Biomechanics Analysis and Reporting

Application

User Manual

Contributors:

Ben Senderling, Boston University

Version 1.0

Contents

[Intent 3](#_Toc121921873)

[Scope 3](#_Toc121921874)

[Overview 3](#_Toc121921875)

[Future Work 4](#_Toc121921876)

[Main Application 4](#_Toc121921877)

[Database Search 4](#_Toc121921878)

[Working Database 5](#_Toc121921879)

[Processing 5](#_Toc121921880)

[Analysis 5](#_Toc121921881)

[Review 5](#_Toc121921882)

[Statistics 5](#_Toc121921883)

[Export 5](#_Toc121921884)

[Configuration 6](#_Toc121921885)

[Analysis Modules 6](#_Toc121921886)

[Custom 6](#_Toc121921887)

[QST 6](#_Toc121921888)

[Assumptions 6](#_Toc121921889)

[Results 6](#_Toc121921890)

[False Nearest Neighbor 7](#_Toc121921891)

[Recurrence Quantification Analysis 7](#_Toc121921892)

[Time Lag 7](#_Toc121921893)

[Loading Data into BAR 7](#_Toc121921894)

[Overview 7](#_Toc121921895)

[Process Modules 7](#_Toc121921896)

[Groupings 7](#_Toc121921897)

[Merge 7](#_Toc121921898)

[Segment 7](#_Toc121921899)

[Treatment 7](#_Toc121921900)

[Review Modules 7](#_Toc121921901)

[Raw 7](#_Toc121921902)

[False Nearest Neighbor 7](#_Toc121921903)

[Recurrence Quantification Analysis 7](#_Toc121921904)

[Time Lag 8](#_Toc121921905)

[Quality Review 8](#_Toc121921906)

[Appendix A Data Types 8](#_Toc121921907)

# Intent

The creation of this application hopes to establish a generalized framework by which scientists in the biomechanics community conduct their data processing and analysis. The intent is that this will facilitate data analysis and reporting by making the process easier for students with varied backgrounds, and by enhancing the repeatability of analytical procedures. The framework provides a base-level application that others can use as-is or build upon with their own sources and methods.

# Scope

The base-level of the application includes 1) loading of data, 2) processing and analysis, 3) graphical and quality review and 4) export of the data. Its design was meant to be open to varied methods used in the study of human movement. Ultimately, these methods are not all that dissimilar to methods in other fields. It is likely those outside the realm of biomechanics would also find this application useful.

Some aspects of data processing are outside the scope of the application. These may require dedicated hardware or complex software solutions, involve high performance computing or complex modeling and simulation.

# Overview

The app has a main application called BAR\_App. This will handle all the configuration settings and dynamic use of modules. Modules are considered any mlapp- or m-file that is dynamically used by the app. Some modules are considered central to the application. These include 1) process\_Groupings, 2) process\_Merge and 3) review\_RAW. Modules fit into categories named 1) Analysis, 2) Load, 3) Process and 4) Review. These are also the major folders used by the app. Another folder called Subroutines is meant to house low level functions that may have widespread use across other modules.

Analysis modules are similar in function to Process modules but differ in scope and intent. Analysis modules are for complex analyses that are more likely to be other mlapp-files. These modules allow users to perform complex operations with significant user input. They often will take time series or aggregate data and produce single metrics or statistics.

Load modules are mostly functions that read data from a file, package it into a BAR App data structure and load it into the app. They are organized using the file and equipment type. The file type must match the file type of the target file however the Equipment type can vary. An example could be load\_h5\_APDM1Lumbar.m and load\_h5\_APDM1Raw.m. Both read data from the same h5 file but the first reads only the lumbar data while the later reads only the raw data.

Process modules are more general purpose analysis methods. These may include segmentation or other treatment like filtering. They are not considered the end of a processing step and would produce data that is later used in an Analysis module. Process modules are more general in that they may proceed multiple Analysis Modules, whereas the later is considered an end point.

Review modules are meant for data visualization. These modules should be mostly mlapp-files that can dynamically produce general or specific figures of the data. For most users the review\_RAW.mlapp module it hoped to be sufficient. For most complex analyses there will be a corresponding Review Module that produces specific figures describing the analysis results.

# Future Work

As of December 12, 2022, the app will only function as hoped within the MATLAB environment. It was hoped that the code could be compiled into an executable that could dynamically use different functions and scripts. This has turned out to be impossible as those scripts need to be compiled at runtime, or a MATLAB server is used. Instead of working on this further the app has been developed as a MATLAB App that can be shared with other MATLAB users.

# Main Application

This section will describe the BAR\_App.mlapp file that is central to the BAR App. It controls the 1) locating and 2) loading of data, 3) processing, 4) analysis, 5) review and 6) export. These functions are mostly accomplished by the different tabs within the app.

## Database Search

Database Search is where all of the file handling will be performed. This starts by setting the Working Database.1 The button will allow a user to select a folder. The app will then check that certain folders exist within it. These include folders named Export, Figures and Results. Export is used as the default target folder when exporting data from the main application. Figures is used by the review\_RAW module and by supported modules as a default folder to save figures to. Results is used to save intermediate data and results. Directories can also be typed into the text field. This will not create subfolders but is useful when quickly changing between Working Directories.

Before files can be loaded the app must know where to look for them. Directories can be entered into the search field2, or added through the button3. The app performs a recursive search algorithm that looks through all listed folders and subfolders for the target files. This is generally quick but will be significantly slower on network drives.

Graphical user interface, text, application, email

Description automatically generated

### Working Database

## Processing

## Analysis

## Review

## Statistics

## Export

## Configuration

# Analysis Modules

The follow section describes the different modules available in the current release. Each section starts with 1) a brief description of the module, 2) assumptions regarding the origin of the data, and 3) a description of the results and what their intended use was.

## Custom

## MATLAB Dependencies

This analysis will determine what MATLAB products are needed from a selection of m- and mlapp-files. This will mostly be useful to programmers who which to communicate to others what MATLAB products are needed to run their code.

### Assumptions

To run this code there may be some products that are required. These would be products needed to run the BAR\_App.mlapp file. Currently only MATLAB 9.11 is required to run the code. The App Designer would be needed to modify the code.

### Results

The script will produce a list of all the MATLAB products needed to run all the analyzed files. This will look like this.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Version | ProductNumber | Certain |
| MATLAB | 9.11 | 1 | TRUE |
| Signal Processing Toolbox | 8.7 | 8 | TRUE |
| Statistics and Machine Learning Toolbox | 12.2 | 19 | TRUE |

## Quantitative Sensory Testing

The Quantitative Sensory Testing (QST) module is a single script that processes specific experimental data recorded at the BU MoveLab. The method itself does not require any user input and allows the results to be collected into a single file.

### Assumptions

Technically this script could run on any data loaded into the BAR App but it is expecting xlsx-Medoc data types. This data was recorded from Medoc software and exported as an xlsx file. The procedure was a pressure-pain sensitivity test. In this test an algometer is used to apply pressure to an anatomical landmark. As that pressure is increased at a set rate the subject presses a trigger to indicate the first sensation of slight pain. The xlsx-file will contain information for the pressure, sampling times and the button press. It will also contain information on the sequence and trial numbers, and meta data related to the test.

### Results

The script will produce the following results.

|  |  |  |
| --- | --- | --- |
| Variable | Intent | Description |
| sequenceN | Experiment descriptive | This is the number of the Program Sequence in Medoc that the data comes from. This allows a user to identify a result and trace which trial it came from. |
| trialN | Experiment descriptive | This is the number of the Program Trial in Medoc that the data comes from. Each Program has a Sequence and within each Sequence are Trials. This allows a user to identify a result and trace which trial it came from. |
| valuePeak | Results metric | This is the absolute peak of the recorded pressure. |
| valueEvent | Results metric | This is the instantaneous pressure at the time of the button press. |
| rSquaredAdjusted | Quality metric | This is an Adjusted R2 that describes how well the pressure data fits a linear line. The application of pressure it meant to increase at a set linear rate. Deviations from a linear rate will result in a lower value and indicate lower quality methods. |
| slope | Quality metric | This is the slope of a linear line fit to the data. The pressure should be applied at a set rate. This slope is that rate. |

## False Nearest Neighbor

## Recurrence Quantification Analysis

## Time Lag

# Loading Data into BAR

### Overview

# Process Modules

### Groupings

### Merge

### Segment

### Treatment

# Review Modules

### Raw

### False Nearest Neighbor

### Recurrence Quantification Analysis

### Time Lag

# Quality Review

# Appendix A Data Types

|  |  |  |
| --- | --- | --- |
| File Type | Equipment Type | Description |
| agd | Actigraph | Actigraph uses agd-files in much of its processing in ActiLife. However, they are not the csv exports with the results. These files are not read by the BAR App but can be copied and moved. |
| csv | Delsys | These are comma separated values exported from Delsys using the Delsys File Utility. When exporting data it is required the option to include headers is checked. Delsys has a number of sensor and file types so this code may not work for all csv-Delsys files. |
| csv | Actigraph | These are the csv exports from Actigraph ActiLife. It does include the spreadsheet exports called: DailyDetailed, DailyTotals, HourlyDetailed, HourlyTotals, SedentaryAnalysis, SleepScores and WearTimeValidation. It does not include the Variables spreadsheet. |
| m | MATLAB | These are the scripts used in MATLAB. |
| mlapp | MATLAB | This is the file type of the MATLAB Applications used in App Designer. |
| xlsx | Medoc | These are Excel files exported from Medoc software. There are very few options or alternate versions but these may vary with the testing equipment and Program configuration. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |