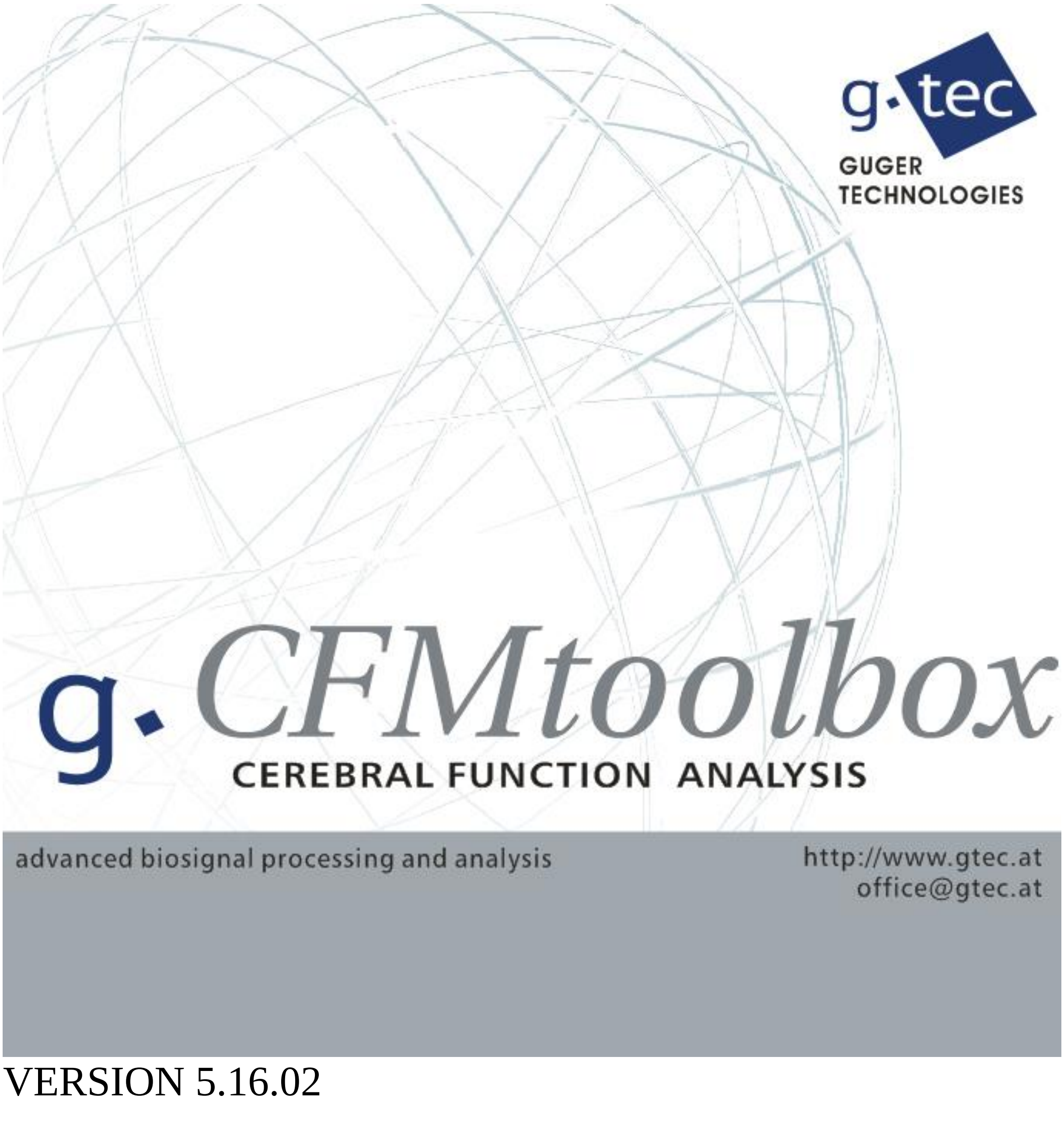


VERSION 5.16.02

USER MANUAL

CONTENT:

To the Reader

Welcome to g.tec’s world of medical and electrical engineering!
Discover the only professional biomedical signal processing platform under MATLAB and Simulink. Your ingenuity finds the appropriate tools in the g.tec elements and systems. Choose and combine flexibly the elements for biosignal amplification, signal processing and stimulation to perform even real-time feedback.

Our team is prepared to find the better solution for your needs.

Take advantage of our experience!

Dr. Christoph Guger   Dr. Guenter Edlinger

Researcher and Developer

Reduce development time for sophisticated real-time applications from month to hours.
Integrate g.tec's open platform seamlessly into your processing system.
g.tec's rapid prototyping environment encourages your creativity.

Scientist

Open new research fields with amazing feedback experiments.
Process your EEG/ECG/EMG/EOG data with g.tec's biosignal analyzing tools.
Concentrate on your core problems when relying on g.tec's new software features like ICA, AAR or online Hjorth's source derivation.

Study design and data analysis

You are planning an experimental study in the field of brain or life sciences? We can offer consultation in experimental planning, hardware and software selection and can even do the measurements for you. If you have already collected EEG/ECG/EMG/EOG, g.tec can analyze the data starting from artifact control, do feature extraction and prepare the results ready for publication.

Preface

This section includes the following topics:

Required Products

Using This Guide - Suggestions for reading the handbook

Conventions - Text formats in the handbook

Required Products

g®.CFMtoolbox uses:

g®.BSanalyze – the advanced biosignal analysis software package from g.tec

MATLAB – as basic matrix operation platform

Signal Processing Toolbox - to give access to standard signal analysis tools

Using This Guide

Chapter "Running g.BSanalyze" shows how to start the Data Editor.

Chapter "Calculating CFM" shows how to calculate the cerebral function monitor signal from EEG data.

Chapter "Classifying CFM" explains the classification of the CFM signals.

Chapter "Help" explains the usage of the on-line help, the printable documentation and the function help.

Chapter "Batch-Mode" shows how to use the g.BSanalyze commands from the MATLAB command line.

Conventions

Item	**Format**	**Example**
MATLAB code	Courier	to start simulink, type `simulink`
String variables	*Courier italics*	`set(P_C,'`*`PropertyName`*`',...)`
Menu items	**Boldface**	Select **Save** from the **File** menu.

Hardware and Software Requirements

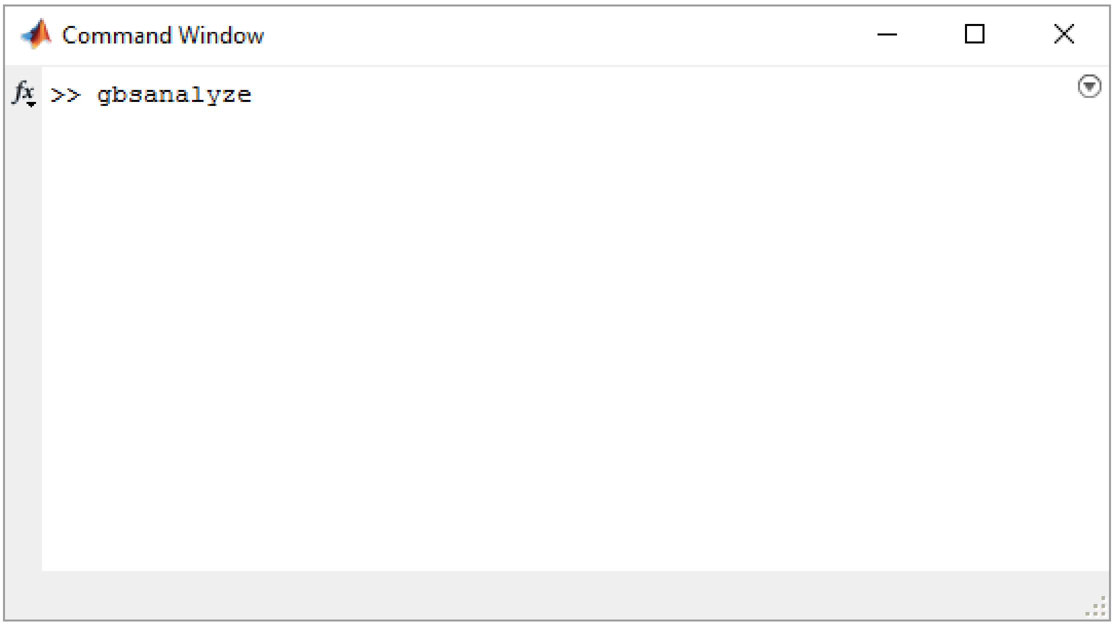
For Hardware and Software Requirements see the g.BSanalyze manual.

Running g.BSanalyze

After starting MATLAB and setting the correct path, type:

```
gbsanalyze
```

in the MATLAB command line



g.BSanalyze starts with a blank data window.

Calculating CFM

The first step is to load EEG data into g.BSanalyze. Then the CFM method is used to calculate the cerebral function monitor (CFM) signal from the EEG data.

The **CFM** window has the following control elements:

Specify data interval:

Start interval at - enter the starting time point for the analysis
End at - enter the end time point of the analysis

Select channels - allows specifying the EEG channel that should be used for the analysis

Specify the parameters:

Method - can be `Traditional` or `HOS`. The CFM algorithm `Traditional` asymmetrically bandpass filters the data. Then the root-mean square function is calculated. This results in the CFM estimation of the EEG data. The `HOS` algorithm uses higher order statistics for the CFM estimation.

FilterBand – specify the **Lowest** and **Highest frequency** for the bandpass filtering (if the `Traditional` method is used)

Classify CFM Pattern – check the box to automatically classify the CFM patterns

Percentile – specify the percentile in percentage to calculate the lower and upper limit of the CFM signal

Epoch – define the epoch length for the automatic classification

Result procedure:

Show with gCFMViewer – check to visualize the CFM signal

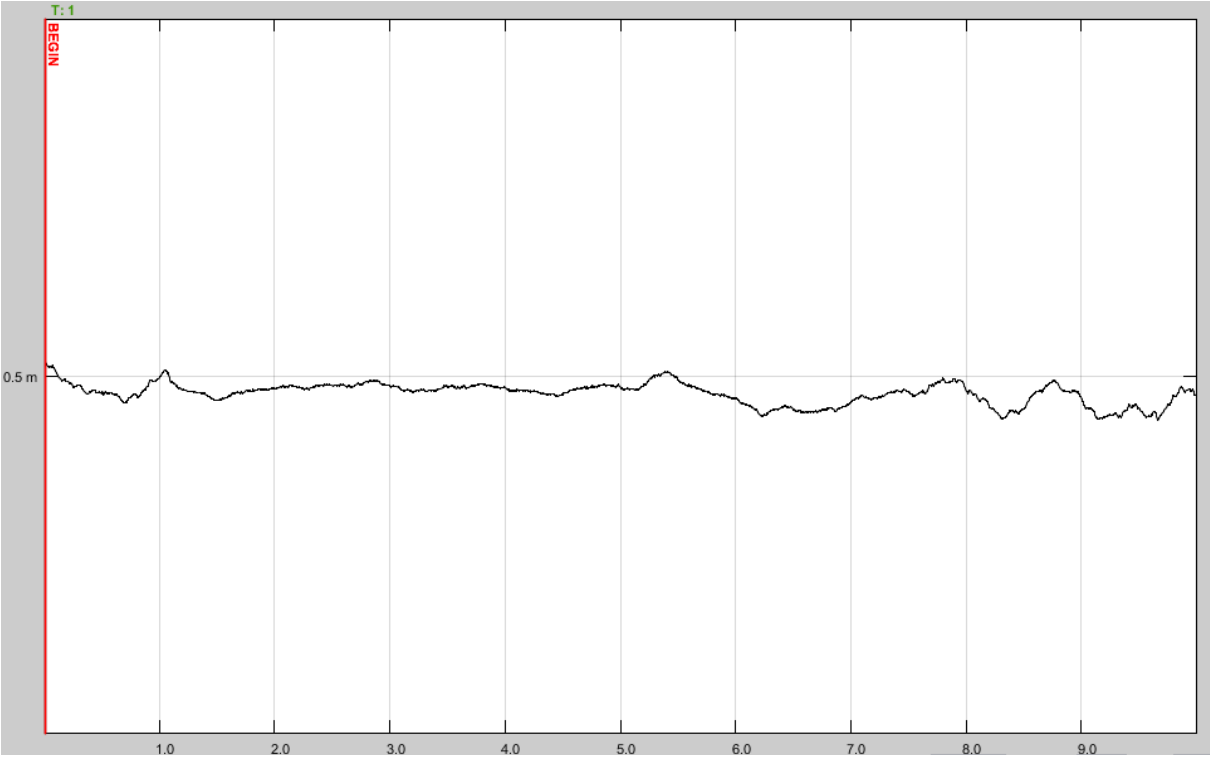
Save result – check to store the calculated CFM signal

Perform the following steps:

1. Click on **Load Data** under the **File** menu of g.BSanalyze and select the file `cfm1.mat` from

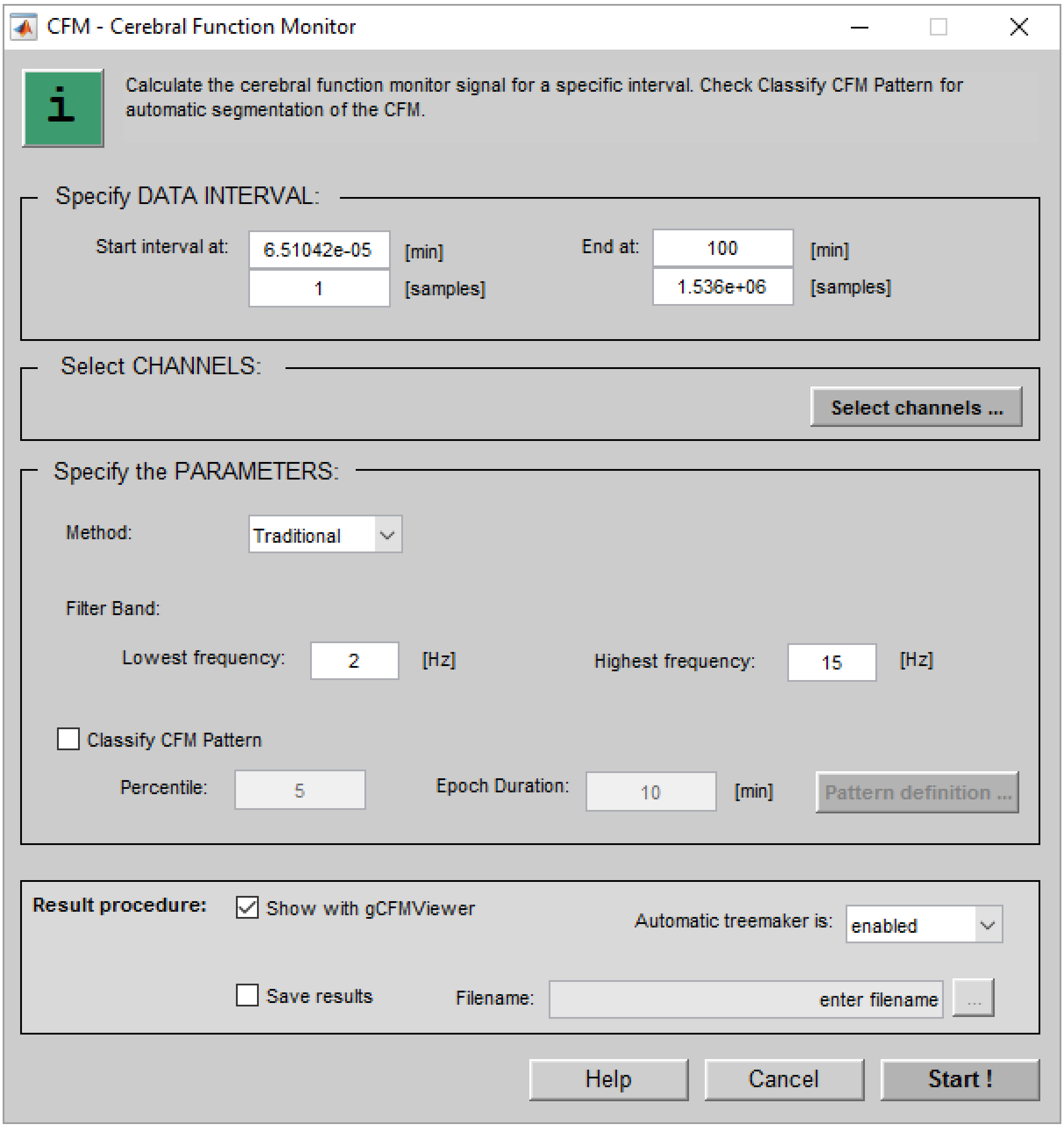
   `Documents\gtec\gBSanalyze\testdata\CFM`

2. Select **Open** to load the EEG data file of a baby. The EEG data is a bipolar recording between C3 and C4 according to the international 10/20 system. The ground electrode was fixed on the forehead. The Data Editor shows the first 10 seconds of the recording.



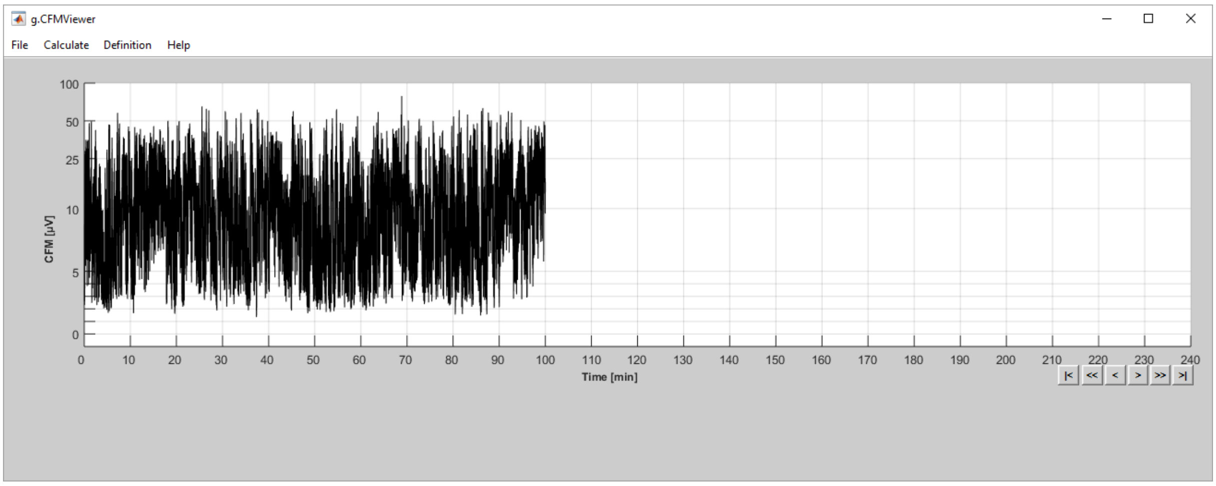
The data is shown with a sensitivity of ±0.5mV.

3. Open the **CFM** window from the **Analyze** menu



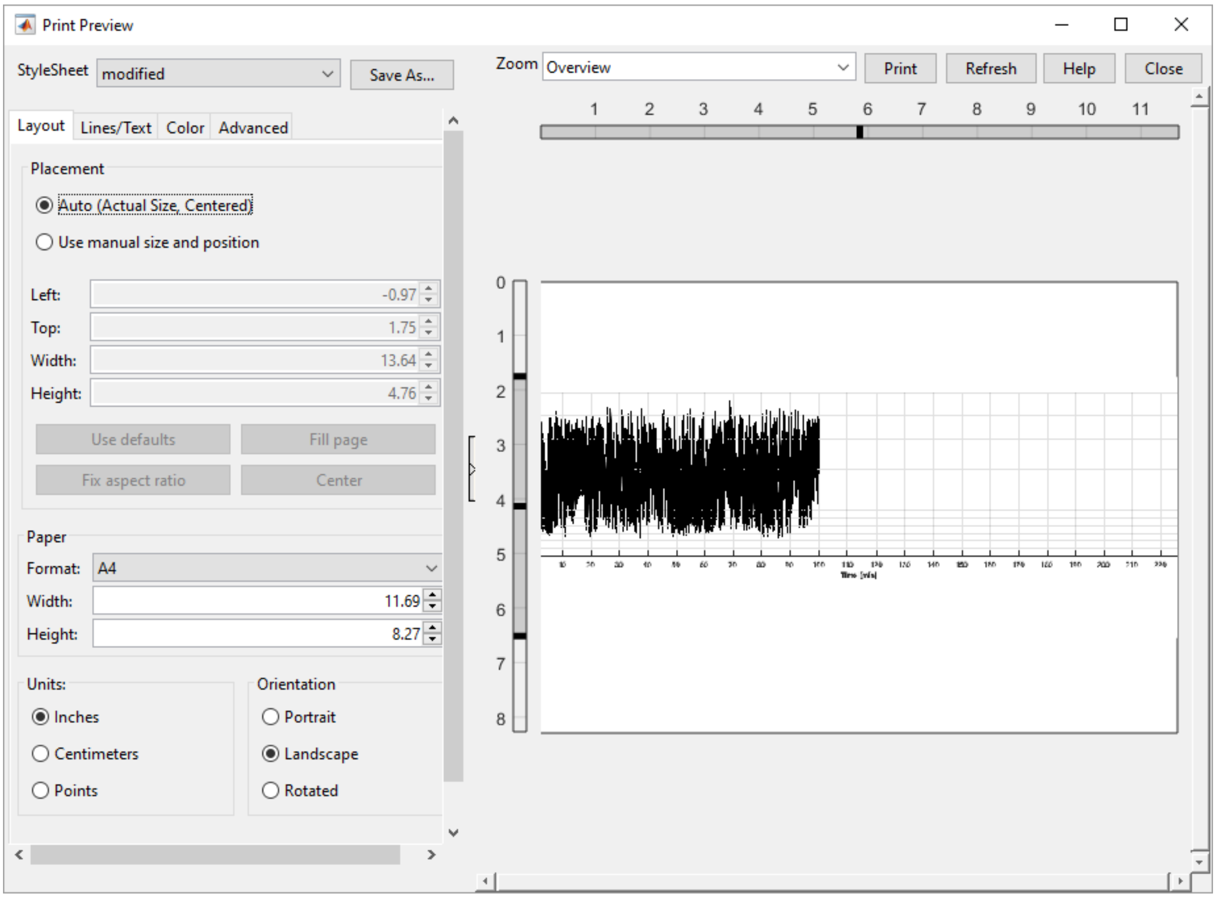
4. To calculate the CFM signal from all `100` minutes with the `Traditional` method and to perform a bandpass filtering between `2` and `15` Hz press **Start**

5. The g.CFMViewer opens with the CFM signal of the EEG channel



The y-axis shows the CFM signal in µV and the x-axis the time in minutes. The signal is mainly between 2 µV and 50 µV. The scroll buttons can be used to navigate through longer data-sets.

6. Click on **Print Preview** in the g.CFMViewer menu to open the following window and click on the **Print** button for a hardcopy.

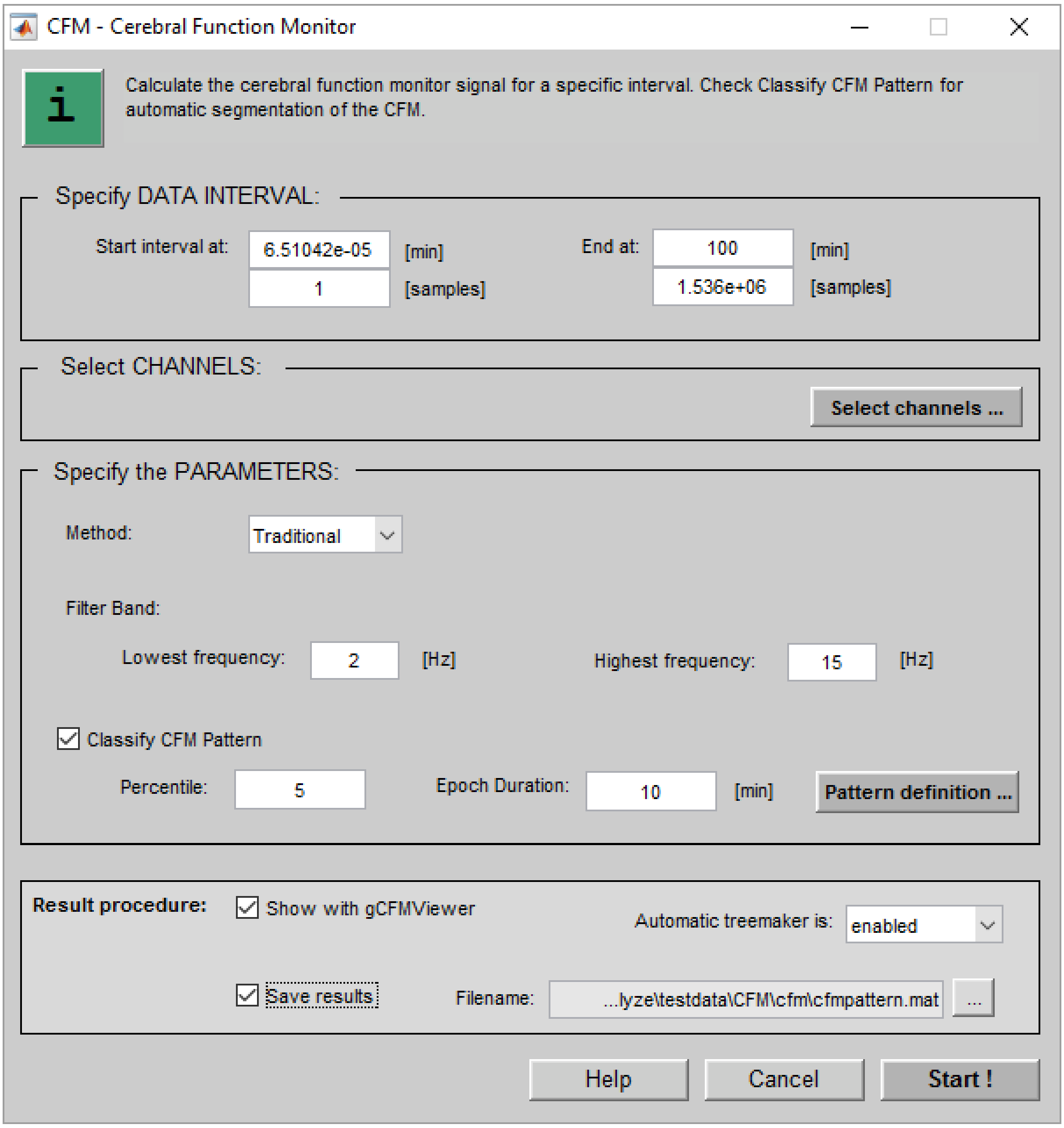


Classifying CFM

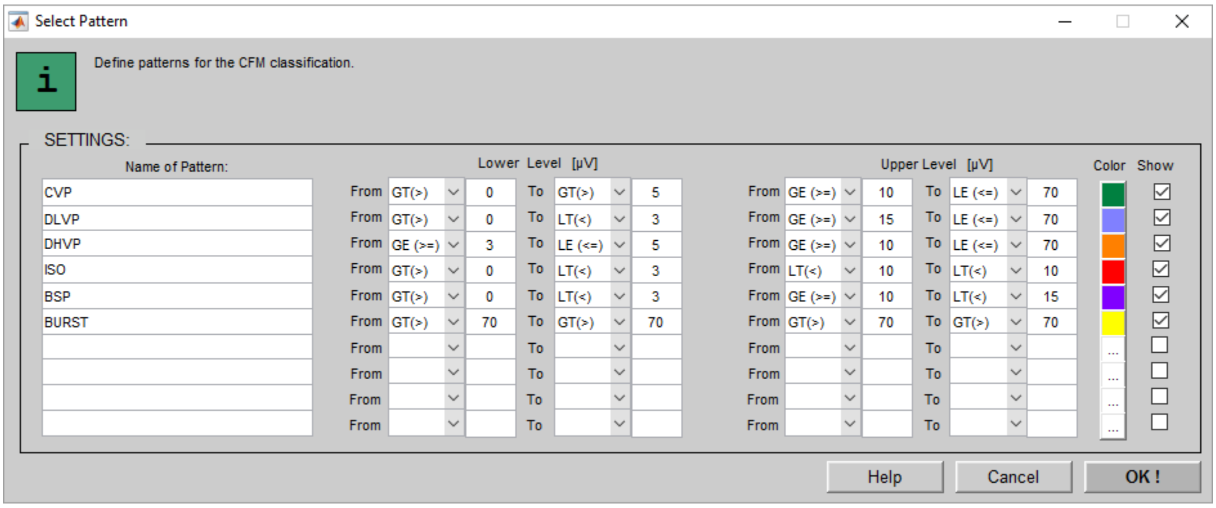
1. Load the data-set into g.BSanalyze as described in the previous section and open the **CFM** window

2. Check **Classify CFM Pattern** to enable the corresponding editor boxes

3. **Epoch Duration** is set to 10 minutes. This divides the CFM signal into segments of 10 minutes and of each segment the upper and the lower percentile is calculated. **Percentile** is set to 5 %. Therefore the upper 5 % percentile and the lower 5 % percentile of each 10 minutes segment are calculated.



4. Open the **Pattern definition …** window to edit the borders for the automatic classification

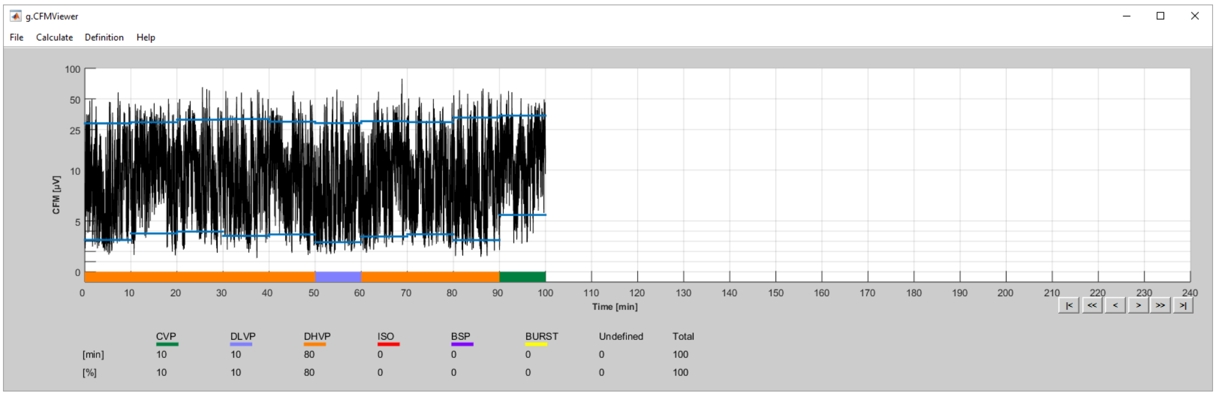


The window allows to specify 10 patterns with a lower and upper level. Enter a name for each pattern and the threshold values of the lower and upper percentiles. Then select a color for the pattern and check the box if this pattern should be used for the analysis. 6 standard patterns are already pre-defined. Press **OK !** to confirm the pattern settings.

5. Check the **Save results** box and enter the filename `cfmpattern.mat` to store the calculated CFM data to harddisk

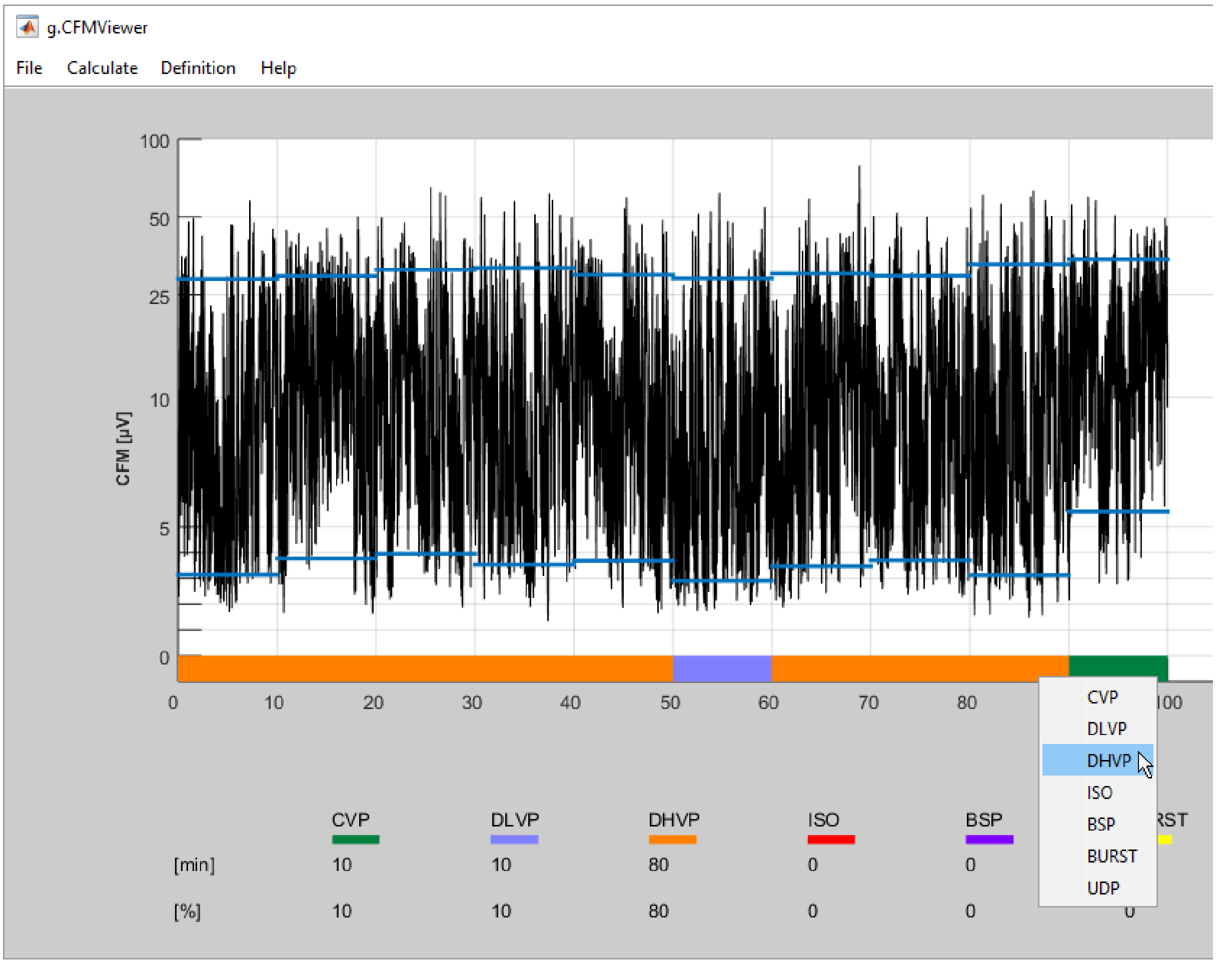
6. Press **Start** to perform the calculation

7. g.CFMViewer opens automatically with the results

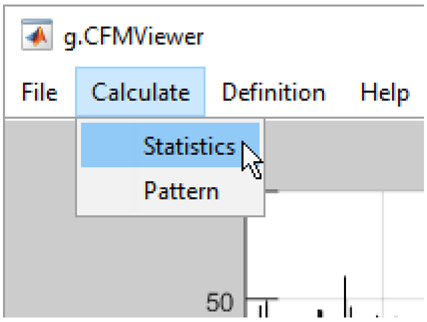


8. Now additionally to the CFM signal the lower and upper percentiles are shown with blue lines for each 10 min segment. These percentiles were used for the automatic classification of the CFM data. The results are shown with the coloured boxes under the CFM signal. In this case the first 50 minutes correspond to `DHVP`, the next segment to `DLVP`,...
   The CFM statistic is presented under the plot. Ten minutes correspond to the class `CVP`, 10 minutes to `DLVP` and 80 minutes to `DHVP`. No CFM traces are available which correspond to `ISO`, `BSP` or `BURST`.
   If no pattern can be assigned according to the pattern definition an `Undefined` is assigned. The last line shows the distribution in percentage.

9. To manually correct the automatic classification right click on the patterns and assign a different pattern name. The color of the segment changes.



10. After performing the corrections of all segments select **Calculate Statistics** to update the pattern distribution

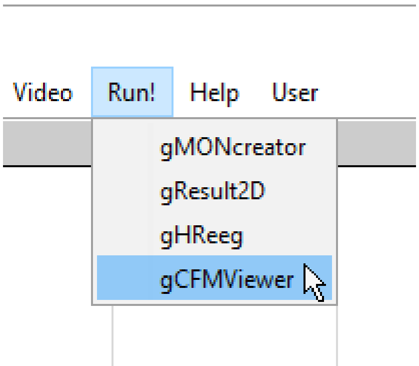


11. Select **Pattern** under the **Definition** menu to change the pattern definition and then start again the classification with **Pattern** from menu **Calculate**

gCFMViewer

The g.CFMViewer is used to view the CFM data of one channel.

To start the g.CFMViewer go to the Data Editor and select `gCFMViewer` under the **Run** menu



or type

```
gCFMViewer
```

into the MATLAB command window.

To view already stored CFM data use the **Load Data** function from the **File** menu.

Perform the following steps to load the data from the MATLAB command line:

Type

```
load cfmpattern.mat
```

into the command line.

This loads the CFM object `CFM_S` into the MATLAB workspace.

Use

```
gCFMViewer(CFM_S)
```

to visualize the data with g.CFMViewer.

To extract the CFM data from the object use

```
y=CFM_S.Out;
plot(y(:,1));
```

Help

g.BSanalyze and the g.CFMtoolbox provide a printable documentation and a function help.

The printable documentation is stored under

```
C:\Program Files\gtec\gBSanalyze\Help
```

as `gCFMtoolbox.pdf`. Use Acrobat Reader to view the documentation.

To view the function help type

```
help gBSfunctionname
```

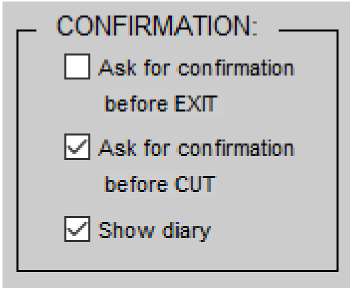
under the MATLAB command window.

To view all functions that are available in batch mode type

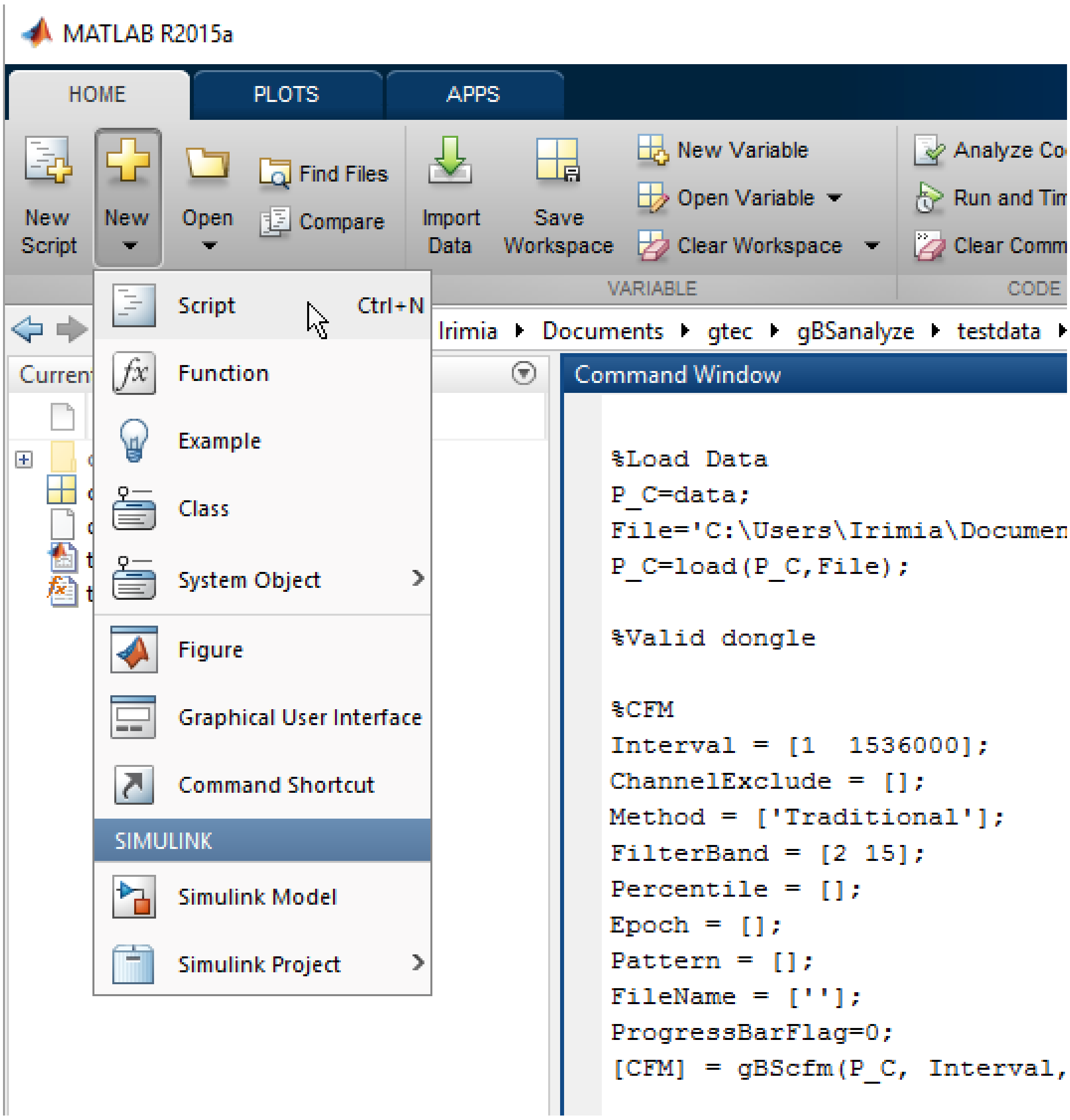
```
gBSfunctions
```

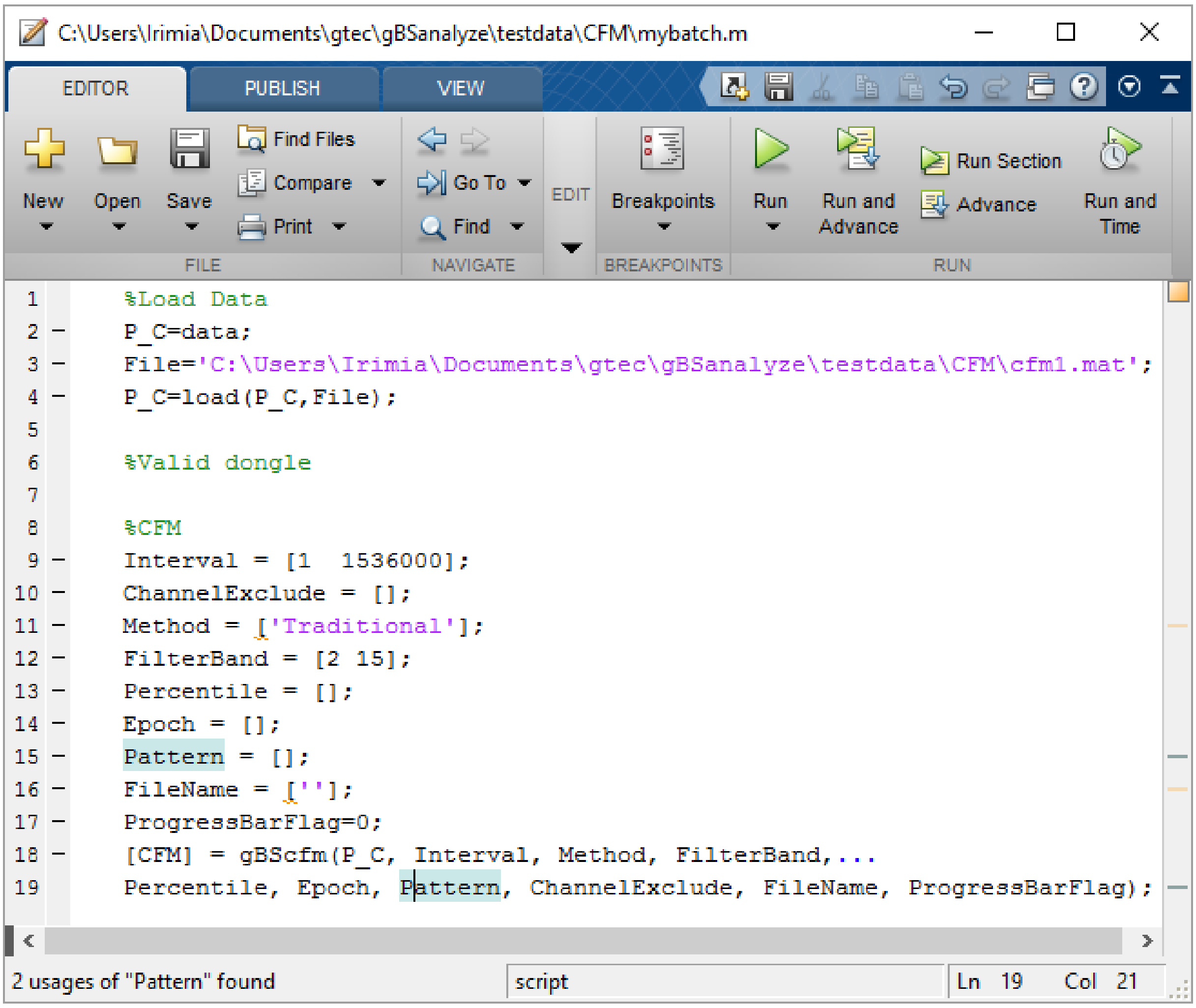
Batch Mode

The easiest way to create a batch for data processing is to perform the analysis under the Data Editor using the graphical user interfaces. Make sure that the **Show diary** checkbox is enabled in **Appearance Settings** under the **Options** menu.



This forces g.BSanalyze to report all calculations in the MATLAB command window. After finishing the analysis open a **New Script** and copy and paste all commands into the file.





Save the batch in your own directory as `mybatch.m` and start the batch under the MATLAB command window with

```
mybatch
```

In order to investigate further data-sets just replace the input data file by the new data file to perform the same analysis.

Product Page

Please visit our homepage www.gtec.at for

- Update announcements
- Downloads
- Troubleshooting
- Additional demonstrations

GUGER TECHNOLOGIES

contact information

g.tec medical engineering GmbH
Sierningstrasse 14
4521 Schiedlberg
Austria

tel. +43 7251 22240
fax. +43 7251 22240 39
web: www.gtec.at
e-mail: office@gtec.at