

# User Manual

---

This document is in “beta” mode. Thus it is a work in progress. If the information that you need isn’t listed or isn’t clearly stated, please contact the author.

Updated – 10/24/2010

**Contact Information:**

John T. Ramshur  
University of Memphis  
Department of Biomedical Engineering  
Memphis, TN  
jramshur@gmail.com  
jramshur@memphis.edu

## Contents

Description .....	3
Installation .....	3
A. Standalone Application.....	3
How to Install .....	3
B. MATLAB Source Code .....	3
How to Install .....	4
Important Notes – Must Read .....	4
Quick Start Guide .....	5
Misc. Information.....	5
Shortcut Keys .....	5
Other .....	5
ECG Data .....	7
ECG Filtering.....	7
Beat Detection .....	8
Building a template .....	11
IBI Filtering .....	11
Exporting.....	12
File Annotations and Database .....	13

## Description

ECG Viewer is a MATLAB GUI for reviewing, processing, and annotating electrocardiogram (ECG) data files. ECG Viewer offers an annotation database, ECG filtering, beat detection using template matching, and inter-beat interval (IBI or RR) filtering. Program was created using MATLAB 2008b.

## Installation

Below there are instructions for the two installation types available for HRVAS. Type A describes how to install HRVAS for use without MATLAB. Type B describes how to install/setup HRVAS if you have already have MATLAB. If you don't know what MATLAB is...then Type A is for you.

### A. Standalone Application

---

Software Dependencies: Software needed to run HRVAS

1. **MATLAB v2008 or higher** - Older versions probably work as long as it's not too old. If you don't have MATLAB, please contact the author about a stand-alone version.
2. **Microsoft Access** – A MS Access database file is needed for storing annotations. MS Access needs to be installed for its drivers and for viewing the database file.

#### How to Install

---

1. Install the MATLAB Compiler Runtime (MCR). Download and run MCRInstaller.exe. MCR is needed to allow HRVAS to run outside of the MATLAB environment.
2. Unzip the ECG\_Viewer zip package. Run ECGViewer.exe to open the user interface.

### B. MATLAB Source Code

---

**These dependencies must be present and/or installed before using ECG Viewer.**

Software Dependencies: Software needed to run HRVAS

1. **MATLAB v2008 or higher** - Older versions probably work as long as it's not too old. If you don't have MATLAB, please use the stand-alone version.
2. **Microsoft Access** – A MS Access database file is needed for storing annotations. MS Access needs to be installed for its drivers and for viewing the database file.

MATLAB Toolbox Dependencies (Optional) : Toolboxes needed to run certain aspects of ECG Viewer

1. MATLAB Database toolbox (needed IF created/vieweing annotations)
2. MATLAB Wavelet Toolbox (needed IF using wavelet filtering)
3. MATLAB Distributed Computing Engine (needed IF using multicore support during beat detection)

File Dependencies: other files needed to run ECG Viewer

1. ecgViewer.m - primary GUI for ECG Viewer
2. fastsmooth.m - signal smoothing function
3. locateOutliers.m - locates IBI outliers or ectopic beats
4. makeTemplate.m - function to create a template
5. matchTemplate.m - template matching function
6. peakDetect.m - function to detect signal peaks
7. annotations.mdb - MATLAB database to hold annotations
8. avg.mexw32 - compiled function for computing the mean (faster than MATLAB's)
9. crosscorr.mexw32 - compiled function for computing cross correlation coef.

## How to Install

---

1. Unzip the ECG\_Viewer zip package containing MATLAB code. Run the main.m file within MATLAB to open the user interface.

## Important Notes – Must Read

---

1. Before using the database feature, you must create a datasource in your Windows environment named "ann\_db". See <http://MATLAB.izmiran.ru/help/toolbox/database/instal2.html#18933>. This is only done once. This datasource must point to your Access database file. Please use the annotations.mdb file supplied as a blank database and rename it if needed.
2. If running ECG Viewer in MATLAB, you may have to give MATLAB administrator rights to access the database. Right click on the MATLAB shortcut and click "Run as administrator".

## Quick Start Guide

1. Select a directory that contains your ECG data files
2. Select an ECG data file from the list.
3. Set the sample rate used during the recording.
4. Select filtering and beat detection options.
5. Click “Preview”.
6. Browse the data, add/read annotations, change options, etc.
7. Export data ... if needed.

## Misc. Information

### Shortcut Keys

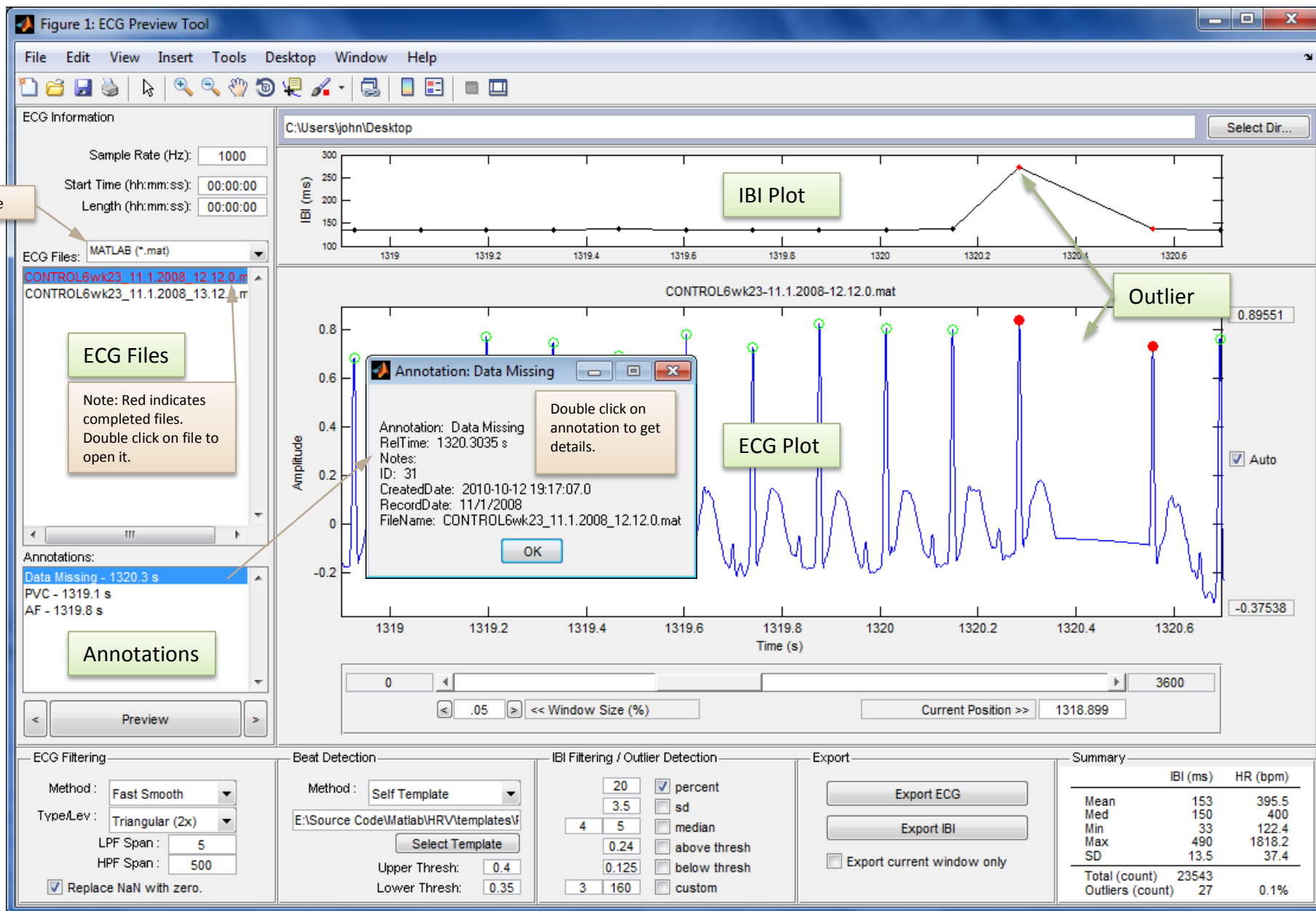
---

- Use the up and down arrow keys to jump to the next/previous IBI outlier.
- Use the left and right arrow keys to jump to the next/previous detection.
- You must click on somewhere on the GUI before using the arrow keys.
- Double click on annotations to see more details.
- Double click on file name to load the ECG.

### Other

---

- Use the arrow buttons next to the “preview” button to automatically load the previous/next ECG data file
- “Start Time” and “Length” options located above the list of ECG files are used to limit the amount of data loaded from the ECG file. “Start time” represents the beginning time. “Length” represents the length of data to load in reference to the starting time. Start time + length = end time.



## ECG Data

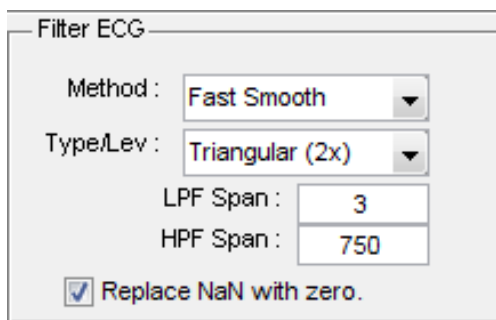
---

As of version 1.1.13, ECG Viewer can read MATLAB binary files (.mat) and ASCII text files (.txt). If using MATLAB binary files (.mat), the software will load the data and then try to find a variable named “ecg” or “ECG”. If not found, the software will use the first variable loaded from the .mat file.

The ECG data is expected to have one or two columns. If one column is used, this column must represent ECG amplitude in arbitrary units. If two columns are used, the first column represents time in units of seconds. The second column represents ECG amplitude in arbitrary units.

## ECG Filtering

---



Two type of ECG filtering are currently available in ECG Viewer.

1. Fast Smooth – Filters ECG base using two moving average filters. User can specify the number of passes from 1x-3x. Low-pass filter (LPF) setting can be adjusted by changing the LPF span. Larger LPF spans eliminate a broader range of high frequency components. High-pass filter (HPF) setting can be adjusted by changing the HPF span. Smaller HPF spans eliminate a broader range of low frequency components. (this need rewording)
2. Wavelet – This filtering method used MATLAB’s wavelet toolbox. The ECG is first decomposition using wavelet decomposition. The user defines the wavelet family to use and the sub-bands (frequency bands) to remove.

## Beat Detection

---

Beat detection is accomplished using a correlation-based template matching algorithm. A sliding window is passed across the ECG signal, and a correlation coefficient,  $r_{xy}$ , is computed between the template and the ECG contained within each window of equal length. Here the correlation coefficient represents the similarity between template and ECG segment. This process produces a new data series containing correlation coefficients  $r_{xy}$ . The correlation coefficient between two  $N$  length segments of  $x$  and  $y$  is computed using the following equation:

$$r_{xy} = \frac{\sum_{n=1}^N [(x(n) - \bar{x})(y(n) - \bar{y})]}{\sqrt{\sum_{n=1}^N [x(n) - \bar{x}]^2 \sum_{n=1}^N [y(n) - \bar{y}]^2}}.$$

If the correlation,  $r_{xy}$ , between template and ECG segment exceed a defined upper threshold value a detection flag is set to high (Figure.1-c). This flag is held in a high state until  $r_{xy}$  drops below a defined lower threshold (Figure.1-d). A beat is set as the largest  $r_{xy}$  peak between any high and low detection flag events (Figure.1-b). Detected beats are marked with green circles. This double threshold method helps eliminate multiple peak detections within a close vicinity of one another.

Computational speed of template matching can be improved using the following three methods (see the function `detectBeats()` within `ecgViewer.m` and see `matchTemplate.m`):

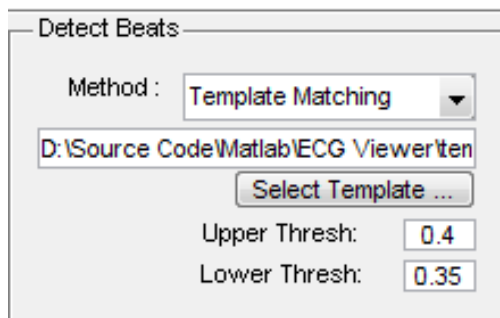
1. Skip Indexing - This method involves reducing the number of times the sliding window translates across the ECG signal. By translating every third ECG sample instead of indexing one sample, 1/3 as many correlation coefficients are computed. Skipped  $r_{xy}$  values are replaced using cubic spline interpolation.
2. Parallelization - This method involves parallelizing a portion of the template matching code for multi-core processor environments.
3. Combination - This method combines both the skip indexing and parallelized methods.



Skipping samples, parallelizing code, and a combination of the two methods reduce computation time (17.2 s original) by approximately 64% (6.2 s), 58% (7.3 s), and 82% (3.16 s) respectively. These data are based on the results of 3 simulations of each technique using an ECG segment of 600,000 samples and a template of 161 samples in length.

Two types of detection methods exist on the ECG Viewer GUI:

1. Template Matching – standard template matching
2. Self Template – This methods using the defined template file to create a new template base on the current ECG file. The current template is used to detect QRS complexes from the first 60,000 samples of ECG. These detected beats are then averaged together to create a “self template”. This self template is used in beat detection. NOTE: The sample length and template width may need to be modified for your preference.



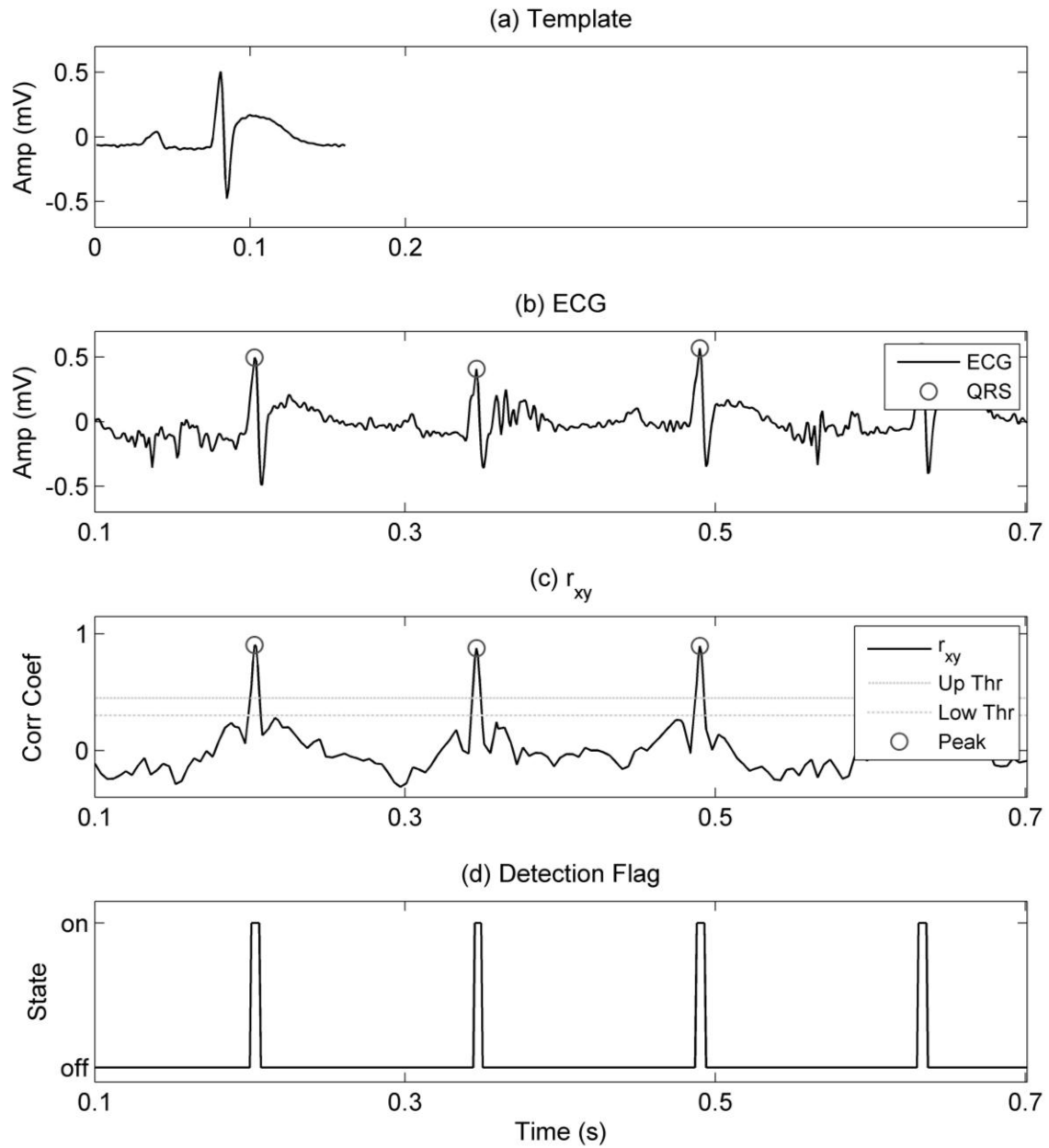


Figure.1 – QRS detection using template matching. Plots include (a) example template used for QRS detection by template matching, (b) rat ECG with detected QRS complexes, (c)  $r_{xy}$  data series and peaks detected using the double threshold technique, and (d) state of the detection flag used in the double threshold technique.

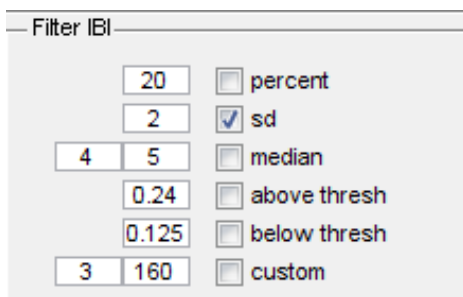
## Building a template

---

- ECG Viewer's template matching marks beats as the center of a matched template. For example, if you make the center of your template correspond with the R-wave peak, then template matching will mark R-waves on your ECG data, etc.
- Averaging several QRS templates together may help create a more universal template

## IBI Filtering

---



Filter IBI

<input type="text" value="20"/>	<input type="checkbox"/> percent
<input type="text" value="2"/>	<input checked="" type="checkbox"/> sd
<input type="text" value="4"/> <input type="text" value="5"/>	<input type="checkbox"/> median
<input type="text" value="0.24"/>	<input type="checkbox"/> above thresh
<input type="text" value="0.125"/>	<input type="checkbox"/> below thresh
<input type="text" value="3"/> <input type="text" value="160"/>	<input type="checkbox"/> custom

IBI filtering is used to mark IBI “outliers”. If an outlier is detected based on the user selected criteria, the beat that initiated the IBI and the IBI itself is marked as an outlier on the plots. Outliers are marked by a filled red circle. Although the term filter is used in this section, no change to the original IBI occurs during ectopic interval detection.

Filter Types:

1. percentage filter – marks intervals that change by more than a user defined percentage (often 20%) from the previous interval <sup>[55]</sup>. This method locates any sudden or abrupt IBI changes.
2. standard deviation filter – marks outliers as being intervals that lie beyond the mean IBI by a user defined value of standard deviations (often 3 SD) <sup>[55]</sup>.
3. median filter – acts as an impulse rejection filter with threshold to delineate ectopic intervals <sup>[50]</sup>. The median filter of a random variable  $x$  of length  $N$  using a threshold of  $\tau$  is given by

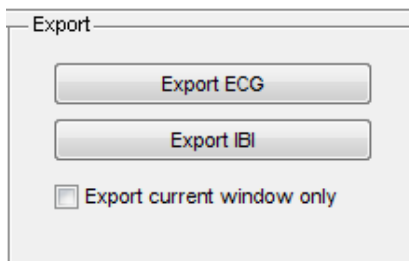
$$D(n) = \frac{|x(n) - med(x)|}{1.483 \cdot med\{|x(n) - med(x)|\}}$$

if  $D(n) \geq \tau$ , then not ectopic; else ectopic

4. above threshold filter – marks any interval lying above the user defined threshold limit as outliers
5. below threshold filter– marks any interval lying below the user defined threshold limit as outliers
6. custom filter – this was a personal filter used to ignore IBI near large spikes in the ECG signal. The first box defines the amplitude limit, and the second box defines the window width. Any IBI within the window width around ECG that has an amplitude above the limit is ignored.

## Exporting

---



Two types of data can be exported using the export button located on the GUI within the export module.

1. Export ECG – this button exports the currently loaded ECG file. File formats available for ECG exporting include MATLAB binary (.mat), 16-bit binary (.dat), text/ASCII (.txt), and Excel (.xls).
2. Export IBI – this button exports the IBI of the currently loaded ECG file. File formats available for IBI exporting include MATLAB binary (.mat), text/ASCII (.txt), and Excel (.xls).

The option labeled “Export current window only” tells ECG Viewer to only export the ECG or IBI currently displayed in the plots. In other words whatever you see on the screen is what gets exported.

## **File Annotations and Database**

---

- ECG Viewer allows users to annotate files and mark files as being “completed”.
- These task can only be completed on the currently loaded ECG file.
- Annotation can be created at any time point on the ECG file currently displayed by double clicking on the desired point on the ECG plot.
- Annotations can be reviewed by double clicking on an annotation label within the annotation list. This list is located below the list of ECG files.
- ECG files can be marked as being “complete” by loading the desired file and right clicking anywhere on the ECG plot.
- To remove annotations you must load the Microsoft Access file and do it manually.

Before using the database feature the first time you must create a datasourse in your Windows environment. See

<http://MATLAB.izmiran.ru/help/toolbox/database/install12.html#18933>. This is only done once.