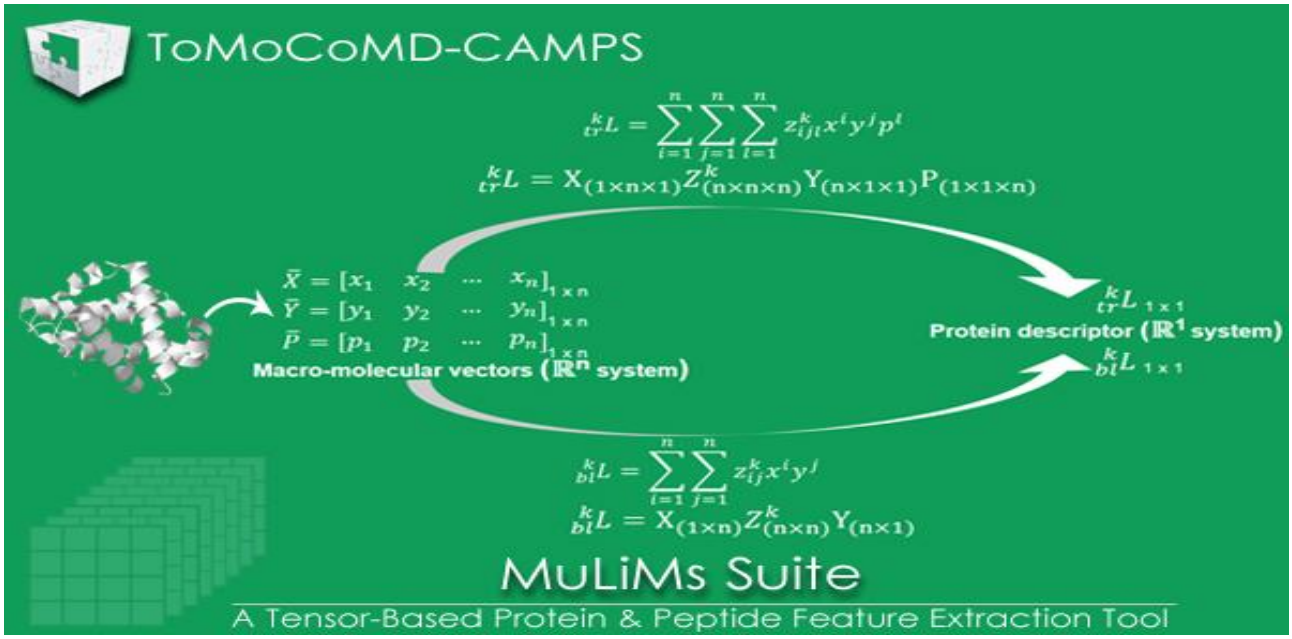
ToMoCoMD-CAMPS

a feature extraction tool

[Suite-Module]
MuLiMs-MCoMPAs v1.0



Software User Manual

ToMoCoMD-CAMPS is a biomacro-molecular (protein and peptide) descriptors computing software composed of one suite named MuLiMs (acronym of Multi-Linear Maps) with parallel functionalities. This suite includes a set of modules derived from algebraic strategies. MuLiMs suite consist of three modules: 1) ***MuLiMs-GCoMPAs*** (acronym of Multi-Linear Maps based on Graph-Theoretic Contact Matrices of 1/2D-Proteins and Amino-Acids Weightings), 2) ***MuLiMs-MCoMPAs*** (acronym of Multi-Linear Maps based on N-Metric and Contact Matrices of 3D-Protein and Amino-Acids Weightings) and 3) ***MuLiMs-SCoMPAs*** (acronym of Multi-Linear Maps based on Surface Contact Matrices of 3D-Protein and Amino-Acid Weightings). In this application, only ***MuLiMs-MCoMPAs*** module is included. ***MuLiMs-MCoMPAs*** constitutes a unique combination of methods for calculating tensor algebra-based 3D-MDs on a sound algebraic basis. These MDs can be used for a wide range of applications in structural bioinformatics and computational biology, in particular in the prediction of proteins biological properties and functions.

USER'S MANUAL

ToMoCoMD-CAMPS
MuLiMs-MCoMPAs v1.0

MuLiMs-MCoMPAs v1.0 is a program that calculates "novel 3D-protein/peptide descriptors based on Multi-Linear Algebraic Maps".

MOLECULAR AND TRASLATIONAL MEDICINE GROUP
Colegio de Ciencias de la Salud (COCSA)
Universidad San Francisco de Quito,
Quito, Ecuador.

July, **2019**

User's Manual
Authorization Memorandum

I have carefully assessed the User's Manual for **ToMoCoMD-CAMPS (MuLiMs-MCoMPAs)**.

MANAGEMENT CERTIFICATION - Please check the appropriate statement.

_____ The document is accepted.

_____ The document is accepted pending the changes noted.

_____ The document is not accepted.

We fully accept the changes as needed improvements and authorize initiation of work to proceed. Based on our authority and judgment, the continued operation of this system is authorized.

_____ _____
NAME DATE
Project Leader

_____ _____
NAME DATE
Operations Division Director

_____ _____
NAME DATE
Program Area/Sponsor Representative

_____ _____
NAME DATE
Program Area/Sponsor Director

USER'S MANUAL

TABLE OF CONTENTS

Page #

1.0 GENERAL INFORMATION

1.0 GENERAL INFORMATION

System Overview

ToMoCoMD-CAMPS is an interactive and user-friendly *free* multi-platform application designed to calculate 1/2/3-D numerical descriptors (indices) for biomacromolecular (protein and peptides) structures, with the objective of characterizing or discriminating among them. This software is comprised of one suite named MuLiMs (acronym of Multi-Linear Maps) with parallel functionalities. This suite consists of three modules derived from algebraic considerations: *1)* ***MuLiMs-GCoMPAs*** (acronym of Multi-Linear Maps based on Graph-Theoretic Contact Matrices of 1/2D-Proteins and Amino-Acids Weightings), *2)* ***MuLiMs-MCoMPAs*** (acronym of Multi-Linear Maps based on N-Metric and Contact Matrices of 3D-Protein and Amino-Acids Weightings) and 3) ***MuLiMs-SCoMPAs*** (acronym of Multi-Linear Maps based on Surface Contact Matrices of 3D-Protein and Amino-Acid Weightings).

In this application, only the MuLiMs-MCoMPAs module is included, which is for the calculation of 3D-biomacromolecular (protein and peptide) descriptors based on the two-linear (bilinear) and three-linear (multi-linear or N-linear) algebraic forms. Thus, is the unique software that computes these kinds of indices, establishing relations among two and three atoms, applying several (dis)similarity metrics or multi-metrics, matrix transformations (simple-stochastic and mutual probability), cut-offs (geometrical and/or topological), group-based calculations and aggregation operators. It was developed using the Java programming language and employs the Chemical Development Kit (CDK) and the Jmol libraries to manipulate the protein structures and visualize the protein structures, respectively. This software is composed by a desktop user-friendly interface and an API library. The GUI was created to ease to the users the configuration of the different options of the biomacromolecular descriptors, while the library was designed to be easily integrated in other software as descriptor calculation component. The calculation of the bio macromolecular descriptors was parallelized using a divide-and-conquer approach to reduce the processing time.

This software has some relevant features, such as:

1) One chemical input format Protein Data Bank (PDB) files.
2) A sub module for the preparation of biomacromolecular datasets to perform calculations (removing hydrogen and HET atoms). This sub module also allows the selection of the model in the case of Nuclear Magnetic Resonance (NMR) experimental data and the chain (s) of interest to the investigator.
3) Four 3D-protein structural representations: Alpha-Carbon, Beta-Carbon, Amide-Bond Carbon and Average to compute descriptors.
4) Visualization of the selected configuration in both plain text (or in 3D) using the jMol library.
5) Sixteen properties as amino acid weightings: side-chain Mass, side-chain Volume, z-scales and son on.
6) Three matrix representations: non-, simple-stochastic and mutual probabilistic.
7) One new matrix form by using cutoffs based on geometrical (Lag R) and/or topological (Lag L) values.
8) Thirty local indices classified into in three categories: *Amino-Acid type*: Apolar, Polar positively charged, Polar negatively charged and so on, *Secondary Structure Preference*:

Alpha helix favoring Amino-Acids, Beta-Sheets favoring Amino-Acids and so on and *R group* (one for each α-aminoacid) including Alanine, Arginine and so on.

9) Three output file formats: Space Delimited Text file, Weka ARFF file and Comma Separated Values file.
10) Notification and information on system error and program exceptions of JRE.
11) Real-time updated logging status (see History Tab windows).
12) Optional generation of Debug Report file.
13) Distances to protein center are computed as diagonal matrix elements.
14) Included twenty-seven aggregation operators (invariants) that generalize the initial form of obtaining indices from atomic (or fragment) contributions, the new indices are obtained by using these invariants on LAIs (Local Amino Acid Invariants).
15) Three protein datasets are provided in the Example Data tool.
16) Brand new Descriptor Search Tool.
17) Include suggested theoretical configurations (XML projects).
18) Enhanced speed for descriptor calculation process with more stability and robustness.
19) A command-line interface (CLI).

System requirements

MuLiMs-MCoMPAs software runs on a wide variety of operating systems and computers including multi-processor clusters, multi-processor or multi-core desktops (PC and MAC), high-performance scientific workstations, and laptops. This release can run either interactively or in batch mode, which permits sequential execution to be distributed across multiple processors (and/or cores) workstations, even in a heterogeneous computing environment. In general terms the minimal and recommended system requirements are:

Hardware:

Processor: All processors developed hereafter by Intel Corp. are supported on the assembly level optimization. All AMD current processors work as old Pentium with higher clock frequency (no special optimization).

Processor Clock Speed: minimum Intel(R) Celeron(R) M processor 1.40GHz or equivalent. Recommended Intel(R) Core2Quad processor 2.5GHz or above.

Memory: 256MB minimum, 512MB default tuning. We strongly recommend 4096 MB or above in order to improve performance.

Software:

Operation system: **ToMoCoMD-CAMPS** is designed to run on any UNIX/LINUX or MAC platforms, as well as on microcomputers running Windows 95, 98, ME, 2000 or XP, Vista, 7, 8, 10 and above. **ToMoCoMD-CAMPS** is platform-independent software.

Operation system extensions: **ToMoCoMD-CAMPS** requires Java(TM) 7 Runtime Environment or above on the target system. It runs under any host operating system, which supports Java(TM) 7 Runtime Environment or above and also works on x86 and x64 based architectures.

Points of Contact

Information

For all comments, suggestions, information, and inquiries about **ToMoCoMD-CAMPS MuLiMs-MCoMPAs** Software please contact:

Prof. Yovani Marrero Ponce, PhD
Colegio de Ciencias de la Salud (COCSA)
Universidad San Francisco de Quito, Quito, Ecuador.
E-mail: ymarrero77@yahoo.es
URL: http://www.uv.es/yoma/

Technical Support

For technical support please contact to:

Prof. Ernesto Contreras Torres, MsC
Grupo de Medicina Molecular y Traslacional (MeM&T)
Universidad San Francisco de Quito, Quito, Ecuador
E-mail: econtrerastorres88@gmail.com

Prof. César Raúl García Jacas, PhD
Departamento de Ciencias de la Computación,
Centro de Investigación Científica y de Educación Superior de Ensenada (CICESE),
Ensenada, Baja California, México.
E-mail: cesarrjacas1985@gmail.com

2.0 SYSTEM SUMMARY

2.0 SYSTEM SUMMARY

System Configuration

The system is prepared to maintain its default configuration regardless of the platform on which is executed. It does not require any parameters or initial configuration file, so it fits natively over the Java™ virtual machine.

The configuration process to start performing calculations of algebraic form descriptors with this application begin on the *"N-Tuple"* panel, located under the tabbed pane menu or simply can be loaded from a preconfigured MuLiMs-MCoMPAs's project file.

Installation of the program

ToMoCoMD-CAMPS MuLiMs suite is available in two different forms, these are:

a) A Java™ portable GUI-based application: *MuLiMs.jar*, for users that frequently use different workstations. No matter the operating system or workstation hardware configuration, ToMoCoMD-CAMPS MuLiMs users always will have this software to one click away.
b) A Java™ portable CLI-based application: *MuLiMs.jar*.

3.0 GETTING STARTED

3.0 GETTING STARTED

This section provides a general walkthrough of the system from initiation through exit.

Loading application

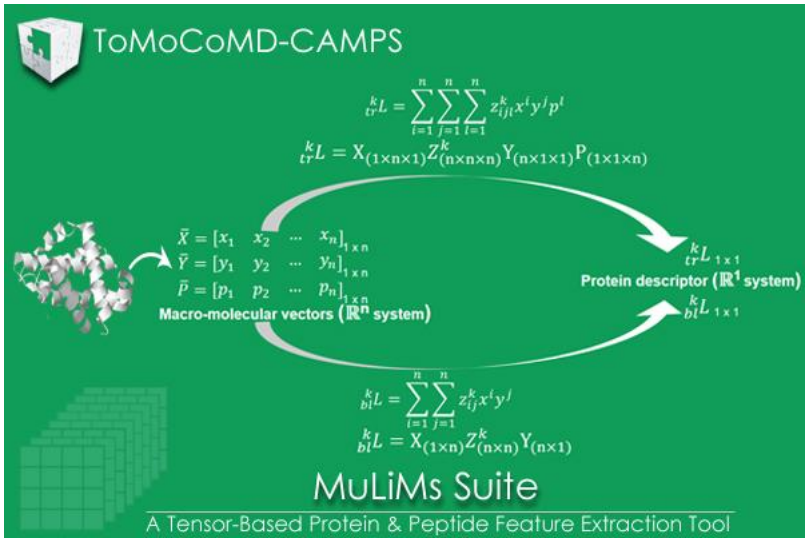


Figure 1. Loading SplashScreen for ToMoCoMD-CAMPS (MuLiMs suite)

The software does not require any additional information to login or warm up, as soon as you execute the main program the Splash Screen is launched instantly.

MuLiMs-MCoMPAs Graphical User Interface (GUI)

*The **MuLiMs-MCoMPAs** GUI has the following screen areas*:

- **Title Bar: Bears the title of program,** ToMoCoMD-CAMPS MuLiMs.
- **Menu Bar:** Menus related to different tasks performed by ToMoCoMD-CAMPS MuLiMs-MCoMPAs.

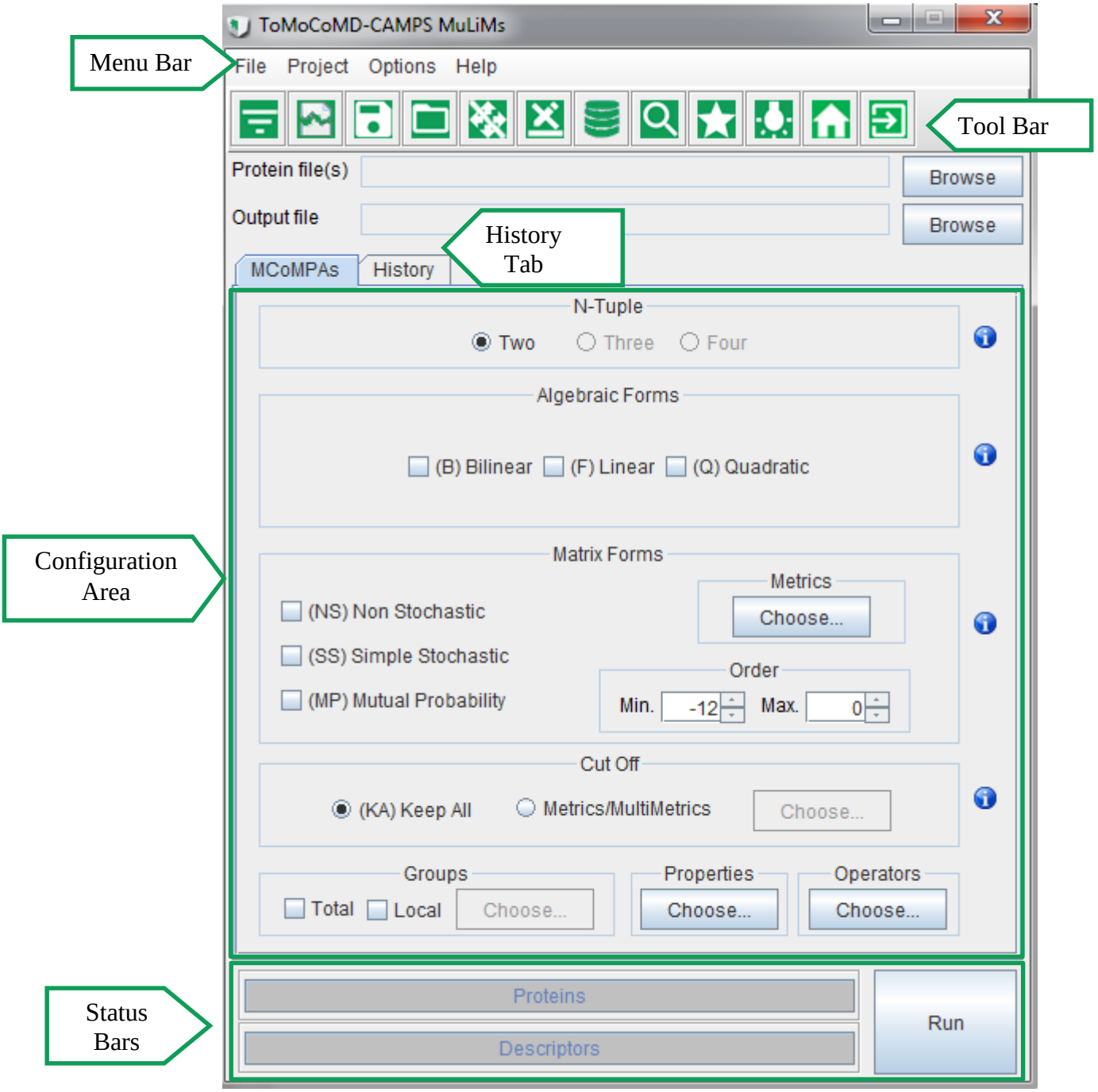


Figure 2. The MuLiMs-MCoMPAs main GUI

- **Tool Bar:** Quick access shortcuts to commonly performed tasks, displayed as graphical icons instead of classical menu items.
- **Status Bar:** Shows the current and remaining proteins and descriptors.
- **Configuration Area:** This is the *main client area*, which contains MDs pane (Algebraic Form descriptors configuration parameters).
- **History Window:** Logging windows for all operations and tasks.

System Menu Bar

This section describes in general terms the system menu first encountered by the user, as well as the navigation paths to functions noted on the screen. Each system function should be under a separate section header.

Figure 1. System Menu Bar

File menu commands

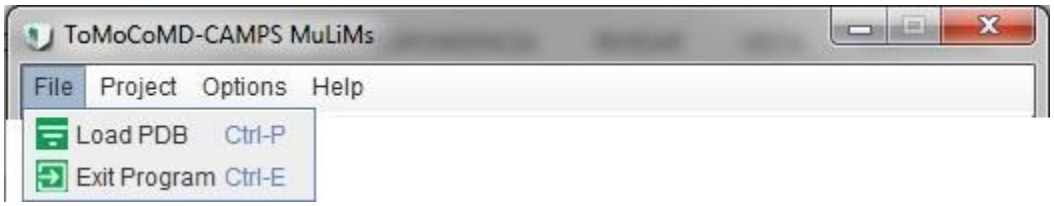


Figure 2. The File menu

Load PDB

Open an input dialog to load PDB files.

Exit Program

Safely close the application with saving option prompt (terminates the current **MuLiMs** session).

Project menu commands

Commands of the *Project menu* allow the user to create configurations and to open and edit existing project files.

Figure 3. The Project menu

New

Creates a brand new empty configuration or Resets the current configuration options of the Algebraic Form Panel.

Save

Export the current configuration and options to a persistent Project Configuration File.

Load

Import configuration and options from a Project Configuration File. That is to say, opens and edit existing project files.

Options menu commands

Commands of the *Options menu* enable the user to set up the next molecular descriptors computation.

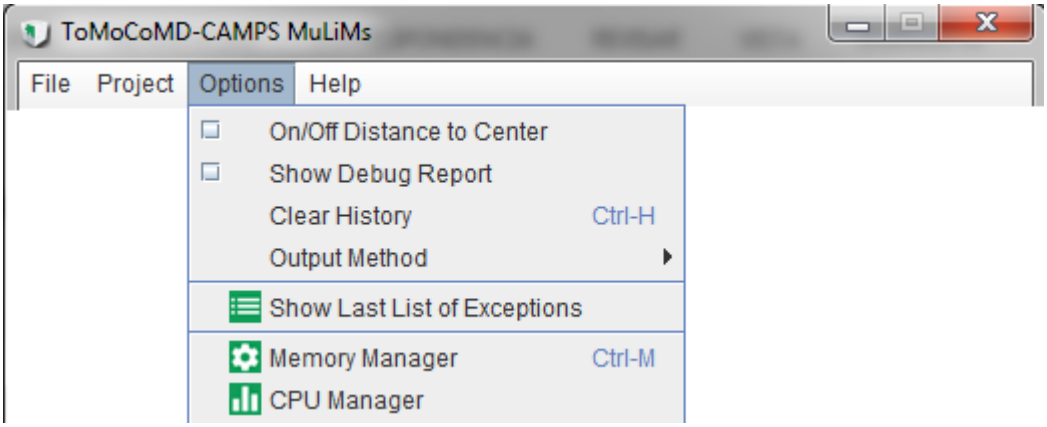


Figure 4. The Option menu

On/Off Distance to Center

Activates or deactivates the use of the (dis)similarity of each amino acid to the protein center in the computation of the tensors.

Show Debug Report

If you check this option, the program generates a new text file with all information concerning the algebraic process that takes place in the calculation.

Clear History

If you press this item, the program cleans the history window.

Output Method

Display the available options to format resulting file with calculations of indices, one can only select one option at a time.

Show Last List of Exceptions

Display the exceptions occurred during the calculation of the biomacromolecular indices.

Memory manager

Display a window where is shown the Random Access Memory (RAM) (measured in Megabytes (MB)) employed by the program.

CPU manager

Display a window to specify the number of CPU cores to use in calculations.

Help menu commands

Moreover, the **MuLiMs-MCoMPAs** main window contains some icons which can be clicked in order to obtain specific information. The *Help menu* contains the following commands:

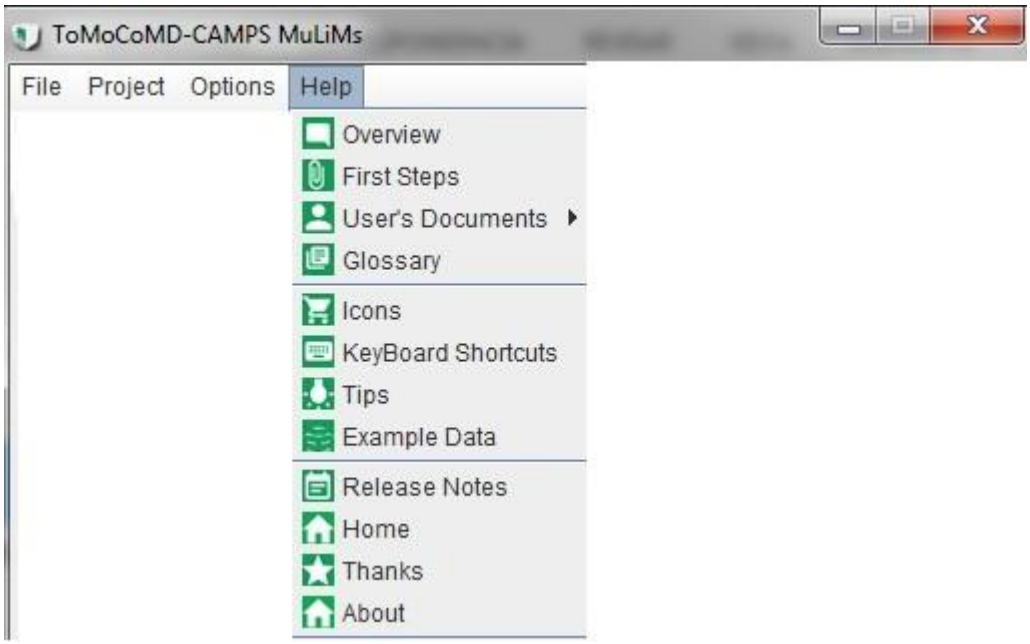


Figure 7. The Help menu

Overviews

Shows an illustrative procedure of how to execute, configure and use the program. MuLiMs module is based on the Chemistry Development Kit (CDK) library.

About the Chemistry Development Kit (CDK).

The CDK an open-source library of algorithms for structural chemo- and bio-informatics, implemented in the programming language Java. It serves as a base for many other applications, including some parts of **MuLiMs** software. For information about CDK, please visit the CDK home page. The CDK library is published under terms of the GNU Lesser General Public License. This project is hosted under http://cdk.sourceforge.net.

Copyright: The CDK is copyrighted by the CDK project, and has been written by Rich Apodaca, Ulrich Bauer, Miguel Rojas Cherto, Fabian Dortu, Martin Eklund, Matteo Floris, Dan Gezelter, Uli Fechner, Rajarshi Guha, Yonquan Han, Thierry Hanser, Tobias Helmus, Kai Hartmann, Christian Hoppe, Oliver Horlacher, Miguel Howard, Violeta Labarta, Nina

Jeliazkova, Geert Josten, Anatoli Krassavine, Stefan Kuhn, Daniel Leidert, Edgar Luttmann, NathanaÃ«l Mazuir, Stephan Michels, Peter Murray-Rust, Irilenia Nobeli, Chris Pudney, Jonathan Rienstra-Kiracofe, David Robinson, Bhupinder Sandhu, Jean-Sebastien Senecal, Sulev Sild, Bradley Smith, Christoph Steinbeck, Stephan Tomkinson, Joerg Wegner, Stephane Werner, Egon Willighagen, and Yong Zhang.

First Steps

Introduce a first time **ToMoCoMD-CAMPS** user into a general overview, system requirements, input and output file modes.

User's Documents

Present three sub-menus related to the MuLiMs-MCoMPAs 3D-Descriptors: Acronyms, Theory and the User's Manual corresponding to this application.

Glossary

Basic **MuLiMs** biomacromolecular descriptors terminology is provided in a tool. This terminology is provided in a window that the user can keep open to be supported through the **MuLiMs** indices setup.

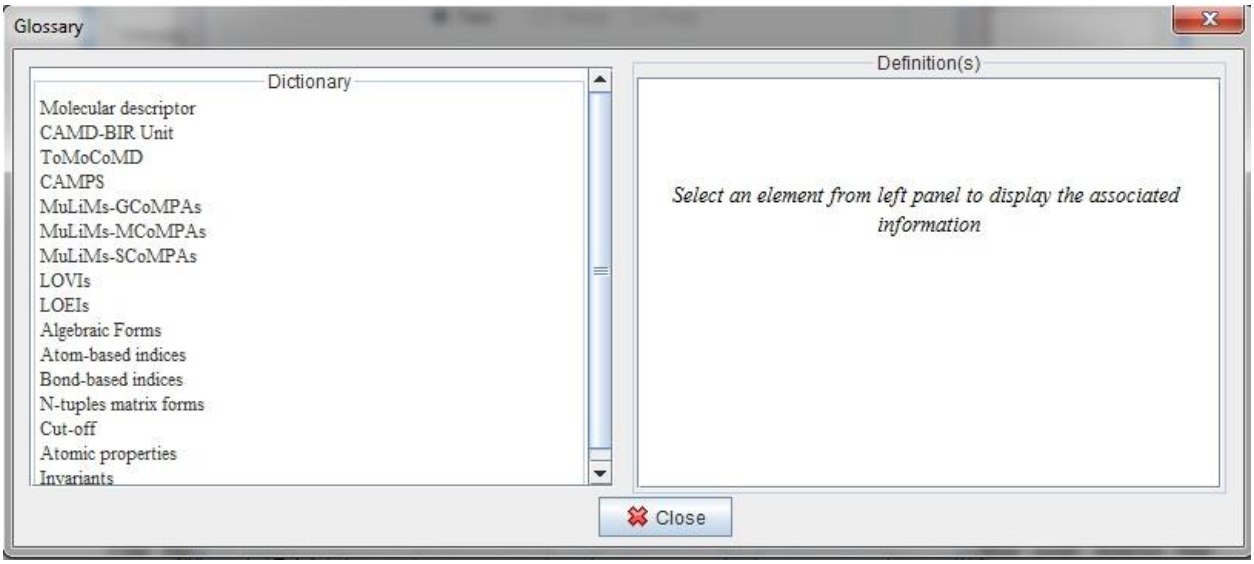


Figure 8. Terms and Concepts Glossary Tool.

Icons

The functionality of all Icon used in the program is described, so that the user learns the meaning of **ToMoCoMD-CAMPS's** icons naturally.

Keyboard Shortcuts

Describe all keyboard accelerators used in the program. These keyboard shortcuts perform the specific commands or replace the equivalent menu items.

Tips

Most frequent first user questions are answered in this section, with 11 tips that provide instant technical support available any time you execute this program.

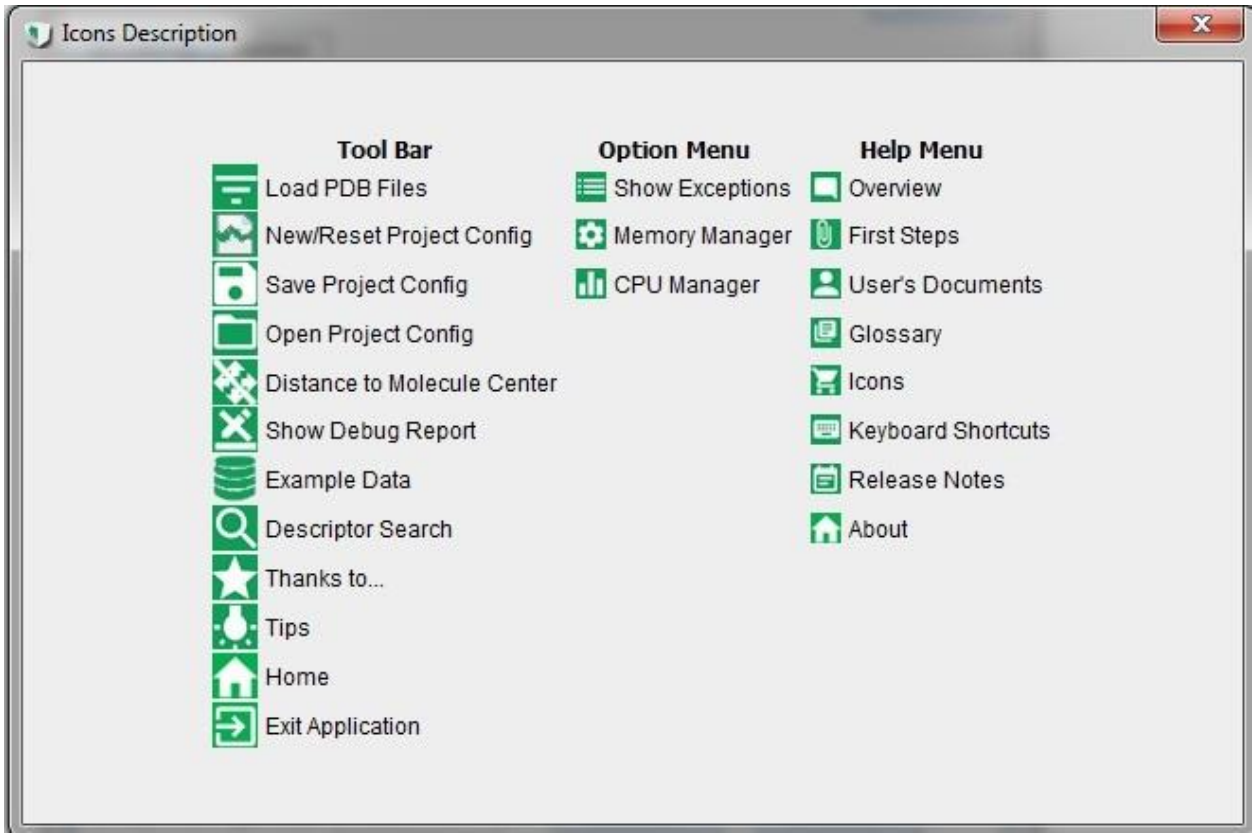


Figure 9. Icons description.

Example Data

The Example Data Tool is a key element for **ToMoCoMD-CAMPS's** first users, in order to test the MDs calculated by this software, three datasets of are provided. Click the example data icon in the tool bar to access these datasets. These datasets will permit to make simple test calculations. To perform descriptor calculations, click the respective checkboxes to select the desired configuration options.

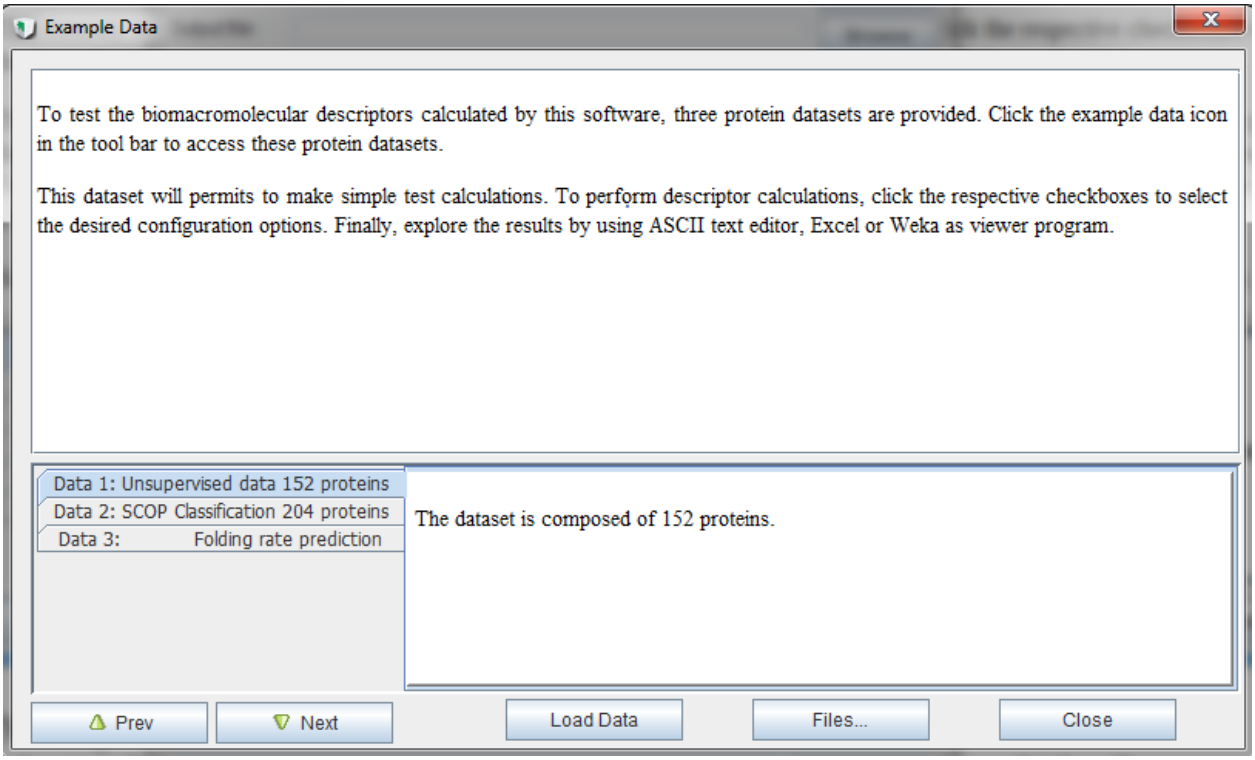


Figure 10. Example Data Tool.

Release Notes

Show the track changes since **MuLiMs-MCoMPAs** was a whiteboard idea.

Home

Bring useful information about **CAMD-BIR Unit**, how to contact us and cite.

Thanks

Recognition to different contributions to the success of this project is offered.

About

Several information about **MuLiMs-MCoMPAs** software and publications.

Tool Bar

Quick access shortcuts for most relevant option and tools. That is, the toolbar icons replace the most important and frequently used **MuLiMs** menu commands. Clicking on toolbar icons enables the user to perform the following commands:

Figure 11. Tool Bar elements.

1. Load PDB.
2. New

3. Save
4. Open
5. On/Off Distance to Protein Center
6. On/Off Generate Debug Report
7. Launch Example data
8. Descriptor Search Tool
9. Thanks
10. Tips
11. Home
12. Exit Program

Descriptor Search

ToMoCoMD-CAMPS includes an optional use tool developed to automatically decode headers assigned to each one of the biomacromolecular indices. *See the picture below*.

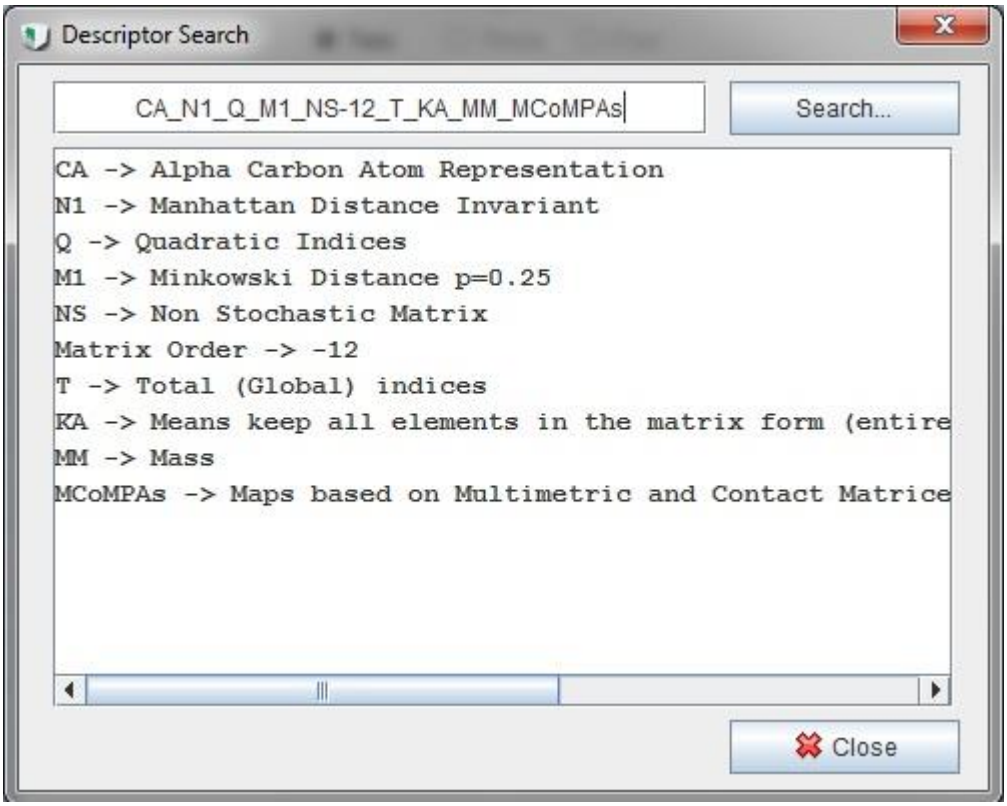


Figure 12. Dialog Search Window.

Status Bar

The status bar located at the bottom of the main window and it shows the protein file name, the algebraic descriptor that is being calculated and the percentage of completion, also displays the name and number for proteins and descriptors.

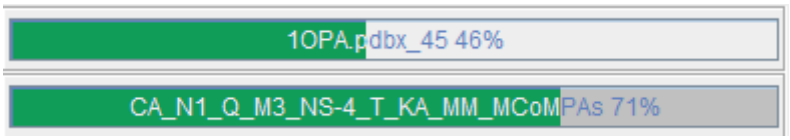


Figure 13. Status Bar.

Configuration Area

This area has seven sub-areas (panes) for MDs configuration. In each pane appears an info button (blue circle) that contains a short description about the theory associated to each part (for more details see **Starting MuLiMs-MCoMPAs** and **Configuring a project** sections).

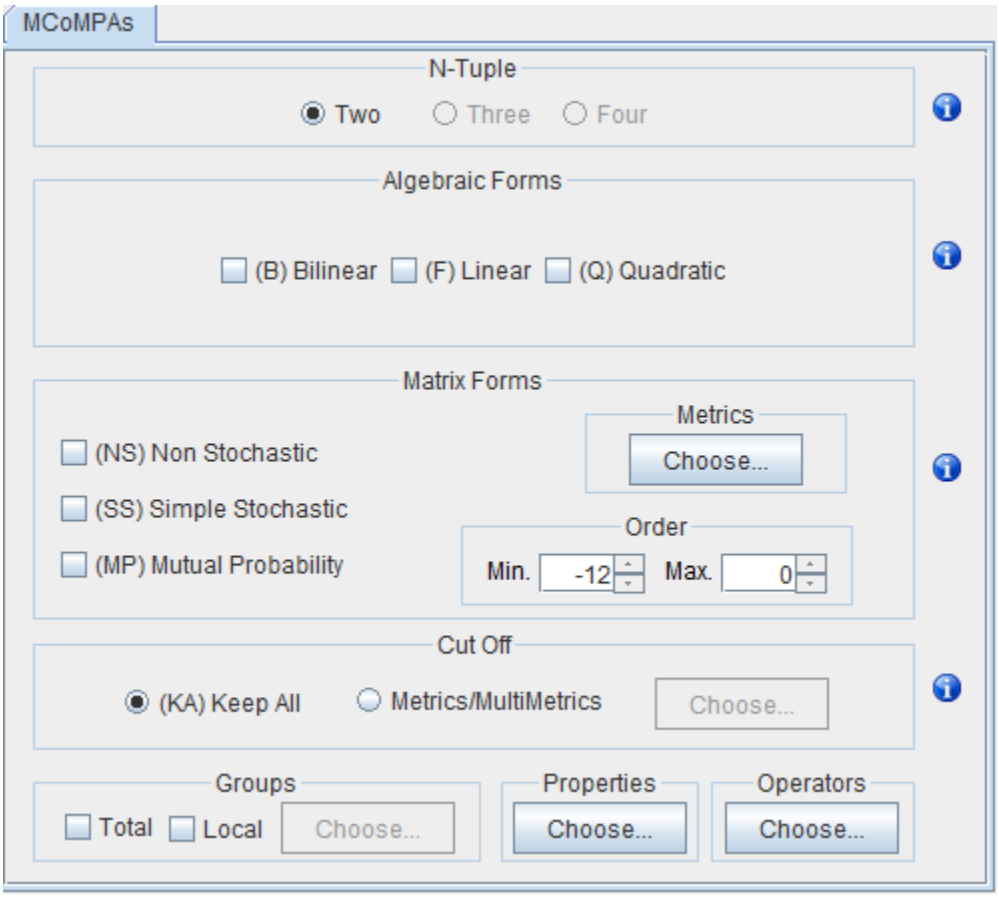


Figure 14. Algebraic Descriptors Configuration Area.

History Window

Logging windows for all operations and task. Besides, after the calculation is finished, the **tab History** (**log file**) shows some details and statistics of the calculation process.

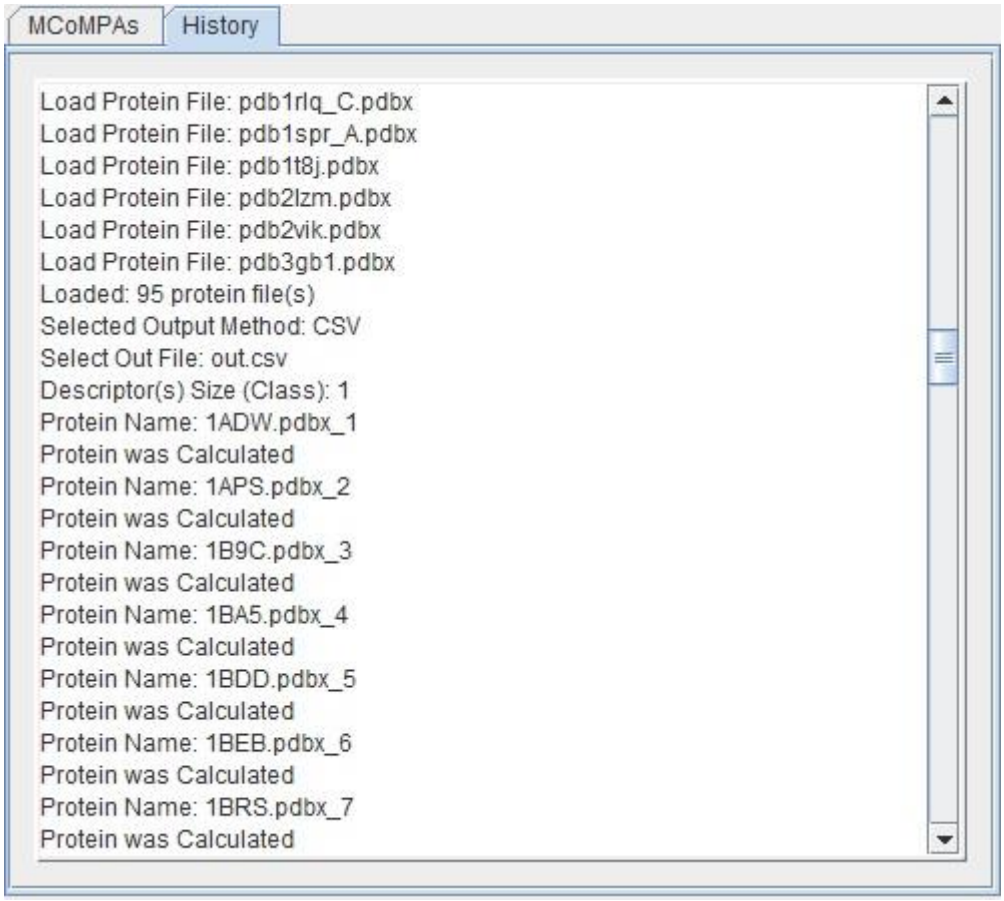


Figure 15. Logging (History) windows.

Exit System

Describe the actions necessary to properly exit the system.

Figure 16. Program Exit Options.

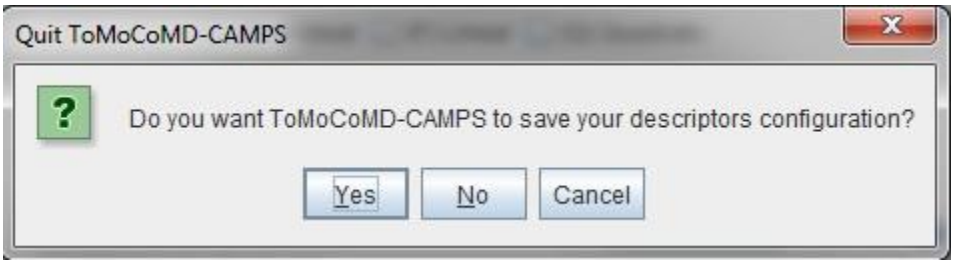


Figure 17. Safe Exit Action prompt.

4.0 USING THE SYSTEM

4.0 USING THE SYSTEM

This section provides a detailed description of the **MuLiMs-MCoMPAs** software from the initial to the final steps, explaining in detail the characteristics of the required input and system-produced output. It covers both calculations of single and multiple molecular datasets and batch mode calculations. Each **MuLiMs-MCoMPAs** function is under a separate section header, and corresponds sequentially to the system functions (menu items) listed in subsections of chapters above.

What you need to know before using MuLiMs-MCoMPAs

To make use of MuLiMs-MCoMPAs calculations, you must generate the 3D-protein representation (s) provided by this software.

MuLiMs-MCoMPAs is not designed as QSAR software; it provides only molecular indices and does not perform QSAR analysis. However, by **MuLiMs-MCoMPAs** it is possible to merge calculated molecular descriptors and user-defined properties for a set of molecules, providing a complete output file which is easily loaded by any correlation analysis application.

MuLiMs-MCoMPAs provides a total of **2 016 740 992** molecular descriptors based on amino acid-pairs relations, **8 142 627 840** molecular descriptors based on ternary relations. In addition, in the **MuLiMs-MCoMPAs software** the 2-tuple cut-off could be used to considerer the most important non-covalent interactions in a biomacromolecular structure according to criteria based on topological and/or geometric distances, as well as, to take into account the distance to protein center as diagonal elements of the matrix forms.

Starting MuLiMs-MCoMPAs

MuLiMs-MCoMPAs is launched by clicking on the configuration files (e.g. .bat files on Microsoft Windows platforms) that are provided. These files allow to improve the performance and speed. These are tweaks for the Java™ VM (JVM), that increase the maximum default limit of JVM heap memory. These preconfigured scripts are located in the root directory of MuLiMs-MCoMPAs program folder. The configurations of JVM heap memory limit are:

- 1 GB
- 2 GB
- 4 GB
- 8 GB
- 16 GB
- 32 GB

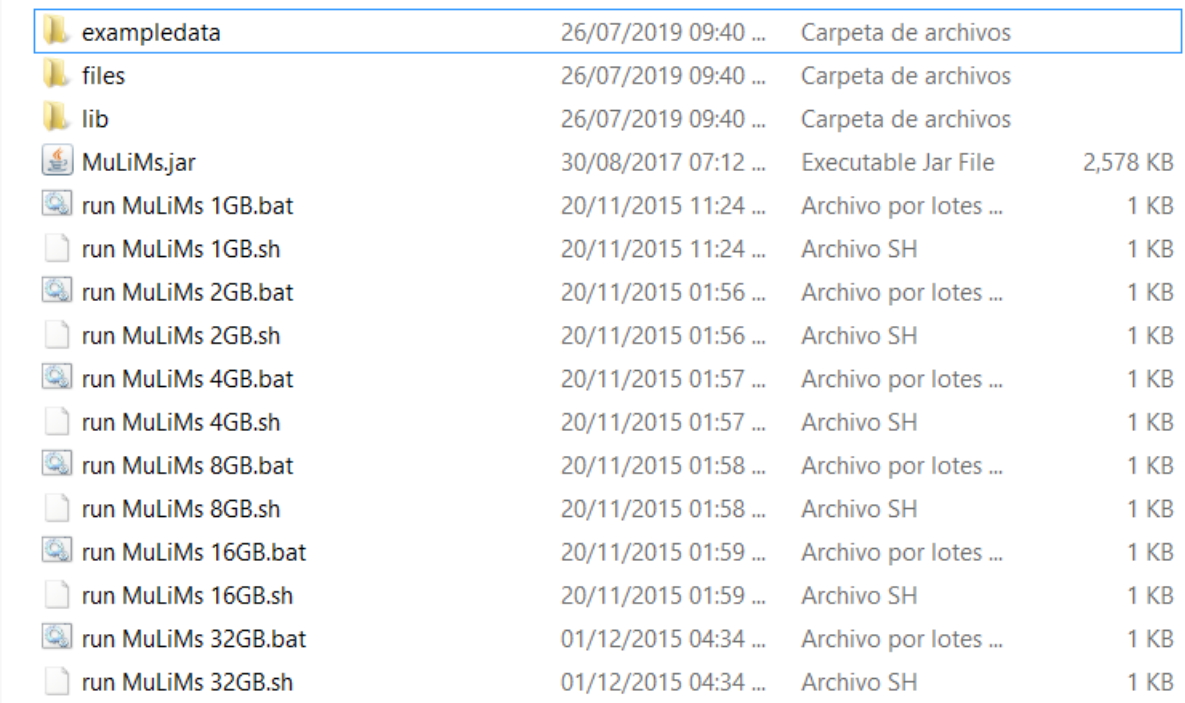


Figure 18. MuLiMs-MCoMPAs Batch Files (.bat or .sh).

Indeed, for each heap memory limit, a command line scripts were targeted for two different kinds of platform:

- Windows Batch File *(.bat*)
- Linux Shell Script *(.sh*).

Otherwise, if the preconfigured command line scripts do not suit your hardware preferences, users can modify the scripts for both platforms to adjust the program JVM heap memory limit according to their system hardware properties, editing these scripts with a text editor program, (i.e. *NotePad* or *WordPad* in Windows, and *GEdit* or *Vi* in Linux). The following example limits de JVM heap memory up to 1024 megabytes:

```
java -Xms256m -Xmx1024m -jar MuLiMs.jar
```

After splash, the main window (GUI) will be displayed on the screen. The follow step is selecting the ***structure input*** and ***descriptors output*** files by pressing the browse button in the *Input and Output* section in the upper right part of the GUI. Next, the user can select the desired descriptors for calculation (see configuring a new project below). Finally, the button "**Starts!"** begins the calculation of the selected descriptors for the structures in the input file(s). After the calculation is finished, an external dialog window appears that shows a message about the successful calculation. This message can be closed by pressing the **Ok** button. Then, the **Exceptions Window** is come into view. This window depicts the list of molecules with structure errors. In addition, the **History Tab** (see *Logging Window*) shows some details and statistics of the run.

Starting a new project

The **New Option** clean all parameters that are used during a **MuLiMs-MCoMPAs** session, *e.g.*, N-tuple, algebraic forms, matrix forms, and so on.

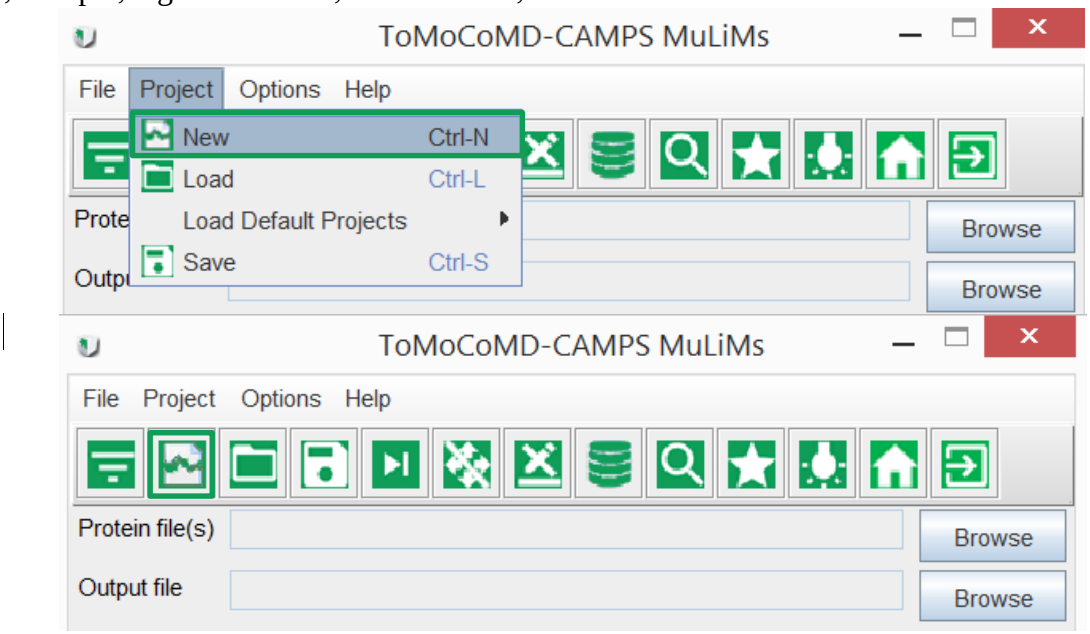


Figure 19. Creating a new project.

Loading a Project File

Project files can be reloaded in order to restore all parameters in a later session or used to execute **MuLiMs-MCoMPAs** in batch mode.

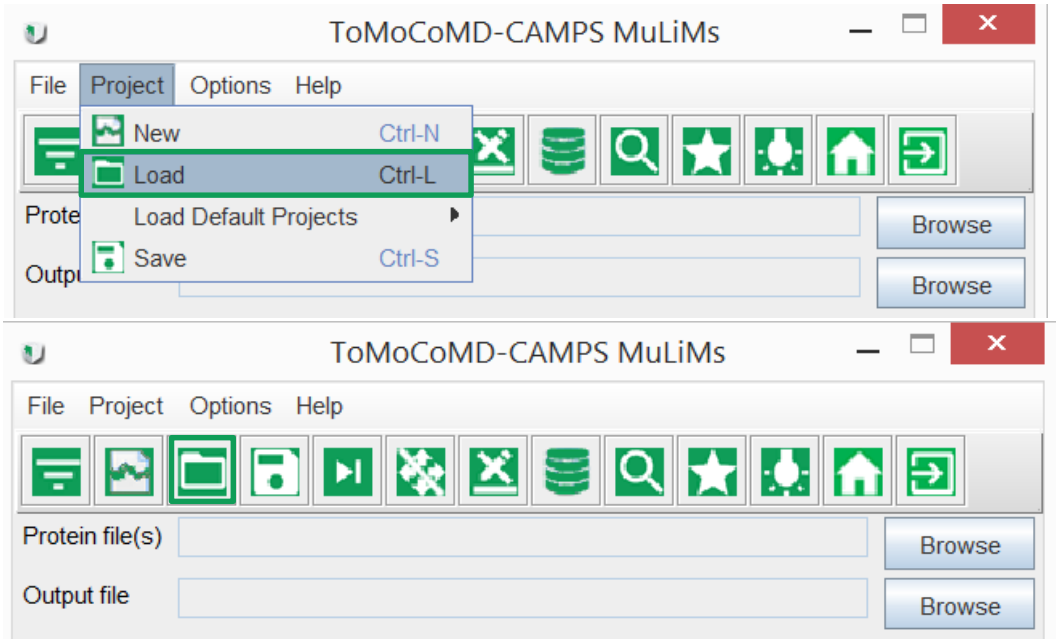


Figure 20. Loading projects button.

Saving a project file

A project file is saved in the menu **Project** in the main menu bar by the menu item **Save**. A dialog box appears where the path and the file name of the project file can be specified.

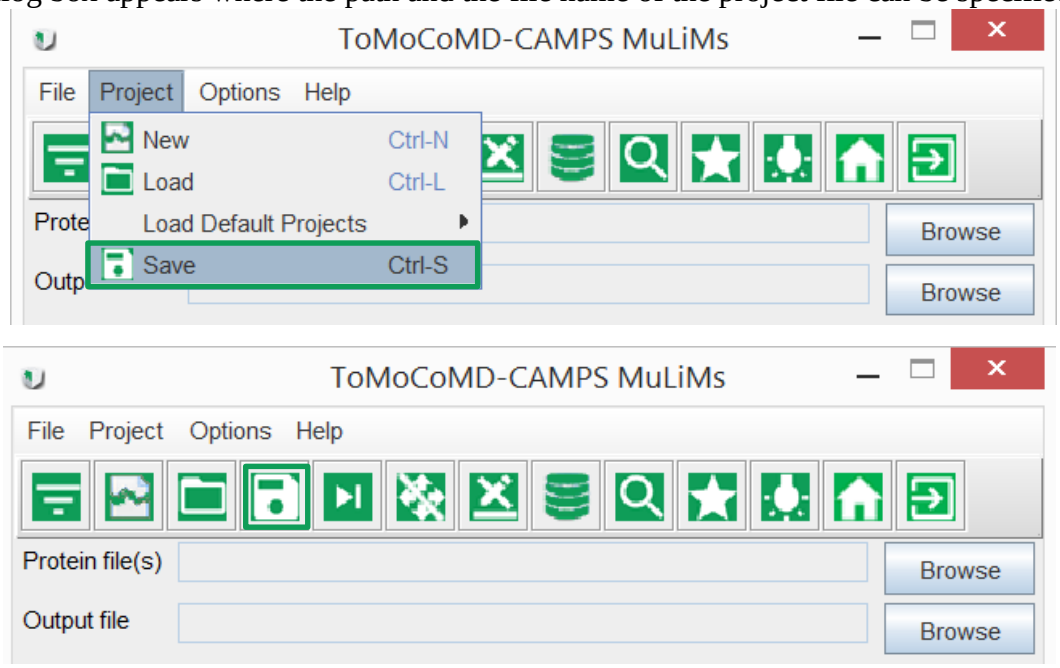


Figure 21. Saving project configuration.

Running a saved project

The previously saved project can be reloaded in order to restore all parts and parameters of a previous session. Finally, the user can run the calculation by clicking the button **Run**. After the calculation is finished, an external dialog window appears that shows a message about successful completion of the calculation. This message can be closed by pressing the **Ok** button.

Program Run Options

The following is a description of the **Calculate** section of the **MuLiMs-MCoMPAs** GUI. The Calculate section consists of the three buttons **Run/ Cancel**, **Exception Window** and **History Tab**. The button **Run/ Cancel** begins the calculation of the selected descriptors. When the calculation is started the **Run** button switches to **Cancel**, allowing it to stop the process. Unless the **Cancel** button is not pressed or the calculation is finished, all remaining buttons and menus are enabled, in case user needs to review the descriptor configuration or access the *Tool Bar* and *Menu Bar* options. In addition, a progress bar appears that shows the protein and the algebraic descriptor that is being calculated at a given time and the percentage of completion, also the name and number for proteins and descriptors are displayed. When the calculation is finished a message appears on the screen that displays "***The process was successfully finished.***"

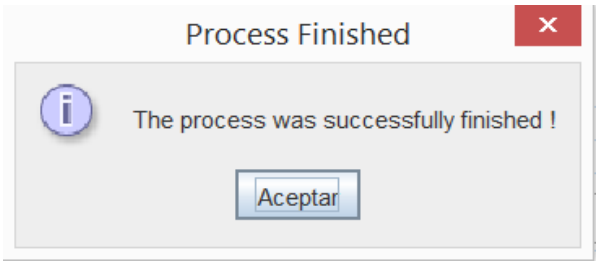


Figure 22. Task finished confirmation window.

Configuring a project

The descriptors section *(Configuration Area)* in the middle part of the **MuLiMs-MCoMPAs** GUI consists of seven different sub-areas. Also, one parameter can be configured by the user out of this area, this is: Distance to Protein Center.

The previous options can be saved in the Project file and are likewise available in the Options Menu and Tool Bars. In each sub-area the user can select the possible parameters. Only in 4 sub-areas is necessary to open a window to select the possible options, namely multi-metric (dis)similarity measures (Figure 41), local indices (Figure 42), amino acid properties (Figure 43) and Aggregation Operators (Figure 44).

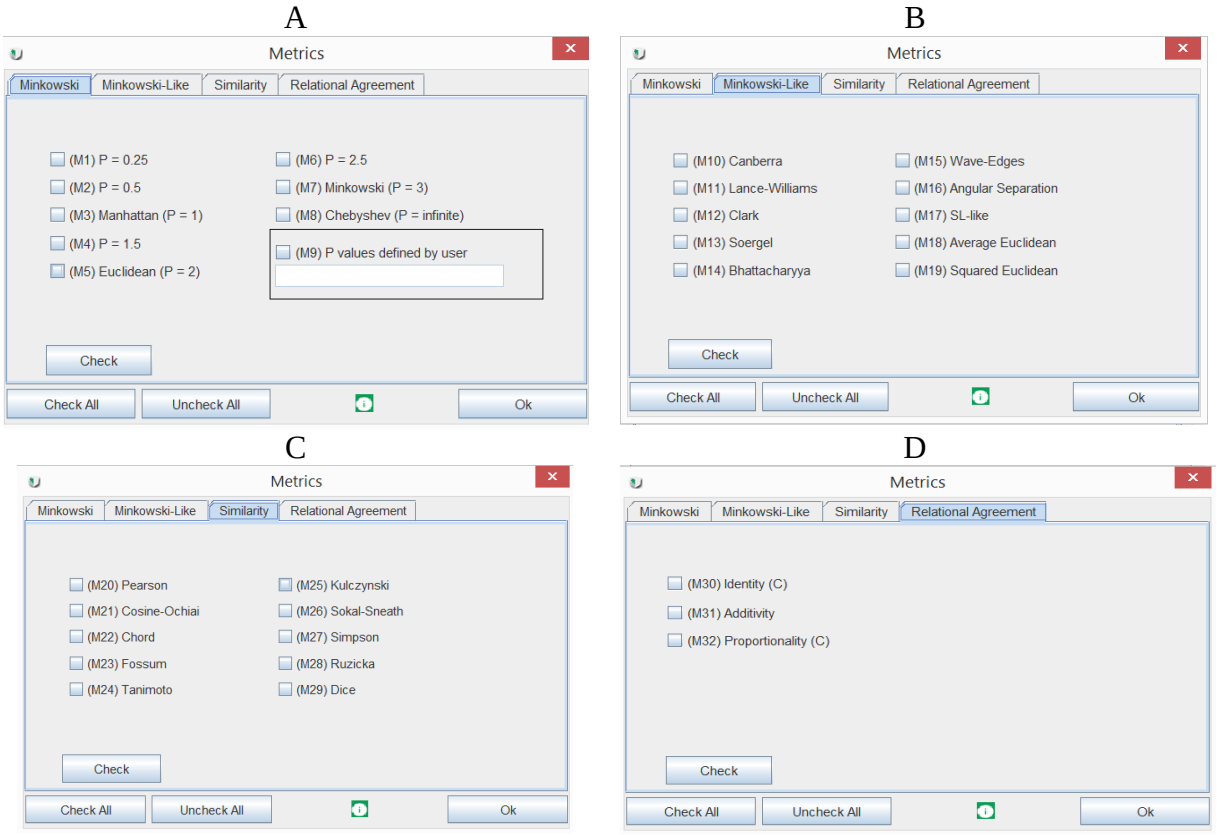
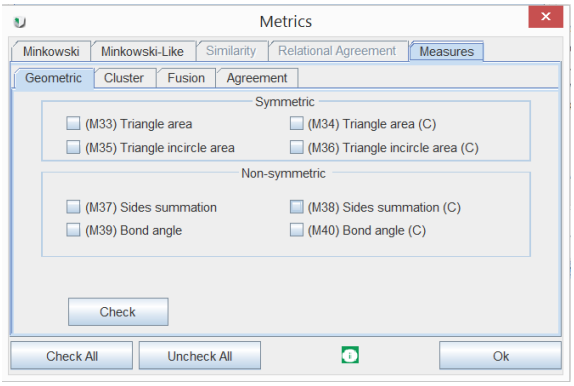
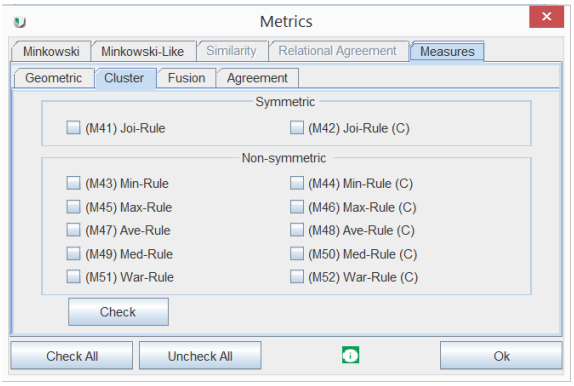


Figure 23. Dialog windows to set the (dis)similarity measures to compute the relations among 2 (A, B, C, D) and 3 (E, F, G, H) amino acids

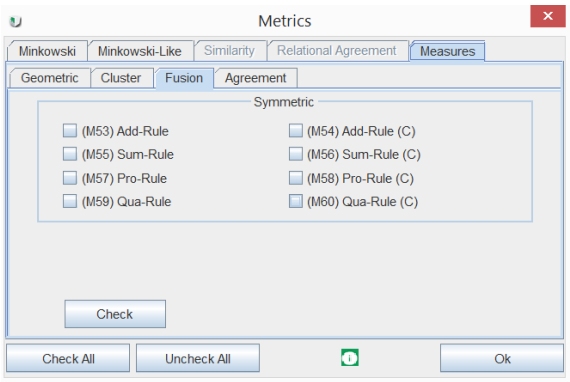
E


Metrics
Minkowski
Minkowski-Like
Similarity
Relational Agreement
Measures
Geometric
Cluster
Fusion
Agreement
Symmetric
(M33) Triangle area
(M34) Triangle area (C)
(M35) Triangle incircle area
(M36) Triangle incircle area (C)
Non-symmetric
(M37) Sides summation
(M38) Sides summation (C)
(M39) Bond angle
(M40) Bond angle (C)
Check
Check All
Uncheck All
Ok

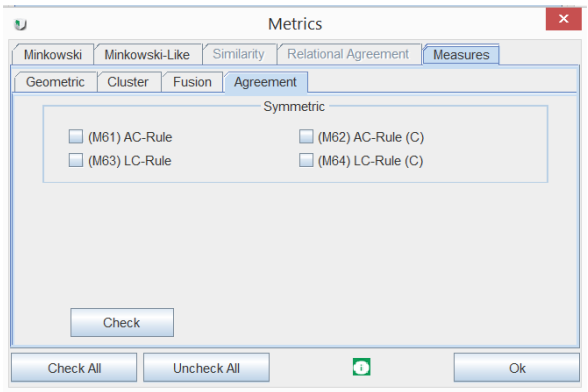
F


Metrics
Minkowski
Minkowski-Like
Similarity
Relational Agreement
Measures
Geometric
Cluster
Fusion
Agreement
Symmetric
(M41) Joi-Rule
(M42) Joi-Rule (C)
Non-symmetric
(M43) Min-Rule
(M44) Min-Rule (C)
(M45) Max-Rule
(M46) Max-Rule (C)
(M47) Ave-Rule
(M48) Ave-Rule (C)
(M49) Med-Rule
(M50) Med-Rule (C)
(M51) War-Rule
(M52) War-Rule (C)
Check
Check All
Uncheck All
Ok

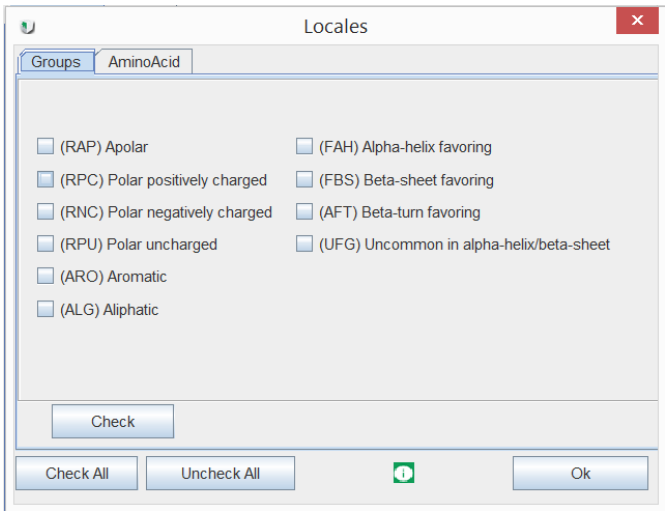
G


Metrics
Minkowski
Minkowski-Like
Similarity
Relational Agreement
Measures
Geometric
Cluster
Fusion
Agreement
Symmetric
(M53) Add-Rule
(M54) Add-Rule (C)
(M55) Sum-Rule
(M56) Sum-Rule (C)
(M57) Pro-Rule
(M58) Pro-Rule (C)
(M59) Qua-Rule
(M60) Qua-Rule (C)
Check
Check All
Uncheck All
Ok

H


Metrics
Minkowski
Minkowski-Like
Similarity
Relational Agreement
Measures
Geometric
Cluster
Fusion
Agreement
Symmetric
(M61) AC-Rule
(M62) AC-Rule (C)
(M63) LC-Rule
(M64) LC-Rule (C)
Check
Check All
Uncheck All
Ok

A



B

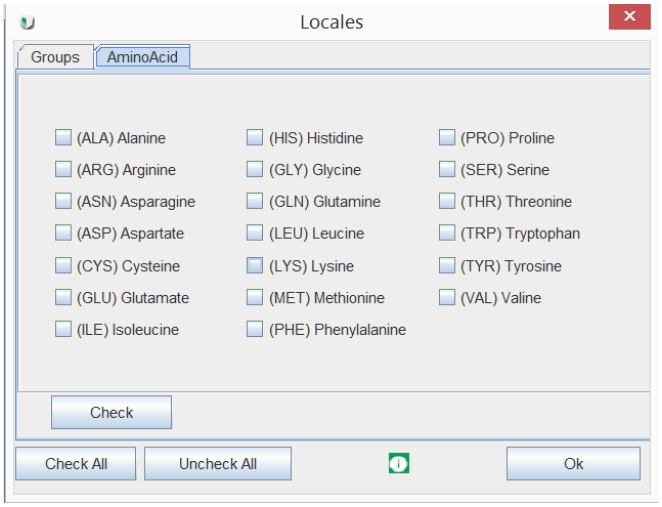


Figure 24. Locales. A) Groups, B) Amino acids.

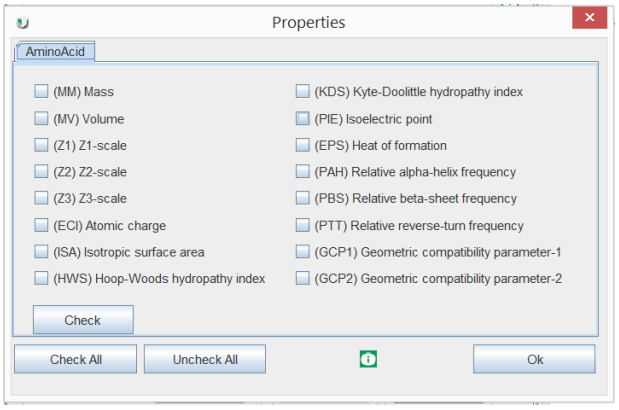


Figure 25. Amino acid-properties (labels)

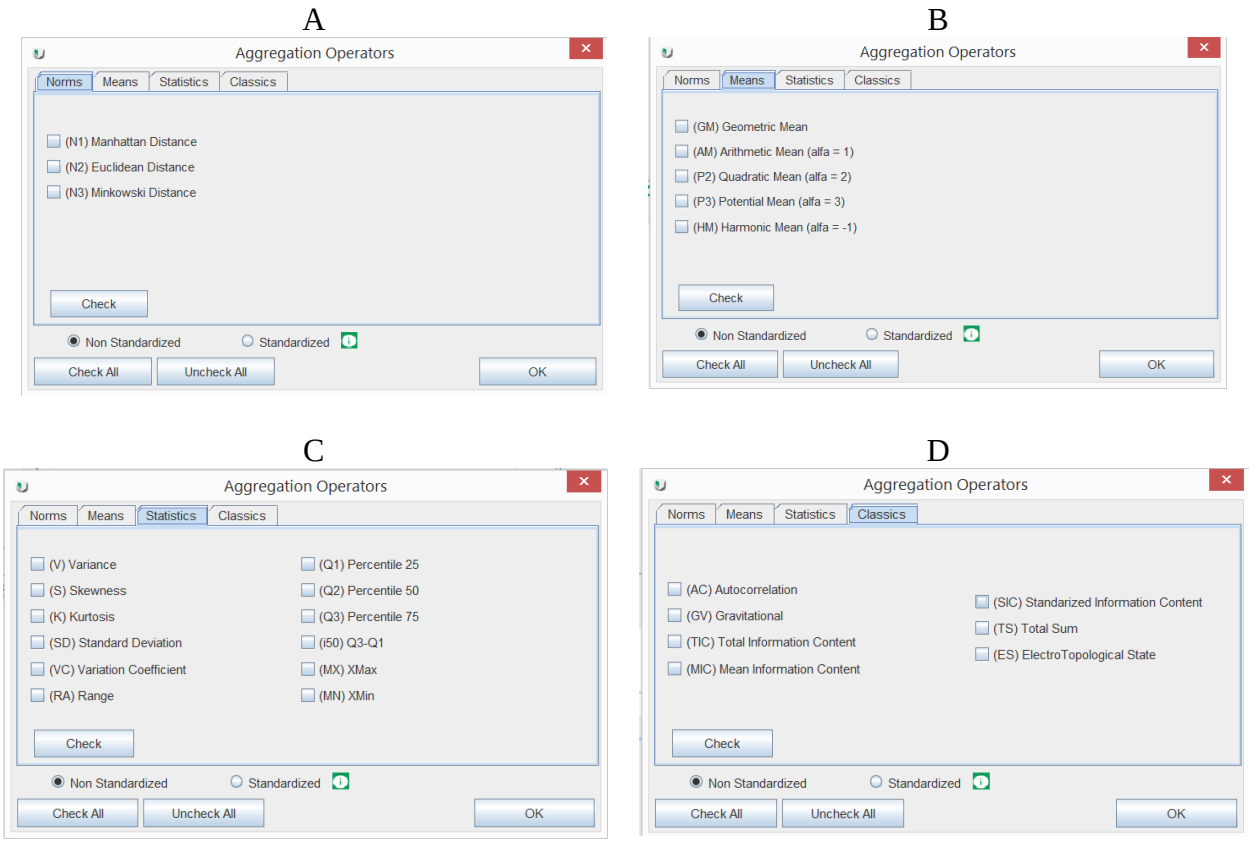


Figure 26. Dialog windows to set the aggregation operator to obtain protein descriptors from amino acidic contributions. A) Norms B) Means C) Statistics D) Classics

N-Tuples

For the **"Two" option**, the index calculations are performed over vertex pairs *i* and *j*. Here, the k^{th} *two-tuple Spatial-Dis-Similarity Tensor* (D-SDST) , ($^{D}\mathbb{Z}$), is used as tensor-based representation of the protein 3D-structure. For this option, *bilinear* (**B**), *linear* (**F**) and *quadratic* (**Q**) indices can be obtained.

For the **"Three" option**, the index calculations are performed over vertex triples *i,j,k*. Here, the k^{th} *three-tuple Spatial-Dis-Similarity Tensor* (T-SDST), ($^{T}\mathbb{Z}$), is used as tensor-based representation of the protein 3D-structure. For this option, *Trilinear Canonical* (***Tr***), *Trilinear linear* (**TrF**), *Trilinear quadratic bilinear* (**TrQB**) and *Trilinear cubic* (**TrC**) indices can be obtained.

Algebraic Forms

The definition for any k^{th} two or three-linear biomacro-molecular descriptors for a protein must consider a canonical basis set and the application of *N*-linear forms (two-linear or three-linear) in a $\mathbb{R}^n$ space; equations (**1**) and (**2**) indicate the mathematical expressions for these definitions:

$$ {}^{k}_{D}L = bl^{k}(\bar{x}, \bar{y}) = \sum_{i=1}^{n}\sum_{j=1}^{n} z_{ij}^{k} x^{i} y^{j} \quad \textbf{(1)} $$

$$ {}^{k}_{T}L = tr^{k}(\bar{x}, \bar{y}, \bar{p}) = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{l=1}^{n} z_{ijl}^{k} x^{i} y^{j} p^{l} \quad \textbf{(2)} $$

where, ${}^{k}_{D}L$ *and* ${}^{k}_{T}L$ are the resulting two-linear and three-linear form MD, *n* is the number of amino acids (*aa*) present in the protein, $[X], [Y], [P]$ are the macro-molecular vectors containing $x^1,\ldots,x^n$, $y^1,\ldots,y^n$ and $p^1,\ldots,p^n$ elements, which are the physicochemical properties of every *aa* present in the protein structure (Section 2). The k^{th} *two and three-tuple- spatial (dis)similarity tensors* (D-SDST and T-SDST) (${}^{D}\mathbb{Z}^{k}$ *and* ${}^{T}\mathbb{Z}^{k}$) are a two and three-order tensors whose elements z_{ij}^{k} *and* z_{ijl}^{k} are calculated by using relationships (metrics and multi-metrics) between two and three *aa*, respectively (see **Figure 27**).

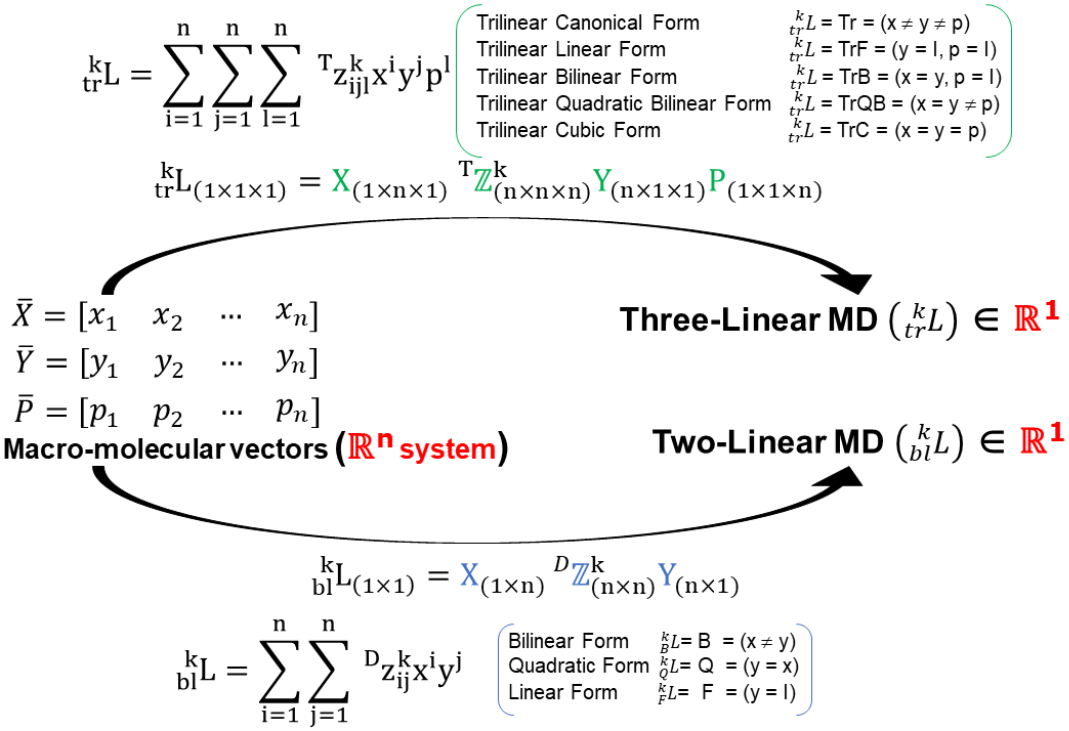


Figure 27. Schematic indication of the transformation of the information contained on macro-molecular vectors using spatial information of the protein (Two and Three-Tuple-Spatial Dis Similarity Tensors, D-SDST (${}^{D}\mathbb{Z}^{k}$) and T-SDST (${}^{T}\mathbb{Z}^{k}$), respectively) and algebraic forms.

Matrix Forms

Three types of matrices of order k ($k \in [-12,12]$) can be used in **MuLiMs-MCoMPAs software**, *namely* Non-stochastic (**NS**), Simple-stochastic (**SS**) and Mutual Probability (**MP**).

Non-stochastic (NS) matrix

The codification of 3D information of the non-covalent interactions of the biomacromolecular (protein or peptide) structure is fulfilled through rules between pairs of amino acids and the values of these rules are the elements of the *k^th^ two* and *three-tuple-spatial (dis)similarity* tensors (D-SDST and T-SDST) ($^{D}\mathbb{Z}^{k}$ *and* $^{T}\mathbb{Z}^{k}$).

When no normalizing procedure is performed over their elements, these tensors is denoted as *k^th^ non-stochastic two* and *three-tuple- spatial (dis)similarity*. The k^th^ non-stochastic two and three-tuple- amino acid based spatial (dis)similarity tensors (D-ASDST and T-ASDST) ($^{D}\mathbb{Z}^{aa,k}$ and $^{T}\mathbb{Z}^{aa,k}$) are two and three-order tensors whose elements $z_{ij}^{aa,k}$ and $z_{ijl}^{aa,k}$ are calculated by using relationships (metrics and multi-metrics, respectively) between two and three *aas*, respectively.

The k^{th} simple-stochastic for two and three-tuple-(dis)similarity tensors $_{ss}^{D}\mathbb{Z}^{k}$ *and* $_{ss}^{T}\mathbb{Z}^{k}$ (SS-D-SDST and SS-T-SDST) and k^{th} mutual probability for two and three-tuple-(dis)similarity tensors $_{mp}^{D}\mathbb{Z}^{k}$ *and* $_{mp}^{T}\mathbb{Z}^{k}$ (MP-D-SDST and MP-T-SDST), can be defined by applying the following equations:

$$ {}_{ss}^{D}z_{ij}^{k} = \frac{{}_{ns}^{D}z_{ij}^{k}}{S_i} = \frac{{}_{ns}^{D}z_{ij}^{k}}{\sum_{j=1}^{n} {}_{ns}^{D}z_{i}^{k}} \quad \textbf{(1)} $$

$$ {}_{ss}^{T}z_{ijl}^{k} = \frac{{}_{ns}^{T}z_{ijl}^{k}}{S_{jl}} = \frac{{}_{ns}^{T}z_{ijl}^{k}}{\sum_{j=1}^{n}\sum_{k=1}^{n} {}_{ns}^{T}z_{ijl}^{k}} \quad \textbf{(2)} $$

$$ {}_{mp}^{D}z_{ij}^{k} = \frac{{}_{ns}^{D}z_{ij}^{k}}{S_{ij}} = \frac{{}_{ns}^{D}z_{ijl}^{k}}{\sum_{i=1}^{n}\sum_{j=1}^{n} {}_{ns}^{D}z_{ij}^{k}} \quad \textbf{(3)} $$

$$ {}_{mp}^{T}z_{ijl}^{k} = \frac{{}_{ns}^{T}zt_{ijl}^{k}}{S_{ijl}} = \frac{{}_{ns}^{T}z_{ijl}^{k}}{\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} {}_{ns}^{T}z_{ijl}^{k}} \quad \textbf{(4)} $$

where, $_{ns}^{D}z_{ijl}^{k}$, $_{ns}^{T}z_{ijl}^{k}$ are the elements of the k^th^ non-stochastic two and three-tuple-spatial (dis)similarity tensors. S_i is the summation of all elements on a row on the two-tuple tensor, S_{jl} is the summation of all entries of the two-tuple tensor corresponding to each *aa* i in a three-tuple matrix for the simple stochastic case. Considering the mutual probability scheme, S_{ij} is the summation of all elements on the two-tuple tensor, S_{ijl} is the summation of all elements of the three-tuple tensor.

Cut-Off Setting

- **KA** (Keep All). All elements in matrices for algebraic forms are taken into consideration.
- **LG**. The *lag* assumes values between 1 and L. The idea of a *lag* is taken from autocorrelation descriptors, only that here the meaning is quite different. This cutoff is applied to the elements in the matrices for the algebraic forms, namely NS, SS and MP. In this software, a range of k values will be used instead of only one value of k, *e.g.*, k= 2-5, k=2;4;7-9.

- **LG[p]** (lag p). In this *lag* the maximum value L can be $n - 1$, where n is the number of atoms. When this lag is applied, the matrix elements are different from zero for relations of atoms that have at less an atom-pair with *topological distance, p,* equal to a specified *k* value.
- **LG[l]** (lag l). In this *lag* the maximum value L will be less than the maximum Euclidean distance between two atoms. When this lag is applied, the matrix elements are different from zero for relations of amino acids that have at less an amino acid-pair with *Euclidean distance, l,* less or equal to a specified *k* value.

Groups Sub-Area

In addition to ***total*** *algebraic indices* computed for the whole-protein, a **local-fragment** formalism can be developed. In this way, the matrix representations of the bio-macromolecular structures can be transformed to considerer information related with groups or amino acid-types belonging to a specific polypeptide fragment (*F*). So, these *local-fragment matrices* are used as matrix forms of the algebraic maps to compute the *local-fragment indices*. The amino acid-type fragments employed in this software are:

- Apolar (RAP)
- Polar positively charged (RPC)
- Polar negatively charged (RNC)
- Polar uncharged (RPU)
- Aromatic (ARO)
- Aliphatic (ALG)

Also we defined groups that include the amino acids that do not favor the folding and/or cannot be commonly found in proteins as part of α-helices or β-sheets (UFG), α-helices favoring amino acids (FAH), β-sheets favoring amino acids (FBS) and β-turns favoring amino acids (AFT). Additionally, groups composed of amino acids of the same kind (R amino acids) in the protein were defined, that is, 20 groups one per each natural α-amino acid, (e.g. F=Ala, F=Arg,…, F=Val). Table 1 shows the amino acidic composition of these local-fragments.

Table 1. Amino acidic composition of the local fragments pre-defined in the MuLiMs-MCoMPAs module of the ToMoCoMD-CAMPS software.

Local-Fragment	Amino acids
RAP	PRO, ILE, ALA, VAL, LEU, PHE, TRP, MET.
RPC	LYS, HIS, ARG.
RNC	ASP, GLU.
RPU	ASN, CYS, GLY, SER, THR, TYR, GLN.
ARO	PHE, TYR, TRP.
ALG	GLY, ALA, PRO, VAL, LEU, ILE, MET.
UFG	GLY, PRO.
FAH	ALA, CYS, LEU, MET, GLU, GLN, HIS, LYS.
FBS	VAL, ILE, PHE, TYR, TRP, THR.
AFT	GLY, SER, ASP, ASN, PRO.

Properties (labels)

Amino acid **properties** are used in **MuLiMs** software as **Amino Acid Weights**

1. Side-chain mass (MM)
2. Side-chain volume (MV)
3. Z1-scale (Z1)
4. Z2-scale (Z2)
5. Z3-scale (Z3)
6. Atomic charge (ECI)
7. Isotropic surface area (ISA)
8. Hoop-Woods hydropathy index (HWS)
9. Kyte-Dolittle hydropathy index (KDS)
10. Isoelectric point (PIE)
11. Heat of formation (EPS)
12. Relative Alpha helix frequency (PAH)
13. Relative Beta-sheet frequency (PBS)
14. Relative Reverse turn frequency (PTT)
15. Geometric compatibility parameter 1(GCP1)
16. Geometric compatibility parameter 2(GCP2)

Aggregation operators to LAIs Vector

The notion of *invariants* (aggregation operators) as a generalization scheme for the linear combination of amino acidic contributions to yield molecular definitions is derived from the hypothesis that the most appropriate global definition of a natural system may not necessarily be additive. Indeed, it was demonstrated by Barigye *et al.*(Barigye 2012, Barigye 2013) that other operators besides the sum could yield better correlations with specific chemical properties. These invariants are applied to the amino acid-level descriptors (LAIs), then the k^{th} *two-* and *three-linear* **MuLiMs MCoMPAs** MDs are obtained using one or several aggregation operators. This program, includes a series of *invariants* that generalize the traditional method of obtaining global (or local) invariants by summation of the LAIs. These are classified in four major groups:

1) **Norms (or Metrics)**:
   - **a)** Minkowski's norms (N1, N2, N3)
2) **Mean Invariants (first statistical moment)**:
   - **a)** Geometric Mean (G),
   - **b)** Arithmetic Mean (M),
   - **c)** Quadratic Mean (P2),
   - **d)** Potential Mean (P3) and
   - **e)** Harmonic Mean (HM)
3) **Statistical Invariants (highest statistical moments):**
   - **a)** Variance (V),
   - **b)** Skewness (S),
   - **c)** Kurtosis (K),
   - **d)** Standard Deviation (SD),

e) Variation Coefficient (VC),
f) Range (R),
g) Percentile 25 (Q1),
h) Percentile 50 (Q2),
i) Percentile 75 (Q3),
j) Inter-quartile Range (I50),
k) X max (MX) and
l) X min (MN)

4) "Classical algorithms" Invariants:

a) Autocorrelations AC(i),
b) Gravitational (GI(i)),
c) Total sum at *k* lags (TSk(i)),
d) Mean information content (MI(i)),
e) Total information content (TI),
f) Standardized information content (SI) and
g) Electrotopological state (ES(i)).

Standardized tab

In the standardization procedure, all values of *original* LAIs are replaced by standardized LAI values which are computed as follows: *Std. LOVIs* = (Original LAI – mean of LAIs)/Std. deviation of original LAIs. With this re-scaling, each new LAI has a mean of 0 and a standard deviation of 1.

Table 1. Norms (Metrics) Invariant.

Name	ID	Formula (Equation)
Minkowski's norms (p = 1) Manhattan norm	N1	$\|x\|_1 = \sum_{i=1}^{n} L_i$
Minkowski's norms (p = 2) Euclidean norm	N2	$\|\bar{x}\|_2 = \sqrt{\sum_{i=1}^{n} \|L_i\|^2}$
Minkowski's norms (p = 3)	N3	$\|\bar{x}\|_3 = \sqrt[3]{\sum_{i=1}^{n} \|L_i\|^3}$

Note 1. The general equation of Minkowski's norms is, $\|\bar{x}\|_p = \sqrt[p]{\sum_{i=1}^{n} |x_i|^p}$.

Note 2. The formulae used in these invariants, are simplified forms of general equations given that the vector $\bar{y}$ is constituted of the coordinates of the origin. For example, in the case of the Euclidean norm (N2), the general formula is: $\|\bar{x}\|_2 = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2 + (x_j - y_j)^2 + (x_z - y_z)^2}$. But given that $\bar{y}$ = (0, 0, 0), this formula reduces to $\|\bar{x}\|_2 = \sqrt{\sum_{i=1}^{n} |x_i|^2}$.

Table 2. Mean (First Statistical Moment) Invariants.

Name	ID	Formula (Equation)
Geometric Mean	G	$\bar{\xi} = \sqrt[n]{\prod_{i=1}^{n} L_i}$
Arithmetic Mean	M	

(power mean with degree α = 1)		
Quadratic Mean (power mean with degree α = 2)	P2	$m_\alpha = \left(\frac{L_1^\alpha + L_2^\alpha + ... + L_n^\alpha}{n} \right)^{\frac{1}{\alpha}}$
Power Mean with degree α = 3	P3	
Harmonic Mean (power mean with degree α = -1)	A	

Table 3 Statistical (Highest Statistical Moments) Invariants

Name	ID	Formula (Equation)
Variance	V	$V = \frac{\sum_{i=1}^{n} \left(L_i - \bar{L}\right)^2}{n-1}$
Skewness	S	$S = n*M_3 /[(n-1)*(n-2)*s^3]$ $M_3 = \sum_{i=1}^{n} \left(L_i - \bar{L}\right)^3$ s^3 *is the standard deviation raised to the 3rd power* *n is the number of atoms.*
Kurtosis	K	$K = [n*(n+1)*M_4 - 3*M_2*M_2*(n-1)] / [(n-1)*(n-2)*(n-3)*s^4]$ $M_j = \sum_{i=1}^{n} \left(L_i - \bar{L}\right)^j$ *n is the number of atoms.* s^4 *is the standard deviation raised to the fourth power*
Standard Deviation	SD	$\sigma = \sqrt{\frac{\left(\sum L_i - \bar{L}\right)^2}{n-1}}$
Variation Coefficient	VC	$c_v = s / \bar{L}$
Range	R	$R = L_{\max} - L_{\min}$
Percentile 25	Q1	$P25 = \left[\frac{N}{4} + \frac{1}{2} \right]$ *N is the number of values*
Percentile 50	Q2	$P50 = \left[\frac{N}{2} + \frac{1}{2} \right]$ *N is the number of values*
Percentile 75	Q3	$P75 = \left[\frac{3N}{4} + \frac{1}{2} \right]$ *N is the number of values*
Inter-quartile Range	I50	$I50 = P75 - P25$
X max	MX	L_i maximum
X min	MN	L_j minimum

Table 4 "Classical" (classical functions to derive MDs from LAIs) Invariants

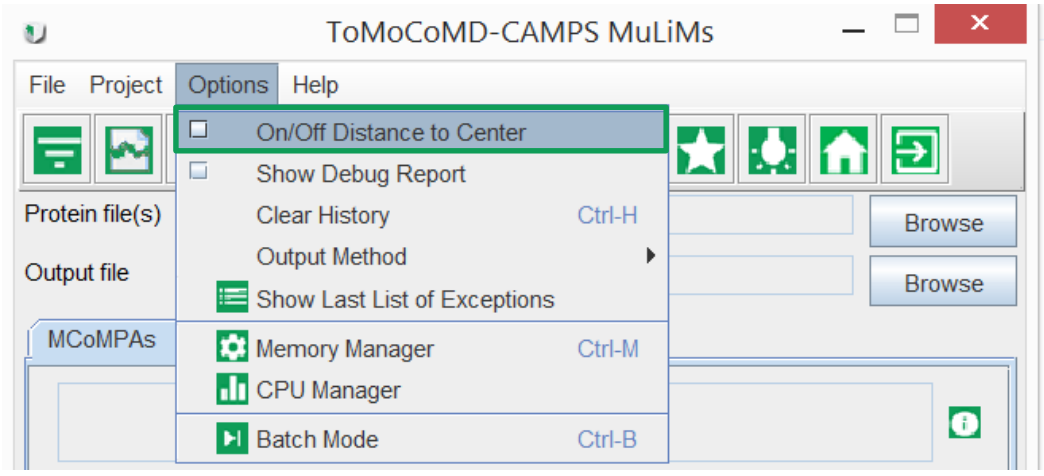
Name	ID	Formula (Equation)
Autocorrelation	$AC^k(i)$	$AUT_k = \sum_{i=1}^{n} \sum_{j \geq 1}^{n} L_i \times L_j \bullet (\delta(d_{ij}, k)), k = 1,2,..7$

Gravitational	$GI^k(i)$	$G_k = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{L_i L_j}{{}^k d_{ij}} \bullet \delta(d_{ij},k)), k = 1,2,..7$
Total sum at lag k	$TS^k(i)$	$TS_k = \sum_{i=1}^{n}\sum_{j=1}^{n} L_{ij} \bullet \delta(d_{ij},k)) = 1,2,...7$
Mean Information Content	$MI(i)$	$MI = -\sum_{i=1}^{A}\frac{N_g}{N_o}\cdot\log_2\frac{N_g}{N_o}$ *where, Ng is the number of atoms with the same LOVI value. N_o is the number of atoms in a molecule*
Total Information Content	TI	$TI = N_0 \cdot \log_2 N_0 - \sum_{g=1}^{G} N_g \cdot \log_2 N_g$
Standardized Information Content	SI	$SI = \frac{IT}{N_0 \cdot \log_2 N_0}$
Electrotopological state (E-state index)	$ES(i)$	$S_i = I_i + \Delta I_i = I_i + \sum_{j=1}^{n}\frac{I_i - I_j}{(d_{ij}+1)^2}$ where, I_i is the intrinsic state of the i^{th} atom and ΔI_i is the field effect on the ith atom calculated as perturbation of the I_i of i^{th} atom by all other atoms in the molecule, d_{ij} is the topological distance between the i^{th} and the j^{th} atoms, and n is the number of atoms. The exponent k is 2.

Additional Configuration Options

Distance to Protein Center

The computations of the distance of each amino acid to the protein center can be used or not to perform the calculations. That is to say, ON (compute) while OFF (does not compute) the distance to protein center. This decision can be configured in options menu.



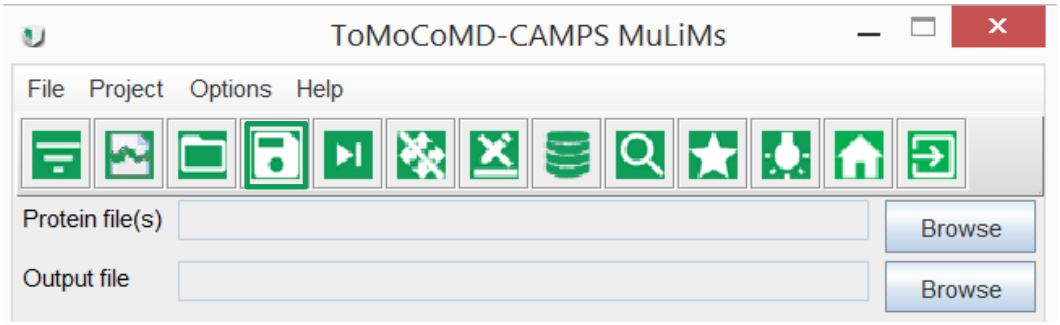


Figure 28. On/Off Distance to Center menu item/tool bar button.

Input and Output Files

The following is a description of the **Input** and **Output** section of the **MuLiMs-MCoMPAs** GUI. In these sections, input structure files can be loaded and the output descriptor files can be chosen.

Structure input files are selected and loaded by clicking the *Browse* button in the **Input Protein (s)** section in the upper right part of the GUI. A dialog box appears displaying the directory that is specified in the input file. The last input folder path is remembered for MuLiMs-MCoMPAs, so you can easy locate your structures files.

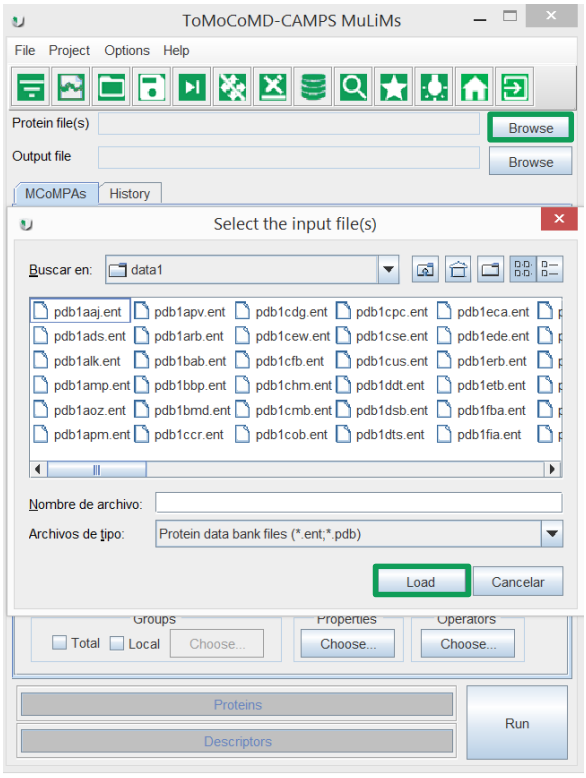


Figure 29. Browsing Input files (*. ent and/or *. pdb).

When you are browsing for the input files, there are two possibilities:

1) The selected file (s) has (have) the extension (s) (*.pdb and/ or *.ent).
2) The selected file (s) has (have) the extension (s) (*.pdbx).

In the first case the *Gathering files information* window is launched where the program acquires information about format of the file, that is, the number of models, the number and names of the chain(s) which are contained in each file.

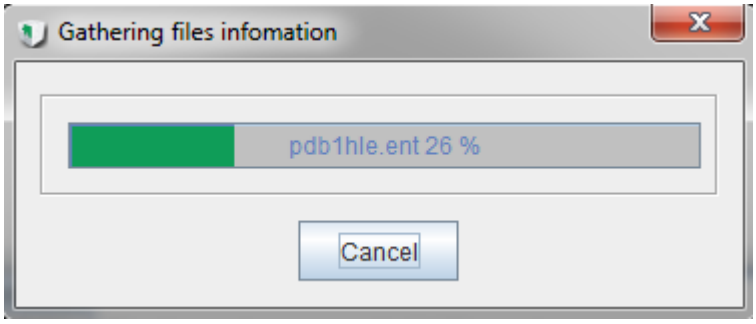


Figure 30. Gathering files information window.

When this process is finished, then if at least one file has a correct format then the *PDB3D-Editor* window is prompted. This sub module allows to the user to *preprocess* the protein dataset before descriptors calculations. The functionalities provided in this sub module are the following:

- Generation of four 3D-protein representations:

1) Alpha-carbon atom (CA)
2) Beta-carbon atom (CB)
3) Amide Bond-carbon atom (AB)
4) Average (AVG)

It is important to highlight that the software automatically removes the hydrogen atoms, ligands, water molecules and any HET atoms. In the hypothetical case that two atom records with the occupancies (i.e. the probability of finding an atom in this coordinates) parameters are reported, then the atom with the *lower* occupancy is deleted. It should be pointed out that the generated *"standardized"* files with extension (*.pdbx) fulfill the guidelines described in the "*Atomic Coordinate Entry Format Description Version 3.30"* and are ready to perform the calculation of the descriptors provided by MuLiMs software. Finally, the standardized files (*.pdbx) are generated into a folder named with the acronym of the selected representation (s) to avoid the mixture of different representations.

- Visualization of a selected representation in both plain text and in 3D (using the Jmol library) prior to saving to a persistent file.
- Selection of a specific model (by double clicking the specific cell).
- Selection of the chain(s) of interest to the user (by double clicking the specific cell).

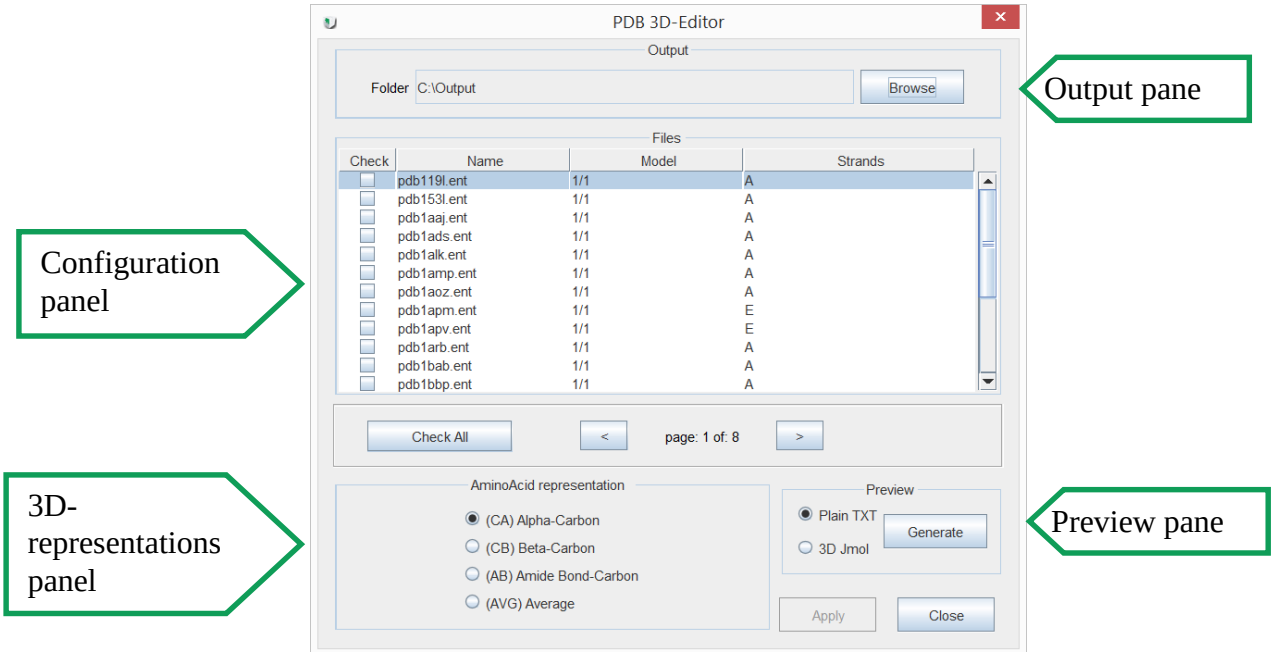


Figure 31. PDB 3D-Editor sub module.

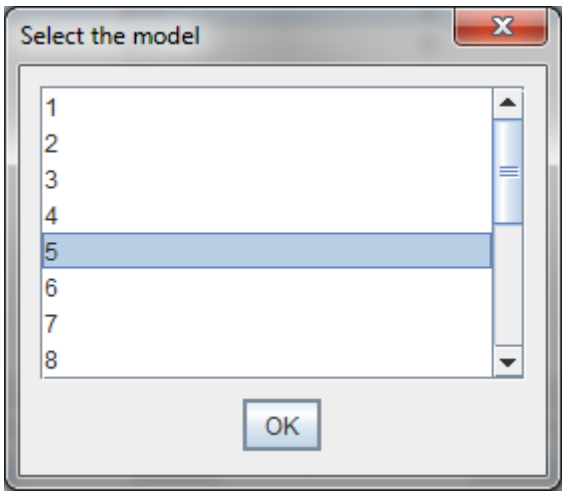


Figure 32. Model selection window.

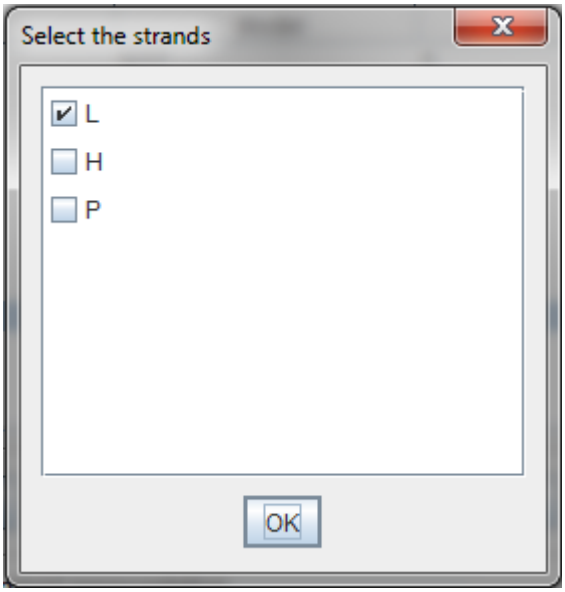


Figure 33. Strand (s) [chain(s)] selection window.

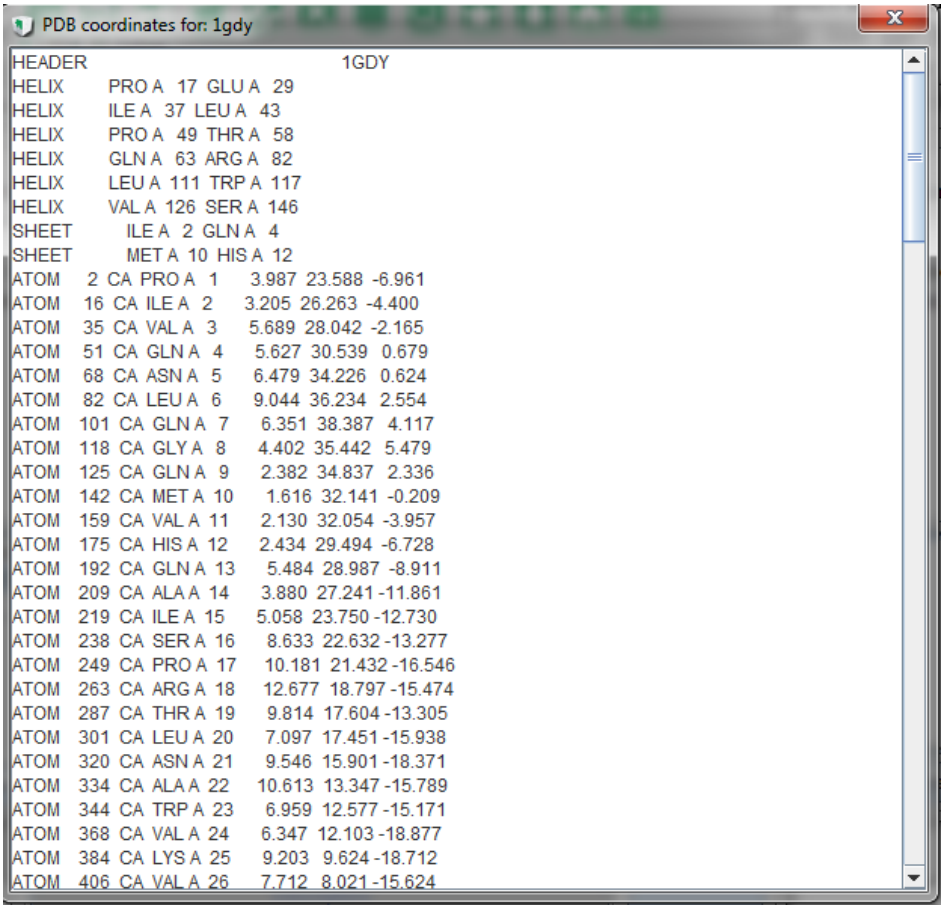


Figure 34. Visualization in Plain Text of [a selected model, strand (s) and representation (CA)] window.

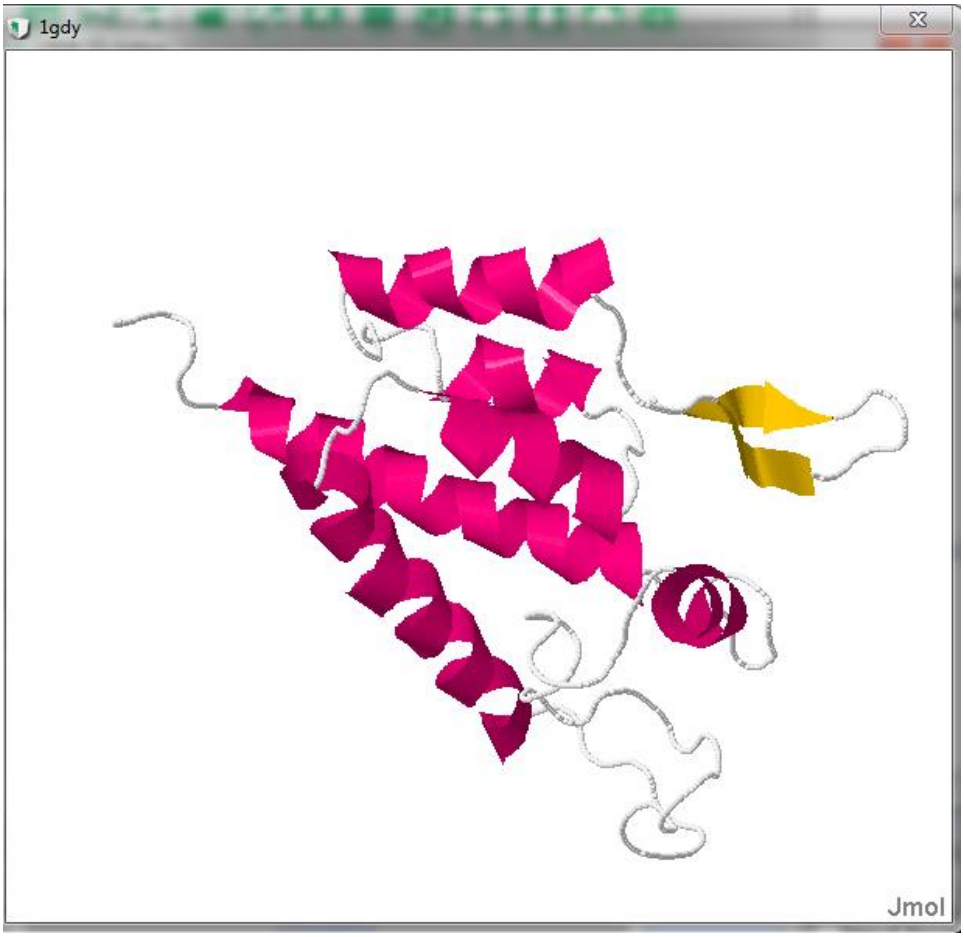


Figure 35. Visualization in 3D of [a selected model, strand (s) and representation (AA)] window.

In the second case the file(s) is (are) automatically loaded by the system because it assumes that this is its internal *standardized* format. It is important to highlight that the selected files must contain the same 3D-protein representation, otherwise the system prompts an error message *"Please provide files with the same amino acid representation"* and no files are loaded.

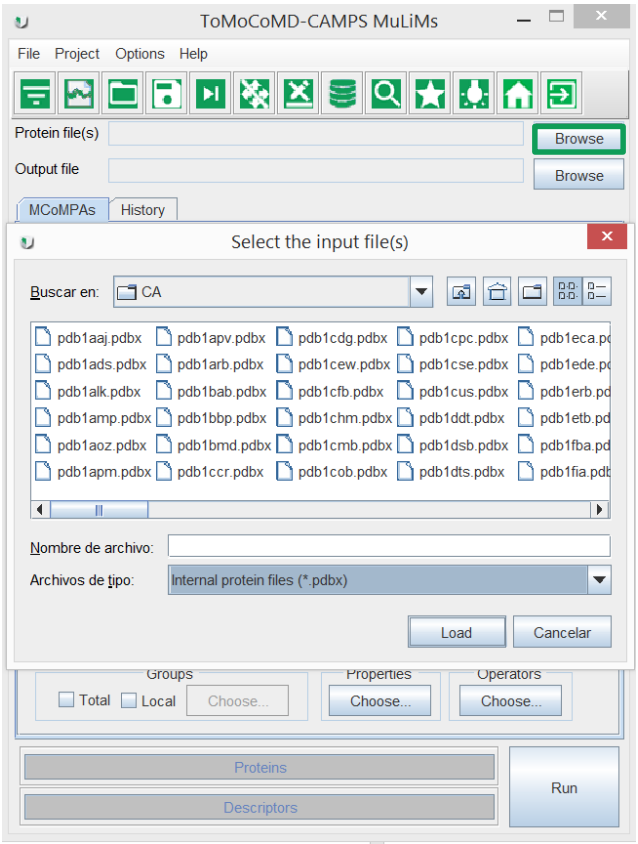


Figure 36. Browsing input standardized files (*.pdbx)

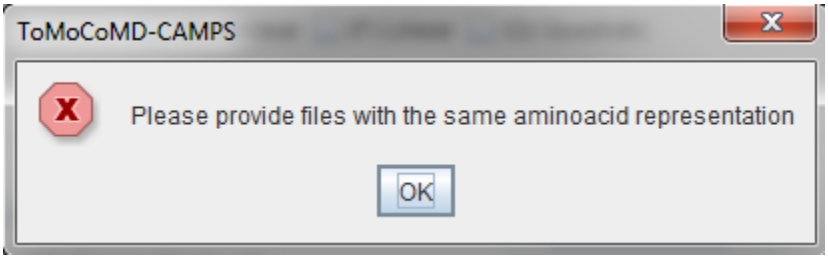


Figure 37. Error message prompted when there are different representations in the selected input files.

The name and path of the descriptor output file is selected by clicking on the *Browse* button in the **Output** section in the upper right part of the GUI.

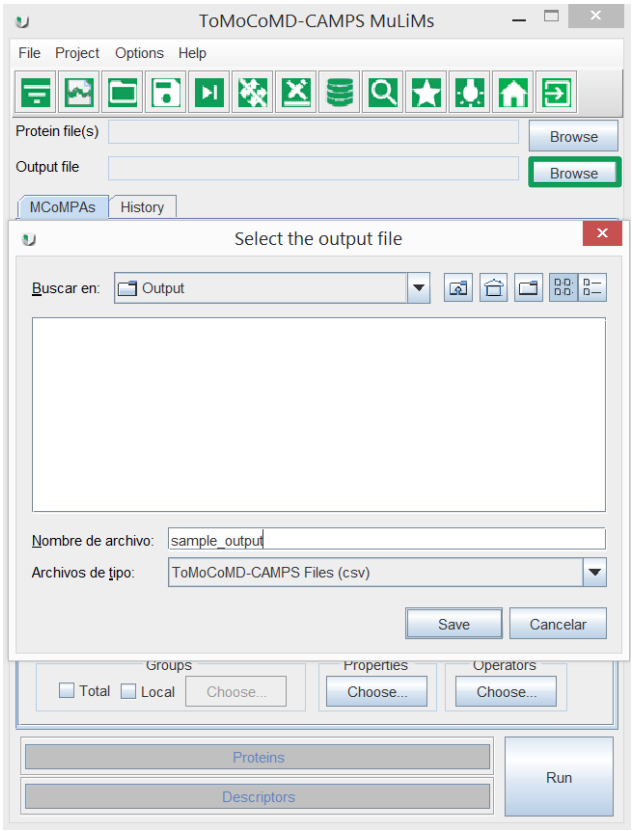


Figure 38. Browsing Output file.

Supported File Formats

INPUT

Protein DataBank File (PDB)

A **PDB** file include atomic coordinates, crystallographic structure factors and NMR experimental data. Aside from coordinates, each deposition also includes the names of molecules, primary and secondary structure information, sequence database references, where appropriate, and ligand and biological assembly information, details about data collection and structure solution, and bibliographic citations.

OUTPUT

Space and Comma Separated Value Files (TXT, CSV)

A **space-separated values** file is a simple text format for a database table. Each record in the table is one line of the text file. Each field value of a record is separated from the next by a space (or blank) character, it is a form of the more general delimiter-separated values format.

As file extension for this output file we choose TXT, because it is a simple file format that is widely supported, so it is often used to move spaced data between different computer programs that support the format. For example, a space-separated file might be used to transfer information from a database program to a spreadsheet.

TXT is an alternative to the common comma-separated values (CSV) format, which often causes difficulties because of the need to escape commas. Literal commas are very common in text data.

A **comma-separated value (CSV)** file stores tabular data (numbers and text) in plain-text form. A plain text form means that the file is a sequence of characters, with no data that has to be interpreted instead, as binary numbers. A CSV file consists of any number of records, separated by line breaks of some kind; each record consists of fields, separated by some other character or string, most commonly a literal comma or tab. Usually, all records have an identical sequence of fields.

CSV is a common, relatively simple file format that is widely supported by consumer, business, and scientific applications. Among its most common uses is moving tabular data between programs that natively operate on incompatible (often proprietary and/or undocumented) formats. This works because so many programs support some variation of CSV at least as an alternative import/export format.

Weka Attribute-Relation File Format (ARFF)

An **ARFF** (Attribute-Relation File Format) file is an ASCII text file that describes a list of instances sharing a set of attributes. ARFF files were developed by the Machine Learning Project at the Department of Computer Science of the University of Waikato for use with the Weka machine learning software (Waikato). A complete specification of ARFF files can be found at http://weka.wikispaces.com/ARFF.

Files Created for MuLiMs-MCoMPAs

MuLiMs-MCoMPAs produces an output file containing the values of the calculated and selected MDs, together with the additional information imported by the user. The Output File can be selected by clicking on the *Browse* button in the *Output* section and MuLiMs-MCoMPAs supports, CSV format (comma separated value), TXT format (space-separated values file) and ARFF (Attribute-Relation File Format) Weka files.

The error and warning messages given below are printed to the Exception Windows (see Exception Windows section) and can be saved by clicking the *Save* button.

The "missing values" is represented by a constant sequence of characters: "*NaN*", stands for *Not a Number* value. For instance, three errors or exceptions types are possible (see Special Instructions and Exceptions section):

1. Errors in calculating the algebraic form descriptors.
2. Unexpected Error in calculating a descriptor.

The **standard MuLiMs-MCoMPAs format** for the output file (.csv) is organized as follows (this format, namely, array of MDs blocks, cannot be changed by the user, see Table 6 for a simple example):

- The *first record* (column) contains the name of the proteins, *that is*: the "name of each file" plus ".pdbx" plus "_" plus the number of protein according to the lexicographic order of its name.

- The following records (columns) contain the **variable labels** *(descriptor headers), i.e.* **CA_N1_Q_M1_NS-4_T_KA_MM_MCoMPAs** (see Descriptor Search Tool in order to automatic translation of each header).

Table 5. An example Output file.

proteins	CA_N1_Q_M1_NS-4_T_KA_MM_MCoMPAs	CA_N1_Q_M1_NS-3_T_KA_MM_MCoMPAs
1ADW.pdbx_1	0.12035372	12.10199362
1APS.pdbx_2	0.15715901	11.34189404
1B9C.pdbx_3	0.30090779	27.9169252
1BA5.pdbx_4	0.05635754	4.937380627
1BDD.pdbx_5	0.05160233	5.079495182

Example Data

Click the example data icon in the tool bar to open these protein datasets. These datasets will permit simple test calculations to be made.

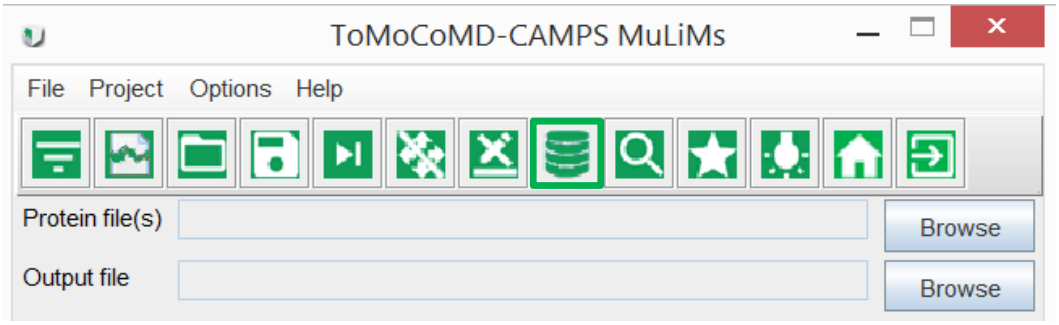


Figure 39. Quick access shortcuts for example data tool.

Searching for Descriptors Headers

By clicking the button 'Descriptor search' a window will appear where the user can enter the symbol (descriptor headers) of an unknown descriptor. A short definition of each part will be returned.

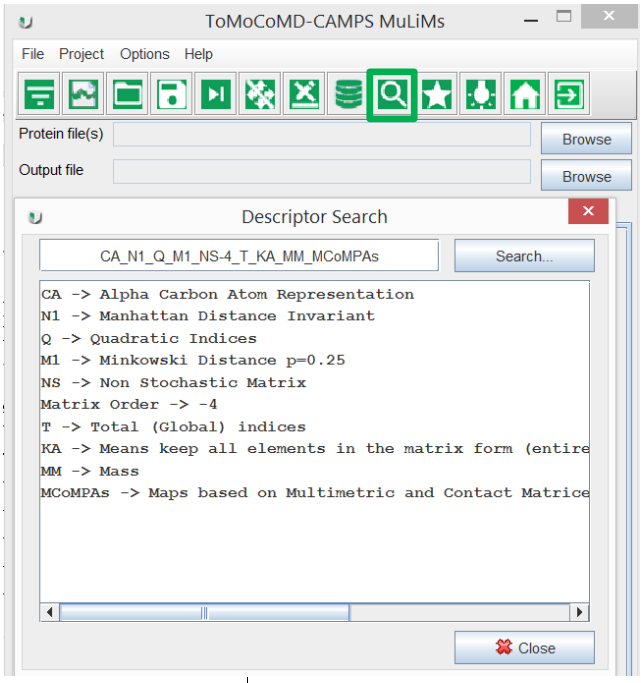


Figure 40. Descriptor Search Tool windows.

Debug Report Capability

This option permits, for each protein in the dataset, the user to save a txt file with the amino acid-property (or two properties for bilinear forms) vector, the *k* order of molecular graph matrices (**NS**, **SS**, or **MP**), and LAIs vectors (for each *k* order).

Property Vector X
Property Vector Y (if bilinear form is present)
Property Vector Z (if any three-linear form is present)

Matrices of order *k* (for a matrix forms: **NS**, **SS**, or **MP)**
LAIs Vectors order *k* (for *two-linear* (**Linear**, **Bilinear**, or **Quadratic**) and three-linear (**Trilinear Canonical**, **Trilinear linear**, **Trilinear bilinear**, **Trilinear quadratic bilinear** and **Trilinear cubic**) algebraic forms)

Matrices of order ±1 (for a matrix forms: **NS**, **SS**, or **MP)**
LAIs Vectors order ±1 (for *two-linear* (**Linear**, **Bilinear**, or **Quadratic**) and three-linear (**Trilinear Canonical**, **Trilinear linear**, **Trilinear bilinear**, **Trilinear quadratic bilinear** and **Trilinear cubic**) algebraic forms)

Matrices of order ±2 (for a matrix forms: **NS**, **SS**, or **MP)**
LAIs Vectors order ±2 (for *two-linear* (**Linear**, **Bilinear**, or **Quadratic**) and three-linear (**Trilinear Canonical**, **Trilinear linear**, **Trilinear bilinear**, **Trilinear quadratic bilinear** and **Trilinear cubic**) algebraic forms)

…

Matrices of order ±12 (for a matrix forms: **NS**, **SS**, or **MP)**
LAIs Vectors order ±12 (for *two-linear* (**Linear**, **Bilinear**, or **Quadratic**) and three-linear (**Trilinear Canonical**, **Trilinear linear**, **Trilinear bilinear**, **Trilinear quadratic bilinear** and **Trilinear cubic**) algebraic forms)

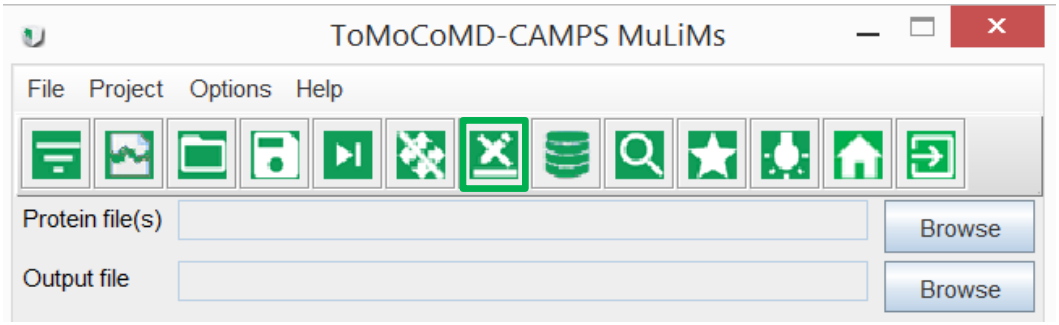


Figure 41. On/Off Generate Debug Report button.

Special Instructions and Exceptions

During the descriptor calculation **MuLiMs-MCoMPAs** writes out a log file that shows some statistics on the program run (History tab-window) and summarizes the errors and critical situations (warnings) encountered during the processing of the input structures (**Exception file**). That is, by checking the tab 'History' in the Configuration frame, a new window will appear during descriptor calculation where the main information concerning the batch in progress is progressively shown. If errors in the structural checking occur for a protein, the protein will be automatically *skipped* and the descriptors for this protein are not calculated. In addition, if an error in descriptor calculation occurs for a protein, then all its descriptor values will be missing in the final output file (*NaN*). Three errors or exceptions types are possible:

1. Errors calculating the algebraic form descriptors. The missing values label *NaN* is placed as descriptor value for each invalid entry.
2. Unexpected Error calculating descriptor. Any other error and exception will be notified through the Exception window, and the output file shows only the name of the protein while the rows for the corresponding descriptor calculations are empty.

The list of proteins with problems for error in calculation is shown in the 'Exception file' window together with the error type. The error and warning messages given below are printed in a log file that can be saved by clicking the **save** button.

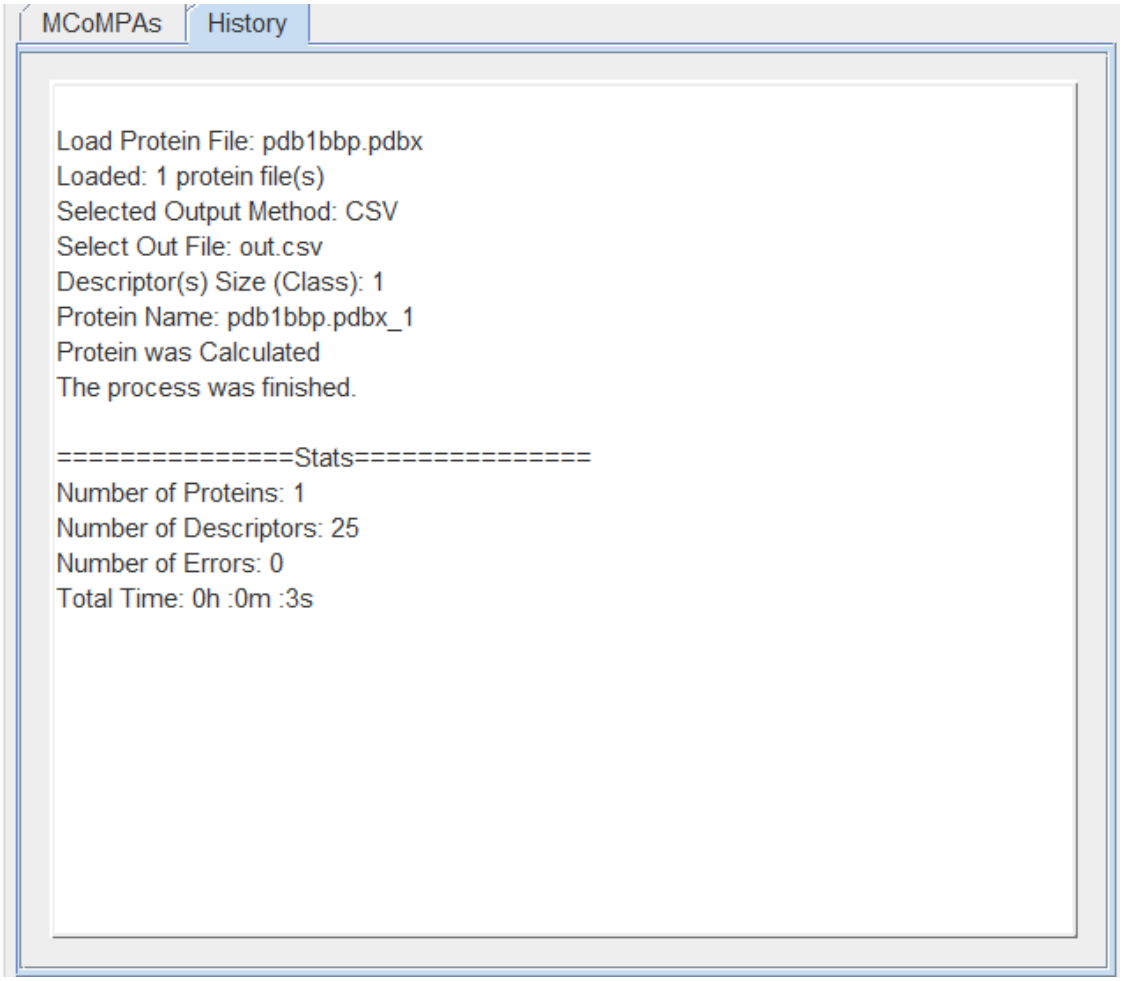


Figure 42. History Panel.

REFERENCES

1. . "ARFF file format." from http://weka.wikispaces.com/ARFF.
2. Barigye, S. J. M. P., Y.; Martínez-López, Y.; Torrens, F.; Artiles-Martínez, L. M.; Pino-Urias, R. W.; Martínez-Santiago, O (2012). "Relations Frequency Hypermatrices in Mutual, Conditional, and Joint Entropy-Based Information Indices." J. Comput. Chem. **34**(9): 259–274.
3. Barigye, S. M.-P., Y.; Santiago, O.; Lopez, Y.; Perez-Gimenez, F.; Torrens, F. (2013). "Shannon's, Mutual, Conditional and Joint Entropy Information Indices: Generalization of Global Indices Defined from Local Vertex Invariants." Curr. Comput. Aided-Drug Des **9**(2): 164–183.
4. Waikato, T. U. o. "Weka machine learning software." from http://www.cs.waikato.ac.nz/~ml/.