
OBM Documentation

AIMLab

Sep 24, 2020

Contents:

1	OBM package	1
1.1	OBM.ComplexityMeasures class	1
1.2	OBM.DesaturationsMeasures class	1
1.3	OBM.HypoxicBurdenMeasures class	2
1.4	OBM.ODIMeasure class	3
1.5	OBM.OverallGeneralMeasures class	3
1.6	OBM.PeriodicityMeasures class	4
1.7	OBM.Preprocessing class	5
2	Indices and tables	7
	Python Module Index	9
	Index	11

1.1 OBM.ComplexityMeasures class

```
class OBM.ComplexityMeasures.ComplexityMeasures (CTM_Threshold=0.25,  
                                                DFA_Window=20,      M_Sampen=3,  
                                                R_Sampen=0.2)
```

Bases: object

Class that calculates Complexity Features from spo2 time series. Suppose that the data has been preprocessed.

:param *signal*: 1-d array, of shape (N,) where N is the length of the signal *CTM_Threshold*: Radius of Central Tendency Measure. *DFA_Window*: Length of window to calculate DFA biomarker. *M_Sampen*: Embedding dimension to compute SampEn. *R_Sampen*: Tolerance to compute SampEn.

compute (*signal*) → OBM._ResultsClasses.ComplexityMeasuresResults

Parameters *signal* – 1-d array, of shape (N,) where N is the length of the signal

Returns

ComplexityMeasuresResults class containing the following features:

- ApEn: Approximate Entropy.
- LZ: Lempel-Ziv complexity.
- CTM: Central Tendency Measure.
- SampEn: Sample Entropy.
- DFA: Detrended Fluctuation Analysis.

1.2 OBM.DesaturationsMeasures class

```
class OBM.DesaturationsMeasures.DesaturationsMeasures (begin, end)
```

Bases: object

Class that calculates the Desaturation Features from spo2 time series. Suppose that the data has been preprocessed.

Parameters

- **begin** – List of indices of beginning of each desaturation event.
- **end** – List of indices of end of each desaturation event.

compute (*signal*) → OBM._ResultsClasses.DesaturationsMeasuresResults

Parameters **signal** – 1-d array, of shape (N,) where N is the length of the signal

Returns

DesaturationsMeasuresResults class containing the following features:

- DL_u: Mean of desaturation length
- DL_sd: Standard deviation of desaturation length
- DA100_u: Mean of desaturation area using 100% as baseline.
- DA100_sd: Standard deviation of desaturation area using 100% as baseline
- DAmx_u: Mean of desaturation area using max value as baseline.
- DAmx_sd: Standard deviation of desaturation area using max value as baseline
- DD100_u: Mean of depth desaturation from 100%.
- DD100_sd: Standard deviation of depth desaturation from 100%.
- DDmx_u: Mean of depth desaturation from max value.
- DDmx_sd: Standard deviation of depth desaturation from max value.
- DS_u: Mean of the desaturation slope.
- DS_sd: Standard deviation of the desaturation slope.
- TD_u: Mean of time between two consecutive desaturation events.
- TD_sd: Standard deviation of time between 2 consecutive desaturation events.

desat_embedding ()

Help function for the class

Returns helper arrays containing the information about desaturation lengths and areas.

1.3 OBM.HypoxicBurdenMeasures class

```
class OBM.HypoxicBurdenMeasures.HypoxicBurdenMeasures (begin, end,
                                                         CT_Threshold=90,
                                                         CA_Baseline=None)
```

Bases: object

Class that calculates Hypoxic Burden Features from spo2 time series. Suppose that the data has been preprocessed.

Parameters

- **begin** – List of indices of beginning of each desaturation event.
- **end** – List of indices of end of each desaturation event.

- **CT_Threshold** – Percentage of the time spent below the “CT_Threshold” % oxygen saturation level.
- **CA_Baseline** – Baseline to compute the CA feature. Default value is mean of the signal.

compute (*signal*)

Parameters **signal** – 1-d array, of shape (N,) where N is the length of the signal

Returns

HypoxicBurdenMeasuresResults class containing the following features:

- CA: Integral SpO2 below the xx SpO2 level normalized by the total recording time
- CT: Percentage of the time spent below the xx% oxygen saturation level
- POD: Percentage of oxygen desaturation events
- AODmax: The area under the oxygen desaturation event curve, using the maximum SpO2 value as baseline and normalized by the total recording time
- AOD100: Cumulative area of desaturations under the 100% SpO2 level as baseline and normalized by the total recording time

1.4 OBM.ODIMeasure class

class OBM.ODIMeasure.ODIMeasure (*ODI_Threshold=3*)

Bases: object

Class that calculates the ODI from spo2 time series. Suppose that the data has been preprocessed.

Parameters **ODI_Threshold** – Threshold to compute Oxygen Desaturation Index.

compute (*signal*) → OBM._ResultsClasses.ODIMeasureResult

Parameters **signal** – The SpO2 signal, of shape (N,)

Returns

ODIMeasureResult class containing the following features:

- ODI: the average number of desaturation events per hour.
- begin: List of indices of beginning of each desaturation event.
- end: List of indices of end of each desaturation event.

1.5 OBM.OverallGeneralMeasures class

class OBM.OverallGeneralMeasures.OverallGeneralMeasures (*ZC_Baseline=None,*
percentile=1,
M_Threshold=2,
DI_Window=12)

Bases: object

Class that calculates Overall General Features from spo2 time series. Suppose that the data has been preprocessed.

Parameters

- **ZC_Baseline** – Baseline for calculating number of zero-crossing points.

- **percentile** – Percentile to perform. For example, for percentile 1, the argument should be 1
- **M_Threshold** – Percentage of the signal M_Threshold % below median oxygen saturation. Typically use 1,2 or 5

compute (*signal*) → OBM._ResultsClasses.OverallGeneralMeasuresResult

Parameters **signal** – 1-d array, of shape (N,) where N is the length of the signal

Returns

OverallGeneralMeasuresResult class containing the following features:

- AV: Average of the signal.
- MED: Median of the signal.
- Min: Minimum value of the signal.
- SD: Std of the signal.
- RG: SpO2 range (difference between the max and min value).
- P: percentile.
- M: Percentage of the signal x% below median oxygen saturation.
- ZC: Number of zero-crossing points.
- DI: Delta Index.

1.6 OBM.PeriodicityMeasures class

class OBM.PeriodicityMeasures.**PRSAmeasures** (*PRSA_Window=10, K_AC=2*)

Bases: object

Function that calculates PRSA Features from spo2 time series. Suppose that the data has been preprocessed.

:param PRSA_Window: Fragment duration of PRSA. K_AC: Number of values to shift when computing autocorrelation

compute (*signal*) → OBM._ResultsClasses.PRSAResults

Parameters **signal** – 1-d array, of shape (N,) where N is the length of the signal

Returns

PRSAResults class containing the following features:

- PRSAc: PRSA capacity.
- PRSAad: PRSA amplitude difference.
- PRSAos: PRSA overall slope.
- PRSAsb: PRSA slope before the anchor point.
- PRSAa: PRSA slope after the anchor point.
- AC: Autocorrelation.

class OBM.PeriodicityMeasures.**PSDMeasures**

Bases: object

Function that calculates PSD Features from spo2 time series. Suppose that the data has been preprocessed.

compute (*signal*) → OBM._ResultsClasses.PSDResults

:param **signal**: The SpO2 signal, of shape (N,)

Returns

PSDResults class containing the following features:

- **PSD_total**: The amplitude of the spectral signal.
- **PSD_band**: The amplitude of the signal multiplied by a band-pass filter between 0.014 and 0.033 Hz.
- **PSD_ratio**: The ratio between PSD_total and PSD_band.
- **PDS_peak**: The max value of the FFT into the band 0.014-0.033 Hz.

1.7 OBM.Preprocessing class

OBM.Preprocessing.**block_data** (*signal*, *threshold=50*)

Apply a block data filter to the SpO2 signal.

Parameters

- **signal** – 1-d array, of shape (N,) where N is the length of the signal
- **(Optional)** (*threshold*) – threshold parameter for block data filter.

Returns preprocessed signal, 1-d numpy array.

OBM.Preprocessing.**delta_filter** (*signal*, *Diff=4*)

Apply Delta Filter to the signal.

Parameters

- **signal** – 1-d array, of shape (N,) where N is the length of the signal
- **Diff** – parameter of the delta filter.

Returns preprocessed signal, 1-d numpy array.

OBM.Preprocessing.**median_spo2** (*signal_spo2*, *FilterLength=9*)

Apply a median filter to the SpO2 signal. Median filter used to smooth the spo2 time series and avoid sporadic increase/decrease of spo2 which could affect the detection of the desaturations. Assumption: any missing/abnormal values are represented as 'np.nan'

Parameters

- **signal** – 1-d array, of shape (N,) where N is the length of the signal
- **(Optional)** (*FilterLength*) – The length of the filter.

Returns preprocessed signal, 1-d numpy array.

OBM.Preprocessing.**resamp_spo2** (*signal*, *OriginalFreq*)

Resample the SpO2 signal to 1Hz. Assumption: any missing/abnormal values are represented as 'np.nan'

Parameters

- **signal** – 1-d array, of shape (N,) where N is the length of the signal
- **OriginalFreq** – the original frequency.

Returns resampled signal, 1-d numpy array, the resampled spo2 time series at 1Hz

`OBM.Preprocessing.set_range` (*signal*, *Range_min=50*, *Range_max=100*)

Range function. Remove values lower than 50 or greater than 100, considered as non-physiological

Parameters **signal** – 1-d array, of shape (N,) where N is the length of the signal

Returns preprocessed signal, 1-d numpy array.

CHAPTER 2

Indices and tables

- `genindex`
- `modindex`
- `search`

O

OBM, [1](#)
OBM.ComplexityMeasures, [1](#)
OBM.DesaturationsMeasures, [1](#)
OBM.HypoxicBurdenMeasures, [2](#)
OBM.ODIMeasure, [3](#)
OBM.OverallGeneralMeasures, [3](#)
OBM.PeriodicityMeasures, [4](#)
OBM.Preprocessing, [5](#)

B

`block_data()` (in module *OBM.Preprocessing*), 5

C

`ComplexityMeasures` (class in *OBM.ComplexityMeasures*), 1

`compute()` (*OBM.ComplexityMeasures.ComplexityMeasures* method), 1

`compute()` (*OBM.DesaturationsMeasures.DesaturationsMeasures* method), 2

`compute()` (*OBM.HypoxicBurdenMeasures.HypoxicBurdenMeasures* method), 3

`compute()` (*OBM.ODIMeasure.ODIMeasure* method), 3

`compute()` (*OBM.OverallGeneralMeasures.OverallGeneralMeasures* method), 4

`compute()` (*OBM.PeriodicityMeasures.PRSAMeasures* method), 4

`compute()` (*OBM.PeriodicityMeasures.PSDMeasures* method), 4

D

`desat_embedding()` (*OBM.DesaturationsMeasures.DesaturationsMeasures* method), 2

`DesaturationsMeasures` (class in *OBM.DesaturationsMeasures*), 1

`delta_filter()` (in module *OBM.Preprocessing*), 5

H

`HypoxicBurdenMeasures` (class in *OBM.HypoxicBurdenMeasures*), 2

M

`median_spo2()` (in module *OBM.Preprocessing*), 5

O

OBM (module), 1

OBM.ComplexityMeasures (module), 1

OBM.DesaturationsMeasures (module), 1

OBM.HypoxicBurdenMeasures (module), 2

OBM.ODIMeasure (module), 3

OBM.OverallGeneralMeasures (module), 3

OBM.PeriodicityMeasures (module), 4

OBM.Preprocessing (module), 5

ODIMeasure (class in *OBM.ODIMeasure*), 3

OverallGeneralMeasures (class in *OBM.OverallGeneralMeasures*), 3

P

PRSAMeasures (class in *OBM.PeriodicityMeasures*), 4

PSDMeasures (class in *OBM.PeriodicityMeasures*), 4

R

`resamp_spo2()` (in module *OBM.Preprocessing*), 5

S

`set_range()` (in module *OBM.Preprocessing*), 5