ICNNA/MENA User Guide

Imperial College London

Felipe Orihuela-Espina

ABSTRACT

Imperial College Near Infrared Spectroscopy Neuroimaging Analysis (ICNNA), formerly known as Imperial College Neuroimage Analysis (ICNA), is a software framework encompassing an Application Programming Interface (API) and a Graphical User Interface (GUI) for Manifold Embedding NIRS Analysis (MENA) under the MATLAB environment.

Imperial College Near Infrared Spectroscopy Neuroimaging Analysis (ICNNA)
ICNNA – Imperial College Neuroimage Analysis
MENA – Manifold Embedding Neuroimage Analysis

User's guide
Release: 1.1.4
Latest revision: 30-Oct-2019

Copyright © 2008-19 Felipe Orihuela-Espina
Author: Felipe Orihuela Espina
Institution: Imperial College London

Contributions:

Research: Dr. Felipe Orihuela Espina, Dr. Daniel R. Leff, Mr. Dave James, Prof. Sir Ara W. Darzi and Prof. Guang-Zhong Yang.

Beta testing: Dr. Felipe Orihuela Espina, Mr. Marco Caproni, Mr. Dave James, Mr. Kunal Shetty, Dr. Samuel A. Montero-Hernández, Dr. Harsimrat Singh and Mrs. Michele Rojas Cisneros.

External sources and algorithms:

MATLAB® and the Statitics Toolbox® is owned by Mathworks

In some cases, ICNNA uses code not only from MATLAB original software package but also from other sources but distributed under free licences. They are however copyright protected, and their right belong to their respective authors and proprietors.

fNIRS data conversion, filtering and detrend algorithms were originally inspired by fOSA package (University College London).

Function cca ⇒ ©SOM Toolbox
Function kldiv ⇒ ©David Fass
Function kmeans, dist2 ⇒ ©netlab, Ian T. Nabney (1996-2001)
Isomap algorithm ⇒ ©Joshua Tenebaum

Every effort is made to recognise other people copyright and intellectual property. If you think I'm missing someone/something please let me know, and I will make the appropriate corrections.

End User Licence Agreement

This software is distributed as copyright freeware under the terms of the GNU General Public Licence. The author reserves the right to change the terms of this licence at any time without prior notification.

ICNNA/MENA is intended for research and academic purposes only. This software may not be used for clinical purposes. This software is not intended for commercial purposes. Profit organizations and companies are allowed to use this software but may not charge anyone for installation or activities that uses ICNNA/MENA.

Disclaimer:

We cannot guarantee that ICNNA/MENA is free of errors. ICNNA/MENA may contain errors that could cause failures or loss of data and may be incomplete or contain inaccuracies. ICNNA/MENA is provided “AS IS” without any warranty, upgrades or support. ICNNA/MENA is provided with no support whatsoever. The authors may at their sole discretion, provide bug reports or upgrades. Imperial College London does not have any obligation to notify users of this occurring. The use of ICNNA/MENA is therefore at the user sole and entire risk. Under no circumstances, shall Imperial College London, or the authors, be liable for any incidental direct or indirect damages arising out of the use of this software.

If you use ICNNA/MENA please cite the following papers:

Orihuela-Espina, Felipe; Leff, Daniel R.; James, David R. C.; Yang, Guang-Zhong; Darzi, Ara W. (2018) *Imperial College Near Infrared Spectroscopy Neuroimaging Analysis (ICNNA) Framework*, Neurophotonics, 5(1):011011

Orihuela-Espina, Felipe; Leff, Daniel R.; James, David R. C.; Darzi, Ara W.; and Yang, Guang-Zhong. (2009) *ICNA: a software tool for manifold embedded based analysis of functional near infrared spectroscopy* In Annual Meeting of the Organization of Human Brain Mapping (OHBM), NeuroImage 47(Suppl1):S59.

Version Control

Document Version Number	Purpose/Change	Author	Date
0.8	Started this document version control. Updated official name from ICNA to ICNNA in this document. Added Neurophotonics paper. Updated to release 1.1.3. Some minor typos updated. Main menu and About dialog figure updated.	FOE	25-Apr-2018
0.8	Added Samuel A. Montero and Kunal Shetty among beta testers recognition.	FOE	26-Apr-2019
0.9	Added Harsimrat Singh and Michele Rojas among beta testers recognition. ICNNA startup script updated.	FOE	30-Oct-2019

Table of Contents

What is ICNNA/MENA?

MENA (Manifold Embedding NIRS Analysis) refers to the analysis of continuous wave functional near infrared spectroscopy data (CW-fNIRS) by means of manifold embedding and dimensionality reduction techniques, together with cluster analysis and distribution similarity measures.

Imperial College Near Infrared Spectroscopy Neuroimaging Analysis (ICNNA) is a software framework encompassing an API (Application Programming Interface) and a GUI (Graphical User Interface) for MENA under the MATLAB® environment. ICNNA make extensive use of the object oriented programming paradigm whilst still relying on MATLAB's matrix manipulation for algebraic computation.

How can MENA help you

Possibly, the biggest strength of MENA is its capability for group analysis. MENA is a powerful analysis tool that goes beyond the signal processing of a single subject over a single session to allow fast exploration and quantification of the general behaviour of groups of subjects, channels, sessions, etc… It incorporates some of the most advanced *manifold embedding* techniques such as Isomap or Curvilinear Distance Analysis, to yield low dimensional representations of the highly complex functional manifold. Furthermore, it offers exploration tools for quickly understanding the distribution of brain haemodynamic patterns within the projection space and their dynamic evolution. Finally, MENA can quantify differences between classes or groups, which in MENA are referred to as *clusters*.

What ICNNA/MENA is not

ICNNA can convert raw light NIRS measurements into changes in haemoglobin concentrations using the modified Beer Lambert law. It can even do some basic signal processing such as decimation and detrending as well as averaging. However, ICNNA main goal is not the signal processing but image understanding. Hence its processing capabilities are reduced to a minimum. For functional NIRS signal processing a few other good packages already exists such as fOSA from Univeristy College London, or HomER from the Photon Migration Lab at the Massachusets General Hospital in Boston. However, in acknowledging this limitation, ICNNA provides easy importing of fOSA data files.

MENA can quantitatively express differences among clusters of haemodynamic patterns through the use of distribution metrics such as Earth Mover's Distance (EMD). MENA can express clusters descriptors such as centroids and average distance to centroid which can be understood as a measure of variance. Additionally, ICNNA can generate tabular database in *.csv* format for further statistical analysis in common statistical packages. Derived from this database,

ICNNA can generate an activity matrix, that is a matrix where rows represents sessions and stimulus and columns represents NIRS channels. Each element of this matrix is assigned a value from 1 to 16 corresponding to 16 different patterns of signal behaviour, indicating which channels are found to be statistically significant active.

Installation Guide

Uncompressed the zipped file in the folder wherever you are planning to use ICNNA/MENA.

Current ICNNA version is v1.1.4. (Released 30-Oct-2019)

Requisites

For current version, ICNNA/MENA requires MATLAB® Release 2010a or after and the Statistics Toolbox 7.0 or after. ICNNA/MENA currently has been tested on Windows Vista, Windows 10 and Mac OS X Leopard. Compatibility with other operating systems is currently not supported.

Welcome to ICNNA

When you have already downloaded and uncompressed ICNNA into a certain folder, you should be able to see the following folders and files:

- **documentation**: ICNNA documentation including this User Guide.
- **GUI**: Contains the functions for the GUI of ICNNA.
- **miscellaneous**: Several functions which does not fit anywhere else.
- **oo**: All the objects of the ICNNA API.
- **plotting**: Several functions related with displaying of information.
- **util**: Include some signal processing functions, histogram manipulation, EMD basic functions, etc. Basic functionality that the basic MATLAB package and/or the Statistics Toolbox do not provide. If the user has access to MATLAB's Signal Processing toolbox ® it is recommended to delete the appropriate functions from this folder, as the implementation in ICNNA may not be optimized. Functions in this folder are intended among other things to reduce the dependence of ICNNA of extra packages or toolboxes.
- **icnna_startup.m**: A script file for initiating the path.

Although the use of ICNNA GUI does not require any knowledge of programming, this User Guide assumes the user is familiar the MATLAB environment. Please refer to MATLAB documentation if you need some particular information about MATLAB.

Opening ICNNA

1. Open MATLAB
2. Select ICNNA's folder as working directory
3. Execute the function `icnna_startup` to initialize the path. This is only necessary once per MATLAB session. Afterwards you can initialize ICNNA without further calling this script.
4. Execute `ICNNA`.

In opening ICNNA, a small menu window appears offering three options as shown in Figure 1. The first allows reading experimental data and its functioning is explain in detail in Section

Data preparation mode. The second is MENA itself. The third option is just a shortcut to obtain a database out of an existing experiment. This option can also be accessed from other parts of the GUI. The menu is completed with an access to the application credits shown in Figure 2.

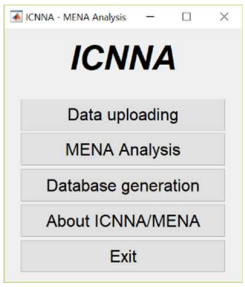


Figure 1 ICNNA main menu

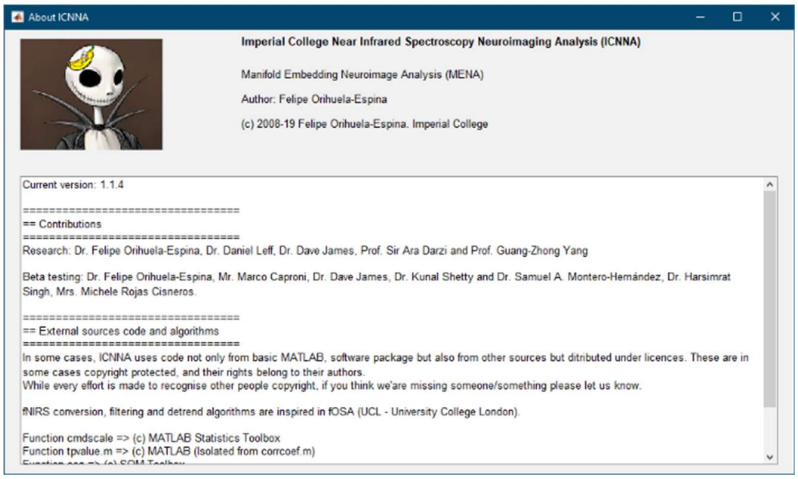


Figure 2 About ICNNA box

ICNNA Graphical User Interface (GUI) vs Application Programming Interface (API)

ICNNA software package holds a library of objects for data storage and manipulation, as well as performing manifold embedding NIRS analysis (MENA). This API is intended for programmers and developers. Information about the API can be found in the technical documentation accompanying each function and method. For the non-developer users, ICNNA also incorporates a GUI. This User Guide describes the usage of ICNNA through the GUI.

Familiarising with ICNNA GUI

ICNNA GUI intends to be both flexible and easy to become familiar with. In fact, the author hopes that the user does not actually need to read through this User Guide, but instead he learns his way through ICNNA naturally by using it straight out of the box. In this sense, this User Guide, does not explain every single option

available in ICNNA, but rather acts as an introduction for the novel user. Two main aspects summarise ICNNA GUI environment.

- Virtually each element is manipulated in its own window, and
- Each window is intended to be self contained enabling full control of the element

Importantly almost every window can be invocated separately from MATLAB's command window if wanted, to explore or modify single pieces of information.

Data preparation mode

Data in ICNNA is collected under an Experiment. The experiment can be regarded as the basic document in the data preparation mode.

Experiment, subjects, sessions and data sources

An Experiment gathers data from a number of subjects over a certain defined sessions. During each session data from several sources or devices, can be collected. Subjects, sessions and data sources form the basics of any experiment in ICNNA as depicted in Figure 3.

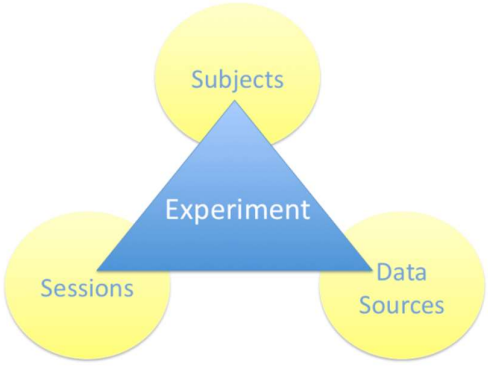


Figure 3 Components of an Experiment

In a sense, an ICNNA experiment data can be understood as a hierarchical tree. An **Experiment** is carried over a number of subjects. Each **Subject** in turn participates in a number of sessions. And during each **Session** a number of data sources can be acquired. This is illustrated in Figure 4. This hierarchy is flexible; not all subjects must undergo all sessions, nor must all sessions incorporate data from all sources.

The concept of **Session** is critical in ICNNA. A session in the strictest sense corresponds to a data collection exercise. But it is by proper definition of the experiment sessions that ICNNA can model virtually any experiment design. An example may be a longitudinal clinical trial in which there is a control group performing a task A, and a second group performing the task B being tested along 3 time points. ICNNA model this experiment by means of six sessions; two tasks at 3 time points. Note that each subject may perform either only one of the tasks or both, thus the subject will be "assigned" to the correct group simply by having the corresponding session/s associated with him.

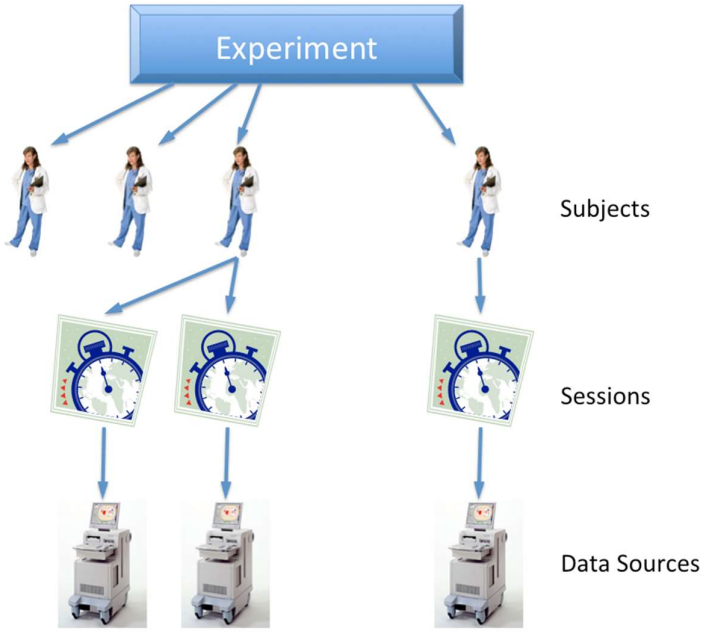


Figure 4 The Experiment element hierarchy

☞ Hint: In principle there's no limit to the number of elements; subjects, session and data sources that can be included in an experiment other than the computer's memory limitation. However in practical terms, ICNNA becomes slower as the number of elements in one single category grows.

A **dataSource**, also referred to simply as data source, in ICNNA represents data collected from a certain device or in other words a recording. A data source is capable of incorporating the data in its raw state or in several different processed stages. At any given time however, only one stage is active. The raw state of the information is referred in ICNNA as **rawData,** whereas the processed stages are referred to as **structuredData**. Any structured data can be thought of as a 3D array; <temporal, spatial, signal> as illustrated in Figure 5. For instance, an fNIRS image from a continuous wave device recording at 2 wavelengths at 10Hz with a 24 channel configuration during 1 minute, may produce a <600 samples, 24 channels, 2 Haemoglobin species> unprocessed structuredData. It is possible to access all this information directly if necessary through the ICNNA API. Please refer to the technical documentation for further details.

☞ Although currently only support for functional NIRS data is given, ICNNA has been thought for future extensions, so that data arising from other sources, such as other neuroimaging modalities, or other devices as for instance a heart rate monitor, will also be easily incorporated. Please refer to the technical documentation on how to expand ICNNA to fulfil your needs.

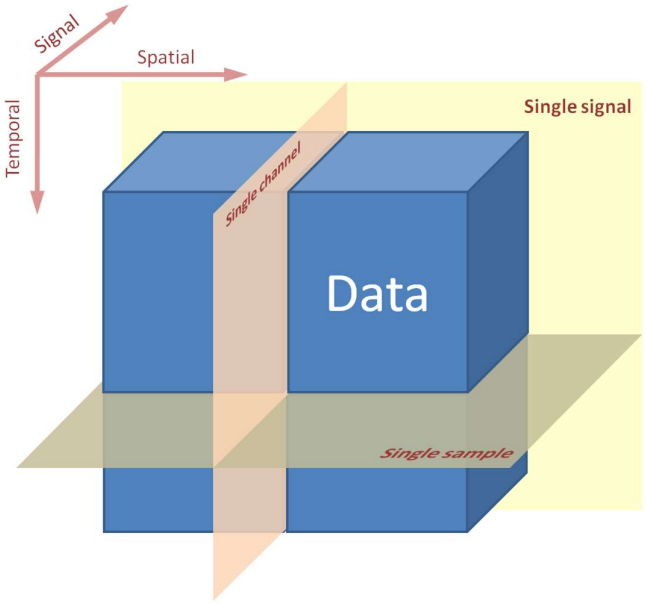


Figure 5 StructuredData as a 3D matrix

A system of identifiers

A name or a tag and a description describe every element in ICNNA. These act more as mnemonics for the user rather than as element descriptors. The user is free to use these in any way s/he likes. More importantly than the name or tag, in ICNNA, every element namely subject, session and data source, is uniquely identified with **a numeric ID**. This ID must is a natural number and ICNNA enforces them to be unique within an experiment for elements of the same kind. For example, there can be only one single subject whose ID is 1, but there may also be a session with an ID equals to 1 as illustrated in Figure 6. It is this numeric identifier that must be indicated when referring to each element. This numeric identifiers permits the user to have repeated names, and can also help in ensuring privacy of subjects which is a common ethical requirement, since now their names are not necessary to identify them.

ICNNA ensures that data sources are consistent across sessions and that sessions are consistent across subjects through session and data source definitions. In other words, all records referring to session *n* across all subjects stick to the same session definition. And analogously, for data source within the sessions, consistency ensures that data source *m* always collects data from the same device. In order to achieve this, the experiment keeps a repository of the ID s of all the subjects, sessions definitions and dataSources definitions as illustrated in Figure 6.

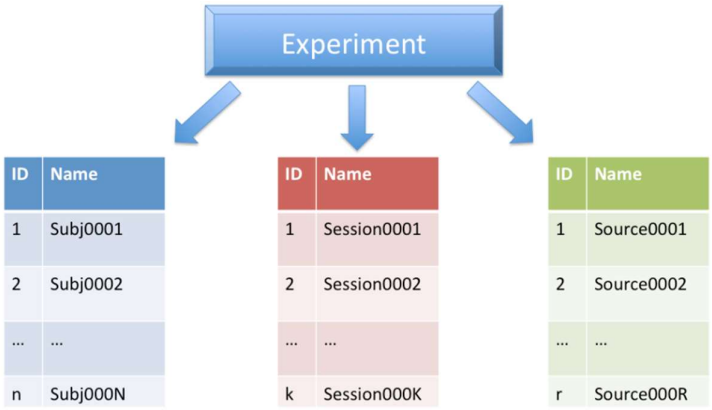


Figure 6 System of identifiers

Timeline

In designing an fNIRS experiment, the researcher frequently thinks in terms of a timeline along which the experimental condition or stimulus are spread at specific timing. This is commonly referred to as the *train of stimulus*. In ICNNA, all structuredData is associated with a timeline, which represents the time course of events associated with the data source during the session. A **timeline** is a collection of events grouped under experimental conditions or stimulus. Each event has an onset linked to a data sample and its corresponding duration. ICNNA can in this way incorporate experiments designed under the block paradigm or under event related paradigm. Tags identify the different conditions, and the user can specified the inter-condition behaviour to be exclusive or non-exclusive. Exclusive behaviours prevent events from mutually exclusive conditions to overlap in time.

The timeline can also be used to mark or represent, not only experimental conditions, but any event of interest occurring during the experiment. In this sense, ICNNA's timeline is much more than the train of stimulus. For instance, in simulating a task, the task can be temporally split in smaller steps by marking each stage with different events. The user of ICNNA may express not only the full task condition, but also create "conditions" to segregate the task into the smaller steps.

> ☞ Timeline length and events are expressed in signal samples, not in units of time. Instantaneous events may be indicated using length 0.

ICNNA attempts to automatically catch the timeline from the raw data if this information is already implicitly or explicitly indicated. Similarly, if data is imported from fOSA, the timeline is constructed from original fOSA files. An example of a timeline with three different conditions is shown in Figure 7.

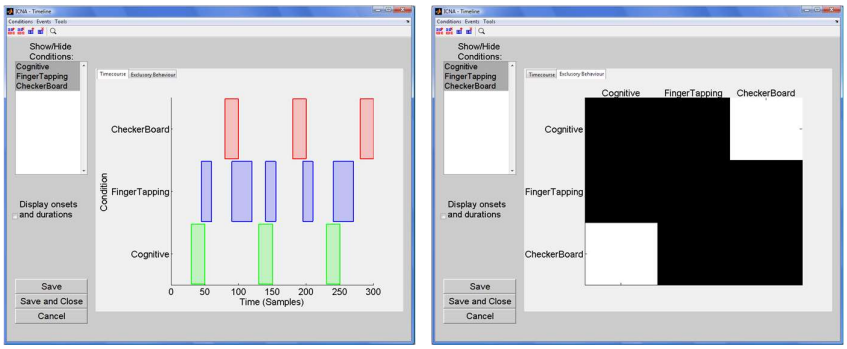


Figure 7 Example of a Timeline. The different conditions and stimulus can be individually manipulated (right) and the conditions exclusory behavior can determine whether the events are allowed to overlap (left).

Learning by example

This section explains how to read data into ICNNA with a figurative example. Let's imagine that a small dataset has been collected with a certain NIRS device. After selecting the first option in the main menu, the user access to the data preparation section. Alternatively this window can be directly invoked from MATLAB's command window by calling `guiExperiment`. In this data preparation section, an Experiment can be understood as the basic document to be saved to a file or recovered from a file.

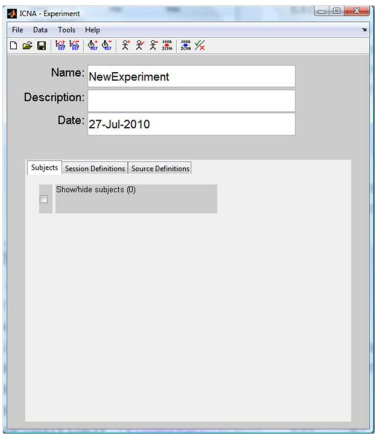


Figure 8 Experiment manipulation window

> ☞ Hint: Virtually every element in ICNNA has an associated manipulation GUI window. These are consistently named `gui`*`ElementName`* and can be accessed directly from MATLAB's command line passing the element as a parameter.

The experiment manipulation window, presented in Figure 8, is composed of the common menu and toolbar and the application area, which can be viewed into two halves. The top half presents some descriptor fields that the user can employ to

name and described the experiment. The bottom half includes three tags for exploring the three basic elements of an experiment, namely the subjects, the sessions and the data sources. For a new experiment all these will be empty. But they will become populated as more data is incorporated to the experiment. In all three cases this information is presented in the form of non-interactive tables. While the information included in each of the columns of the table differs between the different elements, importantly the left side of the table is common to all of them and will be used for displaying the element identifier (see Section A system of identifiers). This window structure with descriptors in the top half and components listed in the lower half is repeated throughout ICNNA GUIs and emphasize the conceptual hierarchy of the elements.

On the fly data acquisition

After typing a name and a description for the experiment, the user can now decide to declare the different data sources and sessions that are part of the experiment, or directly start introducing the experimental information corresponding to data collected from the different subjects. ICNNA attempts to catch the types of sessions and data sources *on the fly*. As new subject information is uploaded, ICNNA automatically matches the sessions declared in the new subject with those already declared in the experiment and ensures that sessions sharing the same ID comply with the same definition, i.e. they indeed refer to the same experimental session across subjects. In addition, if a new session identified by a new ID has been declared in the user, ICNNA automatically adds the new session to the experiment session repository. Similarly occurs with the data sources being acquired during the sessions. This matching and on the fly data acquisition is intended to reduce manual errors and to reduce the time and effort spent in uploading data to ICNNA.

Adding experimental data from new subjects

Let's upload data coming from a new subject then. In order to add a new subject to the experiment, click the button with the stick man and a plus symbol ( ). This will open the subject manipulation window, which again can be directly called from MATLAB's command window as `guiSubject`. The subject manipulation window is shown in Figure 9.

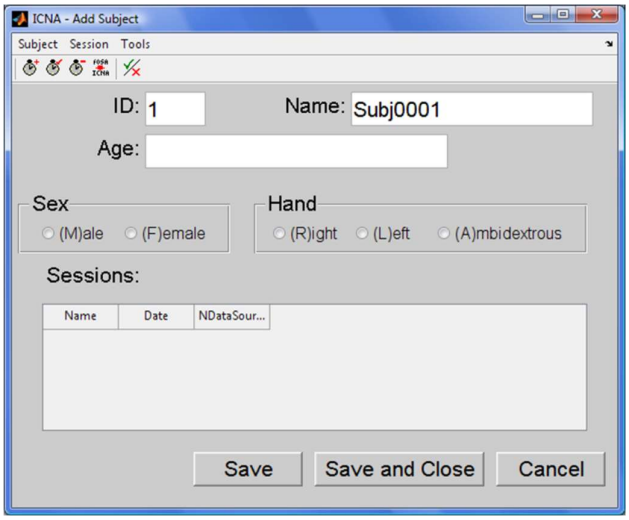


Figure 9 Subject manipulation window

Basic information about the subject can be stored, including handedness, sex and age. To differentiate group of subjects they simply have to undergo different set of sessions during the experiment, e.g. a control task and test task. When a new subject is being added to the experiment the subject is initialized without any session.

Defining new sessions: Session definitions

From the subject manipulation window it is possible to add a new session to the subject by clicking the icon with the clock watch ( ). This opens a new window, similar to the one in Figure 10, for session manipulation. The corresponding command line function call is `guiSession`.

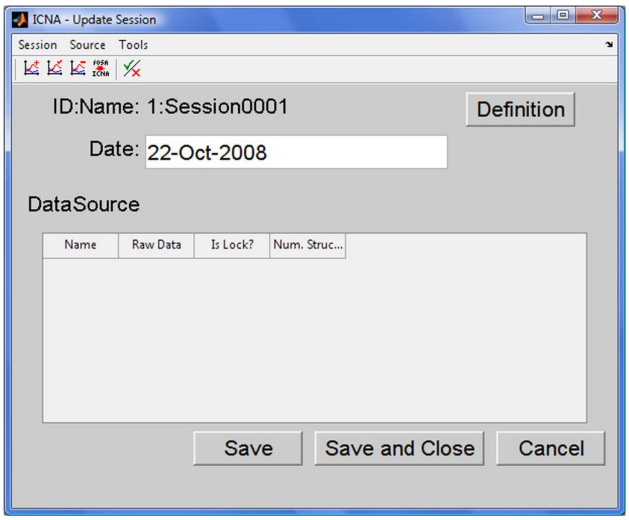


Figure 10 Session manipulation window

The first thing to do in ICNNA when declaring a new session is to establish a definition of it in which an identifier is assigned to the session and the user

declares all the devices from which data is to be collected within this session. If more than one subject participate in a given session, the session identifier is used to trigger a cross-validation. Remember that ICNNA will ensure that sessions with the same ID are consistent i.e., they have the same definition; name, description and list of data sources. In order to define the session, click on the *Definition* button. A new window will pop up to allow the user to establish the definition of the session. This new window is similar to that in Figure 11.

Introduce the session ID, name and description and then define as many new data source as necessary by clicking in the button ( ). It is likely that recorded data may all came from one fNIRS device only. But shall data arise from more than one device; indicate so by incrementing the device number. Once all sources of data for this session have been declared, simply save and close the definition of the session. An example of a data source definition is illustrated in Figure 12.

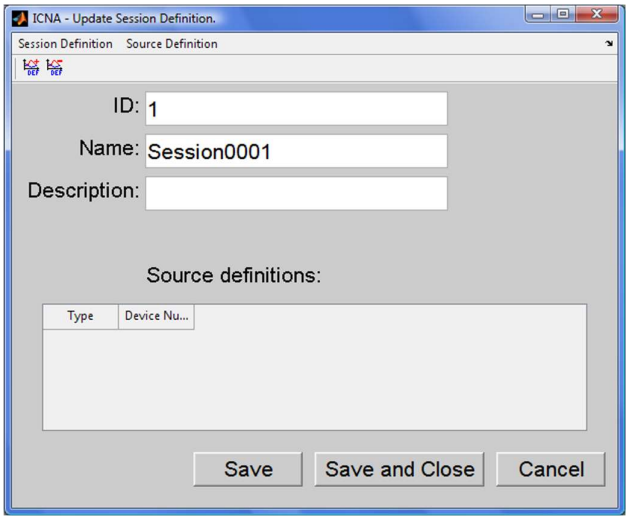


Figure 11 Session definition window

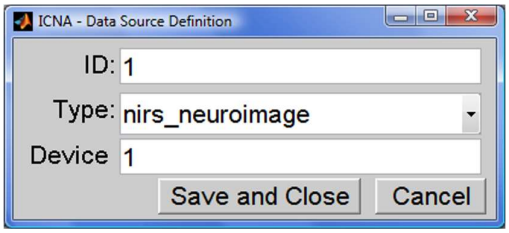


Figure 12 Data source definition window

☞ For every user the session must be defined in the same way. Sessions with the same ID are expected to be EXACTLY the same i.e. they share ID, name, description and definition of sources. If a mistake is made in introducing the information about the session, ICNNA will detect the inconsistency. This however will not happen immediately, but when saving the subject into the experiment.

The session definition may be updated later on if necessary. If the session definition is changed, the session is reset and any existing data is removed from the Experiment. To inform the user about this possible removal of data, a warning dialog appears as shown in Figure 13. It is strongly recommended to ensure that the session definition is correct before proceeding to upload any data to the Experiment. The removal of data only refers to the resetting of the session, but do not panic as the original files are never removed or deleted by ICNNA. You may upload your data again if necessary.

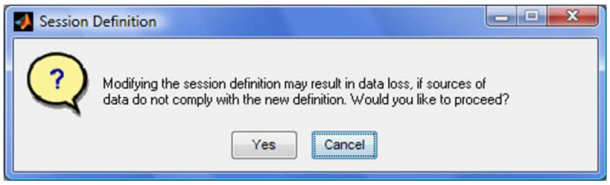


Figure 13 Warning message: Updating session definition

After accepting the changes to the session definition declaring which type of data was recorded during the session, you are now ready to upload the real data to the session.

Uploading NIRS data to ICNNA

Click the button for adding new data source ( ) and type the ID if prompted to do so. Note that the data source must have been previously declared. A new window will be displayed to allow the user to upload data to the Experiment. The window looks like the one in Figure 14.

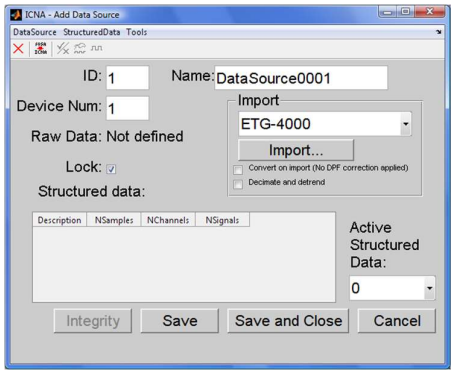


Figure 14 DataSource manipulation window. From here raw data files can be imported, the timeline can be accessed, integrity values can be manually set and timecourses of the signals can be displayed.

After indicating the correct ID and data name, raw data is to be imported or uploaded. Simply selects the device and indicate whether you would like to convert the raw light measurements into changes in haemoglobin. This conversion is made using the modified Beer-Lambert law. Although ICNNA does not focus on signal processing, it allows for decimation and linear detrending to

be applied to the haemodynamic signals. Data can also be imported from fOSA package, which allows for different signal processing. Conversion and/or processing of the raw data into haemodynamic changes, adds a structured data to the data source. Different structured data represents the data at different processing stages. There can be as many structured data as desired, but only one may be active at each moment. If there is only one structured data that becomes the active structured data within the data source. For posterior analysis, only the active data is taken into account.

> ☞ Currently ICNNA only supports HITACHI ETG-4000. Please refer to the technical documentation to create your own classes for importing data from other devices.

From this window you can access to the timeline (see Section Timeline), or display the time courses of the haemodynamic signals (See Figure 15).

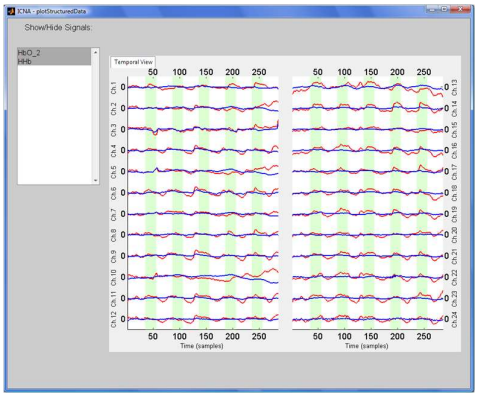


Figure 15 Timecourses of the haemodynamic signals.

> ☞ Hint: If your data has already been processed by fOSA (from UCL) it is not necessary to follow the whole process of declaring, sessions and data sources. At any point you can import your data by clicking in the fOSA to ICNNA button ( ). ICNNA will import all the structuredData and appropriately generate the session and data source definitions.

Removing data affected by artefacts: Integrity checks

Once the data has been uploaded, ICNNA requires that all data undergo a signal quality test known as integrity test. Data integrity tests prevent data affected by artefacts to be passed to further analysis. The integrity checks are made on a per channel basis, and ICNNA allow the integrity marking to be made manually (i.e. following visual assessment), automatically (using implemented algorithms) or a combination of both. The manual set of integrity codes per channel can be accessed from the dataSource dialog from its Tools/Integrity menu. A snapshot of this integrity setting dialog is shown in Figure 16.

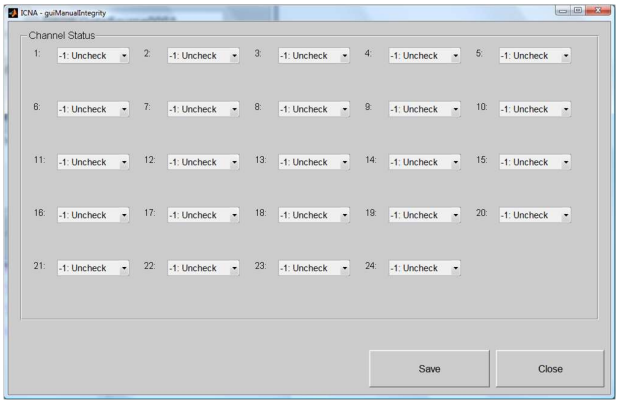


Figure 16 guiManualIntegrity. Set each channel integrity value by choosing the appropriate code in the drop-down menus.

Regarding the automatic integrity checks, the following integrity tests are available:

- Complex numbers
- Saturation related problems such as Apparent Non-Recording and Mirroring
- Optode Movement (not yet available in the present version)

Data integrity tests can be applied every time a subject, session or data source is uploaded, or for the whole experiment at once after al data has been uploaded. Select the integrity button ( ) from any of the dialogs to carry out the integrity checks on the object of interest. An illustration of the automatic integrity checks dialog is shown in Figure 17.

Figure 17 guiCheckIntegrity. Automatic integrity checking available tests

> ☞ MENA systematically ignores all data, which has been left unchecked, i.e. not undergone an integrity check. If you do not want to apply integrity checks to your data, simply deselect all tests and run the integrity without any tests. In fact, this will be applying a mandatory default test for data existence.

Data integrity is extremely time consuming, with the time required for large experiments to last for two or three ***days***. Please be aware that Windows will

automatically kill your process after 5 days by default. It is highly recommended that you make a backup copy of your data before proceeding to run the automatic integrity tests. While it is safe to use the computer for other purposes while tests are running, it is also advisable not to do it to speed up things. If you minimise the MATLAB window during the integrity test it is unlikely that you will be able to maximize it again until the test is finished, giving the appearance that MATLAB is not responding, however this is not the case, just be patient. The verbose option makes the process to inform of the progress through messages in the main command window.

Some final notes on data preparation

Data preparation in ICNNA is a very systematic a repetitive task. Subjects are added to the experiment, sessions are added to the subjects, and data sources are added to the sessions. Do not forget to save all the information as you go back in the chain. If data has already been preprocessed with fOSA, ICNNA provides a batch tool to import all data at once. Place all the fOSA files to be imported into a single directory. This requires the fOSA files to be extracted (or at least temporally copied) from the directory generated by fOSA, and very likely renaming those files. You can import a batch of files as long as they are in the same directory by clicking on the button ().

Basic Visualization and database generation

Yet again, ICNNA main purpose is not to provide an classic statistical analysis software tool. Notwithstanding, ICNNA is capable of limited fNIRS data visualization and statistical analysis.

Basic Visualization Series

From the tools menu in the guiExperiment dialog, select the *Plot generation* option to access the dialog in Figure 18. This dialog permit the generation of a set of image series either temporal or spatial, the latter identified by the prefix TOPO. These series can be:

- Averaged (AVG) or non-averaged across blocks (NBA)
- Present a single channel (SCH) at a time, or all channels at once (ACH) – Temporal only
- Grand averaged series are prefixed by GA. Currently only one series is available.
- Three different output formats are currently supported; *.fig* for MATLAB's original format so any further modification can be made, *.jpg* at 150dpi suitable for online publication, and *.tif* at 300dpi which is a very common accepted standard in scientific journals.

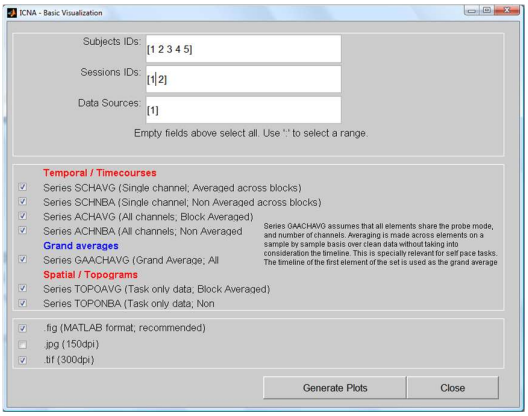


Figure 18 guiBasicVisualization. Several series of figures, both temporal and spatial can be generated, either averaged or non averaged across blocks, for either the whole experiment or any subset of it.

> ☞ Some series can generate a large amount of data. Please ensure that you have enough space in your data drive.

The images can be generated for the whole experiment (default) or any subset of it. Simply indicate the subset of interest in the subjects, sessions and data sources ID selection edit boxes provided. More advance visualization is possible by means of the ICNNA API, pelase refer to the technical documentation for details.

Database generation and the activity matrix

Access to the database generation tool and statistical analysis leading to the activity matrix can be navigated either from ICNNA main menu, from the guiExperiment or from MENA computation of experimentSpace dialog. In all cases, an **experimentSpace** must be generated prior to the obtention of the database. Briefly, the experimentSpace is an intermediate representation of the Experiment data in which active structuredData across subjects, sessions and data sources is further split into block data for different stimulus, trials and signals. Once this partition is available, the database can be generated along with the activity matrix. Please refer to Section Computing the Experiment Space for more information. The concept of experimentSpace is explained in detail in Section Experiment Space, Feature Space and Projection Space.

Database is written to a standard *.csv* file format that can be read in Excel or SPSS, and many other software for further statistical analysis. A crude overview of the activity map across sessions is represented in the activity matrix. The activity matrix compares the baseline of a block versus the task related data of that block. An example of an activity matrix is shown in Figure 19.

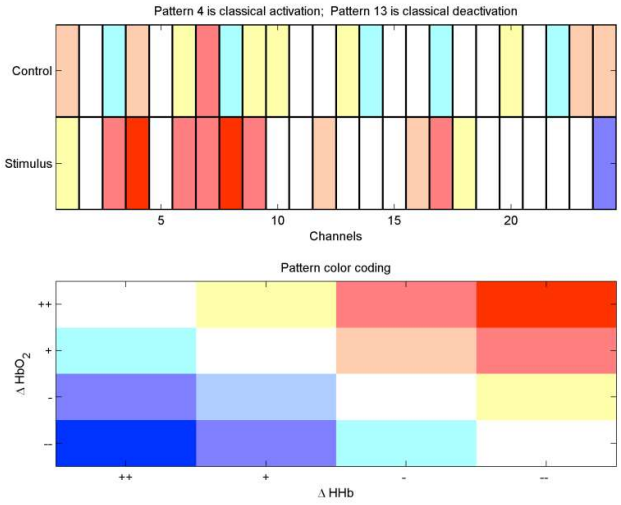


Figure 19 Example of an activity matrix. Rows represents sessions and stimulus (in the above example there is only one condition per session). Columns represents channels. Lower panel shows the legend: (++) Statistically significant signal increment during task with respect to baseline, (+) Signal increment during task with respect to baseline, but did not reached statistical significance, (-) Signal decrement during task with respect to baseline, but did not reached statistical significance, and (--) Statistically significant signal decrement during task with respect to baseline. Color codes sixteen different signal behavior. Redder color corresponds to more active channels. Bluer colors correspond to deactivated channels.

Manifold Embedding Neuroimage Analysis (MENA)

After all data of the Experiment has been uploaded and data integrity checks have been performed, you are ready to analyze your data with MENA. It is very likely that the researcher is familiar with the classical inferential analysis and may it be that s/he is unaware of manifold embedding techniques. The following section describes the basic concepts behind manifold embedding analysis, without detailing the mathematical description.

Charting the Earth: dimensionality reduction

Dimensionality reduction is perhaps the most well known day to day application of manifold embedding techniques, and its most common incarnation is perhaps principal component analysis (PCA). But what exactly is manifold embedding and in particular dimensionality reduction. The classical example to explain the concept of manifold embedding is through cartography. Let's imagine an object in a certain environment. The object is the manifold, and the environment is called a space. The earth is a three dimensional object in a spatiotemporal Euclidean space. When cartographers want to represent parts of the Earth, or the whole of it, into a geographical map, they face the problem of representing the 3D object that is the Earth into a 2D surface that is the paper or map. The cartographers must then attempt to represent the distances as faithfully as possible minimizing the unavoidable distortion, so that the distances in the map are somehow representative of the real distances in the object. They are facing a dimensionality reduction problem. In general, manifold embedding –or by extension, dimensionality reduction- techniques, projects an object into a lower dimensional representation of itself. Importantly the object is not necessarily 3D nor (hyper)-spherical, but can be of any dimensionality, of any shaped, and furthermore be folded over himself in the most entangled ways.

Several possible solutions exist to solve this dimensionality reduction problem, each one with different pros and cons, but importantly they can all be though as a two-stage process. In the first stage a criteria for measuring the distance is imposed. In mathematics the distance criteria is usually referred to as a metric. In the second stage, the distance calculated between landmarks in our object are charted or embedded to create the final map.

Finally, dimensionality reduction is of particular interest when the final map is either 2D or 3D. This is because the latter are the only dimensions that we can perceive and represent.

Defining the distance: Available metrics

In the current version, MENA offers three different basic distances plus the option of computing them either directly in the ambient space or along the manifold. The available distances are:

- *Euclidean* – The ordinary distance. The shortest distance between two points as the crow flies.
- *1-correlation* – A popular measure of signal similarity but has the drawback of not being a formal metric but a pseudometric.
- *JSM* – The square root of the Jensen-Shannon divergence – An entropy based measure of signal similarity.

While most dimensionality reduction work directly on distances computed directly in the ambient space, proper manifold embedding implicitly assumes that distances are those along the manifold or geodesic distances. Thus, ICNNA allow the computation of the geodesic from the basic distances.

- *Geodesic* – The shortest distance between two points when travelling along the manifold's surface.

Note that the geodesic can be computed over any of the basic distances.

Defining the projection: The embedding

In the current version, MENA offers two different embedding techniques:

- *Classical Multidimensional Embedding (cMDS)* – Attempts to find the projection that best preserves the original distances.
- *Curvilinear Component Analysis (CCA)* – Similar in goal to cMDS but the best projection is a function not only of the original distance, but also of the final projected distances.

By combining different metrics with different projection methods, it is possible to obtain different manifold embedding techniques some of which are known by particular names as indicated in Table 1. There is no better or weaker embedding. Different embeddings can unveil different information as they unfold the manifold in different ways. For instance, Isomap will favour a better recovery of the long distances, resulting in a better global projection. CDA on the other hand, will give priority to local distances, and hence allow the user to better explore the neighbourhoods. Friston functional space, projects to a hyper-sphere where the main axis are attentional and intentional.

Table 1 Several Manifold Embedding techniques implemented in ICNNA

Metric	Projection	Embedding	Comments

Euclidean	cMDS	PCA	PCA and cMDS are a mathematical duality.
1-correlation	cMDS	Friston's functional space	Projection axes unveils attentional and intentional
Geodesic (Euclidean)	cMDS	Isomap	
Geodesic (Euclidean)	CCA	CDA	

Experiment Space, Feature Space and Projection Space

In understanding MENA, it is necessary to recognise three different representations of the *same* information, or spaces.

First, the **Experiment Space**, is a 7 dimensional lattice where the dimensions arise from the concepts explained in the data manipulation section. These dimensions are:

1. Subject
2. Session
3. Data Source
4. Stimulus or experimental condition
5. Signal e.g. chromophore
6. Channel or spatial location
7. Trial or block

The coordinates of the points in this space are the identifiers of the different elements where suitable, or a certain default ordering. The points in this space are a temporal signal, which extract single trial information composed of baseline, task and rest or recovery chunk. In this representation, all the information of the Experiment is present in the space, but the information has been split to its *quantum* division by trial and homogenized in its temporal variation.

From the small temporally homogeneous block pieces, the user then indicates how he desires to explore its data by grouping pieces, either by channel, or by signal, and filtering which subjects and sessions he wants to look at. The result of this filtering and grouping generates the Feature Space. The **Feature Space** is a high dimensional space where each dimension is a feature or in other words, a sample of a single chromophore at a certain location at a certain time. The number of dimensions varies depending on the grouping that has been selected. The number of patterns or points existing in this space depends on the filtering. This space is the one to be charted to a lower dimensional representation.

The **Projection Space** or the Embedded Space is the low dimensional projection of the Feature Space. The key concept behind MENA is that the Euclidean distances in the Projection Space are representative of those distances as measured by any metric, in the high dimensional Feature Space. Normally, the Projection Space is

of a dimensionality small enough to be represented on the computer screen, i.e. is either 2D or 3D. While the dimensionality of this space is smaller than in the Feature Space, there exist a one-to-one relation of the patterns.

Group analysis: Clusters and database generation

Very often in research analysis is carried out by understanding the differences between different groups, e.g. those with a treatment against a control group. In MENA, this concept of study groups is represented by clusters. A cluster is a group of points in the Projection Space. It can be constructed out from grouping subjects, sessions, channels, or stimulus, or trials in any combination. Note that MENA is not equipped with clustering tools; it is the user who defines the clusters.

MENA is not an inferential statistics tool. It will not give you the statistical significance of the differences between groups. However, this does not mean that MENA is not capable of explaining differences among groups. For this MENA uses a measure of distribution difference known as *Earth Mover's Distance* (EMD). EMD is capable of quantifying the differences between two clusters in the Projection Space. In addition, ICNNA provides a convenient database generation tool, which converts the Experiment Space into a tabular form that can be exported into comma separated values (.csv) format. The .csv format can be later imported into any common spreadsheet such as Microsoft Excel, or some statistical packages such as SPSS.

Learning by example

Open ICNNA as indicated in Section Opening ICNNA and select the MENA analysis option to enter the MENA analysis. It is also possible to access MENA by directly typing the command `guiAnalysis`. If you have not yet uploaded your dataset into ICNNA, please refer to Section Data preparation mode.

The analysis window, presented in Figure 20, is composed of the common menu and Toolbar and the application area, which can be viewed into three parts. The top part present some descriptor fields that the user can employ to name and described the experiment and holds the indicator of whether the Experiment Space has been computed. In the middle of the analysis window, a panel of tags allow the user to configure the analysis that s/he wants to perform. Finally in the bottom part a table lists all the clusters declared for the analysis.

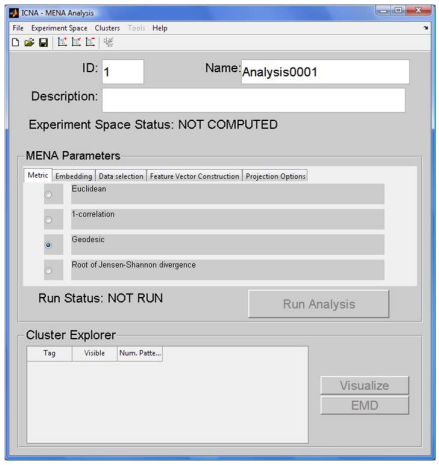


Figure 20 MENA main window

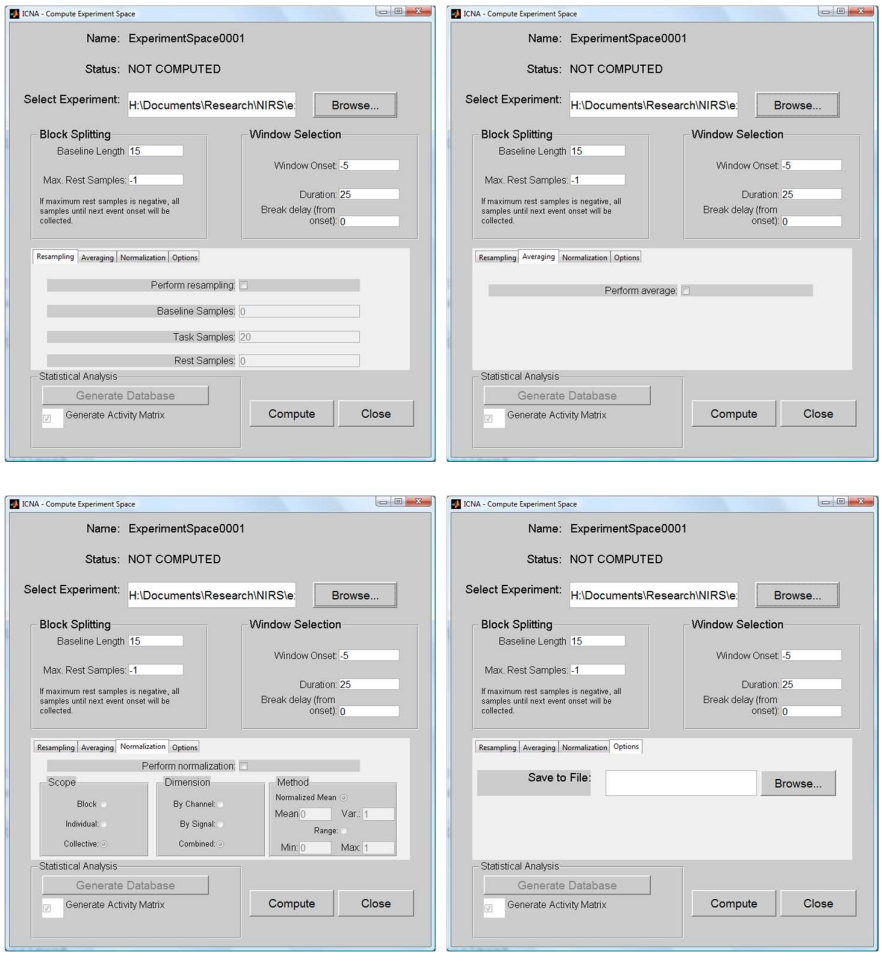


Figure 21 Experiment Space definition window with the database generation tool. The construction of the Experiment space is flexible; blocks can be resampled and/or averaged and data may be normalized.

Computing the Experiment Space

The first step in MENA is defining the **ExperimentSpace** or simply experiment space. Go to the menu Experiment Space and select the option *Compute…*. This opens a dialog as the one in Figure 21. The Experiment Space breaks the experimental data into small blocks of data as explained in Section Experiment Space, Feature Space and Projection Space customizing which data are to be analyzed. This dialog has two panels, four tags and three buttons. Start by selecting the Experiment for analysis in the edit box in the top of the dialog. The two panels are:

- *Block Splitting*: Prior to any processing e.g. resampling, averaging or normalization, data is partitioned in raw blocks. Each block arise from a given event in the timeline and in all cases is composed by three chunks; baseline, task and rest or recovery periods, as illustrated in Figure 22.

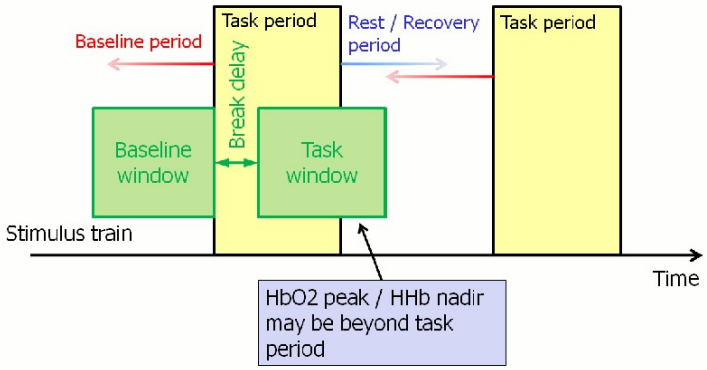


Figure 22 Understanding block splitting and window selection.

- *Window Selection*: After processing, the information selected for analysis out from every block can be finely customized to how many samples of baseline, task and rest following the task are selected. This is expressed in terms of a time window onset and duration. The onset can be negative to include some baseline values, zero to match the task init or a positive integer to leave a delay from the onset of the task.

The dialog tags permits processing of the blocks after splitting and prior to window selection, as well as providing some other options, as follows:

- *Resampling*: Recommended for self paced task. It ensures that the blocks are all sequenced to the same length by normalizing the time length to a number of samples.
- *Averaging*: Averages all trials of a certain session. A common operation to increase the signal to noise ratio at the cost of losing statistical power. If selected, the resulting single trial is identified as block 1.
- *Normalization*: Permit signal normalization either to a fixed range or the a certain mean/variance. Normalization can be collective for the whole experiment or as an individual basis. Finally, normalization can be made across signal, channel or combined.

- *Options*: If desired, the resulting Experiment Space can be saved to a file for later reuse.

Finally, the three buttons perform the following tasks:

- Generate database: Display a dialog for selecting a destination folder. Then it generates a tabular database in *.csv* format, ready to be imported in external spreadsheets or statistical packages. It also generates a helper file indicating the contents of the database. This button is initially disabled, and will enable only when the Experiment Space has been defined. Tick the small checkbox if you want the activity matrix also to be generated.
- Compute: Generate the Experiment Space with the indicated features. Fill the information in the tags before proceeding to compute the Experiment Space. After the Experiment Space has been defined, and computed, the database generation button will become enabled.
- Close: Close the dialog.

A progress bar will inform of the progress in the construction of the Experiment Space as shown in Figure 23.

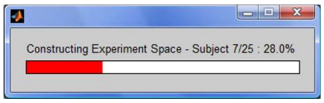


Figure 23 Progress monitor during construction of Experiment Space

Running the Analysis

Once the Experiment Space has been computed, it is now possible to construct the Feature Space, and finally embed the data in the Projection Space. Even though the Experiment Space actually contain all your dataset, your MENA analysis can be restricted to a small subset of your data to explore certain particularities. Start by indicating which subjects and sessions do you want to include in your analysis. Use the subject and session identifiers in the tag Data selection as shown in Figure 24. You can use MATLAB range notation for short, e.g. [1:4,6,7] selects the elements 1, 2, 3, 4, 6 and 7.

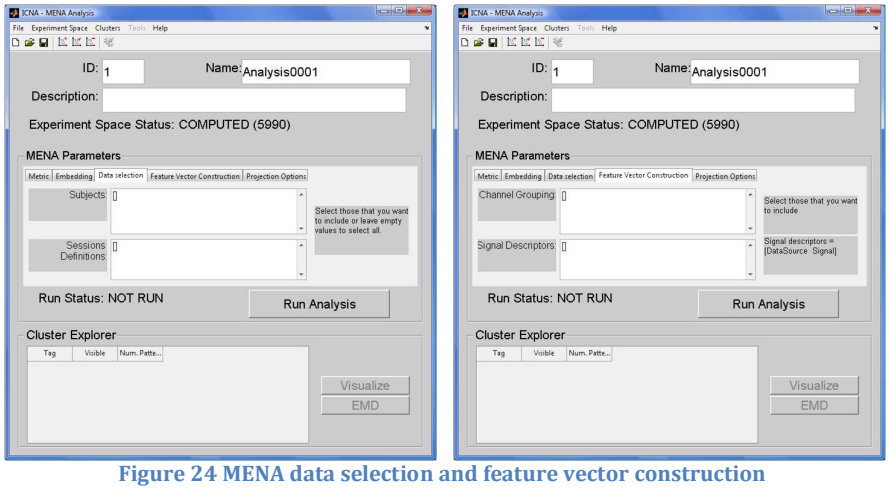
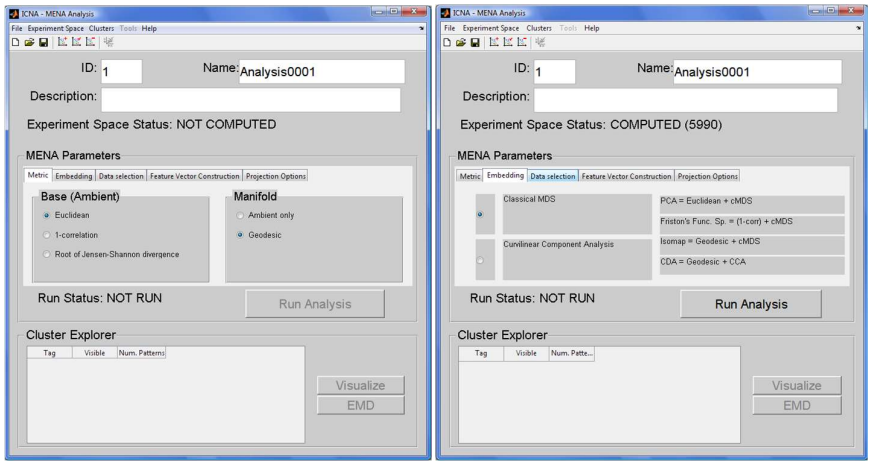


Figure 24 MENA data selection and feature vector construction

After indicating the filter of your data, indicate how you want the atomic trial data blocks to be reorganized into patterns. You can do this in the Feature Vector Construction Tab as illustrated in Figure 24. It is necessary to indicate the channel grouping and the signal selection. One pattern will be created for each channel group. Each channel group can either be a single channel or a set of channels. The only restriction is that if more than one group is to be built, then they must be composed of the same number of channels. This will enable to carry out analysis by channel, or by region or even full behaviour analysis. Indicate each group by stating the channels to be included in that group and use a semicolon ';' to separate among groups. Again MATLAB range notation will be accepted. Groups will automatically be assigned sequential identifiers. A neuroimage device such as a near infrared spectroscopy collects data from more than one signal at a time, e.g. the oxygenated and deoxygenated haemoglobin species. It is therefore necessary to select which signals from which devices are to be included in the analysis. Since each signal is generated from a data source, the signal is uniquely identified by a pair [*dataSource signalNumber*]. More than one signal can be included in the analysis by using a semicolon. For instance, [1 1; 1 2] will select signals 1 and 2 – perhaps oxy and dexoy haeomoglobin- from data source 1. Remember that regardless of your choice, only data that has passed the integrity check will be included in the analysis.



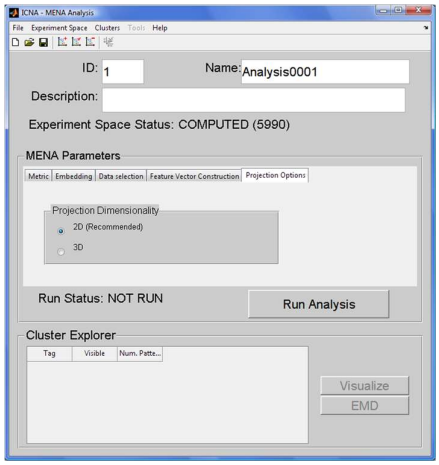


Figure 25 MENA metric and embedding selection

The two steps above, permits to define the construction of the Feature Space out from the Experiment Space. The next step is to select the metric or base distance to impose a topology on the Feature Space. Go to the Metric tab to choose from the available metrics. Then choose the embedding technique of your interest in the Embedding tab. And finally in the Projection Options tab, select the dimensionality of the Projection Space. This is illustrated in Figure 25. The last step is to actually run the analysis by pressing the *Run Analysis* button.

> ☞ MENA is not restricted to the 2D or 3D projection offered by the GUI. However it is only possible to visualize plots in either two or three dimensions. For higher dimensional projections, it is necessary to use MENA from the command window.

The time necessary to perform the analysis highly depends on the number of patterns and the metric selected, with the geodesic metric being the slowest to compute, and may vary from a few seconds up to an hour or so. Progress is reported through the command window and a final dialog will indicate when the analysis has been done (see Figure 26).

Figure 26 Analysis finished

The goodness of the embedding: The distance distortion plot

Different embedding techniques have different ways of evaluating the goodness of a projection. Despite particular evaluation measures, the goodness of the projection can be estimated by looking at the distance distortion plot, which in addition can be constructed for any embedding. The distance distortion plot can be displayed by clicking on the button provided ( ). The distance distortion plot

is a deceivingly simple plot of the distances in the low dimensional Projection Space against its corresponding original high dimensional Feature Space. Because one of the goals of dimensionality reduction techniques is to allow visualization of high dimensional data, the intrinsic dimensionality of the original manifold is perhaps not as relevant as it is understanding the distortion suffered by the manifold during the embedding process. The distance distortion plot is indicative of whether local or global distances have been better preserved during the embedding. An example of the distance distortion plot can be found in Figure 27.

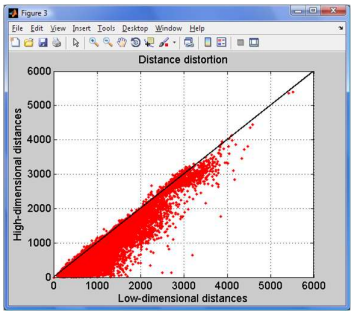


Figure 27 An example of the distance distortion plot

Before visualizing the results: Defining the clusters

In MENA, the results can only be visualized by means of clusters. After the analysis has been run, patterns have been formed and therefore it then possible to define your clusters. There is no restriction on how the clusters can be defined. Indeed, patterns belonging to a cluster may be picked "by hand" one by one. But ICNNA interface provides a convenient way of quickly defining your cluster. Click on the toolbar button for adding a new cluster ( ). The dialog for defining a new cluster, in Figure 28, appears. As with any other element in ICNNA, clusters are also distinguished by an identifier, which must be unique within a particular analysis.

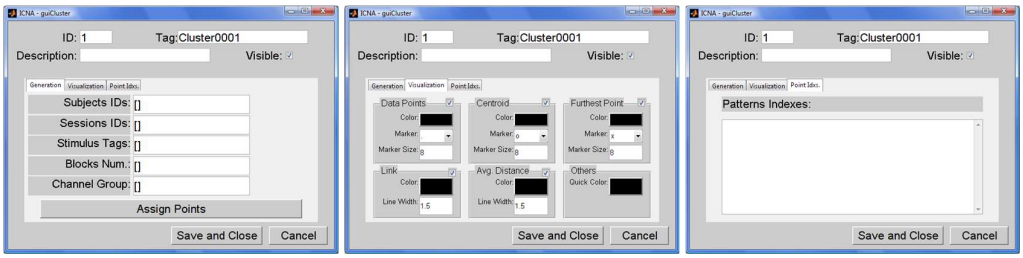


Figure 28 Cluster definition window

The first tag of this dialog permits to establish the criteria for grouping patterns in a cluster. Simply indicate those subjects, sessions, etc that are to be part of your cluster. Remember that channel refers to channel groups rather than to individual channels. Leave text boxes empty for selecting all possible values. After the criteria has been establish click the button *Assign Patterns* for ICNNA to assign the relevant points to the cluster. You can check these points in the last tag. The second tag allows you to customize the representation of your cluster. An important property

of the cluster is the *Visible* property. Although the embedding has been done with all the points selected during the process of running the analysis, the visualization of the points can be enabled of disabled to facilitate the exploration of your data. Use the *Visible* property to change the visibility status of a cluster. There is no limit in the number of clusters that you can define. Since clusters are defined after the analysis has been run, and it is not hard coded in the experiment space, there is no restriction to have a point belonging to more than one cluster at the same time, e.g. a point representing the channel 14 of subject 3, can be grouped in clusters 'Channel 14', 'Left temporal' or 'Treatment dose A' to allow different explorations over the same analysis.

> ☞ If analysis is re-run at any point, defined clusters will be eliminated as existing patterns in the Feature and Projection Spaces may change.

An example of an analysis with clusters already defined is shown in Figure 29.

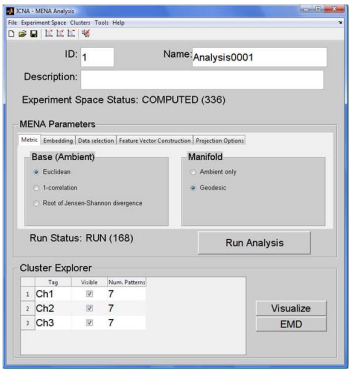


Figure 29 Example of an analysis with three clusters

Visualizing the results on the Projection Space

Once the clusters have been defined MENA offers the user with two possible ways of looking at your data. A qualitative spatial visualization of the Projection Space, which is mainly useful for data understanding and a quantitative determination of cluster difference as measured by the EMD.

Click the *Visualize* button to access the spatial visualization. An example of the visualization window can be seen in Figure 30. The visualization window holds a main axis showing the Projection Space. Understanding the spatial distribution of the different patterns and the clusters they form in the projection is crucial to maximize the appreciation of the analysis results.

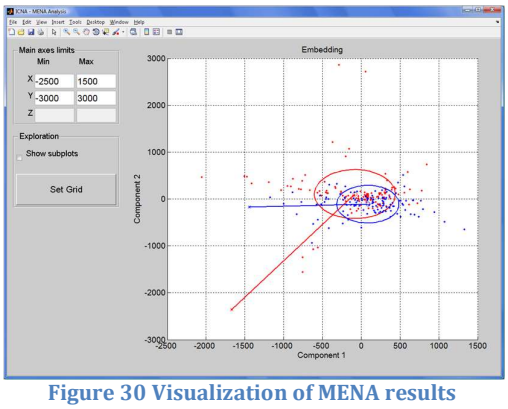


Figure 30 Visualization of MENA results

In the visualization window it is possible to zoom in and out. Furthermore, if the embedding has been projected into 2D space, then it is further possible to explore the attached feature vector linked to every point in the projection, or to explore regional average behaviour by over-imposing a grid on top of the Projection Space as shown in Figure 31. The subplots present the feature vector, which depends on the definition stated for the analysis (see Figure 24). The grid definition window is illustrated in Figure 32. If a 3D projection has been selected, the exploration tools will not be available, but instead it is possible to rotate the view of the Projection Space.

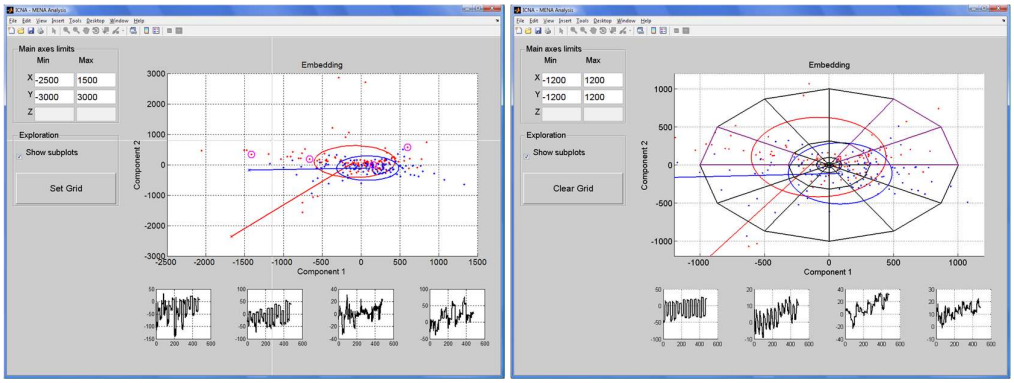


Figure 31 Exploration tools

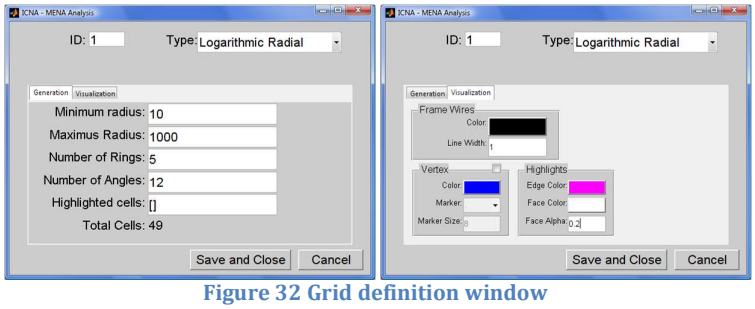


Figure 32 Grid definition window

To access the quantitative determination of cluster difference, click the EMD button in the analysis window (see Figure 29). An example of the EMD quantitative assessment of an analysis is illustrated in Figure 33. The quantitative

analysis show EMD distances between visible clusters in two views: numerical or an intensity coloured matrix. The lower the EMD value, the more similar are two clusters. Note how the main diagonal of the matrix i.e. the distance of a cluster to itself is 0. Remember that EMD values do not inform you about the statistical significance of the results. Similar to any metric distance, an EMD value is not large or short by itself, but it is only meaningful when considered as a relative value to other distances. For example, an EMD value 20 is neither high or low, it is higher than 2 but lower than 40 and roughly similar to 20.5. Finally some statistical information, such as the histrogram, the minimum, the maximum, the range, the mean and the standard deviation, about the EMD distances obtained can be accessed in the histogram tab.

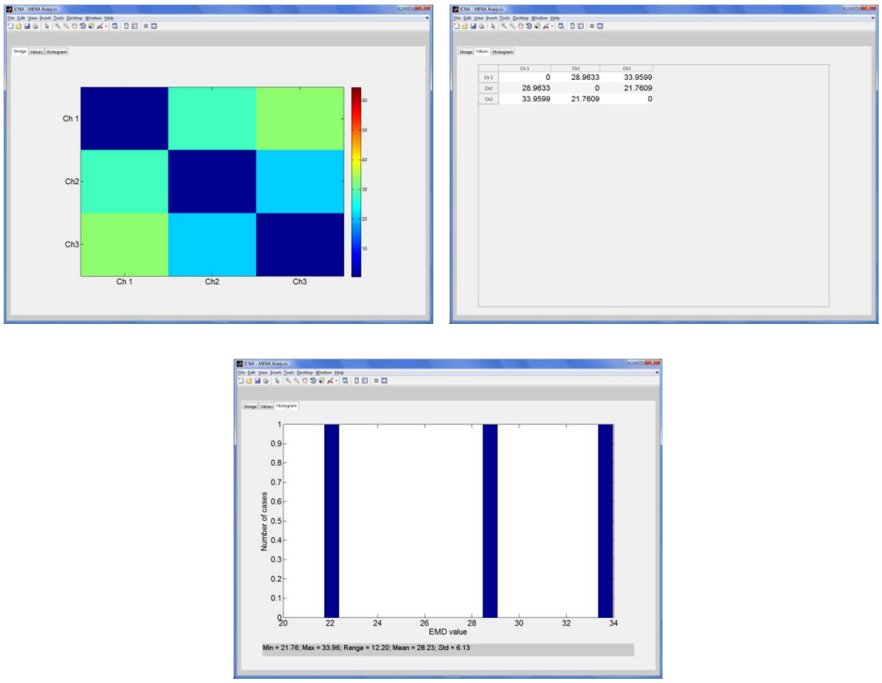


Figure 33 Example of the EMD results between three clusters

Advance use of MENA

This User Guide has presented the rudiments of using MENA through its user interface and ICNNA. The ICNNA GUI should be enough for most analysis purposes but in some specific occasions a finer control may be required. In those cases, ICNNA/MENA can be fully controlled using the command window if necessary. This however requires knowledge of MATLAB programming, and is beyond this User Guide. Please refer to the technical documentation for how to make use of ICNNA/MENA API.

Registration tool also released with ICNNA/MENA is not covered in this guide. Please execute `guiRegistration` to launch the registration tool.