BioSigKit

Contents of **BioSigKit**:

BioSigKit is both a directory and a function.

```
  ============== BioSigKit ==================
  BioSigKit is a set of useful signal processing tools that are either
  developed by me personally or others in different fields of biosignal
  processing. BioSigKit is a wrapper with a simple visual interface that
  gathers this tools under a simple easy to use platform. This tool might
  be only used for non-commercial, academic, research and learning
  purposes.

 % ============== How to start ===================
  example:: obj = BioSigKit(Sig, Fs, Gr)
  Where:::
  Sig: is the signal
  Fs : Sample Rate
  Gr : Flag for showing the interface or not
  Then you can call any subroutine
 % ====List Of Subroutines that you can call for QRS detection =========
  ------- Algorithm -------------------- How to Call ---------------
  (1) Pan-Tompkins Algorithm :        obj.PanTompkins
  (2) Phase Space Reconstruction :    obj.PhaseSpaceAlg
  (3) RST State-Machine :             obj.StateMachine
  (4) Filter Bank:                    obj.FilterBankQRS
  (5) MTEO qrstAlg:                   obj.MTEO_qrstAlg
  (6) AMPD:                           obj.AMPD_PAlg
 % ==== List of all subroutines for ACC, EMG and etc processing ========
  (7) Activity Detection Hilbert:     obj.Env_hilbert
  ---------------- Inputs ------------------ %
  Smooth_window : Length of smoothing window in nr of sample
  threshold_style : Set 0 for Automatic, Set 1 for Manual
  DURATION : The number of samples for signal to be above
  threshold to be considered active
  ---------------- Output ------------------ %
  alarm :  Pattern of activities
  ---------------- Demo -------------------- %
  v = repmat([.1*ones(200,1);ones(100,1)],[10 1]);                    % generate true variance profile
  obj.sig = sqrt(v).*randn(size(v));
  obj.Env_hilbert;
 -----------------
  (8) Comp Mobility and Complexity:   obj.ComputeHjorthP
 -----------------
  (9) Posture detection 3 Chann ACC:  obj.ACC_Act
  ------------- Inputs ----------------------- %
  obj.sig : 3 axis Accelerometer signal where, each row is an axis
  and each column a sample (e.g. (3,:))
  obj.Fs : Sampling frequency of the Accelerometer
  ------------- Output ----------------------- %
  output : adaptively filtered ACC channels based on activity
  state : activity level (0:steady,1:walking,2:joggin)
  EE : Energy Expenditure over 1 min (or length sig)
  F : Bandpass filter in Hz
  SMA : Signal Magnitude area
 ------------------
  (10)PsuedoCorr template matching :  obj.TemplateMatch
  ------------------- Inputs -------------------- %
  template :  A template in the form of a vector, the length of
```

```
 the template should be smaller than the signal.
 lag : a lag in terms of nr of samples to move the template,
 it should be smaller than the length of the template
 ------------------- Outputs -------------------- %
 PsC_s : template matching score in range [0,1].
 best_lag: the lag that gave the highest correlation score.
-----------------
 (11)ECG derived respiration :       obj.EDR_comp
-----------------
 (12)ACC derived respiration :       obj.ADR_comp
% ===== General Projective, linear and nonlinear filterings ============ %%
 (13)Real-time neural PCA:           obj.neural_pca
 ----------------- Inputs ------------------- %
 X : Multi-channel signal, each row represents a channel and
 each column a sample.
 nPCA: Number of PCAs to extract
 nit: Number of iterations to go through the whole signal
 T : Learning rate in range [0,1], default:0.9
 ---------------- Outputs ------------------- %
 EigVec: Eigen vectors
 PC : PCs
 Eigval : Eigenvalues
-----------------
 (14)Adaptive Filtering:             obj.adaptive_filter
                 * RLS : Recursive Least Squares Filter
                 * ALE : Adaptive Line Enhancer (Delayed Filter)
                 * VLALE : Variable Leaky Adaptive Line Enhancer
                 * NLMS_ecg : Normalized Least Mean Squares filter for artficat
                   removal in ECG based on 3 channel Accelerometer recordings
-------------------- Inputs ---------------------------%
 type: type of the filter (numeric):
       (1) RLS : Recursive Least Squares Filter
       (2) ALE : Adaptive Line Enhancer (Delayed Filter)
       (3) VLALE : Variable Leaky Adaptive Line Enhancer
       (4) NLMS_ecg : Normalized Least Mean Squares filter for
       artficat removal in ECG based on 3 channel Accelerometer
       recordings
 ref: Reference signal:
        * For RLS filter it is single channel (1*N)
        * For NLMS_ECG filter it should be 3 Channel
        Accelerometer (3*N)
 obj.Sig: input signal to clean (single channel vector)
 order : order of the filter (for VLALE and ALE also delay)
 lambda : learning rate(0<= lambda <=1, usually close to 1)
-------------------- Outputs --------------------------%
 output: cleaned signal
 error_sig : error signal
-----------------
 (15)Nonlinear phasespace filtering: obj.nonlinear_phase_filt
-------------------- Method ---------------------------%
 Employs nonlinear phasespace filter to clean up the signal. This
 method is very strong and even able to extract foetal ecg from
 single channel maternal recordings. Please refer to examples of
 BioSigKit for further details.
-------------------- Inputs ----------------------------%
 sig : Signal to be analyzed (single channel)
 t : Number of samples for computing delayed phase space (def: 1)
 d : Embedding dimension to consider (def: 50)
 m : dimension of null space (def: 49)
 r : number of nearest neighbors to consider
 (normally a large number def: length(sig)/4)
 --------------------- Output ------------------------ %
 output : Cleaned Signal
 --------------------- example ----------------------- %
 output = projective(foetal_ecg(:,2), 1, round(Fs/5), round(Fs/6.25), 1500);
 (16)Teager-Keiser energy operator:  obj.TK_comp
```

Class Details

Superclasses	handle
Sealed	false
Construct on load	false

Constructor Summary

BioSigKit	--------- Check Inputs -------------%

Property Summary

ACC	3 channel ACC
Alg	Alg type from interface
FreqValHolder	Holds SampFreq from interface
Fs	Sample Freq
Gr	Flag for showing the interface
LoadedSig	Loaded Sig
PhasePeriod	Phase Period Used for Phase Space
PlotResult	Verbose results of each algorithm
Results	Struct holding results
ScalogramL	Scalogram Length
Sig	Signal
SigView	Signal View Axis
Status	Loader
complexity	Complexity based on Hjorth Param
mobility	mobility based on Hjorth
panel	Main Visualization Panel
path_alg	Path of Algorithm
slashchar	OS dependent /
statsC	Holding Stats

Method Summary

	ACC_Act	------------- Inputs ----------------------- %
	ADR_comp	------------ Method -----------------------------%
	AMPD_PAlg	
protected	BioSigKitPanel	================= Sets up BioSigKit GUI ====================== %%
	ComputeHjorthP	---------------- Hjorth Parameters ------------------- %
	EDR_comp	--------------------- Method ------------------------ %
	Env_hilbert	---------------- Inputs ------------------ %
	FilterBankQRS	
	ImportSig	
	MTEO_qrstAlg	
	PanTompkins	
	PhaseSpaceAlg	
	ResetStats	
	RunAlg	
	StateMachine	
	TK_comp	--------------- Outputs ----------------%
	TemplateMatch	------------------- Inputs --------------------- %
	UpdateFig	
	adaptive_filter	-------------------- Inputs --------------------------%
	addlistener	Add listener for event.
	delete	Delete a handle object.
	eq	== (EQ) Test handle equality.
	findobj	Find objects matching specified conditions.
	findprop	Find property of MATLAB handle object.

	ge	>= (GE) Greater than or equal relation for handles.
	gt	> (GT) Greater than relation for handles.
Sealed	isvalid	Test handle validity.
	le	<= (LE) Less than or equal relation for handles.
	lt	< (LT) Less than relation for handles.
	ne	~= (NE) Not equal relation for handles.
	neural_pca	----------------- Inputs ------------------- %
	nonlinear_phase_filt	-------------------- Method ---------------------------%
	notify	Notify listeners of event.
	visualizeResults	--------------- First the Figure ------------------%

Event Summary

ObjectBeingDestroyed	Notifies listeners that a particular object has been destroyed.