for the user of the feature library as well as for the developer. In order to understand the meaning of a certain feature it is not necessary anymore to recover the extraction procedure from its implementation.

1 General remarks

In the feature library the voltage trace is represented by two vectors, the voltage vector `V` and the corresponding time vector `T`. *Voltage trace indices* are indices of these two vectors. Some features require additional trace data, this is:

- `stim_start`, the time at the beginning of the stimulus current.
- `stim_end`, the time at the end of the stimulus current.

Some features require parameters in order to determine the extraction procedure.

Feature values are represented as vectors as well. All elementary features which describe properties of action potentials have entries corresponding to one action potential. Other features have only one entry containing the feature value. For some features the extraction procedure can *fail*. In case of failure the vector containing the feature values is *empty* and an error message is issued. Note that a successful calculation of a feature can also result in an empty vector (e.g. `peak indices` on a trace without action potentials).

2 Elementary features

2.1 peak indices

The voltage trace indices at the voltage maxima of the peaks

namespace / identifier	`LibV1:peak_indices`
unit	(index)
required features	none
required trace data	V
required parameters	Threshold

Operating on the voltage trace starting at index 0, each upwards crossing of `V` and the value of `Threshold` is considered a peak onset, and each downwards crossing a peak offset respectively. The peak index is the index of the maximum in between.

```
v_0,...,v_{n-1} = V
FOR i = 0,...,n-1 DO
  IF v_i < Threshold AND v_{i+1} > Threshold THEN
    APPEND i TO upwardsIndices
  ENDIF
  IF v_i > Threshold AND v_{i+1} < Threshold THEN
    APPEND i TO downwardsIndices
  ENDIF
ENDFOR
IF length of upwardsIndices ≠ length of downwardsIndices THEN
  FAIL "Bad trace shape."
ENDIF
u_0,...,u_{n-1} = upwardsIndices
d_0,...,d_{n-1} = downwardsIndices
FOR j = 0,...,n-1 DO
  APPEND i TO peak_indices WITH v_i maximal AND u_j ≤ i < d_j
ENDFOR
```

remarks: The usage for experimental traces is not recommended. For noisy traces small fluctuations around the value of `Threshold` are counted as peaks. Also traces have been observed where the minima between peaks laid above typical values of `Threshold`, sometimes even above the maxima of other peaks. This results in flawed peak count. These issues have been addressed in `LibV4:peak_indices`.

2.2 peak indices (2nd)

The voltage trace indices at the voltage maxima of the peaks, noise save

namespace / identifier	`LibV4:peak_indices`
unit	(index)
required features	none
required trace data	V T
required parameters	min spike height threshold

The nulls of the first derivative of `V` with a change of sign from $-$ to $+$ are the minima. The maxima between adjacent minima are presumable peaks. The left-hand (right-hand) height of a peak is the difference of the maximum and the left-hand (right-hand) minimum. A peak is kept if boths heights are bigger than `min spike height` *or* the maximum is bigger than `threshold` and one height is bigger than `min spike height`.

```
Δv_0, ..., Δv_{n-1} = ΔV
APPEND 0 TO minima
FOR i = 0, ..., n - 1 DO
  IF Δv_i < 0 AND Δv_{i+1} > 0 THEN
    APPEND i + 1 TO minima
  ENDIF
ENDFOR
APPEND n - 1 TO minima
min_0, ..., min_{n-1} = minima
FOR i = 0, ..., n - 2 DO
  max_i = j WITH v_j maximal AND min_i ≤ j < min_{i+1}
  h1 = V[max_i] - min_i
  h2 = V[max_i] - min_{i+1}
  IF h1 > min_spike_height AND h2 > min_spike_height
  OR
  V[max_i] > threshold AND (h1 > min_spike_height OR h2 > min_spike_height) THEN
    APPEND max_i TO peak_indices
  ENDIF
ENDFOR
```

2.3 peak voltage

The voltages at the maxima of the peaks

namespace / identifier	`LibV1:peak_voltage`
unit	mV
required features	peak indices
required trace data	V
required parameters	none

Iterating over the `peak indices` peak voltage yields `V` at each index.

```
p_0, ..., p_{n-1} = peak_indices
FOR i = 0, ..., n - 1 DO
  APPEND V[p_i] TO peak_voltage
ENDFOR
```

2.4 peak time

The times of the maxima of the peaks

namespace / identifier	`LibV1:peak_time`
unit	ms
required features	peak indices
required trace data	T
required parameters	none

Iterating over the `peak indices`, yield `T` at each index.

```
p_0,...,p_{n-1} = peak_indices
FOR i = 0,...,n-1 DO
  APPEND T[p_i] TO peak_time
ENDFOR
```

2.5 trace check

Causes feature extraction failure with error message when peaks before or after the stimulus are detected

namespace / identifier	`LibV1:trace_check`
unit	none
required features	peak time
required trace data	stim_start stim_end
required parameters	

Iterating over the values `peak time`, yield 0 as long as all the time values are bigger than `stim start` and smaller than `stim end`. Otherwise fail.

```
pt_0,...,pt_{n-1} = peak_time
FOR i = 0,...,n-1 DO
  IF pt_i < stim_start OR pt_i > 1.05· stim_end THEN
    FAIL "Trace sanity check failed, there were spike outside the stimulus interval."
  ENDIF
ENDFOR
APPEND 0 TO trace_check
```


2.6 ISI values

The interspike intervals (i.e. time intervals) between adjacent peaks, starting at the second peak

namespace / identifier	`LibV1:ISI_values`
unit	ms
required features	peak time
required trace data	none
required parameters	none

```
pt_0,...,pt_{n-1} = peak_time
IF n < 3 THEN
  FAIL "Three spikes required for calculation of ISI_values."
ENDIF FOR i = 2,...,n-1 DO
  APPEND pt_i - pt_{i-1} TO ISI_values
ENDFOR
```

2.7 doublet ISI

The time interval between the first too peaks

namespace / identifier	`LibV1:doublet_ISI`
unit	ms
required features	peak time
required trace data	none
required parameters	none

```
pt_0,...,pt_{n-1} = peak_time
IF n < 2 THEN
  FAIL "Need at least two spikes for doublet_ISI."
ENDIF
APPEND pt_1 - pt_0 TO doublet_ISI
```


2.8 burst ISI indices

ISI indices of those ISIs which are the beginning of a burst

namespace / identifier	`LibV1:burst_ISI_indices`
unit	(index)
required features	ISI values peak indices
required trace data	none
required parameters	burst factor (optional)

The median of the ISIs is determined. The `burst factor` defaults to 2. Each ISI bigger than `burst factor` times median divides two successive bursts, if it is also bigger than the following ISI times `burst factor`.

```
isi_0,...,isi_{n-1} = ISI_values
c = 0
FOR i = 1,...,n-2 DO
  median = median of { isi_c, ..., isi_{n-1}}
  IF isi_i > burst_factor · median
  AND isi_{i+1} < isi_i / burst_factor THEN
    APPEND i+1 TO burst_ISI_indices
  ENDIF
  c = i
ENDFOR
```

2.9 mean frequency

The mean frequency of the firing rate

namespace / identifier	`LibV1:mean_frequency`
unit	Hz
required features	peak time
required trace data	stim start stim end
required parameters	none

Yield the number of peaks divided by the time to the last spike.

remarks: The resulting value might be unexpected for bursting or irregular spiking cells. Regard `ISI CV` and `adaptation index` to assure that the cell is firing uniformly during the stimulus.

```
pt_0,...,pt_{n-1} = peak_time
c = 0
FOR i = 0,...,n-1 DO
  IF pt_i ≥ stim_start AND pt_i ≤ stim_end THEN
    c = c+1
  ENDIF
ENDFOR
APPEND 1000 · c/(time_to_last_spike - stim_start) TO mean_frequency
```


2.10 time to first spike

Time from the start of the stimulus to the maximum of the first peak

namespace / identifier	LibV1:time_to_first_spike
unit	ms
required features	peak time
required trace data	stim start
required parameters	none

```
pt_0,...,pt_{n-1} = peak_time
IF n < 1 THEN
  FAIL "One spike required for time_to_first_spike."
ENDIF
APPEND pt_0 - stim_start TO time_to_first_spike
```

2.11 min AHP indices

Voltage trace indices at the after-hyperpolarization

namespace / identifier	LibV1:min_AHP_indices
unit	(index)
required features	peak indices
required trace data	V T stim end
required parameters	none

Yield the indices at the voltage minima between two peaks. For the last peak yield the minimum between the last peak and the end of the stimulus.

```
pi_0,...,pi_{n-1} = peak_indices
IF n < 1 THEN
  FAIL "At least one spike required for calculation of min_AHP_indices."
ENDIF
t_0,...,t_{n-1} = T
end_index = minimal i WITH t_i ≥ stim_end
IF end_index > pi_{n-1} + 5 THEN
  pi' = (pi_0, ..., pi_{n-1}, end_index)
ENDIF
m = length of pi'
FOR i = 0,...,m-2 DO
  APPEND j TO min_AHP_indices WITH V[j] minimal AND pi'_i ≤ j < pi'_{i+1}
ENDFOR
```


2.12 min AHP values

Voltage values at the after-hyperpolarization

namespace / identifier	`LibV1:min_AHP_values`
unit	mV
required features	min AHP indices
required trace data	V T stim end
required parameters	none

Iterate over `min AHP indices`. Yield `V` at every index.

```
ahp_0, ..., ahp_{n-1} = min_AHP_indices
FOR i = 0, ..., n-1 DO
  APPEND V[ahp_i] TO min_AHP_values
ENDFOR
```


2.13 adaptation index

Normalized average difference of two consecutive ISIs

namespace / identifier	`LibV1:adaptation_index`
unit	none
required features	peak time
required trace data	stim start stim end
required parameters	spike skipf max spike skip offset (optional)

All peaks in the time interval of `stim start` $-$ `offset` and `stim end` $+$ `offset` are regarded, `offset` defaults to zero. The adaptation index is zero for a constant firing rate and bigger than zero for a decreasing firing rate:

$$A = \frac{1}{N-k-1}\sum_{i=k}^{N}\frac{\mathrm{ISI}_i - \mathrm{ISI}_{i-1}}{\mathrm{ISI}_i + \mathrm{ISI}_{i-1}} = \frac{1}{N-k-1}\sum_{i=k}^{N}\frac{t^{\mathrm{peak}}_{i+1} - 2t^{\mathrm{peak}}_{i} + t^{\mathrm{peak}}_{i-1}}{t^{\mathrm{peak}}_{i+1} - t^{\mathrm{peak}}_{i-1}}$$

with

the interspike intervals: $\mathrm{ISI}_i = t^{\mathrm{peak}}_{i+1} - t^{\mathrm{peak}}_{i}$,
the number of peaks: N.

The first k peaks are skipped. The parameter `spike skipf` is the fraction of skipped peaks, k is the minimum of `spike skipf` times N and `max spike skip`.

```
pt_0,...,pt_{n-1} = peak_time
pt'_0, ..., pt'_{m-1} = { pt_i | pt_i ≥ stim_start − offset AND pt_i ≤ stim_end + offset }
k = min{ spike_skipf ·m, max_spike_skip}
pt''_0, ..., pt''_{l-1} = (pt'_k, ..., pt'_m)
IF l < 4 THEN
  FAIL "Minimum 4 spike needed for feature [adaptation_index]."
ENDIF
isi_0, ..., isi_{j-1} = pt''_1− pt''_0, ..., pt''_{l-1}− pt''_{l-2}
sub_0, ..., sub_{i-1} = isi_1− isi_0, ..., isi_{j-1}− isi_{j-2}
sum_0, ..., sum_{i-1} = isi_1+ isi_0, ..., isi_{j-1}+ isi_{j-2}
APPEND 1/(i-1) Σ_{n=0}^{i-1} sub_n/sum_n TO adaptation_index
```


2.14 adaptation index 2

Normalized average difference of two consecutive ISIs

namespace / identifier	`LibV1:adaptation_index2`
unit	none
required features	peak time
required trace data	stim start stim end
required parameters	offset (optional)

The extraction is identical to the one of `adaptation index` for `spike skipf` equal zero.

```
pt_0,...,pt_{n-1} = peak_time
pt'_0, ..., pt'_{m-1} = { pt_i | pt_i ≥ stim_start AND pt_i ≤ stim_end }
IF m < 4 THEN
  FAIL "Minimum 4 spike needed for feature [adaptation_index]."
ENDIF
isi_0, ..., isi_{j-1} = pt'_1 - pt'_0, ..., pt'_{m-1} - pt'_{m-2}
sub_0, ..., sub_{i-1} = isi_1 - isi_0, ..., isi_{j-1} - isi_{j-2}
sum_0, ..., sum_{i-1} = isi_1 + isi_0, ..., isi_{j-1} + isi_{j-2}
APPEND 1/(i-1) Σ_{n=0}^{i-1} sub_n/sum_n TO adaptation_index
```


2.15 spike width 2

The FWHM of each peak

namespace / identifier	LibV1:spike_width2
unit	ms
required features	min AHP indices
required trace data	V T
required parameters	none

The peak onset is defined as the maximum of the second derivative. For the calculation of the full width at half maximum the height of the peak is taken relative to the voltage at peak onset. As one peak often contains only a little number of data points, the time vector is linearly interpolated in the rising and the falling flank.

```
pi_0,...,pi_{n-1} = peak_indices
ahp_0,...,ahp_{n-1} = min_AHP_indices
FOR i = 0,...,n-1 DO
  onset_index = arg max_j d^2V[j] WITH ahp_i <= j <= pi_{i+1}
  onset_voltage = V[onset_index]
  peak_voltage = V[pi_{i+1}]
  half_voltage = (onset_voltage + peak_voltage) / 2
  *** rising phase ***
  half_index = min j WITH V[j] > half_voltage AND ahp_i <= j <= pi_{i+1}
  t_0 = T[half_index - 1]
  v_0 = V[half_index - 1]
  v_1 = V[half_index]
  Δt = T[half_index] - T[half_index - 1]
  t_1 = t_0 + (half_voltage - v_0)/(v_1 - v_0) Δt
  *** falling phase (buggy) ***
  half_index = min j WITH V[j] < half_voltage AND pi_{i+1} <= j <= pi_{i+1}
  IF half_index = pi_{i+1} THEN
    FAIL "Falling phase of last spike is missing."
  ENDIF
  t_0 = T[half_index - 1]
  v_0 = V[half_index - 1]
  v_1 = V[half_index]
  Δt = T[half_index] - T[half_index - 1]
  t_2 = t_0 + (half_voltage - v_0)/(v_1 - v_0) Δt
  APPEND t_2 - t_1 TO spike_width2
ENDFOR
```


2.16 AP width

Width of each peak at the value of `Threshold`

namespace / identifier	`LibV1:AP_width`
unit	ms
required features	peak indices min AHP indices
required trace data	V T stim start
required parameters	Threshold

The peak onset (offset) is determined as the upwards (downwards) crossing of the `V` and the value of `Threshold`. AP width yields the time difference between peak onset and peak offset.

```
ahp_0,...,ahp_{n-1} = min_AHP_indices
pi_0,...,pi_{m-1} = peak_indices
start_index = minimal i WITH T[i] ≥ stim_start
ahp'_0,...,ahp'_{m-1} = start_index, ahp_0,...,ahp_{n-1}
FOR j = 0,...,m-2 DO
  onset_index = minimal i WITH V[i] ≥ Threshold AND ahp'_j ≤ i < ahp'_{j+1}
  offset_index = minimal i WITH V[i] ≤ Threshold AND pi_j ≤ i < ahp'_{j+1}
  APPEND T[offset_index] - T[onset_index] TO AP_width
ENDFOR
```


2.17 spike half width

The FWHM of each peak

namespace / identifier	`LibV1:spike_half_width`
unit	ms
required features	min AHP indices peak indices
required trace data	V T stim start
required parameters	none

The height of the peak is defined relative to `V` at the subsequent `min AHP index`. As one peak often contains only a little number of data points, the time vector is linearly interpolated in the rising and the falling flank.

```
start_index = minimal i WITH T[i] ≥ stim_start
ahp_0,...,ahp_{n-1} = min_AHP_indices
pi_0,...,pi_{m-1} = peak_indices
ahp'_0,...,ahp'_{m-1} = start_index, ahp_0,...,ahp_{n-1}
FOR i = 1,...,m-1 DO
  half_voltage = (pi_{i-1} + ahp'_i) / 2
  rise_index = min j WITH V[j] > half_voltage AND ahp'_{i-1} ≤ j ≤ pi_{i-1}
  δv = half_voltage - V[rise_index]
  Δv = V[rise_index] - V[rise_index - 1]
  Δt = T[rise_index] - T[rise_index - 1]
  δt_1 = Δt δv/Δv
  fall_index = min j WITH V[j] < half_voltage AND pi_{i-1} ≤ j ≤ ahp'_i
  δv = half_voltage - V[fall_index]
  Δv = V[fall_index] - V[fall_index - 1]
  Δt = T[fall_index] - T[fall_index - 1]
  δt_2 = Δt δv/Δv
  APPEND T[fall_index] + δt_1 - T[rise_index] + δt_2 TO spike_half_width
ENDFOR
```


2.18 burst mean frequency

The mean frequency during a burst for each burst

namespace / identifier	`LibV1:burst_mean_freq`
unit	Hz
required features	burst ISI indices peak time
required trace data	none
required parameters	none

Iterate over the `burst ISI indices` and yield the number of peaks divided by the length of the burst.

```
isi_0,...,isi_{n-1} = burst_ISI_indices
isi'_0,...,isi'_{m-1} = (0, isi_0,...,isi_{n-1})
FOR i = 0,...,m-2 DO
  IF isi'_{i+1}- isi'_i = 1 THEN
    ν = 0
  ELSE
    Δt = peak_time[isi'_{i+1} - 1] - peak_time[isi'_i]
    ν = 1000 (isi'_{i+1} - isi'_i + 1) / Δt
  ENDIF
  APPEND ν TO burst_mean_frequency'
ENDFOR
Δt = peak_time[last] - peak_time[isi'_i]
ν = 1000 (length of peak_time - isi'_i) / Δt
APPEND ν TO burst_mean_frequency'
burst_mean_frequency = {ν|ν IN burst_mean_frequency AND ν ≠ 0}
```

2.19 interburst voltage

The voltage average in between two bursts

namespace / identifier	`LibV1:interburst_voltage`
unit	mV
required features	peak indices burst ISI indices
required trace data	V T
required parameters	none

Iterating over the `burst ISI indices` determine the last peak before the burst. Starting 5 ms after that peak take the voltage average until 5 ms before the first peak of the subsequent burst.

```
isi_0,...,isi_{n-1} = burst_ISI_indices
IF n < 2 THEN
  RETURN
ENDIF
FOR i = 0,...,n-1 DO
  start_index = peak_indices[isi_i - 1]
  t_start = T[start_index] + 5
  end_index = peak_indices[isi_i]
  t_end = T[end_index] - 5
  start_index = -1 + minimal j WITH j ≥ start_index AND T[j] > t_start
  end_index = 1 + maximal j WITH j < end_index AND T[j] < t_end
  APPEND mean V[j] WITH start_index < j < end_index TO interburst_voltage
ENDFOR
```


2.20 voltage base

The membrane resting potential

namespace / identifier	LibV1:voltage_base
unit	mV
required features	none
required trace data	V T stim start
required parameters	none

Yield the average voltage during the time interval $\frac{1}{4}$ times `stim start` and $\frac{3}{4}$ times `stim start` well before the stimulus.

```
start_time = stim_start · 0.25
end_time = stim_start · 0.75
t_0,...,t_{n-1} = T
FOR i = 0,...,n − 1 DO
  IF t_i ≥ start_time THEN
    sum = sum + V[i]
    size = size + 1
  ENDIF
  IF t_i > end_time THEN
    EXIT FOR
  ENDIF
ENDFOR
APPEND sum / size TO voltage_base
```

2.21 AP height

The voltages at the maxima of the peaks

namespace / identifier	LibV1:AP_height
unit	mV
required features	peak voltage
required trace data	none
required parameters	none

Identical to peak voltage.

remarks: This feature exists for reasons of compatibility. I recommend the usage of `peak voltage` instead. `AP height` should rather yield the height of the peak relative to the membrane resting potential.

```
p_0,...,p_{n-1} = peak_indices
FOR i = 0,...,n − 1 DO
  APPEND V[p_i] TO AP_height
ENDFOR
```


2.22 AP amplitude

The relative height of the action potential

namespace / identifier	`LibV1:AP_Amplitude`
unit	mV
required features	AP begin indices peak voltage
required trace data	V
required parameters	none

Yield the difference of `peak voltage` and `V` at `AP begin indices` for each peak.

```
pv_0,...,pv_{n-1} = peak_voltage
b_0,...,b_{n-1} = AP_begin_indices
FOR i = 0,...,n-1 DO
  APPEND pv_i - b_i TO AP_amplitude
ENDFOR
```

2.23 AHP depth abs

Voltage values at the after-hyperpolarization

namespace / identifier	`LibV1:AHP_depth_abs`
unit	mV
required features	min AHP values
required trace data	none
required parameters	none

Identical to `min AHP values`

remarks: This feature exists for reasons of compatibility. I recommend the usage of `min AHP values` instead.

```
ahp_0,...,ahp_{n-1} = min_AHP_indices
FOR i = 0,...,n-1 DO
  APPEND V[ahp_i] TO AHP_depth_abs
ENDFOR
```


2.24 AHP depth abs slow

Voltage values at the "slow" after-hyperpolarization

namespace / identifier	`LibV1:AHP_depth_abs_slow`
unit	mV
required features	peak indices
required trace data	V T
required parameters	none

Starting at the second peak iterating over `peak indices` find the minimum of `V` between each pair of peaks. For each pair of peaks start the search 5 ms after the first peak.

```
pi_0,...,pi_{n-1} = peak_indices
IF n < 3 THEN
  FAIL "At least 3 spikes needed for AHP_depth_abs_slow and AHP_slow_time."
ENDIF
(pi'_0, ..., pi'_{m-1}) = (pi_1, ..., pi'_{n-2})
FOR i = 0,...,m-1 DO
  start_time = T[pi'_i] + 5
  start_index = minimal j WITH T[j] ≥ start_time AND p'_i ≤ j < p'_{i+1}
  min_voltage = min_j V[j] WITH start_index ≤ j < p'_{i+1}
  APPEND min_voltage TO AHP_depth_abs_slow
ENDFOR
```

2.25 AHP slow time

Relative timing of the "slow" after-hyperpolarization

namespace / identifier	`LibV1:AHP_slow_time`
unit	none
required features	AHP depth abs slow
required trace data	V T
required parameters	none

Starting at the second peak iterating over `peak indices` find the minimum of `V` between each pair of peaks. For each pair of peaks start the search 5 ms after the first peak. Yield the time at the minimum divided by the length of the interspike interval.

```
pi_0,...,pi_{n-1} = peak_indices
IF n < 3 THEN
  FAIL "At least 3 spikes needed for AHP_depth_abs_slow and AHP_slow_time."
ENDIF
(pi'_0, ..., pi'_{m-1}) = (pi_1, ..., pi'_{n-2})
FOR i = 0,...,m-1 DO
  start_time = T[pi'_i] + 5
  start_index = minimal j WITH T[j] ≥ start_time AND p'_i ≤ j < p'_{i+1}
  min_index = arg min_j V[j] WITH start_index ≤ j < p'_{i+1}
  APPEND (T[min_index] - T[pi'_i]) / (T[pi'_{i+1}] - T[pi'_i]) TO AHP_slow_time
ENDFOR
```

2.26 time constant

The membrane time constant

namespace / identifier	`LibV1:time_constant`
unit	ms
required features	none
required trace data	V T stim start stim end
required parameters	none

The extraction of the time constant requires a voltage trace of a cell in a hyperpolarized state. Starting at `stim start` find the beginning of the exponential decay where the first derivative of `V(t)` is smaller than -0.005 $\frac{\mathrm{V}}{\mathrm{s}}$ in 5 subseequent points. The flat subsequent to the exponential decay is defined as the point where the first derivative of the voltage trace is bigger than -0.005 and the mean of the follwowing 70 points as well. If the voltage trace between the beginning of the decay and the flat includes more than 9 points, fit an exponential decay. Yield the time constant of that decay.

```
min_derivative = 0.005
decay_length = 10
min_time = 70
start_index = 10 + minimal i WITH T[i] ≥ stim_start
middle_index = minimal i WITH T[i] ≥ (stim_start + stim_end) / 2
dvdt = ΔV_i/ΔT_i WITH start_index ≤ i < middle_index
*** find the decay ***
decay_index = 0
WHILE THERE IS x IN (dvdt[decay_index], ..., dvdt[decay_index + 5]) DO
  decay_index = decay_index + 1
ENDWHILE
IF decay_index + 5 = length of dvdt - 1 THEN
  FAIL "Could not find the decay."
ENDIF
*** find the flat ***
i = decay_index WHILE T[i] < T[middle_index] + min_time DO
  IF dvdt[i] > - min_derivative THEN
    j = minimal j WITH T[j] - T[i] > min_time
    mean = mean of (dvdt[i], ..., dvdt[j])
    IF mean > - min_derivative THEN
      EXIT WHILE
    ENDIF
  ENDIF
  i = i + 1
ENDWHILE
flat_index = i
IF flat_index - decay_index < decay_length THEN
  FAIL "Trace fall time too short."
ENDIF
(v_0, ..., v_{n-1}) = (v[decay_index], ..., v[flat_index])
(t_0, ..., t_{n-1}) = (t[decay_index], ..., t[flat_index])
x = 0.38
golden section search:
FOR i = 0, ..., n - 1 DO
  logv_i = log(v_i - v_{n-1} + x)
ENDFOR
slope, residuals = fit_straight_line(t, logv)
repeat golden section search to minimize residuals
APPEND -1 / slope TO time_constant
```


2.27 voltage deflection

The relative steady state voltage in a hyperpolarized state

namespace / identifier	`LibV1:voltage_deflection`
unit	mV
required features	none
required trace data	V T stim start stim end
required parameters	none

Calculate the base voltage as the mean of `V` before `stim start`. Calculate the steady state voltage as the mean of 5 values of `V` 10 points before `stim end`. Yield the difference of steady state voltage and base voltage.

```
base = mean of V[i] WITH 0 ≤ T[i] < stim_start
end_index = minimal i WITH t_i ≥ stim_end
ss = mean of V[i] WITH end_index - 10 < i < end_index - 5
APPEND ss - base TO voltage_deflection
```

2.28 ohmic input resistance

The ohmic input resistance R_{in} of the cell

namespace / identifier	`LibV1:ohmic_input_resistance`
unit	MΩ
required features	voltage deflection
required trace data	stimulus current
required parameters	none

Yield `voltage deflection` divided by `stimulus current`.

remarks: The regular feature E29 determins R_{in} far more accurately.	

```
APPEND voltage_deflection / stimulus_current TO ohmic_input_resistance
```

2.29 maximum voltage

The maximum voltage during a stimulus

namespace / identifier	`LibV1:maximum_voltage`
unit	mV
required features	none
required trace data	V T stim start stim end
required parameters	none

Find the maximum of `V` between `stim start` and `stim end`.

```
max = maximal V[i] WITH i:  stim_start ≤ T[i] < stim_end
APPEND max TO maximum_voltage
```

2.30 steady state voltage

Average voltage after the stimulus

namespace / identifier	`LibV1:steady_state_voltage`
unit	mV
required features	none
required trace data	V T stim end
required parameters	none

Yield the average of `V` after `stim end`.

```
mean = mean V[i] WITH i:  stim_end < T[i]
APPEND mean TO steady_state_voltage
```

2.31 ISI CV

The coefficient of variation of the ISIs

namespace / identifier	`LibV1:ISI_CV`
unit	none
required features	ISI values
required trace data	none
required parameters	none

Yield the coefficient of variation:

$$c_v = \frac{\sigma}{\mu}$$

with

$$\text{standard deviation: } \sigma = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(\text{ISI}_i - \mu)^2},$$

$$\text{mean: } \mu = \frac{1}{N}\sum_{i=1}^{N}\text{ISI}_i.$$

```
μ = mean of ISI_values
σ = standard deviation of ISI_values
APPEND σ/μ TO ISI_CV
```

2.32 Spikecount

The number of peaks during stimulus

namespace / identifier	`LibV1:Spikecount`
unit	none
required features	peak indices trace check
required trace data	none
required parameters	none

Yield the length of `peak indices`.

```
APPEND length of peak_indices TO Spikecount
```


2.33 AHP depth

Relative voltage values at the after-hyperpolarization

namespace / identifier	`LibV1:AHP_depth`
unit	mV
required features	voltage base min AHP values
required trace data	none
required parameters	none

Iterate over `min AHP values` and yield the difference of the value and `voltage base`.

```
ahp_0, ..., ahp_{n-1} = min_AHP_values
FOR i = 0, ..., n - 1 DO
  APPEND ahp_i - voltage_base TO AHP_depth
ENDFOR
```

2.34 AP begin indices

Voltage trace indices at the onset of each action potential

namespace / identifier	`LibV2:AP_begin_indices`
unit	(index)
required features	min AHP indices interpolate
required trace data	V T stim start stim end
required parameters	none

Iterate over `min AHP indices`. If there is no AHP for the last peak, add the index at `stim end` to the indices. Yield the action potential onsets where the first derivative of the voltage trace is higher than $12\frac{\mathrm{V}}{\mathrm{s}}$, for at least 5 points.

```
min_derivative = 12.0
start_index = minimal i WITH T[i] ≥ stim_start
ahp_0, ..., ahp_{n-1} = min_AHP_indices
min_0,, ..., min_{m-1} = (start_index, ahp_0, ..., ahp_{n-1})
IF T[min_{m-1}] < stim_end THEN
  end_index = minimal i WITH t_i ≥ stim_end
  APPEND end_index TO min
ENDIF
dvdt = Δ V
FOR i = 0 TO length of min - 2 DO
  IF x ≥ min_derivative FOR ALL x IN (dvdt[min_i], ..., dvdt[min_i + 5]) THEN
    APPEND min_i TO AP_begin_indices
  ENDIF
ENDFOR
```


2.35 AP rise indices

Voltage trace index at the rising flank of each action potential.

namespace / identifier	`LibV2:AP_rise_indices`
unit	(index)
required features	peak indices AP begin indices
required trace data	V
required parameters	none

Yield the indices of the voltage trace after each `AP begin index` where `V` reaches half the maximum of the amplitude of the action potential. The amplitude of the action potential is taken relative to `V` at `AP begin indices`.

```
begin_0,...,begin_{n-1} = AP_begin_indices
pi_0,...,pi_{n-1} = peak_indices
FOR i = 0,...,n-1 DO
  half_voltage = (begin_i + pi_i) / 2
  rise_index = arg min_j |V[j] - half_voltage| WITH begin_i <= j < pi_i
  APPEND rise_index TO AP_rise_indices
ENDFOR
```

2.36 AP end indices

Voltage trace indices at the offset of each action potential

namespace / identifier	`LibV2:AP_end_indices`
unit	(index)
required features	peak indices
required trace data	V T
required parameters	none

Iterate over `peak indices` and find after each index where the first derivative of the voltage trace exceeds $-12\frac{\mathrm{V}}{\mathrm{s}}$.

```
pi_0,...,pi_{n-1} = (peak_indices, length of V - 1)
min_derivative = -12.0
dvdt = Δ V
FOR i = 0,...,n-2 DO
  end_index = minimal j WITH pi_i + 1 <= j < pi_{i+1} AND dvdt[j] > min_derivative
ENDFOR
```


2.37 AP fall indices

Voltage trace index at the falling flank of each action potential

namespace / identifier	`LibV2:AP_fall_indices`
unit	(index)
required features	peak indices AP begin indices AP end indices
required trace data	V
required parameters	none

Yield the indices of the voltage trace after each `peak index` where `V` falls down to half the maximum of the amplitude of the action potential. The amplitude of the action potential is taken relative to `V` at `AP begin indices`.

```
begin_0,...,begin_{n-1} = AP_begin_indices
end_0,...,end_{n-1} = AP_end_indices
pi_0,...,pi_{n-1} = peak_indices
FOR i = 0,...,n-1 DO
  half_voltage = (begin_i + pi_i) / 2
  fall_index = arg min_j |V[j] - half_voltage| WITH pi_i <= j < end_i
  APPEND fall_index TO AP_fall_indices
ENDFOR
```

2.38 AP duration

Duration of an action potential from onset to offset

namespace / identifier	`LibV2:AP_duration`
unit	ms
required features	AP begin indices AP end indices
required trace data	T
required parameters	none

Iterate over `AP begin indices` and return the difference of the time at the `AP end index` and the `AP begin index`.

```
begin_0,...,begin_{n-1} = AP_begin_indices
end_0,...,end_{n-1} = AP_end_indices
FOR i = 0,...,n-1 DO
  APPEND T[end_i] - T[begin_i] TO AP_duration
ENDFOR
```


2.39 AP duration half width

FWHM of each action potential

namespace / identifier	`LibV2:AP_duration_half_width`
unit	ms
required features	AP rise indices AP fall indices
required trace data	T
required parameters	none

Iterate over `AP rise indices` and return the difference of the time at the `AP fall index` and the `AP rise index`.

```
rise_0,...,rise_{n-1} = AP_rise_indices
fall_0,...,fall_{n-1} = AP_fall_indices
FOR i = 0,...,n-1 DO
  APPEND T[fall_i] - T[rise_i] TO AP_duration_half_width
ENDFOR
```

2.40 AP rise time

Time from action potential onset to the maximum

namespace / identifier	`LibV2:AP_rise_time`
unit	ms
required features	AP begin indices peak indices
required trace data	T
required parameters	none

Iterate over `AP begin indices` and return the difference of the time at the `peak index` and the `AP begin index`.

```
pi_0,...,pi_{n-1} = peak_indices
begin_0,...,begin_{n-1} = AP_begin_indices
FOR i = 0,...,n-1 DO
  APPEND T[pi_i] - T[begin_i] TO AP_rise_time
ENDFOR
```


2.41 AP fall time

Time from action potential maximum to the offset

namespace / identifier	`LibV2:AP_fall_time`
unit	ms
required features	peak indices AP end indices
required trace data	T
required parameters	none

Iterate over `peak indices` and return the difference of the time at the `end index` and the `peak index`.

```
pi_0,...,pi_{n-1} = peak_indices
end_0,...,end_{n-1} = AP_end_indices
FOR i = 0,...,n-1 DO
  APPEND T[end_i] - T[pi_i] TO AP_fall_time
ENDFOR
```

2.42 AP rise rate

Voltage change rate during the rising phase of the action potential

namespace / identifier	`LibV2:AP_rise_rate`
unit	$\frac{\mathrm{V}}{\mathrm{s}}$
required features	AP begin indices peak indices
required trace data	V T
required parameters	none

Iterate over `AP begin indices` and return the ratio of the voltage difference and the time difference at the `peak index` and the `AP begin index`.

```
pi_0,...,pi_{n-1} = peak_indices
begin_0,...,begin_{n-1} = AP_begin_indices
FOR i = 0,...,n-1 DO
  APPEND (V[pi_i] - V[begin_i]) / (T[pi_i] - T[begin_i]) TO AP_rise_rate
ENDFOR
```

2.43 AP fall rate

Voltage change rate during the falling phase of the action potential.

namespace / identifier	`LibV2:AP_fall_rate`
unit	$\frac{\mathrm{V}}{\mathrm{s}}$
required features	peak indices AP end indices
required trace data	V T
required parameters	none

Iterate over `peak indices` and return the ratio of the voltage difference and the time difference at the `AP end index` and the `peak index`.

```
pi_0,...,pi_{n-1} = peak_indices
end_0,...,end_{n-1} = AP_end_indices
FOR i = 0,...,n-1 DO
  APPEND (V[end_i] - V[pi_i]) / (T[end_i] - T[pi_i]) TO AP_fall_rate
ENDFOR
```

2.44 fast AHP

Voltage value of the action potential onset relative to the subsequent AHP

namespace / identifier	`LibV2:fast_AHP`
unit	mV
required features	AP begin indices min AHP indices
required trace data	V
required parameters	none

Iterate over `AP begin indices` and yield the difference of `V` at the `AP begin index` and the `min AHP index`.

```
begin_0,...,begin_{n-1} = AP_begin_indices
ahp_0,...,ahp_{n-1} = min_AHP_indices
FOR i = 0,...,n-1 DO
  APPEND V[begin_i] - V[ahp_i] TO fast_AHP
ENDFOR
```


2.45 AP amplitude change

Difference of the amplitudes of the second and the first action potential divided by the amplitude of the first action potential

namespace / identifier	LibV2:AP_amplitude_change
unit	none
required features	peak voltage
required trace data	none
required parameters	none

```
amp_0, ..., amp_{n-1} = AP_amplitude_change
FOR i = 0, ..., n - 2 DO
  APPEND (amp_{i+1} - amp_0) / amp_0 TO AP_amplitude_change
ENDFOR
```

2.46 AP duration change

Difference of the durations of the second and the first action potential divided by the duration of the first action potential

namespace / identifier	LibV2:AP_duration_change
unit	none
required features	AP duration
required trace data	none
required parameters	none

```
dur_0, ..., dur_{n-1} = AP_duration
FOR i = 0, ..., n - 2 DO
  APPEND (dur_{i+1} - dur_0) / dur_0 TO AP_duration_change
ENDFOR
```

2.47 AP rise rate change

Difference of the rise rates of the second and the first action potential divided by the rise rate of the first action potential

namespace / identifier	LibV2:AP_rise_rate_change
unit	none
required features	AP rise rate
required trace data	none
required parameters	none

```
rr_0, ..., rr_{n-1} = AP_rise_rate
FOR i = 0, ..., n - 2 DO
  APPEND (rr_{i+1} - rr_0) / rr_0 TO AP_rise_rate_change
ENDFOR
```

2.48 AP fall rate change

Difference of the fall rates of the second and the first action potential divided by the fall rate of the first action potential

namespace / identifier	`LibV2:AP_fall_rate_change`
unit	none
required features	AP fall rate
required trace data	none
required parameters	none

```
fr_0, ..., fr_{n-1} = AP_fall_rate
FOR i = 0, ..., n - 2 DO
  APPEND (fr_{i+1} - fr_0) / fr_0 TO AP_fall_rate_change
ENDFOR
```

2.49 fast AHP change

Difference of the `fast AHP` of the second and the first action potential divided by the `fast AHP` of the first action potential

namespace / identifier	`LibV2:fast_AHP_change`
unit	none
required features	fast AHP
required trace data	none
required parameters	none

```
fahp_0, ..., fahp_{n-1} = fast_AHP
FOR i = 0, ..., n - 2 DO
  APPEND (fahp_{i+1} - fahp_0) / fahp_0 TO fast_AHP_change
ENDFOR
```

2.50 AP duration half width change

Difference of the FWHM of the second and the first action potential divided by the FWHM of the first action potential

namespace / identifier	`LibV2:AP_duration_half_width_change`
unit	none
required features	AP duration half width
required trace data	none
required parameters	none

```
dhw_0, ..., dhw_{n-1} = AP_duration_half_width
FOR i = 0, ..., n - 2 DO
  APPEND (dhw_{i+1} - dhw_0) / dhw_0 TO AP_duration_half_width_change
ENDFOR
```

2.51 steady state hyper

Steady state voltage during hyperpolarization

namespace / identifier	LibV2:steady_state_hyper
unit	mV
required features	none
required trace data	V T stim end
required parameters	none

Find the voltage trace index at `stim end`. Yield the average of `V` between that index minus 35 and that index minus 5.

```
end_index = minimal i WITH t_i ≥ stim_end
mean = mean of V[i] WITH end_index - 35 ≤ i < end_index - 5
APPEND mean TO steady_state_hyper
```

2.52 amp drop first second

Difference of the amplitude of the first and the second peak

namespace / identifier	LibV2:amp_drop_first_second
unit	double
required features	peak voltage
required trace data	none
required parameters	none

```
IF length of peak_voltage < 2 THEN
  FAIL "At least 2 spikes needed for the calculation of amp_drop_first_second."
ENDIF
APPEND peak_voltage[0] - peak_voltage[1] TO amp_drop_first_second
```

2.53 amp drop first last

Difference of the amplitude of the first and the last peak

namespace / identifier	LibV2:amp_drop_first_last
unit	double
required features	peak voltage
required trace data	none
required parameters	none

```
IF length of peak_voltage < 2 THEN
  FAIL "At least 2 spikes needed for the calculation of amp_drop_first_last."
ENDIF
APPEND peak_voltage[0] - peak_voltage[last] TO amp_drop_first_last
```


2.54 amp drop second last

Difference of the amplitude of the second and the last peak

namespace / identifier	LibV2:amp_drop_second_last
unit	double
required features	peak voltage
required trace data	none
required parameters	none

```
IF length of peak_voltage < 3 THEN
  FAIL "At least 3 spikes needed for the calculation of amp_drop_second_last."
ENDIF
APPEND peak_voltage[1] - peak_voltage[last] TO amp_drop_second_last
```

2.55 max amp difference

Maximum difference of the height of two subsequent peaks

namespace / identifier	LibV2:max_amp_difference
unit	double
required features	peak voltage
required trace data	none
required parameters	none

```
IF length of peak_voltage < 2 THEN
  FAIL "At least 2 spikes needed for the calculation of max_amp_difference."
ENDIF
APPEND max Δpeak_voltage TO max_amp_difference
```


3 Regular features

3.1 back-propagating AP attenuation

Ratio of relative heights of somatic peak and dendritic peak

namespace / identifier	`LibV2:BPAPatt2`
unit	none
required features	peak voltage;location_soma voltage base;location*
required trace data	V;location_dend620
required parameters	none

Inject a short square pulse at soma that invokes exactly one action potential. The relative height of the action potential at soma is the difference of `peak voltage;location_soma` and `voltage base;location_soma`. The recording at the dendrite takes place at the thickest apical dendrite with a distance of 620 μm from soma. The relative height is the difference of the maximum of `V;location_dend620` and `voltage base;location_dend620`. Yield the relative height at soma divided by the relative height at the dendritic location.

remarks: There exist a *hoc* implementation under the same name, where instead of a ratio the actual relative height of the peak at the dendrite is returned.

```
bpapatt = (peak_voltage[0];location_soma - voltage_base;location_soma) / (max_i V[i];location_dend - voltage_base;location
APPEND bpapatt TO BPAPatt2
```

3.2 back-propagating AP attenuation (2nd)

Ratio of relative heights of somatic peak and dendritic peak

namespace / identifier	`LibV2:BPAPatt3`
unit	none
required features	peak voltage;location_soma voltage base;location*
required trace data	V;location_dend800
required parameters	none

Inject a short square pulse at soma that invokes exactly one action potential. The relative height of the action potential at soma is the difference of `peak voltage;location_soma` and `voltage base;location_soma`. The recording at the dendrite takes place at the thickest apical dendrite with a distance of 800 μm from soma. The relative height is the difference of the maximum of `V;location_dend800` and `voltage base;location_dend800`. Yield the relative height at soma divided by the relative height at the dendritic location.

```
bpapatt = (peak_voltage[0];location_soma - voltage_base;location_soma) / (max_i V[i];location_dend - voltage_base;location
APPEND bpapatt TO BPAPatt3
```


3.3 E2

Difference of the amplitude of the first and the second peak

namespace / identifier	`LibV2:E2`
unit	mV
required features	amp_drop_first_second;APDrop*
required trace data	none
required parameters	none

Take the mean of `peak voltage` over all repetitions of the stimulus protocol `APDrop` for the first and the second peak. Yield the difference.

```
APPEND mean of amp_drop_first_second;APDrop* TO E2
```

3.4 E3

Difference of the amplitude of the first and the last peak

namespace / identifier	`LibV2:E3`
unit	mV
required features	amp_drop_first_last;APDrop*
required trace data	none
required parameters	none

Take the mean of `peak voltage` over all repetitions of the stimulus protocol `APDrop` for the first and the last peak. Yield the difference.

```
APPEND mean of amp_drop_first_last;APDrop* TO E3
```

3.5 E4

Difference of the amplitude of the second and the last peak

namespace / identifier	`LibV4:E4`
unit	mV
required features	amp_drop_second_last;APDrop*
required trace data	none
required parameters	none

Take the mean of `peak voltage` over all repetitions of the stimulus protocol `APDrop` for the second and the last peak. Yield the difference.

```
APPEND mean of amp_drop_second_last;APDrop* TO E4
```


3.6 E5

Maximum difference of the height of two subsequent peaks

namespace / identifier	`LibV2:E5`
unit	mV
required features	max_amp_difference;APDrop*
required trace data	none
required parameters	none

Take the mean of `peak voltage` over all repetitions of the stimulus protocol `APDrop` for each peak. Yield the biggest difference between two peaks.

```
APPEND mean of max_amp_difference;APDrop* TO E5
```

3.7 E6 (AP amplitude)

Relative height of the first action potential

namespace / identifier	`LibV2:E6`
unit	mV
required features	AP amplitude;APWaveForm*
required trace data	none
required parameters	none

Take the mean of the first `AP amplitude` over all repetitions and iterations of the stimulus protocol `APWaveForm`.

```
APPEND mean of AP_amplitude;APWaveForm* at index 0 TO E6
```

3.8 E7 (AP duration)

Duration of the first action potential

namespace / identifier	`LibV2:E7`
unit	ms
required features	AP duration;APWaveForm*
required trace data	none
required parameters	none

Take the mean of the first `AP duration` over all repetitions and iterations of the stimulus protocol `APWaveForm`.

```
APPEND mean of AP_duration;APWaveForm* at index 0 TO E7
```


3.9 E8 (AP duration half width)

FWHM of the first action potential

namespace / identifier	`LibV2:E8`
unit	ms
required features	AP duration half width;APWaveForm*
required trace data	none
required parameters	none

Take the mean of the first `AP duration half width` over all repetitions and iterations of the stimulus protocol `APWaveForm`.

```
APPEND mean of AP_duration_half_width;APWaveForm* at index 0 TO E8
```

3.10 E9 (AP rise time)

Time from onset of the first action potential to the maximum

namespace / identifier	`LibV2:E9`
unit	ms
required features	AP rise time;APWaveForm*
required trace data	none
required parameters	none

Take the mean of the first `AP rise time` over all repetitions and iterations of the stimulus protocol `APWaveForm`.

```
APPEND mean of AP_rise_time;APWaveForm* at index 0 TO E9
```

3.11 E10 (AP fall time)

Time from maximum of the first action potential to offset

namespace / identifier	`LibV2:E10`
unit	ms
required features	AP fall time;APWaveForm*
required trace data	none
required parameters	none

Take the mean of the first `AP fall time` over all repetitions and iterations of the stimulus protocol `APWaveForm`.

```
APPEND mean of AP_fall_time;APWaveForm* at index 0 TO E10
```


3.12 E11 (AP rise rate)

Voltage change rate during the rising phase of the first action potential

namespace / identifier	`LibV2:E11`
unit	$\frac{V}{s}$
required features	AP rise rate;APWaveForm*
required trace data	none
required parameters	none

Take the mean of the first `AP rise rate` over all repetitions and iterations of the stimulus protocol `APWaveForm`.

```
APPEND mean of AP_rise_rate;APWaveForm* at index 0 TO E11
```

3.13 E12 (AP fall rate)

Voltage change rate during the falling phase of the first action potential

namespace / identifier	`LibV2:E12`
unit	$\frac{V}{s}$
required features	AP fall rate
required trace data	none
required parameters	none

Take the mean of the first `AP fall rate` over all repetitions and iterations of the stimulus protocol `APWaveForm`.

```
APPEND mean of AP_fall_rate;APWaveForm* at index 0 TO E12
```

3.14 E13 (fast AHP)

Voltage value of the onset of the first action potential relative to the subsequent AHP

namespace / identifier	`LibV2:E13`
unit	mV
required features	fast AHP;APWaveForm*
required trace data	none
required parameters	none

Take the mean of the first `fast AHP` over all repetitions and iterations of the stimulus protocol `APWaveForm`.

```
APPEND mean of fast_AHP;APWaveForm* at index 0 TO E13
```


3.15 E14 (AP amplitude)

Relative height of the second action potential

namespace / identifier	`LibV2:E14`
unit	mV
required features	AP Amplitude;APWaveForm*
required trace data	none
required parameters	none

Take the mean of the second `AP amplitude` over all repetitions and iterations of the stimulus protocol `APWaveForm`.

```
APPEND mean of AP_amplitude;APWaveForm* at index 1 TO E14
```

3.16 E15 (AP duration)

Duration of the second action potential

namespace / identifier	`LibV2:E15`
unit	ms
required features	AP duration;APWaveForm*
required trace data	none
required parameters	none

Take the mean of the second `AP duration` over all repetitions and iterations of the stimulus protocol `APWaveForm`.

```
APPEND mean of AP_duration;APWaveForm* at index 1 TO E15
```

3.17 E16 (AP duration half width)

FWHM of the second action potential

namespace / identifier	`LibV2:E16`
unit	ms
required features	AP duration half width;APWaveForm*
required trace data	none
required parameters	none

Take the mean of the second `AP duration half width` over all repetitions and iterations of the stimulus protocol `APWaveForm`.

```
APPEND mean of AP_duration_half_width;APWaveForm* at index 1 TO E16
```

3.18 E17 (AP rise time)

Time from onset of the second action potential to the maximum

namespace / identifier	`LibV2:E17`
unit	ms
required features	AP rise time;APWaveForm*
required trace data	none
required parameters	none

Take the mean of the second `AP rise time` over all repetitions and iterations of the stimulus protocol `APWaveForm`.

```
APPEND mean of AP_rise_time;APWaveForm* at index 1 TO E17
```

3.19 E18 (AP fall time)

Time from maximum of the second action potential to offset

namespace / identifier	`LibV2:E18`
unit	ms
required features	AP fall time;APWaveForm*
required trace data	none
required parameters	none

Take the mean of the second `AP fall time` over all repetitions and iterations of the stimulus protocol `APWaveForm`.

```
APPEND mean of AP_fall_time;APWaveForm* at index 1 TO E18
```

3.20 E19 (AP rise rate)

Voltage change rate during the rising phase of the second action potential

namespace / identifier	`LibV2:E19`
unit	$\frac{V}{s}$
required features	AP rise rate;APWaveForm*
required trace data	none
required parameters	none

Take the mean of the second `AP rise rate` over all repetitions and iterations of the stimulus protocol `APWaveForm`.

```
APPEND mean of AP_rise_rate;APWaveForm* at index 1 TO E19
```

3.21 E20 (AP fall rate)

Voltage change rate during the falling phase of the second action potential

namespace / identifier	`LibV2:E20`
unit	$\frac{V}{s}$
required features	AP fall rate
required trace data	none
required parameters	none

Take the mean of the second `AP fall rate` over all repetitions and iterations of the stimulus protocol `APWaveForm`.

```
APPEND mean of AP_fall_rate;APWaveForm* at index 1 TO E20
```

3.22 E21 (fast AHP)

Voltage value of the onset of the second action potential relative to the subsequent AHP

namespace / identifier	`LibV2:E21`
unit	mV
required features	fast AHP;APWaveForm*
required trace data	none
required parameters	none

Take the mean of the second `fast AHP` over all repetitions and iterations of the stimulus protocol `APWaveForm`.

```
APPEND mean of fast_AHP;APWaveForm* at index 1 TO E21
```

3.23 E22 (AP amplitude change)

Difference of the amplitudes of the second and the first action potential divided by the amplitude of the first action potential

namespace / identifier	`LibV2:E22`
unit	none
required features	AP amplitude change;APWaveForm*
required trace data	none
required parameters	none

Take the mean of `AP amplitude change` over all repetitions and iterations of the stimulus protocol `APWaveForm`.

```
APPEND mean of AP_amplitude_change;APWaveForm* at index 0 TO E22
```

3.24 E23 (AP duration change)

Difference of the durations of the second and the first action potential divided by the duration of the first action potential

namespace / identifier	`LibV2:E23`
unit	none
required features	AP duration change;APWaveForm*
required trace data	none
required parameters	none

Take the mean of `AP duration change` over all repetitions and iterations of the stimulus protocol `APWaveForm`.

```
APPEND mean of AP_duration_change;APWaveForm* at index 0 TO E23
```

3.25 E24 (AP duration half width change)

Difference of the FWHM of the second and the first action potential divided by the FWHM of the first action potential

namespace / identifier	`LibV2:E24`
unit	none
required features	AP duration half width change;APWaveForm*
required trace data	none
required parameters	none

Take the mean of `AP duration half width change` over all repetitions and iterations of the stimulus protocol `APWaveForm`.

```
APPEND mean of AP_duration_half_width_change;APWaveForm* at index 0 TO E24
```

3.26 E25 (AP rise rate change)

Difference of the rise rates of the second and the first action potential divided by the rise rate of the first action potential

namespace / identifier	`LibV2:E25`
unit	none
required features	AP rise rate change;APWaveForm*
required trace data	none
required parameters	none

Take the mean of `AP rise rate change` over all repetitions and iterations of the stimulus protocol `APWaveForm`.

```
APPEND mean of AP_rise_rate_change;APWaveForm* at index 0 TO E25
```


3.27 E26 (AP fall rate change)

Difference of the fall rates of the second and the first action potential divided by the fall rate of the first action potential

namespace / identifier	`LibV2:E26`
unit	none
required features	AP fall rate change;APWaveForm*
required trace data	none
required parameters	none

Take the mean of `AP fall rate change` over all repetitions and iterations of the stimulus protocol `APWaveForm`.

```
APPEND mean of AP_fall_rate_change;APWaveForm* at index 0 TO E26
```

3.28 E27 (fast AHP change)

Difference of the `fast AHP` of the second and the first action potential divided by the `fast AHP` of the first action potential

namespace / identifier	`LibV2:E27`
unit	none
required features	fast AHP change;APWaveForm*
required trace data	none
required parameters	none

Take the mean of `fast AHP change` over all repetitions and iterations of the stimulus protocol `APWaveForm`.

```
APPEND mean of fast_AHP_change;APWaveForm* at index 0 TO E27
```

3.29 E29

Ohmic input resistance

namespace / identifier	`LibV2:E29`
unit	MΩ
required features	steady state hyper;IV*
required trace data	stimulus current
required parameters	none

Get the points (`stimulus current`, `steady state hyper`) for all repetitions and iterations of the stimulus protocol `IV` and fit a straight line. Yield the slope.

```
FOR suffix IN IV* DO
  APPEND stimulus_current;suffix TO x
  APPEND steady_state_hyper;suffix TO y
ENDFOR
slope, residuals, R^2 = fit_straight_line(x, y)
APPEND slope TO E29
```

3.30 E29: coefficient of determination

The coefficient of determination (often: R^2) of the straight line fit according to E29

namespace / identifier	`LibV2:E29_cod`
unit	none
required features	E29
required trace data	none
required parameters	none

```
FOR suffix IN IV* DO
  APPEND stimulus_current;suffix TO x
  APPEND steady_state_hyper;suffix TO y
ENDFOR
slope, residuals, R^2 = fit_straight_line(x, y)
APPEND R^2 TO E29_cod
```

3.31 E39

The slope of a linear fit of the curve `mean frequency` vs. `stimulus current`

namespace / identifier	`LibV2:E39`
unit	$\frac{\mathrm{Hz}}{\mathrm{nA}}$
required features	mean frequency;IDthreshold*
required trace data	stimulus current;IDthreshold*
required parameters	none

Get the points (`stimulus current`, `mean frequency`) for all repetitions and iterations of the stimulus protocol `IDthreshold` and fit a straight line.

remarks: A straight line generally is not an appropriate fit to the described curve.

```
FOR suffix IN IDthreshold* DO
  APPEND stimulus_current;suffix TO x
  APPEND mean_frequency;suffix TO y
ENDFOR
slope, residuals, R^2 = fit_straight_line(x, y)
APPEND slope TO E39
```


3.32 E39: coefficient of determination

The coefficient of determination (often: R^2) of the straight line fit according to E39

namespace / identifier	`LibV2:E39_cod`
unit	none
required features	E39
required trace data	none
required parameters	none

```
FOR suffix IN IDthreshold* DO
  APPEND stimulus_current;suffix TO x
  APPEND steady_state_hyper;suffix TO y
ENDFOR
slope, residuals, R^2 = fit_straight_line(x, y)
APPEND R^2 TO E39_cod
```

3.33 E40 (time to first spike)

Average time from the begin of the stimulus to the maximum of the first peak

namespace / identifier	`LibV2:E40`
unit	ms
required features	time to first spike;IDrest*
required trace data	none
required parameters	none

Take the mean of `time to first spike` over all repetitions and iterations of the stimulus protocol `IDrest`

```
APPEND mean of time_to_first_spike;IDrest* at index 0 TO E40
```

3.34 time constant mean

Membrane time constant

namespace / identifier	`LibV2:time_constant_mean`
unit	ms
required features	time constant;IV*
required trace data	none
required parameters	none

Take the mean of `time constant` over all repetitions and iterations of the stimulus protocol `IV`

```
APPEND mean of time_consant;IV* at index 0 TO time_constant_mean
```


3.35 time constant standard deviation

Standard deviation of the membrane time constant

namespace / identifier	`LibV2:time_constant_std`
unit	ms
required features	time constant mean time constant;IV*
required trace data	none
required parameters	none

Calculate the standard deviations of all values of `time constant` obtained on different repetitions and iterations of the stimulus protocol `IV`.

```
APPEND standard deviation of time_constant;IV* at index 0 TO time_constant_std
```

4 LibV5

4.1 AP begin voltage

Voltage values at the onset of each action potential

namespace / identifier	`LibV5:AP_begin_voltage`
unit	mV
required features	AP_begin_indices
required trace data	V T stim start stim end
required parameters	none

Return the voltage levels at `AP_begin_indices`.

```
begin_0,...,begin_{n-1} = AP_begin_indices
FOR i = 0,...,n-1 DO
  APPEND V[begin_i] TO AP_begin_voltage
ENDFOR
```


4.2 AHP time from peak

Time between AP peaks and AHP depths

namespace / identifier	`LibV5:AHP_time_from_peak`
unit	ms
required features	min_AHP_indices peak_indices
required trace data	V T stim start stim end
required parameters	none

Obtain the `min_AHP_indices` and `peak_indices`, and calculate the time between these indices in the T array.

```
peak_0, ... ,peak_{m-1} = peak_indices
ahp_0, ... ,ahp_{n-1} = min_AHP_indices
IF m > n THEN
  FAIL
ENDIF
FOR i = 0, ... , m - 1 DO
  APPEND T[ahp_i] - T[peak_i] TO AHP_time_from_peak
ENDFOR
```

4.3 AP amplitude from voltagebase

The height of the action potential measured from voltage base

namespace / identifier	`LibV1:AP_amplitude_from_voltagebase`
unit	mV
required features	voltage base peak voltage
required trace data	V
required parameters	none

Yield the difference of `peak voltage` and `voltage base` for each peak.

```
pv_0, ... ,pv_{n-1} = peak_voltage
FOR i = 0, ... , n - 1 DO
  APPEND pv_i - voltage_base TO AP_amplitude_from_voltagebase
ENDFOR
```

