Bonemat

User Guide

v 3.2

Abstract

This is a brief manual of the Bonemat software. It is intended to be a guide to the user, and we hope it can help rather than confuse you. Please be aware that the manual is work in progress. It may not be complete or updated, and we cannot ensure all the information in it is 100% correct.

Bonemat is not intended for clinical use, nor for commercial use. It cannot be used for commercial purposes (e.g. consultancy), nor redistributed. You use it at your own risk, and you are responsible for the interpretation of any results.

Being a research tool under continuous development, Bonemat may not work exactly as documented, and sometimes it may not work at all (loop, crash, etc.)! Please report to us if you find any bugs (developers@bonemat.org), or would like to suggest improvements (info@bonemat.org).

Contents

1 License

Bonemat v3.2 use license agreement License grant and restrictions on use

The software named Bonemat v3.2 and any accompanying documentation, manuals or data (hereafter collectively "SOFTWARE") is property of Istituto Ortopedico Rizzoli and the copyright law protects it.

A revocable, limited, non-transferable, non-assignable, non-exclusive use license is hereby granted by Istituto Ortopedico Rizzoli, (hereafter "LICENSOR") to the user named below (hereafter "USER") for the limited use of the Software as long as the Software is used in accordance with all the provisions of this Agreement.

SOFTWARE is intended only for non-commercial research and education. It may not be used in relation to any commercial or military activity. SOFTWARE use must be acknowledged in all publications, presentations, or documents describing work in which SOFTWARE is used.

It is understood by USER that SOFTWARE is made available only on the condition that SOFTWARE will not be modified in any way or otherwise transformed into any other form. USER shall not modify, translate, reverse engineer, decompile, or disassemble SOFTWARE or any portion of SOFTWARE. USER shall not distribute SOFTWARE or create and distribute derivative works based on SOFTWARE. USER shall not duplicate in any form, except for archival purposes, the programs, program statements or any other materials relating to SOFTWARE supplied by LICENSOR to USER.

USER shall not remove any copyright notices or other proprietary notices or labels from SOFTWARE.

The rights and license granted USER hereunder to use SOFTWARE are restricted solely and exclusively to USER and may not be assigned, rented, subleased, sub-licensed, sold, offered for sale, transferred, disposed of, encumbered or mortgaged.

Title and all ownership and proprietary rights to the SOFTWARE and any derivative works thereof (including without limitation copyright, patent, trademark, and trade secret rights) shall remain in LICENSOR.

SINCE THE SOFTWARE IS PROVIDED FREE OF CHARGE, THE SOFTWARE IS PROVIDED ON AN "AS IS" BASIS. ACCORDINGLY, LICENSOR DOES NOT MAKE AND HEREBY EXPRESSLY DISCLAIMS, ANY AND ALL REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE SOFTWARE, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT, AND ANY WARRANTY WITH RESPECT TO THE ACCURACY OR APPLICABILITY OF THE SOFTWARE OR OF DATA OR ALGORITHMS CONTAINED IN THE SOFTWARE.

LICENSOR accepts no responsibility for, and makes no representations as to, the accuracy or reliability of any report, output, or data prepared by or with the assistance of SOFTWARE. The entire risk as to the use, quality and performance of SOFTWARE is born by USER. Should SOFTWARE prove defective in any way, USER, and not LICENSOR, assumes the entire cost of any service and repair. LICENSOR is not obligated to render any technical assistance or support in the use or operation of SOFTWARE or to correct or fix any defects or bugs in the SOFTWARE.

Under no circumstances shall LICENSOR or its affiliates, officers, directors, contractors, students, employees and agents, be liable to Licensee or any other person for any damages, losses or liabilities whatsoever or any general, special, incidental or consequential damages of any character, including without limitation damages for loss of good will, lost profits, work stoppage, computer failure or malfunction, or any and all other monetary damages or losses. In no event shall LICENSOR or its affiliates, officers, directors, contractors, students, employees, and agents be liable to USER with respect to any claim or demand made against USER by any other person.

Licensee agrees to indemnify and hold LICENSOR, and its affiliates, officers, directors, contractors, students, employees, and agents, harmless of, from and against all claims, costs, expenses, damages, and liabilities, including reasonable attorney's fees, actually or allegedly resulting from or incident to the use, operation, or storage of SOFTWARE.

In view of the unique and proprietary nature of SOFTWARE, it is understood and agreed that LICENSOR's remedies at law for a breach by USER of USER's obligations under this Agreement may be inadequate and that LICENSOR shall, in the event of any such breach, be entitled to equitable relief (including, without limitation, provisional and permanent injunctive relief) in addition to all other remedies provided under this Agreement or available to LICENSOR at law.

The License under this Agreement may be terminated at the sole discretion of LICENSOR at any time by LICENSOR. The License under this Agreement will automatically terminate upon USER'S failure to comply with any of the terms and conditions of this Agreement. Upon termination of the License under this Agreement for any reason, USER shall make no further use of SOFTWARE and USER shall destroy all copies of SOFTWARE.

Bonemat v3.2 is partially based on VISUALIZATION TOOL KIT (VTK) (**1**)

Bonemat v3.2 is partially based on WXWINDOWS LIBRARY (**2**)

Bonemat v3.2 is partially based on DCMTK OFFIS LIBRARY (**3**)

1. **The Visualization Toolkit (VTK) copyright**
   This is an open-source copyright as follows: Copyright © 1993-2002@ Ken Martin, Will Schroeder, Bill Lorensen. All rights reserved.
   Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. * Neither name of Ken Martin, Will Schroeder, or Bill Lorensen nor the names of any contributors may be used to endorse or promote products derived from this software without specific prior written permission. * Modified source versions must be plainly marked as such, and must not be misrepresented as being the original software.
   THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS 'AS IS' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,

EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

2. **wxWindows Library Licence**
   Copyright (C) 1998 Julian Smart, Robert Roebling [, ...]
   Everyone is permitted to copy and distribute verbatim copies of this licence document, but changing it is not allowed.
   WXWINDOWS LIBRARY LICENCE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION
   This library is free software; you can redistribute it and/or modify it under the terms of the GNU Library General Public Licence as published by the Free Software Foundation; either version 2 of the Licence, or (at your option) any later version.
   This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. see the GNU Library General Public Licence for more details.
   EXCEPTION NOTICE

   - As a special exception, the copyright holders of this library give permission for additional uses of the text contained in this release of the library as licenced under the wxWindows Library Licence, applying either version 3 of the Licence, or (at your option) any later version of the Licence as published by the copyright holders of version 3 of the Licence document.
   - The exception is that you may use, copy, link, modify and distribute under the user's own terms, binary object code versions of works based on the Library.
   - If you copy code from files distributed under the terms of the GNU General Public Licence or the GNU Library General Public Licence into a copy of this library, as this licence permits, the exception does not apply to the code that you add in this way. To avoid misleading anyone as to the status of such modified files, you must delete this exception notice from such code and/or adjust the licensing conditions notice accordingly.
   - If you write modifications of your own for this library, it is your choice whether to permit this exception to apply to your modifications. If you do not wish that, you must delete the exception notice from such code and/or adjust the licensing conditions notice accordingly.

3. **DCMTK Offis Library**

   Copyright (C) 1994-2015, OFFIS e.V. All rights reserved.

   This software and supporting documentation were developed by OFFIS e.V. R&D Division Health Escherweg 2 26121 Oldenburg, Germany

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- Neither the name of OFFIS nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

(a) THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRAN-TIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

2 What is Bonemat?

Bonemat, a software developed at Istituto Ortopedico Rizzoli in Bologna, Italy, maps on a Finite Element mesh the bone elastic properties derived from Computed Tomography images. Bonemat can import CT images and FE models, interactively visualize them, fuse them into a coherent representation, and export the updated FE mesh, to some of the most used FE solvers (Ansys, Abaqus) or in generic format, once bone properties have been mapped. Though developed for computational bone biomechanics, it may be intended as a more general tool for the numerical integration of entities from a regular to an unstructured grid.

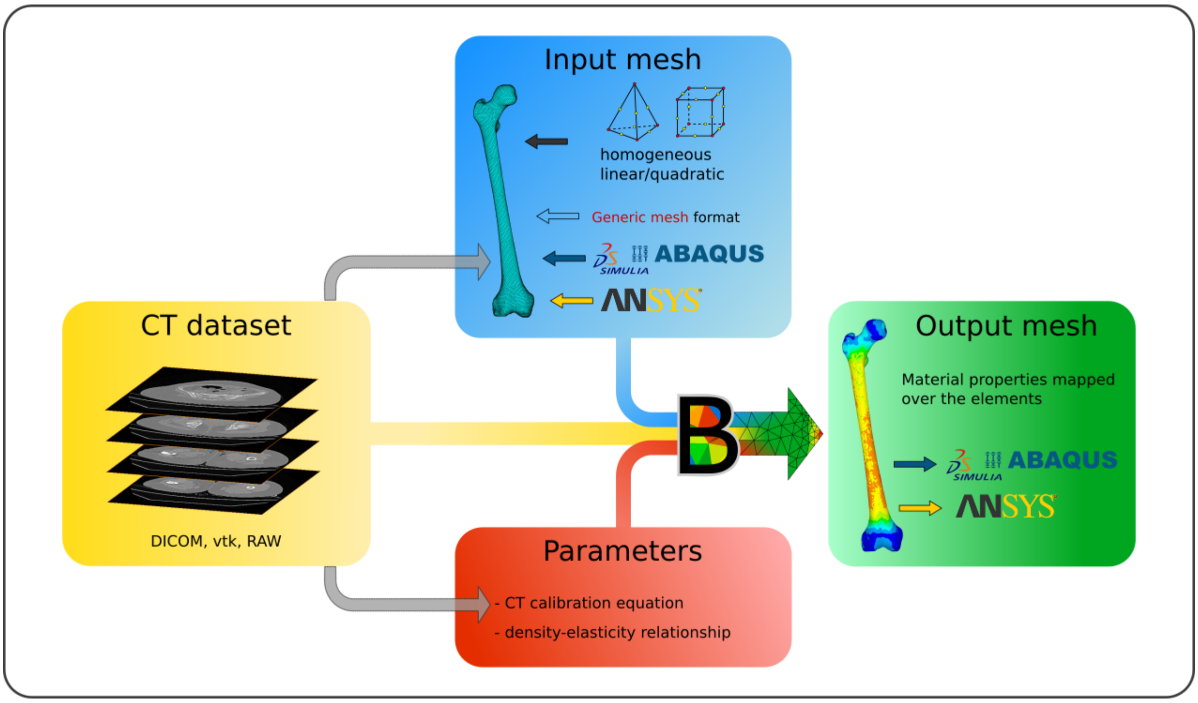


Figure 1

3 System Requirements

To use Bonemat you just need a PC running Microsoft Windows (we are sorry but currently other operative systems are not supported). The minimum system requirements are:

- Operating System: Windows (XP or later versions)
- Processor: 1.0 GHz
- RAM: 1GB (3 GB recommended)
- Graphics Card: Compatible with OpenGL 1.2
- Hard Disk: 40MB

4 Bonemat basics: software framework

To allow the user to interact with and visualise data, we embedded the core algorithm of Bonemat in a software framework based on the Visualisation Toolkit (www.vtk.org) and other specialised libraries. Within this framework, the main entities exposed to users are:

- Data objects, called also Virtual Medical Entities, VMEs for short in the following;
- Views, that provide interactive visualisation of the VMEs;
- Operations, that create new VMEs or modify existing ones (special Operations are the Importers, that let you import and convert into a VME almost any biomedical dataset, and the Exporters that can convert the VME into files formatted according to common standards);
- Graphical User Interface (GUI) components;

Each session of Bonemat, including data, can be saved and reloaded. When saving, object properties and relationships among objects are stored in an XML file with extension .msf. This .msf file links to a complementary folder, in which the datasets relative to each object are stored in binary format.

4.1 VMEs

VMEs provide the data representation. A VME represents a data entity. There are many VME types, that can be used to represent almost any biomedical image or signal. The VME types Bonemat can directly handle are: VMEVolume, VMEMesh. Other VME types can be imported through the “Import vtk” function, and some of them can be visualized (e.g. VMESurface in the View Surface), but no interaction through Operations is foreseen for them in Bonemat. VMEs are organized in a hierarchical tree (VME Tree or data Tree) and they are all composed by a dataset, a matrix that defines the pose of the VME with respect to its parent in the VME Tree, and a set of metadata that provides all the textual attributes of the VME.

4.2 Views

Views allow data examination. There are various views that permit to examine the VMEs. A View provides an interactive representation of the VME Tree. For each view the system maintains a display list of the VMEs that are currently rendered in that view. The Views available in Bonemat are: Arbitrary, Orthoslice, RX-CT, Surface (Figure 2).

The user can control several properties of the active View through the View Settings tab in the upper part of Control Bar.

4.3 Operations

Operations create new VMEs or modify existing ones. At any time the user can select a VME and then choose one of the available Operations. An Operation may accept only

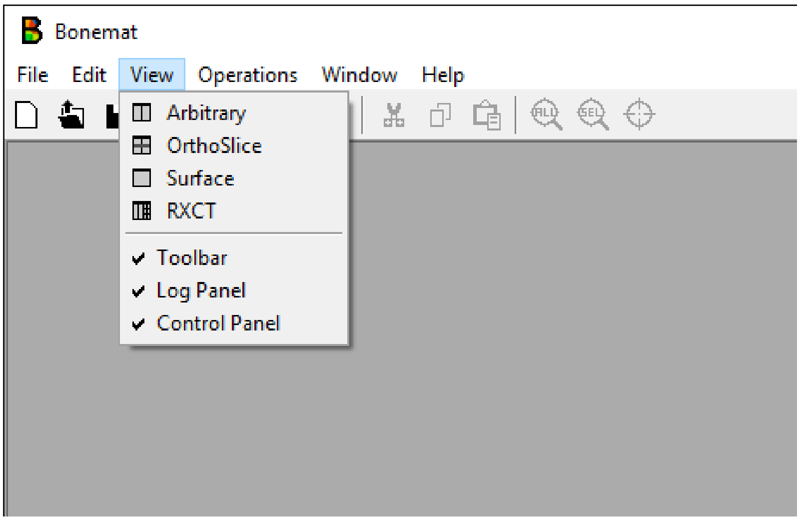


Figure 2

a particular type of VME as primary input, e.g. the Bonemat operation accepts only VMEMesh. At any time, it is possible to run only those operations that accept the currently selected VME (Figure 3).

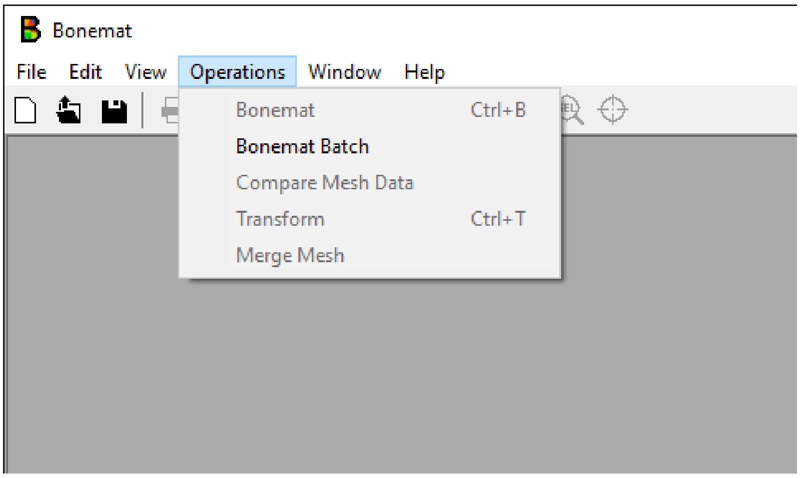


Figure 3

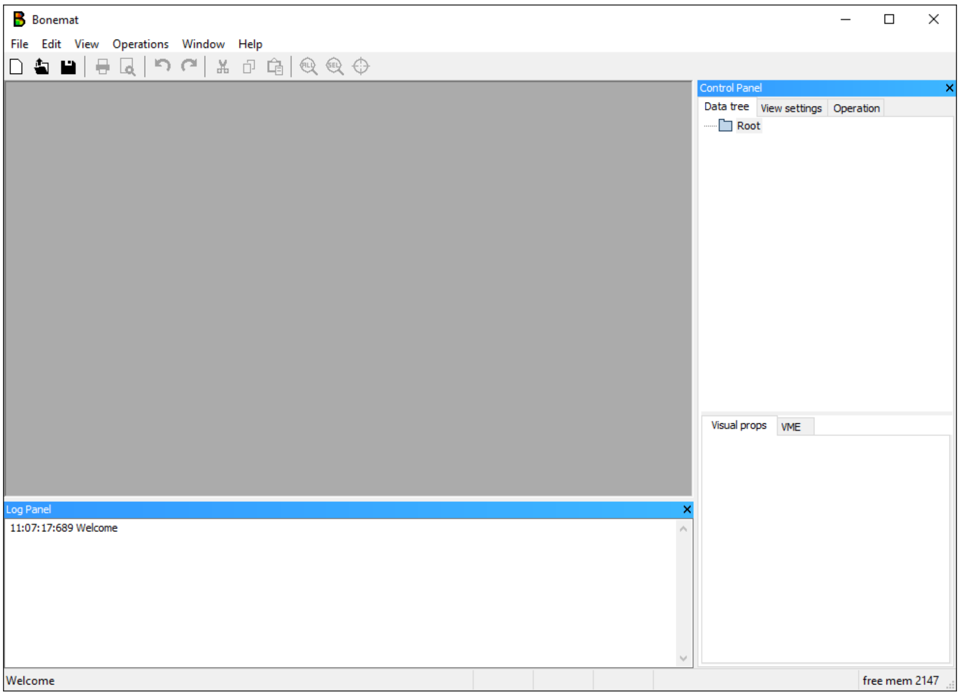
Operations that need more than one input may request additional VMEs to the user, e.g. the Bonemat operation requires as additional input a VMEVolume. The Operations available in Bonemat are: Bonemat, Bonemat Batch, Compare Mesh Data, Transform, and Merge Mesh.

Operations are modal: when an operation is running, it is not possible to run another operation and it is not possible to change the selected VME.

The user can control the behavior of the operation through the Control Bar. Operations support the Undo/Redo feature.

5 Bonemat basics: Graphical User Interface (GUI)

When launched the application interface looks like Figure 4.



```
11:07:17:689 Welcome
```

```
Welcome                                                        free mem 2147
```

Figure 4

5.1 User Interaction

In all visualization windows the mouse interaction with the available objects happens as follows:

- left mouse button to rotate
- central mouse button (or press down the scroll wheel) to translate/pan
- right mouse button to zoom in/out.

If a laptop is used, please look up in the manual for how to simulate from keyboard the central button of the mouse.

5.2 GUI environment

The basic structure of the GUI is composed of:

- Menu Bar
- Main Working Area

- Lateral Control Panel, showing the VME hierarchical structure
- Log Panel, for the system messages.

Contextual menus (upon clicking right mouse button) are also available for Views, VMEs, and for the VME tree.

5.2.1 Menu Bar

5.2.1.1 File This item contains all the commands related to input/output operations; in the application skeleton the basic features are open/save/new commands to respectively load, store, or initialize a new MSF session file. All importers/exporters will be described in detail in the following.

5.2.1.2 Edit This item contains the commands to cut/copy/paste/delete any VME from the tree. Undo/redo commands are also available. Find VME allows the user to search a VME by name in the data tree. Each command has a keyboard shortcut, displayed in the menu. Finally, some settings of the application can be chosen (changes will take effect when the application restarts).

5.2.1.3 View This item contains the list of the available views. To add a new view, simply select Add View and select the desired view. It is also possible to select which of the other bars (Control Bar, Log Bar, Tool Bar and Time Bar) should be visible in the global application window. All views will be described in detail in the following.

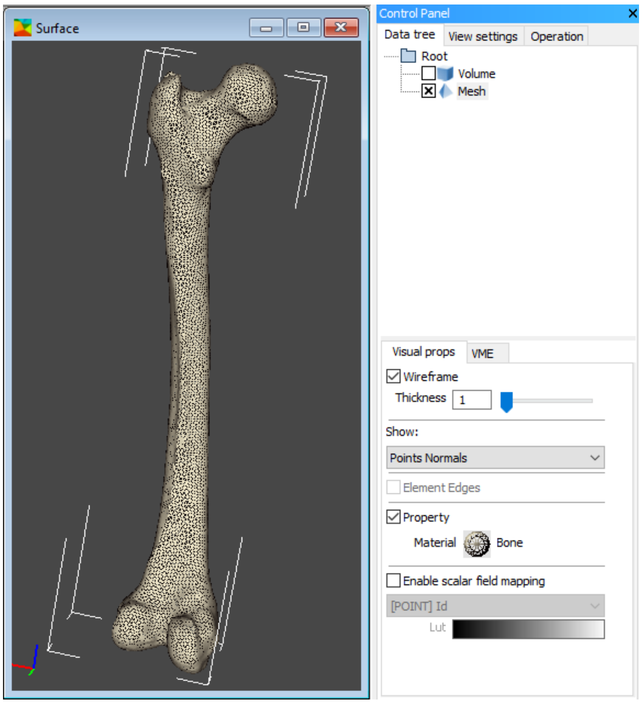
5.2.1.4 Operations This item contains a list of available operations within the application; to perform an operation, first select your input VME (if any), then select the desired Operation in Operations menu (if an operation can not be run with the selected VME as input, the operation name appears in grey). All operations will be described in detail in the following.

5.2.2 Control Panel

The Control Panel is formed by three sub-windows:

5.2.2.1 Data Tree This sub-window shows the loaded VMEs, with their hierarchical structure. For the selected VME three other tabs are active (in the bottom part of the Control bar): it shows all VMEs loaded in the application, with their hierarchical structure. The user can select only one VME at a time. For the selected VME, two other tabs, shown below the tree, are active:

- *Visual props* shows the rendering properties of the selected VME according to the active view (Figure 5).
- *vme* shows the attributes of selected VME (such as the name, the encryption, the number of sub-entities such as cells or points it is compesed of, etc) (Figure 6).



☐ Volume

☒ Mesh

☑ Wireframe

Thickness 1

Show: Points Normals

☐ Element Edges

☑ Property

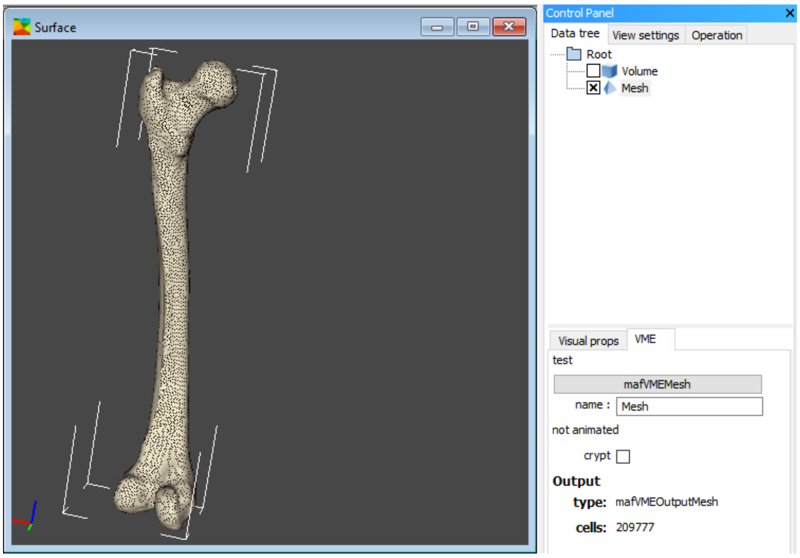
Material Bone

☐ Enable scalar field mapping

[POINT] Id

Lut

Figure 5



☐ Volume

☒ Mesh

test

mafVMEMesh

name : Mesh

not animated

crypt ☐

Output

type: mafVMEOutputMesh

cells: 209777

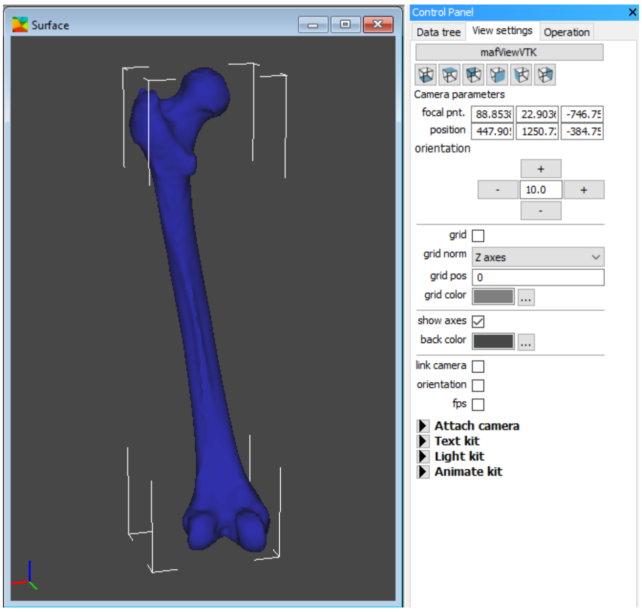
Figure 6

Any VME is represented by:

- colour
    - Blue: the correspondent VME is loaded in memory and can be used
    - Grey: the VME cannot not be displayed in the active view
- icon
    - Cube: if the data structure is a VME Volume (e.g. CT data)
    - Tetrahedron: if the data structure is a VME Mesh (e.g. FE model)

The display check box near the VME icon can be a square or a circle. If it is a circle, the view is mutually exclusive for that type of VME, which means that only one VME of that type can be display in the selected view.

5.2.2.2 View Settings This sub-window shows the settings of the active view. These can be consistently different depending on the type of active view. For example, camera parameters will be shown when a Surface view is active, slice or look-up table settings when an Orthoslice view is active. (Figure 7).



focal pnt. 88.853 22.903 -746.7
position 447.90 1250.7 -384.7
10.0
grid ☐
grid norm Z axes
grid pos 0
show axes ☑
link camera ☐
orientation ☐
fps ☐

Figure 7

5.2.2.3 Operation It shows the settings of the operation in progress.

5.2.3 Contextual Menus (i.e. mouse right-click)

5.2.3.1 View contextual menu It can be activated pressing with right mouse button in any view background. It provides specific actions for the active view (Figure 8):

- *Rename view* allows to assign a specific name to the current view;
- *Normal size/maximize* allows to restore the size view to maximised or normal; in compound views this command is available for the view as a whole or for the selected sub panel;
- *Save as image* allows to save the content of the view in a standard image format (bmp, jpg, tiff, etc.); in compound views this command is available for the view as a whole or for the selected sub panel;
- *Export as VRML* allows to save the content of the rendered view in VRML format.

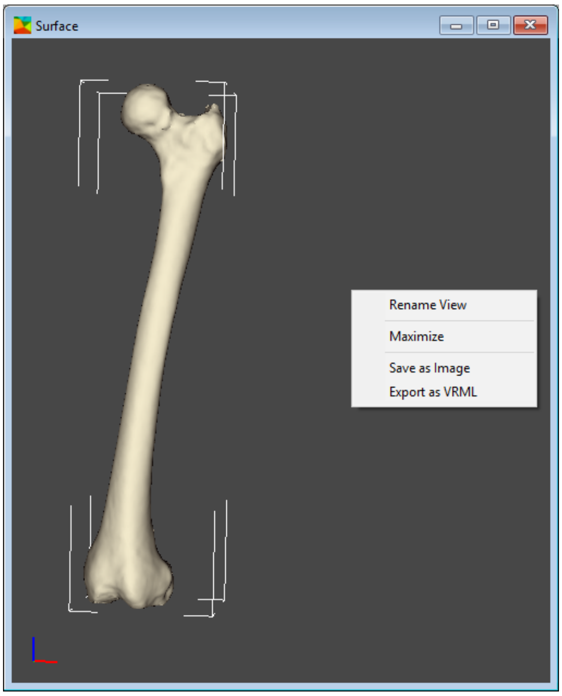
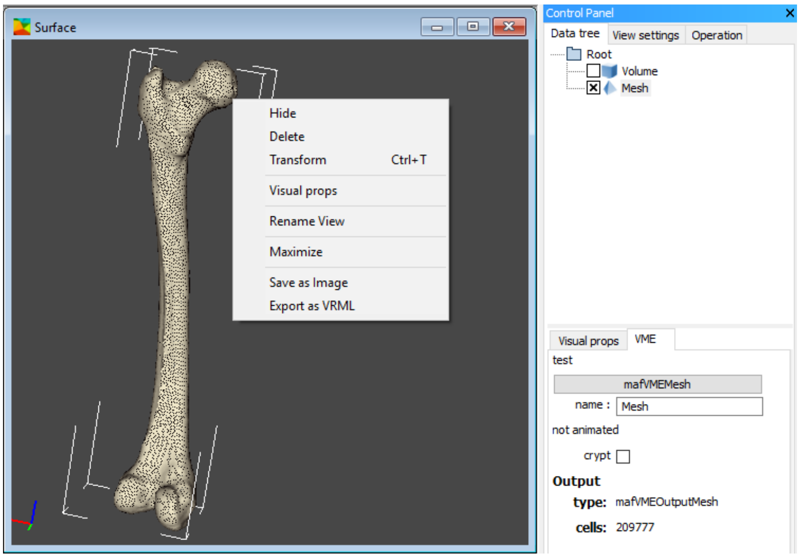


Figure 8

5.2.3.2 VME contextual menu It can be activated pressing with right mouse button on any VME displayed in any view (provided it is also the one selected in the VME tree) (Figure 9). It adds to the actions of the View contextual menu, some other actions that are specific for the VMEs:

- *Hide* removes from the display list the selected VME;
- *Delete* deletes from the Data Tree the selected VME;

- *Transform* launches the Transform operation on the selected VME.
- *Visual props* opens a GUI exposing the visual properties of the VME according to the selected View, or (in compound views) the view sub-panel.



test

mafVMEMesh

name : Mesh

not animated

crypt ☐

Output

type: mafVMEOutputMesh

cells: 209777

Figure 9

5.2.3.3 Tree contextual menu

It can be activated by pressing with the right mouse button on any VME in the VME tree (Figure 10). It provides some visualisation shortcuts:

- *Hide/show (present if the VME can be displayed in the active view)* hides/shows the selected VME;
- *Show sub-tree (present if a view is opened)* displays in the current view all the VMEs of the sub-tree related to the selected VME;
- *Show same type* it is present if the selected VME can be displayed in the active view. It displays in the current view all the VMEs of the same type of the selected one;
- *Hide sub-tree* removes from visualization in the active view all the VMEs of the sub-tree of the selected VME;
- *Hide same type* removes from visualization in the current view all the VMEs of the same type of the selected one;
- *Crypt* applies encryption to the selected VME;
- *Enable crypt sub-tree* applies encryption to all VMEs of a sub-tree.
- *Disable crypt sub-tree* removes encryption from all VMEs of a sub-tree.

- *Sort children nodes* allows to sort children nodes in alphabetical order.
- *Keep tree nodes sorted* allows to keep the alphabetical sorting even when new VMEs are added to the data tree.

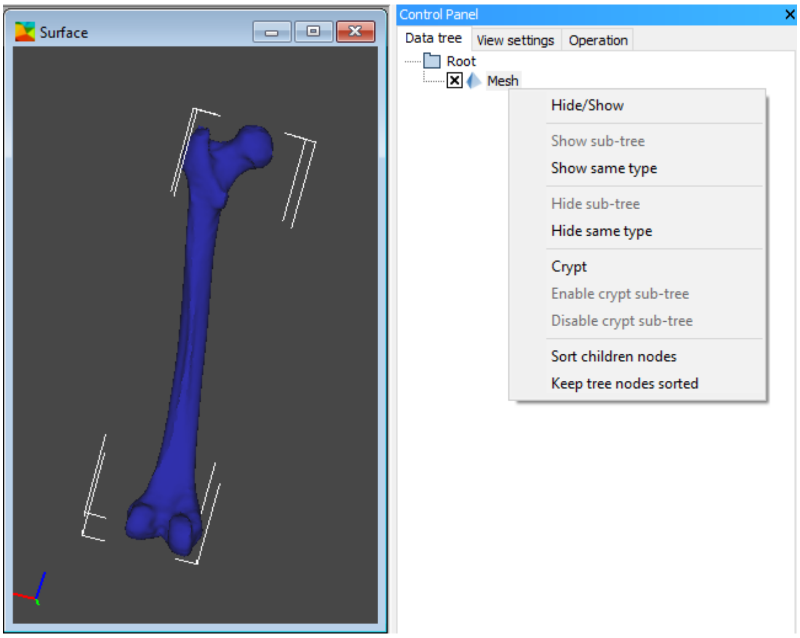


Figure 10

6 Bonemat Key Commands

6.1 Data Importers and Exporters (menu File)

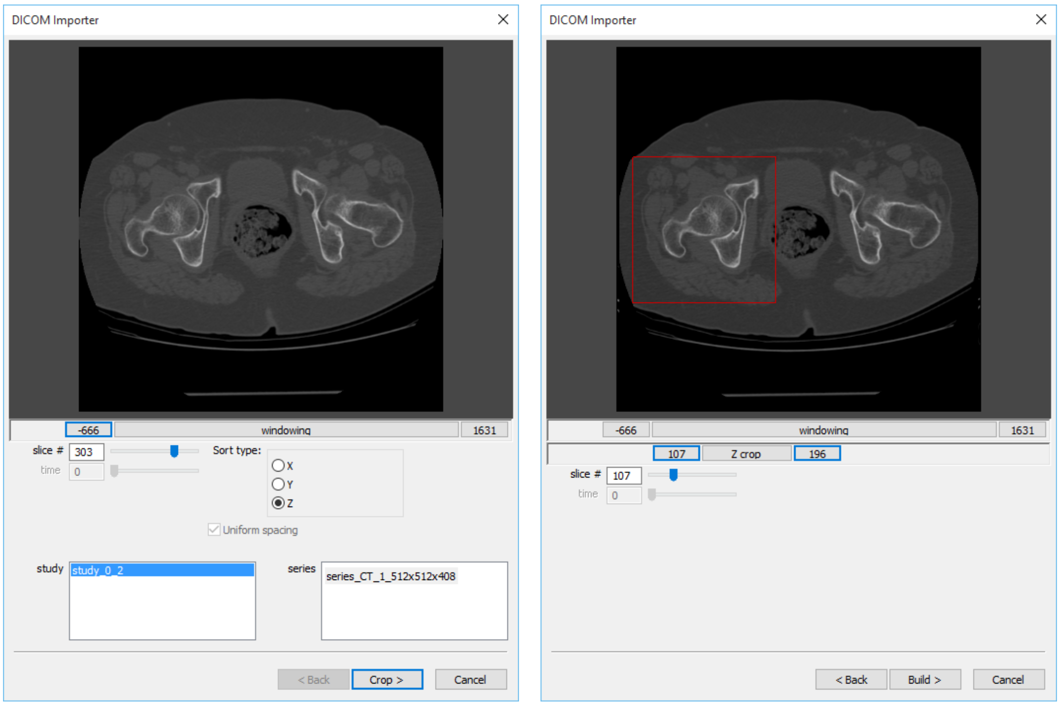
The Import and Export classes, accessible through the menu File, allow the user to import/export data to and from the application, in different formats. When in use in the Bonemat application, the files are stored in the internal MSF format.

Importers/Exporters are grouped according to the type of data they are dealing with. In Bonemat, the user can import medical images, finite element meshes, or previously stored vtk files. Imported entities can also be exported.

6.1.1 Import Dicom

Path: File → Import → Images → DICOM
This importer loads medical images stored in DICOM format, and creates as output either a Volume or Image VME (most times, a Volume VME).

The user has first to select, from a standard window for folder selection, the directory in which the Dicom files are stored. Once the folder has been selected, a new wizard window appears in which the user can subsequently:



-666 windowing 1631

slice # 303

time 0

study study_0_2

series series_CT_1_512x512x408

-666 windowing 1631

107 Z crop 196

slice # 107

time 0

(a) (b)

Figure 11

1. see the images preview (with some options to interact: choose the number of study id if multiple ones were present in the folder, slide across the slices with appropriate sliders, and perform windowing on the images) (Figure 11a).

2. proceed to the optional "Crop" stage, in which cropping is performed in the slices plane by drawing with the mouse the crop area, and in the direction orthogonal to the slices by moving the "z-crop" slider. The in-plane crop area can be changed by moving each side of the area, and controlled over the different slices with the "slice num" slider (Figure 11b).

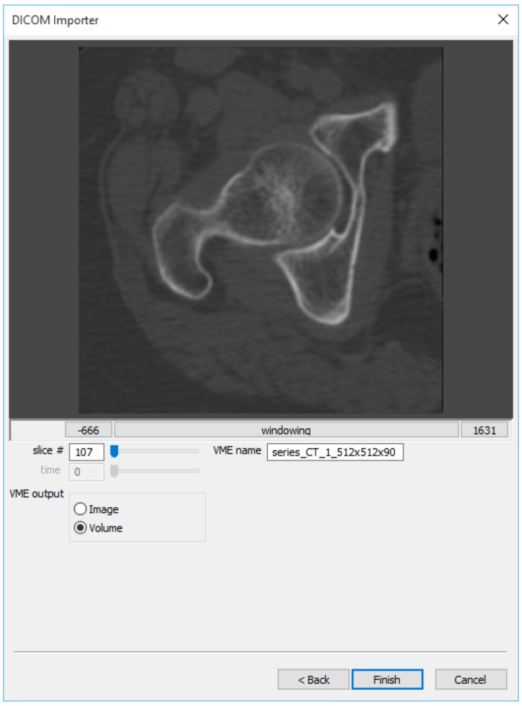
3. proceed to the "Build" stage, when only the cropped area is kept. The user can review the selected and cropped images, and choose among "Volume" (default) and "Image" the type of VME produced by the Import Operation, before completing the process by clicking "Finish" (Figure 12).

6.1.2 Import Raw Volume

Path: File → Import → Images → Raw Volume

This importer can load a raw binary volume file, i.e. a stack of images already stored in a single file (Figure 13).

A VMEVolume is created in the data tree as a result. The user needs to choose the file name (browsing with the "open" button), specify the endianity, the scalar type (char, short, int, float, double) through which the images are represented, the number of compo-



-666 windowing 1631

slice # 107

time 0

VME name series_CT_1_512x512x90

VME output
○ Image
● Volume

Figure 12

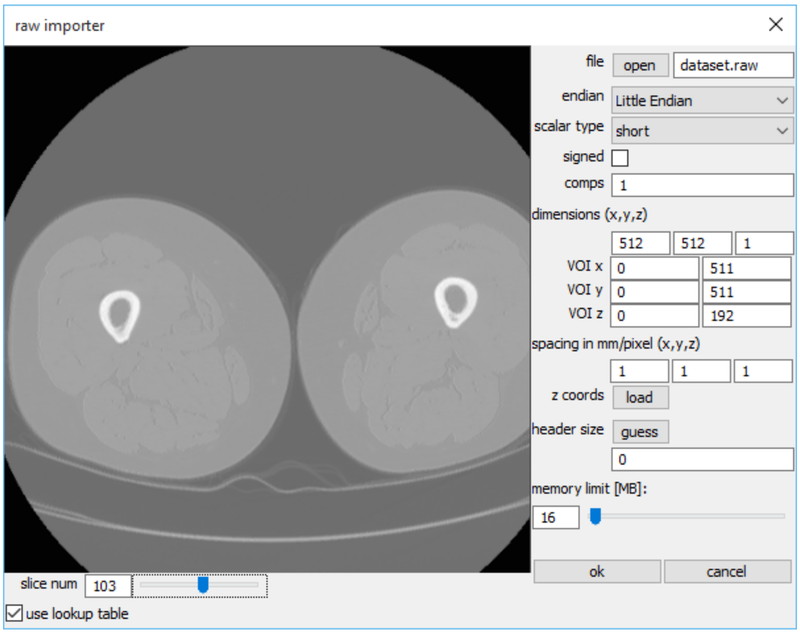
nents for each pixel value (e.g. 1 in a greyscale image, 3 in a RGB Volume), and to specify if data are signed or not (check-box). The user has also to enter the dataset dimensions (number of pixels) and the spacing in mm/pixel in all directions. The import can be limited to a part of the original volume, defining a volume of interest (VOI) through x, y and z pixel intervals (VOI x, VOI y, VOI z fields). If the spacing along the z-spacing axis is not uniform the user can specify a text file from which the Z-coordinates are read (browsing with "z-coord load" button). The Z coordinates file should be a text file whose format is like the following (the coordinates should be written in ascending order):

```
Z coordinates:
-109.1
-107.8
-104.0
...
```

Moreover, the user can set the header size to be skipped in the import process. A "guess header" function is available through the "guess" button. Finally, a VMEVolume is created in the tree upon pressing the ok button.

6.1.3 Import Raw Images

Path: File → Import → Images → Raw Images
This importer can load a stack of raw binary images, automatically creating a VMEVolume (Figure 14).. The raw images should be stored in a single directory and should be



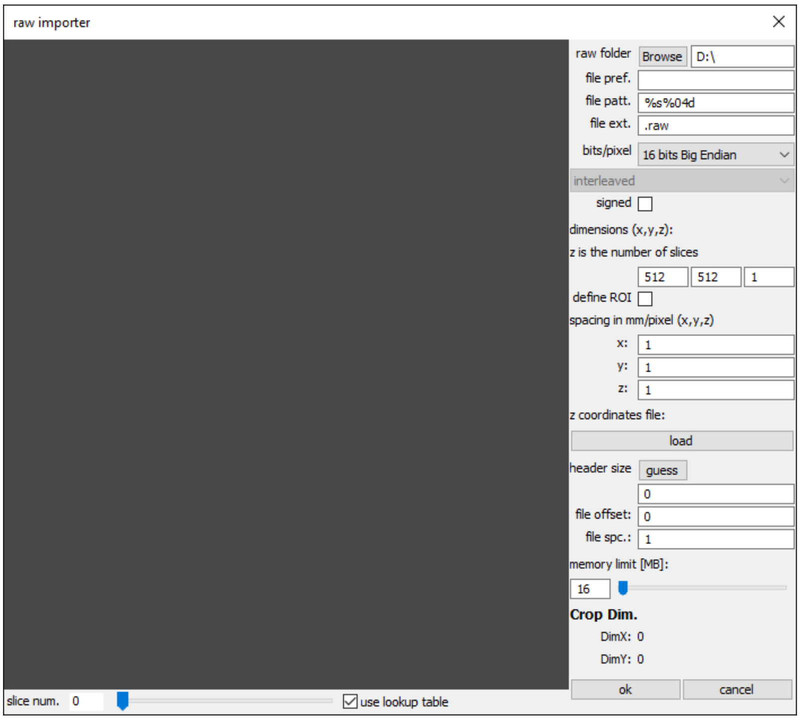
file: dataset.raw
endian: Little Endian
scalar type: short
signed ☐
comps: 1
dimensions (x,y,z): 512 512 1
VOI x: 0 511
VOI y: 0 511
VOI z: 0 192
spacing in mm/pixel (x,y,z): 1 1 1
header size: 0
memory limit [MB]: 16
slice num: 103
☑ use lookup table

Figure 13

named with the same prefix and a progressive number (e.g. RawImage_0006.raw, RawImage_0007.raw, etc.). The user needs to specify the name of single files in the folder through the prefix (e.g. RawImage_), the pattern (e.g. %s%04d) and the extension of the files (e.g. .raw). The user should also specify if the data stored in the images are 8bits, 16 bits Little Endian, 16 bits Big Endian, 24 bits RGB, and whether they are signed or not. In case of 24 bits images also the presence of interleaving should be specified. The user should then set the overall dimensions of the volume in pixels and the spacing along the axes in mm/pixel. Ticking on a checkbox, a ROI can optionally be specified on the XY plane, drawing and adjusting a red box with left mouse button (ROI dimensions are shown underneath). If the spacing of the images is not uniform in the Z direction, the user can import a text file with the Z coordinates, just like in the Import Raw Volume. Moreover, the user can set the header size to be skipped in the import. The user can set an offset for the first slice number ("file offset" field) and a spacing in the numbering of the images ("file spc" field) if different from unity.
If all the information provided are compatible with the actual files fed in the application, a preview of the images will appear in the window, to be optionally scrolled through the "slice num" slider. Finally, a VMEVolume is created in the tree upon pressing the ok button.

6.1.4 Import VTK

Path: File → Import → Other → vtk
The VTK importer loads data stored in vtk format. The type of VME created into the data tree depends on the information contained into the vtk file.



raw folder: D:\
file pref.:
file patt.: %s%04d
file ext.: .raw
bits/pixel: 16 bits Big Endian
interleaved
signed ☐
dimensions (x,y,z):
z is the number of slices
512 512 1
define ROI ☐
spacing in mm/pixel (x,y,z)
x: 1
y: 1
z: 1
z coordinates file:
header size: 0
file offset: 0
file spc.: 1
memory limit [MB]: 16
DimX: 0
DimY: 0
slice num. 0
☑ use lookup table

Figure 14

6.1.5 Import Generic Mesh

Path: File → Import → Finite Element → Generic Mesh
This importer allows the creation of a VMEmesh from two files: (i) (nodes file) a matrix of nodes IDs and coordinates, (ii) (elements file) a matrix of element IDs and nodal connectivity. (Figure 15)

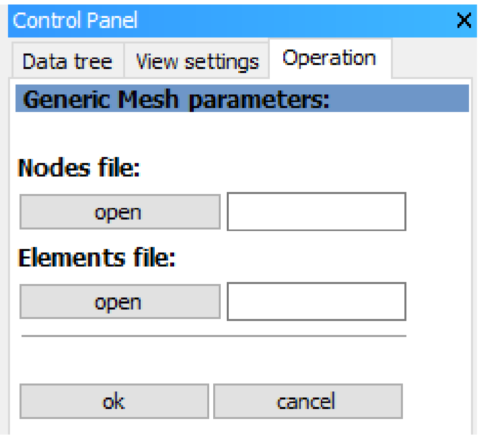


Figure 15

i. The file defining the list of nodes should be a matrix, where: the first column indicates the node number, columns from second to fourth define the x,y,z, nodal coordinates.

ii. The file defining the list of elements should be a matrix, where: the first column indicates the ID of the element; the following columns indicate the nodal connectivity. Depending on the number of nodes listed for each element, elements will be interpreted as linear tetrahedra (4 nodes per element), linear hexahedra (8 nodes), or quadratic tetrahedra (10 nodes) or hexahedra (20 nodes). Please consider that currently meshes should be homogenous, i.e. each model should be made of elements of the same type.

6.1.6 Import Ansys CDB File

Path: File → Import → Finite Element → Ansys CDB File
This importer creates a VMEmesh directly from an Ansys ASCII archive (.cdb file). The importer can handle an FE mesh composed by N components. In this case, N VMEs will be created in the VME tree, corresponding to the N components of the mesh. The user can eventually merge them by using the Merge Mesh Operation, if he needs to have a single VMEmesh as the original model (for more detail please look at section 6.3.5). The importer currently supports only the following entities: nodes, linear and quadratic solid elements, and material cards. In particular, the following ANSYS elements are supported: SOLID285 (for 4-noded tetrahedral elements), SOLID45 (for 8-noded hexahedral elements), SOLID187 (for 10-noded tetrahedral elements) and SOLID186 (for 20-noded hexahedral elements).

6.1.7 Import Ansys Input File

Path: File → Import → Finite Element → Ansys Input File
This importer creates a VMEmesh directly from an Ansys ASCII input file. The importer can handle an FE mesh composed by N components. In this case, N VMEs will be created in the VME tree, corresponding to the N components of the mesh. The user can eventually merge them by using the Merge Mesh Operation, if he needs to have a single VMEmesh as the original model (for more detail please look at section 6.3.5). The importer currently supports only the following entities: nodes, linear and quadratic solid elements, and material cards. In particular, the following ANSYS elements are supported: SOLID285 (for 4-noded tetrahedral elements), SOLID45 (for 8-noded hexahedral elements), SOLID187 (for 10-noded tetrahedral elements) and SOLID186 (for 20-noded hexahedral elements).

6.1.8 Import Abaqus File

Path: Import → Finite Element → Abaqus File
The ABAQUS input file importer has been designed to import an ABAQUS Input File in the same format that is written automatically by the ABAQUS CAE. This means that the input file must contain part, assembly and instance definitions, but please do not use all capitals for the keywords (e.g. use *Assembly instead of *ASSEMBLY). If the instance has been transformed (rotated or translated) within the assembly, the script will note the transform information from the assembly parts of the input file and calculate the material assignment using transformed nodal co-ordinates. Any defined regions (such as node or

element sets), or boundary conditions (such as constraints or loads), can be left in the file – these will not influence the node or element import and the lines will simply be re-written in the output file.

6.1.9 Export Generic Mesh

Path: File → Export → Finite Element → Generic Mesh
This exported is symmetric to the Import Generic Mesh feature. A VMEmesh is translated into two (or three if materials are present) files: (i) a matrix of nodes IDs and coordinates, (ii) a matrix of element IDs, element properties and nodal connectivity, and optionally (iii) an array of material properties, formatted according to Ansys format for material card definition, but easy to interpret for any user. All rules for the sintaxis of the nodes and elements files are the same as for the importer *Import Generic Mesh.*

6.1.10 Export Ansys CDB File

Path: File → Export → Finite Element → Ansys CDB File
This exporter creates an ASCII file that can be read by Ansys as an Ansys archive file (.cdb). The exporter can deal with the basic finite element entities supported by the Bonemat application: nodes, linear and quadratic solid elements, and material cards usually containing Young's modulus, density, and Poisson's ratio values. In the exported file, elements are grouped into Ansys components according to either element type and material properties. Currently, Bonemat supports the following ANSYS elements: SOLID285 (for 4-noded tetrahedral elements), SOLID45 (for 8-noded hexahedral elements), SOLID187 (for 10-noded tetrahedral elements) and SOLID186 (for 20-noded hexahedral elements).

6.1.11 Export Ansys Input File

Path: File→ Export → Finite Element → Ansys Input File
This exporter creates an ASCII input file (.inp) that can be directly read by Ansys program. The exporter can deal with the basic finite element entities supported by the Bonemat application: nodes, linear and quadratic solid elements, and material cards usually containing Young's modulus, density, and Poisson's ratio values. In the exported file, elements are grouped into ANSYS components according to either element type and material properties. Currently, Bonemat supports the following ANSYS elements: SOLID285 (for 4-noded tetrahedral elements), SOLID45 (for 8-noded hexahedral elements), SOLID187 (for 10-noded tetrahedral elements) and SOLID186 (for 20-noded hexahedral elements).

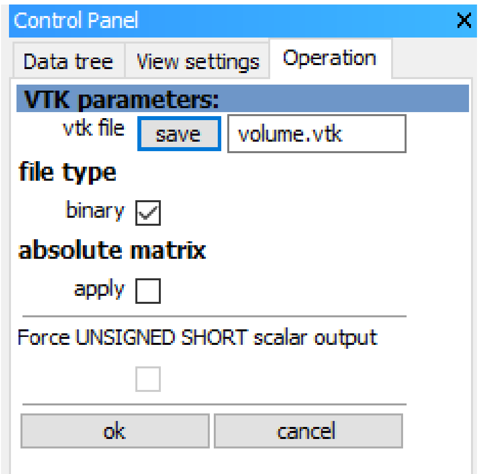
6.1.12 Export Abaqus File

Path: File →Export →Finite Element →Abaqus File
The ABAQUS exporter will create an ABAQUS input file which will be a copy of the one used for the input file, but with extra lines containing the material definitions. It will have the same name but with 'MAT' added to the end. The extra lines which define the new element sets (which specify which elements correspond to which material section) and the new element sections will be included just before the *End Part keyword. Extra

lines defining the materials and their properties will be added after the *End Assembly keyword. All materials will be named ‘BoneMat_’ followed by a reference number, and all element sets will be named ‘BoneMatSet_’ followed by a reference number. To avoid a conflict, please ensure that you do not create element sets or materials using the same naming convention when creating your ABAQUS input file.

6.1.13 Export VTK

Path: File → Export → Other → VTK
Each VME can be exported as its vtk representation. The user can choose whether to export the binary or the ASCII file (Figure 16).



VTK parameters:
vtk file: volume.vtk
file type
binary ☑
absolute matrix
apply ☐
Force UNSIGNED SHORT scalar output
☐

Figure 16

6.2 Views (Menu View)

Every VME can be visualized and managed through a combination of views and operations. Many instances of the same view with different VMEs displayed and/or different views can be opened at the same time according to the user’s needs, and to memory usage limits of the computer. Views can be single, or composed by different sub-views (visualization panels), (Figure 17). Each visualization panel views can be (i) perspective, i.e visualizing a VME in space through a camera that can be roto-translated or zoomed, or (ii) slice, i.e. slicing a VME through a defined plane (Figure 17).

Special interaction objects (Gizmos, i.e. particular handles instantiated by a View or an Operation to allow direct interaction with the mouse) are present in the slice views to facilitate the definition and the fine tuning of the view settings (Figure 18).

6.2.1 View Arbitrary

The Arbitrary View is a compound view made of two panels. In the left panel, the user can change using gizmos the position and the orientation of an arbitrary plane; in the right

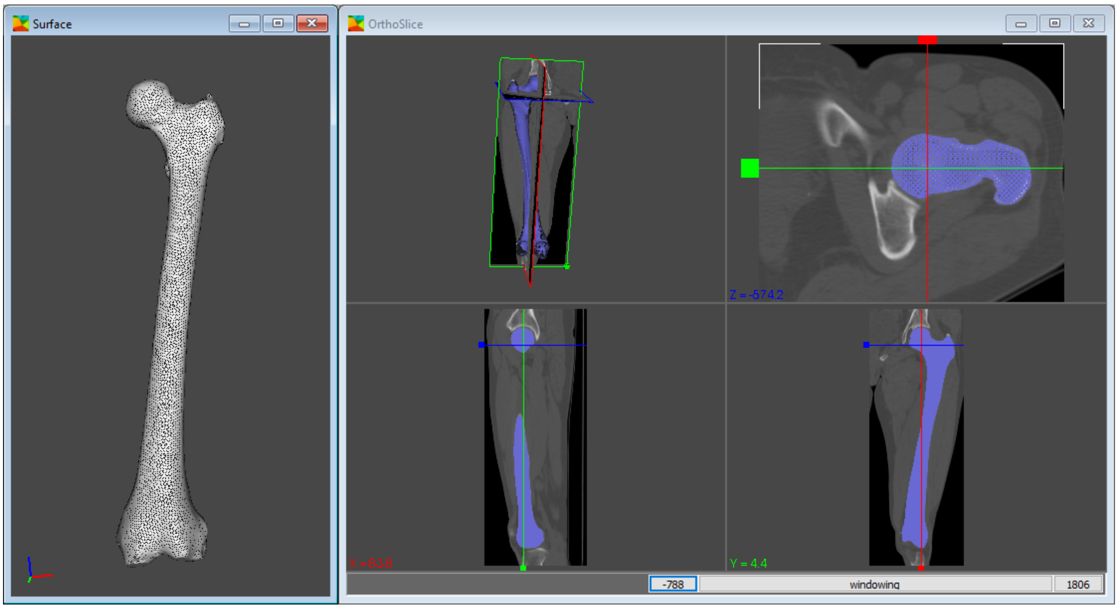


Figure 17

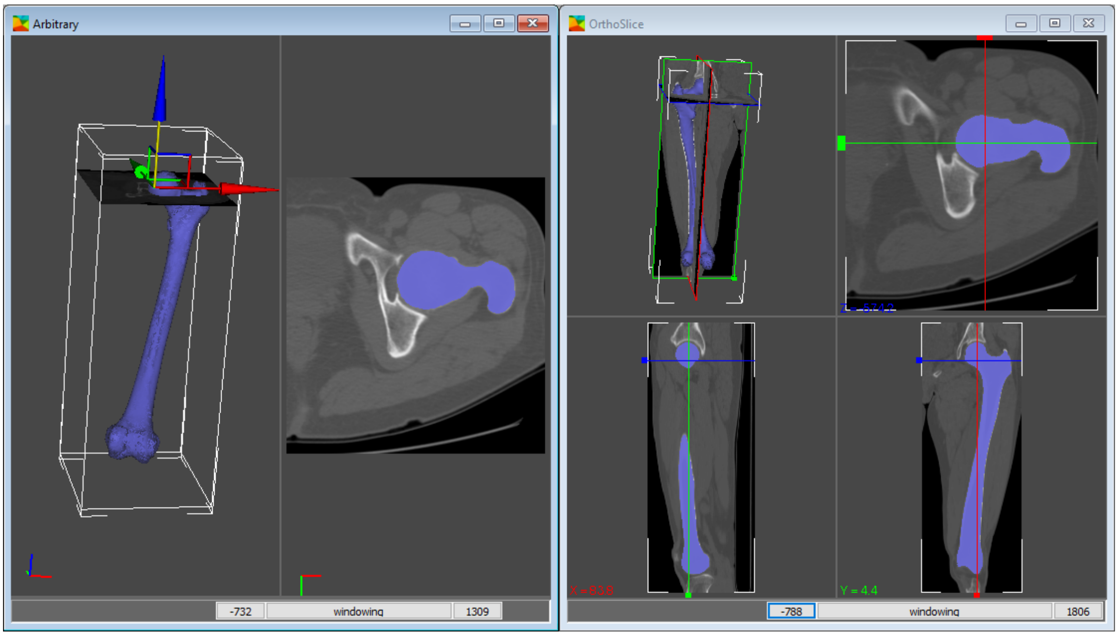


Figure 18

panel the corresponding volume section is showed in a 2D parallel (slice) view (Figure 19).

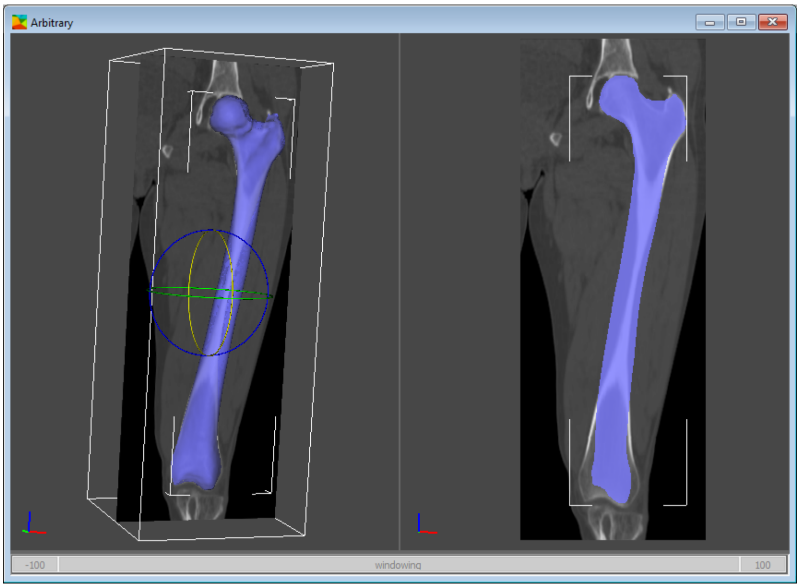


Figure 19

The Arbitrary View supports the visualization of:

- VMEvolume, one at a time. Windowing for the volume scalar is available. It applies to both view panels.
- VMEmesh, even more than one simultaneously, with a 3D rendering in the left volumetric view panel, and the sliced contour in the right slice view panel.
  VMEmesh Visual properties ("Visual props" panel accessible through contextual menu): in this panel, the user manages the visualization options of the selected VMEmesh. The following command/options are available (Figure 21).
  - Wireframe: this checkbox enables wireframe visualization of the mesh. The user can adjust the edges thickness, by editing the number or acting on the dedicated slider.
  - Lighting: from the drop down menu, the user can change the way normals are calculated (on the cells or on the vertices) when displaying the mesh. "Points Normals" (default) gives a smoothed visualization, while "Cells Normals" shows discontinuities at edges.
  - Element edges: this checkbox enables the visualization of the edges of the mesh.
  - Property: this checkbox applies several rendering properties, called "Material" for brevity. Under the checkbox there is a circle with the material preview. Upon clicking on it, another window is opened (Figure 20), in which it is possible to:

        - Choose a material from a library
        - Modify the current material
        - Store or remove the current material from the library
        - Reset the library at the default list

    The changes at the current material are saved in the list only by clicking "Store current material in Library" but the material appearance is automatically applied to the mesh by clicking "OK". It is possible to modify the following material properties:

        - Name of the material
        - Ambient color and intensity
        - Diffuse color and intensity
        - Specular color, intensity and power
        - Opacity
        - Wireframe

    - Enable scalar field mapping: this checkbox is active if the selected VMEmesh has one or more scalar fields associated to its vertices/cells. If checked, it enables the mapping of a scalar field onto the mesh. The user can select from the drop down menu the scalar field to visualize. Then, clicking on the Lut (Look-up table) bar a window opens, letting the user choose/modify the contouring color map, value range, number of contours, and color/value associations.

The view properties can be changed from the "View settings" panel (Figure 22):

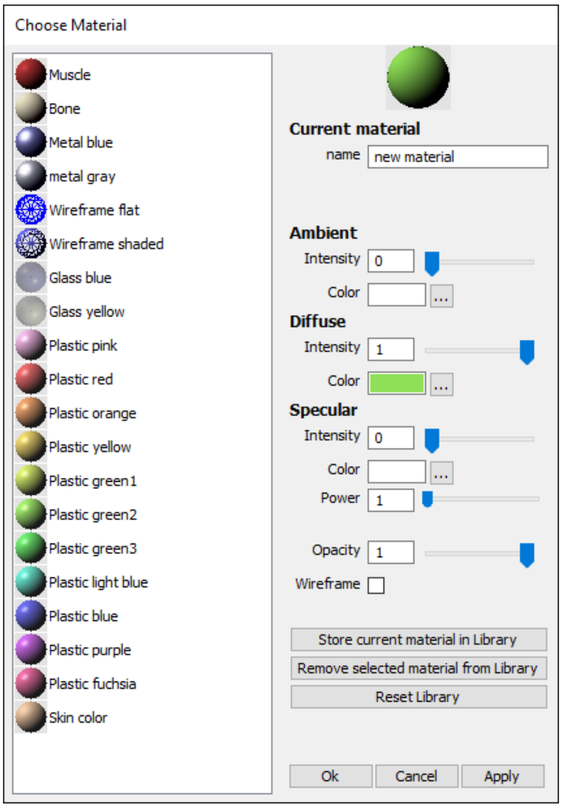
i. The user can change the position and orientation of the visualized slice through Gizmo or Text interaction.
"Gizmo Interaction": it is possible to translate or rotate the slice with three arrows (for the translation) or three rings (for the rotations). By selecting the desired gizmo and holding pressed the left button of mouse, the gizmo moves and the plane is updated accordingly.
"Text Interaction": The user can write the value in the corresponding text box, and restore the initial plane position with the "reset" button.

ii. "Reset" button: the user can reset the initial position of the slice

iii. "LUT": the user can set or define the look-up table to be assigned to the scalars of the VMEVolume currently visualized.

6.2.2 View OrthoSlice

The OrthoSlice View is a compound view made of four panels and three gizmos. A perspective panel shows a volume intersected by the three orthogonal planes and the other three panels show a parallel view of each other slice (Figure 23a). The Orthoslice View supports the visualization of:



Choose Material

Current material
name: new material

Ambient
Intensity: 0
Color: ...

Diffuse
Intensity: 1
Color: ...

Specular
Intensity: 0
Color: ...
Power: 1

Opacity: 1
Wireframe ☐

Figure 20

- VMEvolume, one at a time. Windowing for the volume scalar is available. It applies to all view panels.
  VMEvolume visual properties ("Visual props" panel accessible through contextual menu): the user can change or define the LUT of the volume scalars, and its opacity in the visualization; the x,y,z quotes can also be changed manually (Figure 23b)

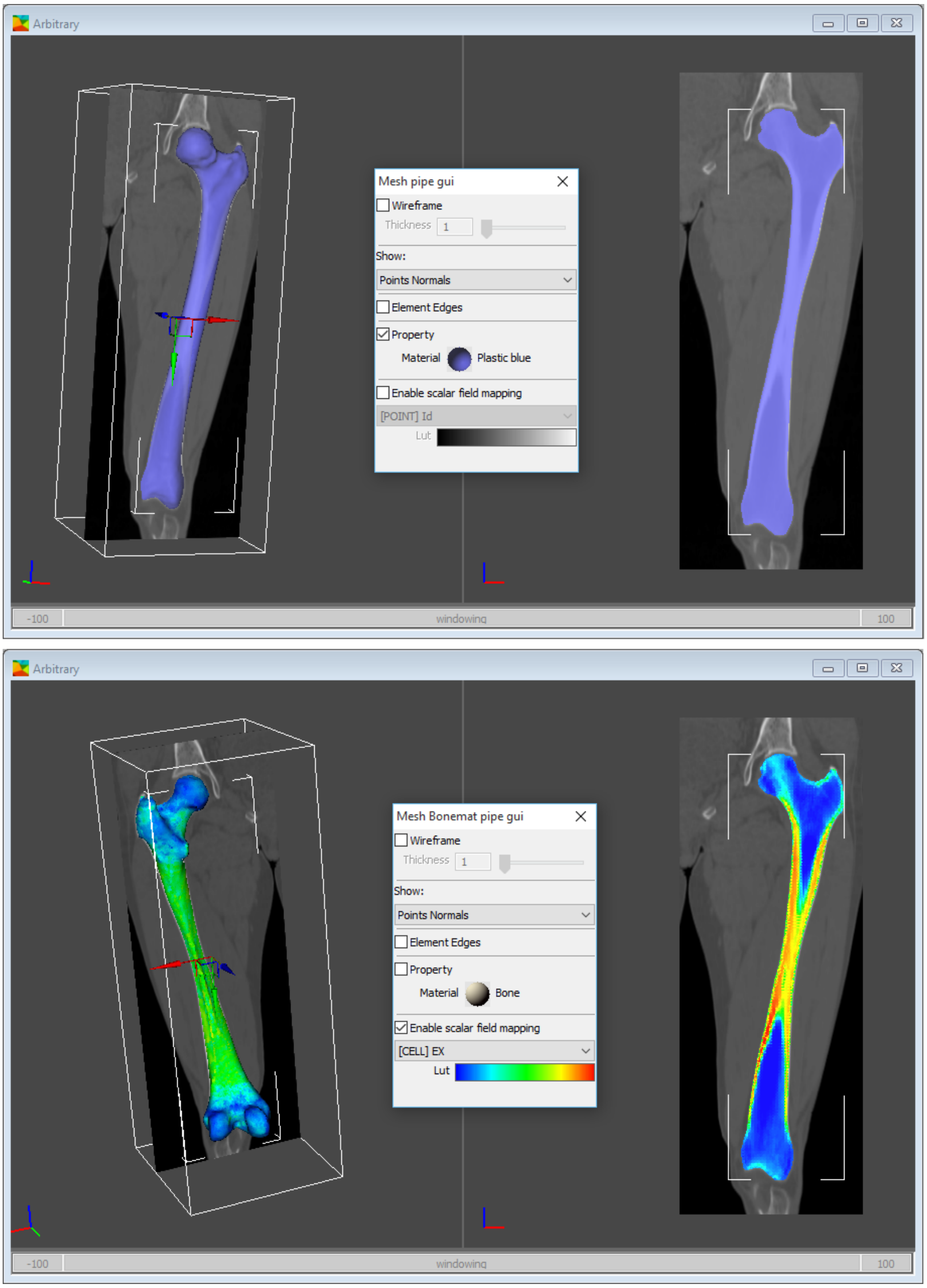
- VMEmesh, even more than one simultaneously, with a 3D rendering in the perspective view panel, and the sliced contour in the orthogonal slice panels.
  VMEmesh Visual properties: for further explanation look at the paragraph 6.2.1.

The view properties can be changed from the "View settings" panel (Figure 24):

i. "layout": the user can change the layout of the panels in the view.

ii. "LUT": the user can set or define the look-up table to be assigned to the scalars of the VMEVolume currently visualized.

iii. "Snap on grid": the user can select whether to avoid interpolation between slices and perform slice cuts only at values of the original grid.



Mesh pipe gui

☐ Wireframe

Thickness 1

Show:

Points Normals

☐ Element Edges

☑ Property

Material Plastic blue

☐ Enable scalar field mapping

[POINT] Id

Lut

Mesh Bonemat pipe gui

☐ Wireframe

Thickness 1

Show:

Points Normals

☐ Element Edges

☐ Property

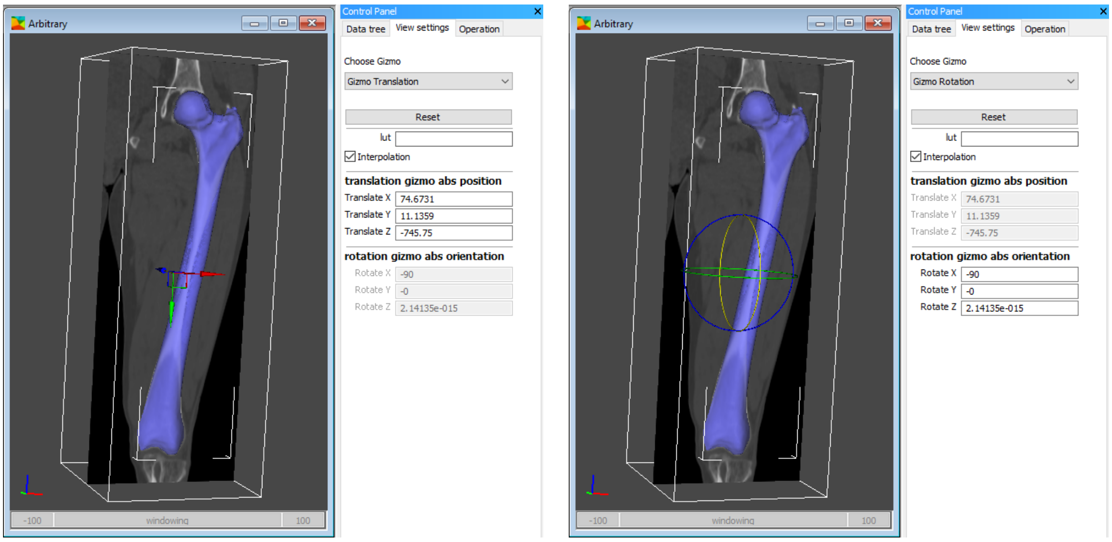
Material Bone

☑ Enable scalar field mapping

[CELL] EX

Lut

Figure 21



Choose Gizmo: Gizmo Translation
lut:
☑ Interpolation
translation gizmo abs position
Translate X: 74.6731
Translate Y: 11.1359
Translate Z: -745.75
rotation gizmo abs orientation
Rotate X: -90
Rotate Y: -0
Rotate Z: 2.14135e-015

Choose Gizmo: Gizmo Rotation
lut:
☑ Interpolation
translation gizmo abs position
Translate X: 74.6731
Translate Y: 11.1359
Translate Z: -745.75
rotation gizmo abs orientation
Rotate X: -90
Rotate Y: -0
Rotate Z: 2.14135e-015

Figure 22

6.2.3 View RX-CT

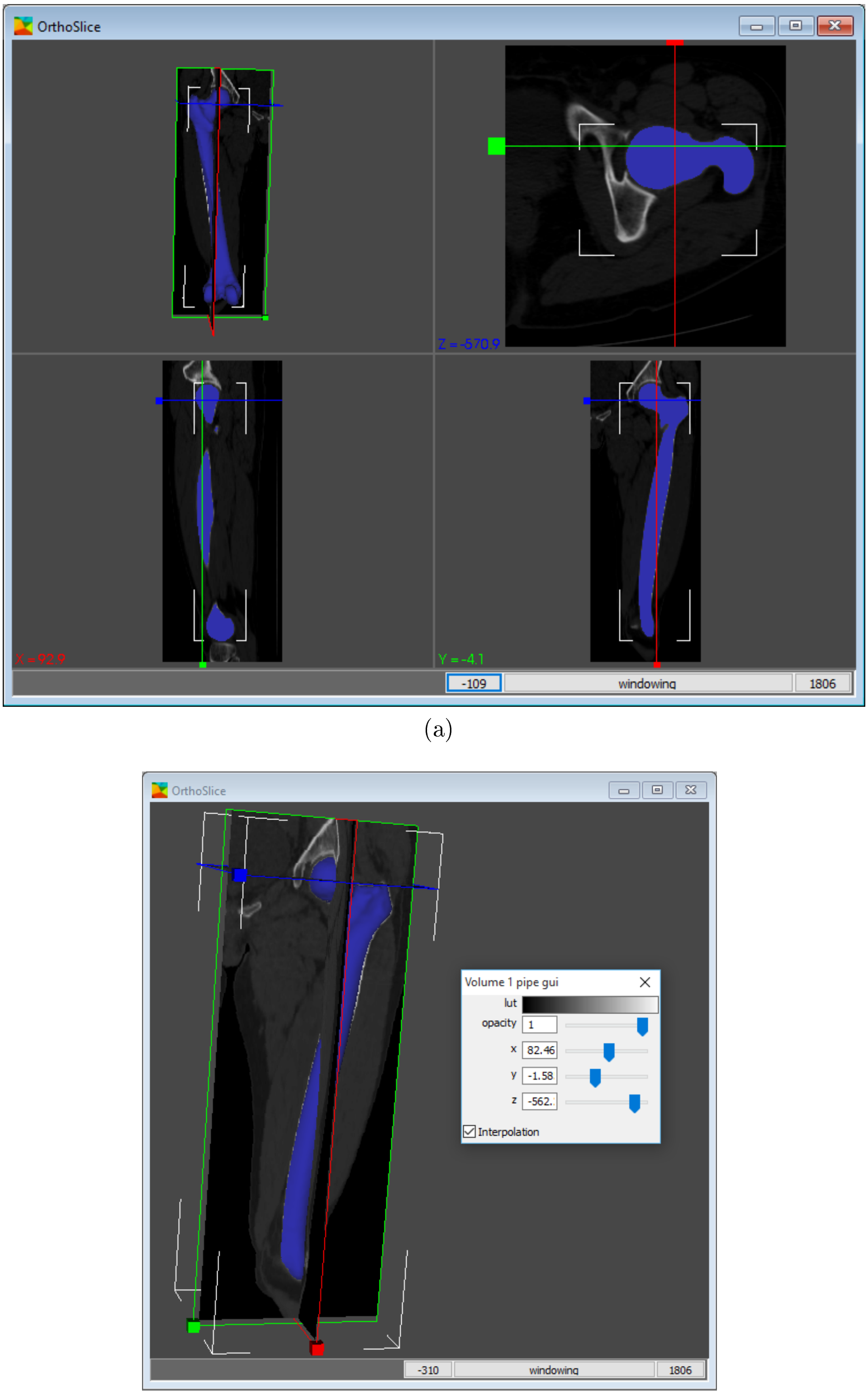
The RXCT View is a compound view made of eight 2D panels: two planar projections in the XZ and YZ planes, and six slices in the XY plane, whose Z-quotes are identified in the two planar projections by six gizmos (Figure 25).

The RX-CT View supports the visualization of:

- VMEvolume (one at a time), displayed in the projection panels as a digitally reconstructed radiograph, usually resembling anatomical planes projections (e.g. Anterior/Posterior and Medial/Lateral), and sliced along the Z direction in the slice panels.
  VMEvolume visual properties ("Visual props" panel accessible through contextual menu): the user can change or define the LUT of the volume scalars, and its opacity in the visualization; the z quote can also be changed manually (Figure 26).
- VMEmesh (even more than one simultaneously), as a 3D rendering in the projection panels and as sliced contours in the slice panels.
  VMEmesh Visual properties: for further explanation look at the paragraph 6.2.1.

The view properties can be changed from the "View setting" panel (Figure 27):

- "side" (Left/right) changes the projection direction of the lateral projection
- "snap on grid": the user can select whether to avoid interpolation between slices and perform slice cuts only at values of the original grid. The user can change the layout of the panels in the view
- "move all": by checking it, the user moves all gizmos simultaneously
- "reset slices" brings back all slices to their original positions



opacity 1

x 82.46

y -1.58

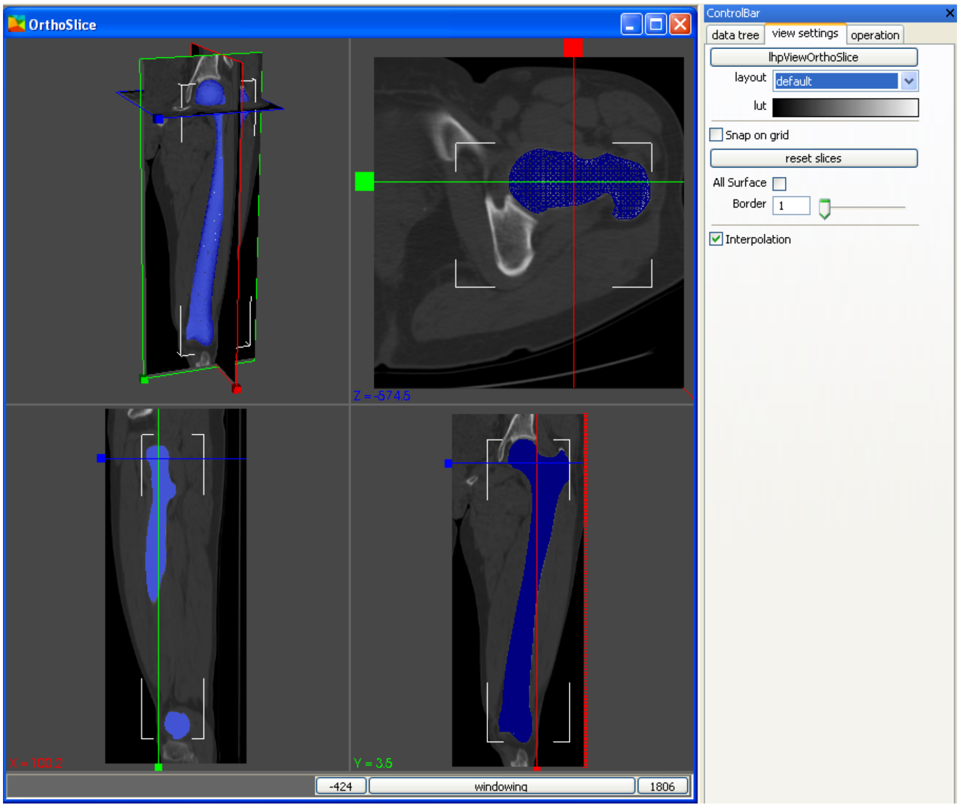
z -562.

☑ Interpolation

(a)

(b)

Figure 23



layout: default

☐ Snap on grid

All Surface ☐

Border: 1

☑ Interpolation

-424 windowing 1806

Figure 24

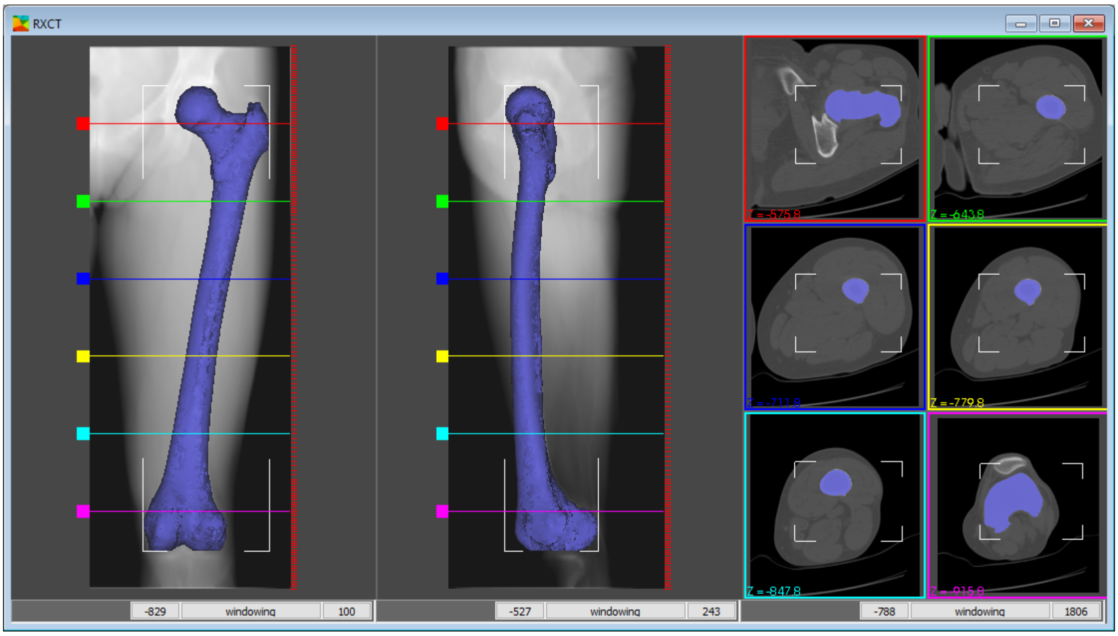
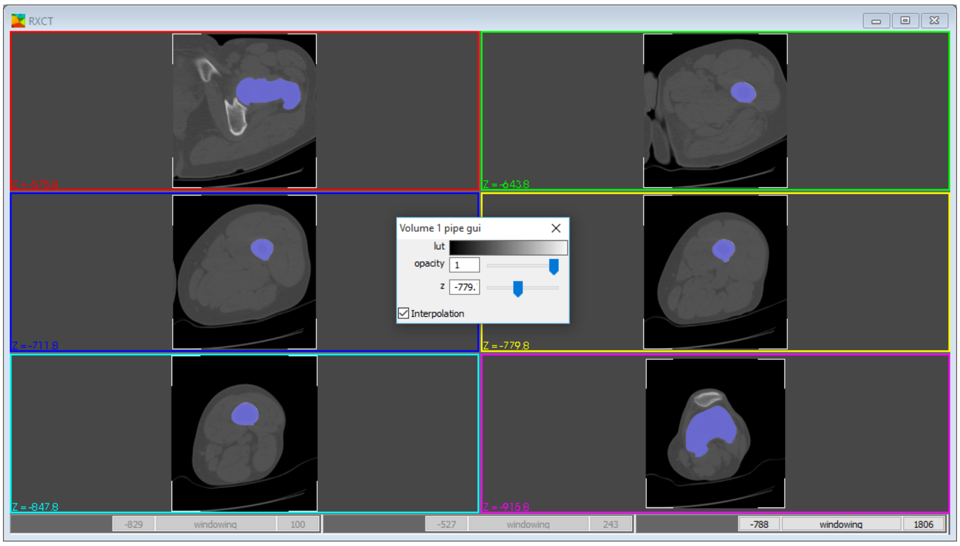


Figure 25



Volume 1 pipe gui

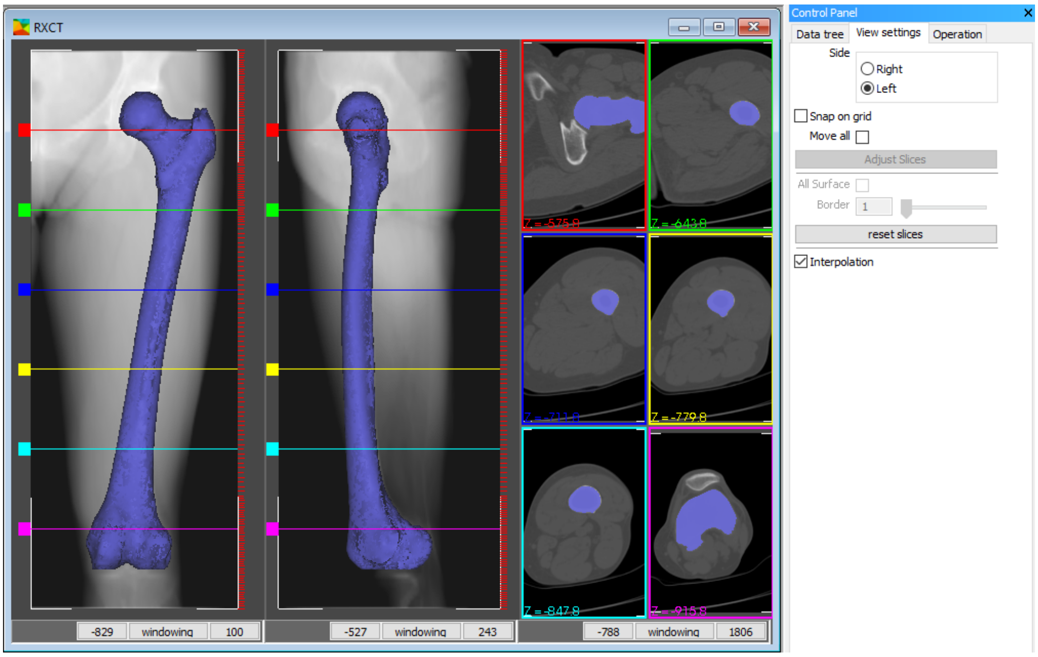
lut

opacity 1

z -779.

☑ Interpolation

Figure 26



Side

○ Right

● Left

☐ Snap on grid

Move all ☐

All Surface ☐

Border 1

☑ Interpolation

Figure 27

6.2.4 View Surface

The Surface View is a three-dimensional perspective render window (Figure 28).

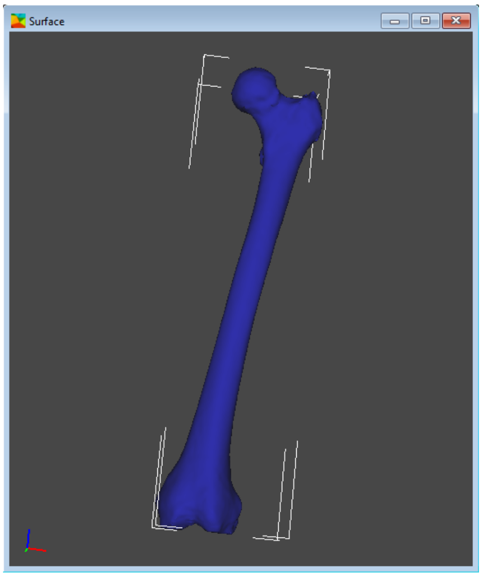


Figure 28

The Surface View supports in Bonemat the visualization of:

- VMEvolumes, displayed only with their bounding box.
- VMEmesh ("Visual props" panel of the control bar, below the VME tree, or "Visual props" in the contextual menu). For further explanation see VMEmesh visual properties in the paragraph 6.2.1.

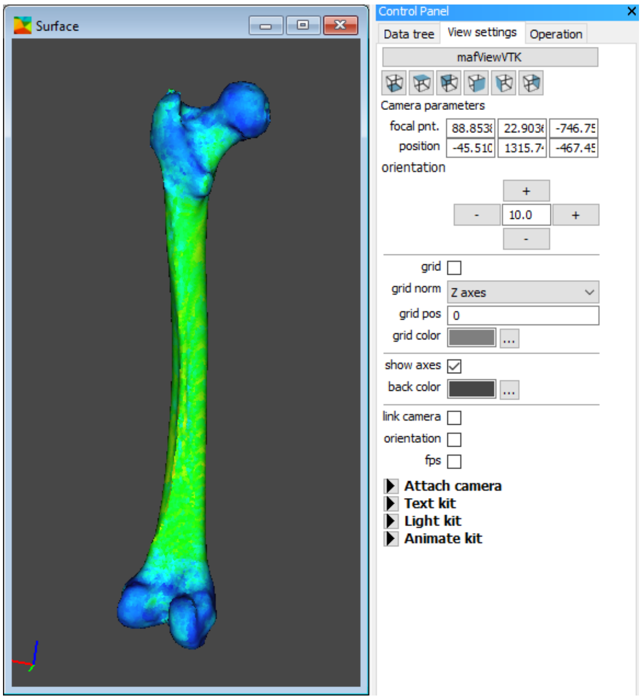
The view properties can be changed from the "View settings" panel (Figure 29):

- "camera parameters": the user can control the camera with text entries rather than with the mouse, if a predefined exact camera position has to be set;
- "grid" commands make it possible to visualize the grid or the axes and modify their colours
- "show axes" shows or hides the reference system gizmo
- "back color" lets the user set the background colour. The other settings below are part of Surface View in MAF but are of little use in Bonemat.

6.3 Operations (menu Operations)

6.3.1 Bonemat

This Operation is the core of the application. The Bonemat operation allows the user to assign to each finite element of a bone mesh an average material property derived from



focal pnt. 88.853 22.903 -746.75
position -45.51 1315.7 -467.45
10.0
grid ☐
grid norm Z axes
grid pos 0
grid color ▨ ...
show axes ☑
back color ■ ...
link camera ☐
orientation ☐
fps ☐

Figure 29

the Hounsfield Unit (HU) of the tissue in that region, as reported in the CT dataset, thus generating an inhomogeneous FE model based on the density information contained in the CT. The most common (and here implemented) procedure to relate CT data to bone mechanical properties is to extract a density value from the CT numbers (densitometric calibration achieved by scanning a phantom) and from this calculating an elastic modulus by applying a density-elasticity relationship. An additional correction to the densitometric calibration parameters may be introduced, since it has been shown that densitometric phantoms are not free of errors when mimicking bone characteristics. The core of the Bonemat algorithm is the numerical integration that maps the voxel-wise properties of the CT grid to element-wise properties of a (unstructured) mesh grid.

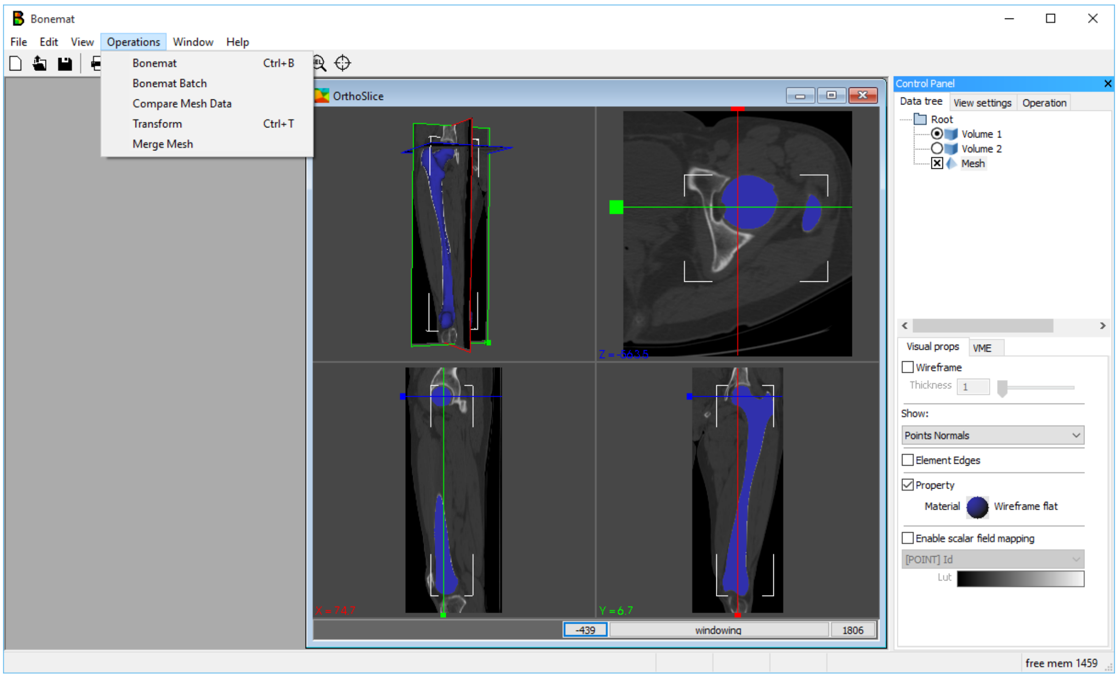
The input data are a VMEvolume and a VMEmesh (the mesh has to fit within the volume bounding box). The operation is accessible when a VMEmesh is selected (Figure 30).

The output of the operation is an updated VMEmesh in which each element has been assigned an elastic modulus value on the basis of CT data. The elements are grouped by their material card (Figure 31).

An additional output is given, in terms of a so called frequency file, which lists how many elements share the same elastic modulus, for all the sampled elastic moduli in the model.

In the graphical interface, the selected VMEmesh is taken as primary input. Then the user has to specify the secondary input (VMEvolume). If a unique VMEvolume is present in the whole MSF file, that VMEvolume is automatically taken as secondary input, otherwise the user can select a VMEvolume browsing the data tree (Figure 32).

The user has then to define some operation parameters, related either to the material



☐ Wireframe

Thickness 1

Show: Points Normals

☐ Element Edges

☑ Property

Material Wireframe flat

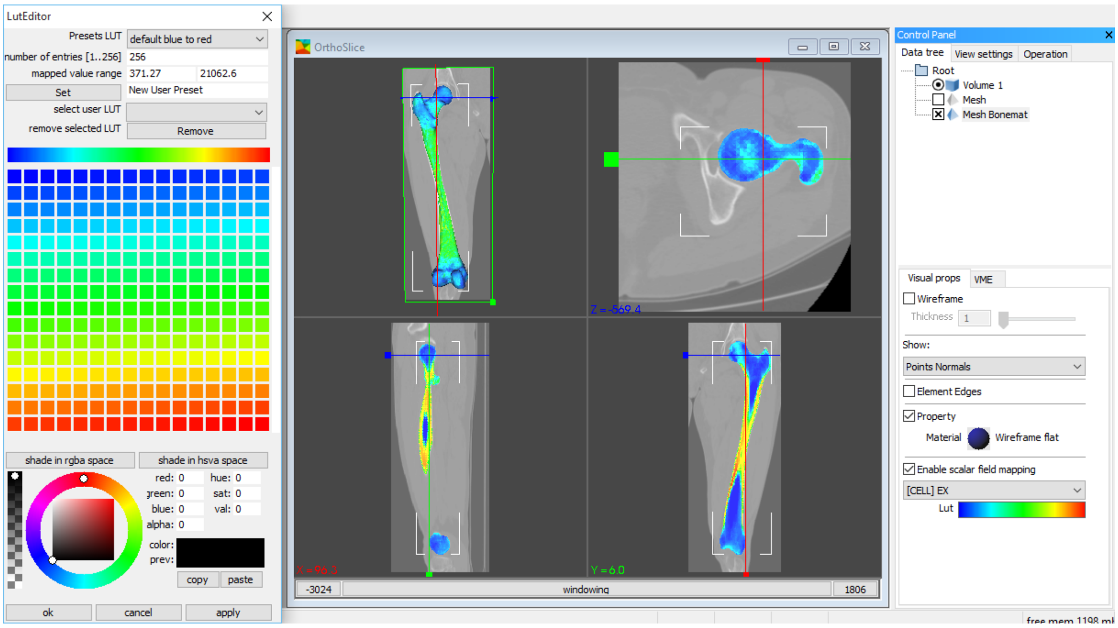
☐ Enable scalar field mapping

[POINT] Id

windowing -439 1806

free mem 1459

Figure 30



Presets LUT: default blue to red

number of entries [1..256]: 256

mapped value range: 371.27 21062.6

New User Preset

select user LUT

remove selected LUT

red: 0 hue: 0

green: 0 sat: 0

blue: 0 val: 0

alpha: 0

color:

prev:

☐ Wireframe

Thickness 1

Show: Points Normals

☐ Element Edges

☑ Property

Material Wireframe flat

☑ Enable scalar field mapping

[CELL] EX

windowing -3024 1806

Figure 31

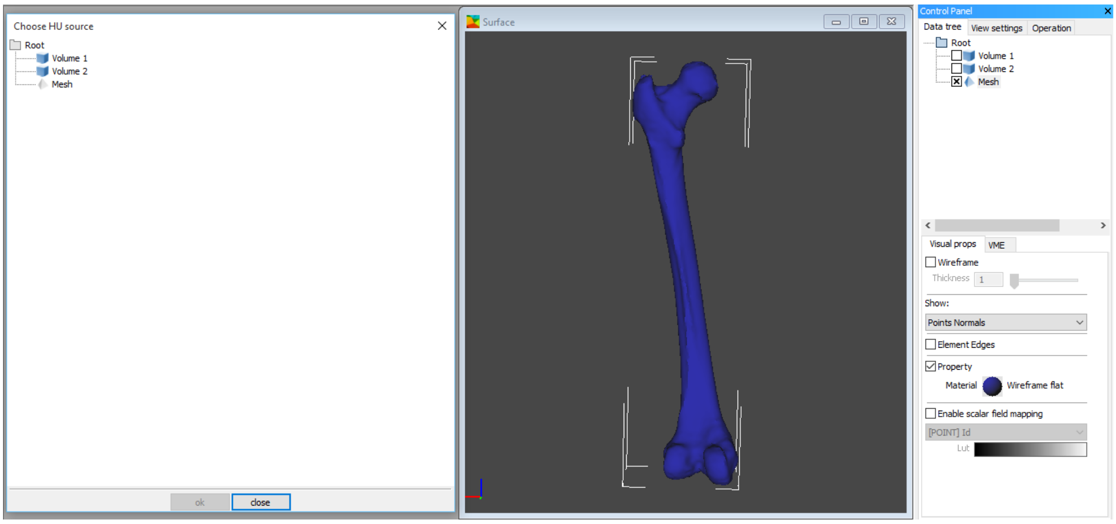


Figure 32

modeling relationships to be used and to the algorithm. The specified parameters can be saved ("save configuration file" command) in a XML file which is by default stored in the parent directory. The configuration file can be reloaded ("open configuration file" command) or updated ("save configuration file as" command) (Figure 33).

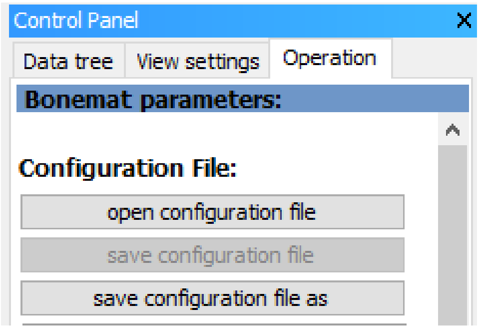


Figure 33

Please consider that the application itself is adimensional, so the user is responsible for setting coherent measurement units to the variables.

6.3.1.1 Material modeling parameters

- CT densitometric calibration ($\rho_{QCT} = a + bHU$) specifies CT densitometric calibration parameters (usually obtained through a calibration phantom)
- (optional) Correction of the calibration ($\rho_{Ash} = a + b\rho_{QCT}$) applies a linear correction rule, if available, to the CT densitometric calibration. There is the possibility of specifying up to three different linear corrections on three different density ranges. See the following reference for details on the opportunity of using such a correction: Schileo E, Dall'Ara E, Taddei F, Malandrino A, Schotkamp T, Baleani M, Viceconti

M. An accurate estimation of bone density improves the accuracy of subject-specific finite element models.J Biomech. 2008;41(11):2483-91.

- density-elasticity relationship ($E = a+b{\rho_{Ash}}^c$) specifies the law relating bone density to bone elastic modulus, usually taken from the many empirical density-elasticity relationships available in the literature. There is the possibility of specifying up to three different power laws on three different density ranges, so as to account for discontinuities (e.g. one for cortical and one for trabecular bone, given a threshold). Density-elasticity relationships have been extensively reported and discussed in the literature, see e.g. the following references, for a critical review and a FE application respectively:

  Helgason B, Perilli E, Schileo E, Taddei F, Brynjólfsson S, Viceconti M. Mathematical relationships between bone density and mechanical properties: a literature review. Clin Biomech. 2008 Feb;23(2):135-46.

  Schileo E, Taddei F, Malandrino A, Cristofolini L, Viceconti M. Subject-specific finite element models can accurately predict strain levels in long bones. J Biomech 2007;40(13):2982-9.

  The user can set a minimum value for the elastic modulus to be assigned (optional). All the material modelling parameters are shown in Figure 34.

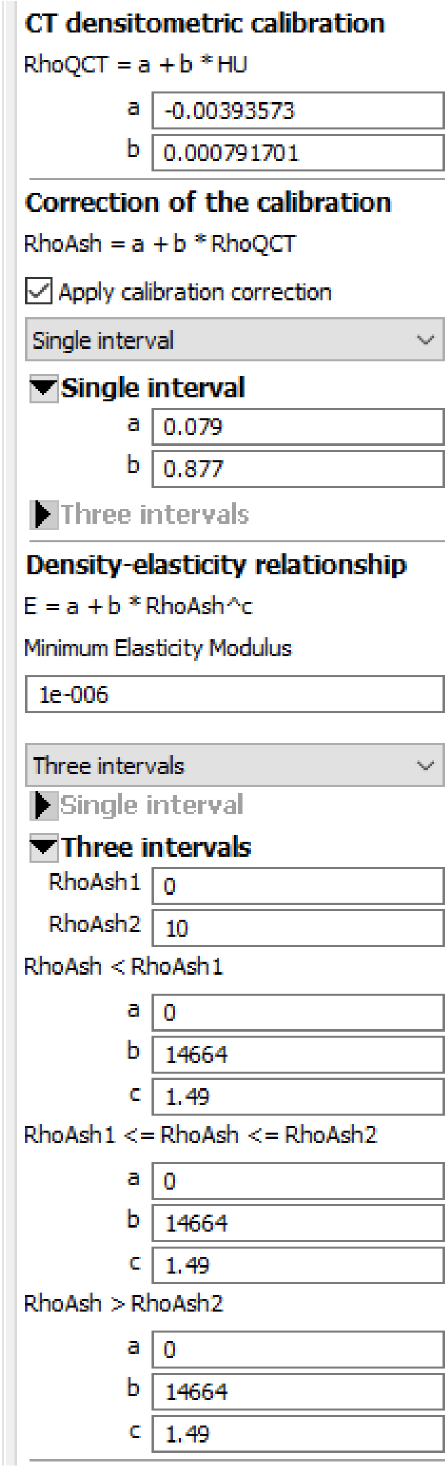
6.3.1.2 Algorithm parameters

- HU integration vs. E integration: specifies whether performing the integration on the native HU field on each element and then calculate the E values (called HU integration), or vice versa (E integration). See the following references to choose among the averaging strategy:

  Taddei F, Schileo E, Helgason B, Cristofolini L, Viceconti M. The material mapping strategy influences the accuracy of CT-based finite element models of bones: An evaluation against experimental measurements. Med Eng Phys. 2007 Nov;29(9):973-9.

  Taddei F, Pancanti A, Viceconti M. An improved method for the automatic mapping of computed tomography numbers onto finite element models. Med Eng Phys. 2004 Jan;26(1):61-9.

  Zannoni C, MantovaniR, Viceconti M. Material properties assignment to finite element models of bone structures: a new method. Med EngPhys. 1998 Dec;20(10):735-40.



CT densitometric calibration

RhoQCT = a + b * HU

a -0.00393573

b 0.000791701

Correction of the calibration

RhoAsh = a + b * RhoQCT

☑ Apply calibration correction

Single interval

Single interval

a 0.079

b 0.877

Three intervals

Density-elasticity relationship

E = a + b * RhoAsh^c

Minimum Elasticity Modulus

1e-006

Three intervals

Single interval

Three intervals

RhoAsh1 0

RhoAsh2 10

RhoAsh < RhoAsh1

a 0

b 14664

c 1.49

RhoAsh1 <= RhoAsh <= RhoAsh2

a 0

b 14664

c 1.49

RhoAsh > RhoAsh2

a 0

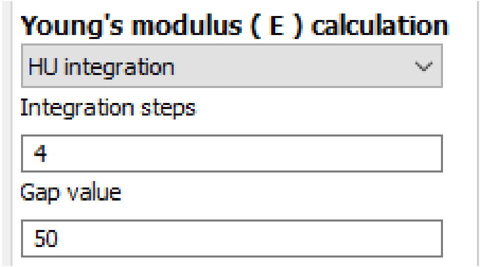
b 14664

c 1.49

Figure 34

- Integration steps:specifies how many integration points the application will set within each element
- Gap value:specifies the minimum gap between two subsequent material cards (e.g. given a typical elastic modulus range in the order of 50 to 20000 MPa, the user may want to avoid defining 19950 different materials but rather group them each e.g. 500 MPa, resulting in maximum 40 material cards).

All the algorithm parameters are shown in Figure 35.



Young's modulus (E) calculation

HU integration

Integration steps

4

Gap value

50

Figure 35

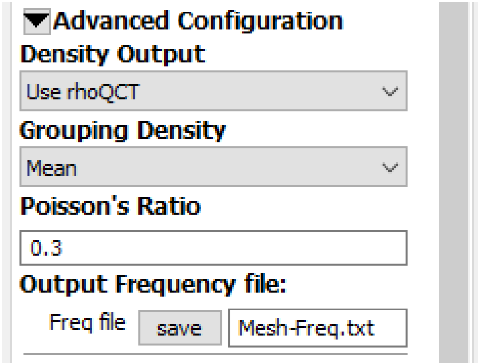
6.3.1.3 Advanced configuration settings

This small sub-menu lets the user set some minor, yet possibly interesting, parameters:

- Density Output: since a list of bone density values comes as output with the list of elastic moduli, and different density measurements can be set in the application, the user can decide whether to list the density coming directly from the densitometric calibration (named “Rho-QCT”, default), or that resulting from the (optional) correction to the densitometric calibration.
- Grouping Density: by default, elastic moduli bins are assigned by Bonemat starting from the highest modulus among each element and then grouping to that value all elements within the specified “gap value”. Then the algorithm searches again for the highest value among the remaining elements, and the game goes on until all elements are assigned an elastic modulus, or the minimum value specified above (optional) is reached. This means that the distance from two consecutive elastic moduli bins is not necessarily the gap value. The “Grouping Density” options lets the user decide whether to assign to the bin the highest elastic modulus value contained in it, or the mean one. This option does not usually imply big or mechanically relevant differences in the output when e.g. analyzing a whole bone ranging from 50 to 20000 MPa in Young’s modulus, but can be relevant in niche applications where small ranges are analyzed, or if few material bins are used.
- Poisson’s ratio: while waiting for the implementation of a consistent relationship (if any) between density and Poisson’s ratio, a fixed value is set. The default is 0.3, here the user can specify a different value.

- Output frequency file: specifies the output filename (and folder address, default on the install directory) of the file that lists how many elements share the same elastic modulus, for all the sampled elastic moduli in the model.

All the advanced configuration settings are shown in Figure 36.



Advanced Configuration

Density Output

Use rhoQCT

Grouping Density

Mean

Poisson's Ratio

0.3

Output Frequency file:

Freq file save Mesh-Freq.txt

Figure 36

6.3.2 Bonemat batch

The Bonemat batch Operation allows a user to run a series of Bonemat operations on multiple volumes and multiple meshes using a single text file (.txt format) (Figure 37).

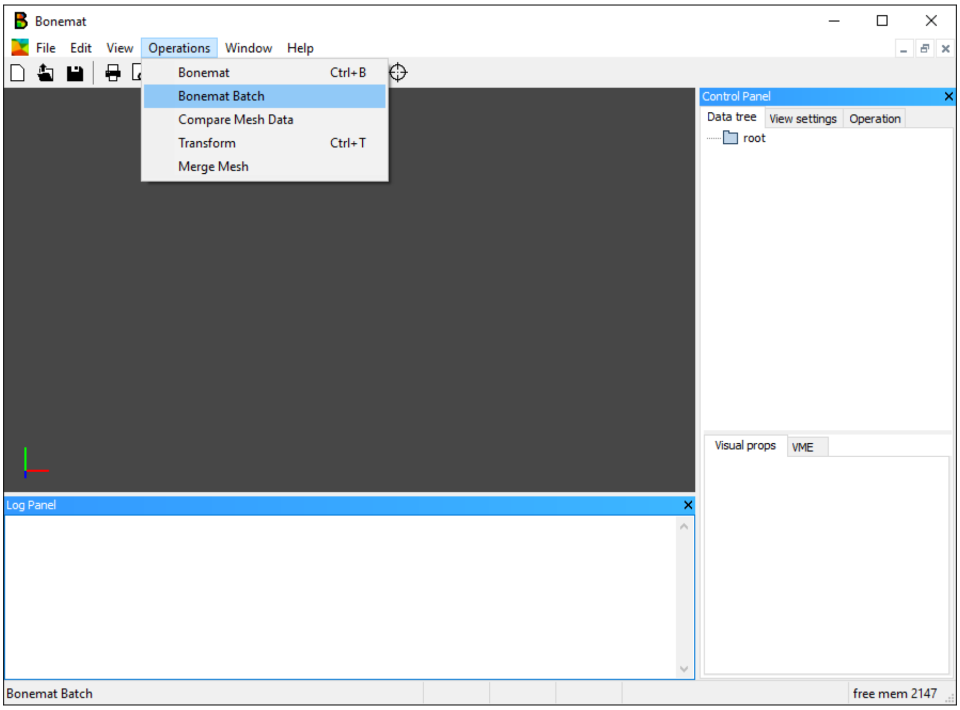


Figure 37

The text file has to include, for each Bonemat operation, the following information:

- path of the VMEmesh (FE model, .cdb or .inp or .vtk file formats)
- path of the VMEvolume (CT volume, .vtk file format)
- path to the calibration file (.conf or .xml file formats)
- path for the output FE model (FE model, .inp or .cdb or .vtk file formats)

Paths can be "absolute", for example the text file can be formatted as follows:

```
C:\Windows\mesh\mesh.cdb
C:\Windows\volume\volume.vtk
C:\Windows\calibration_files\calibration.xml
C:\Windows\results\output.inp
```

Even "relative" paths can be used, as in the following example:

```
mesh\mesh.cdb
volume\volume.vtk
calibration_files\calibration.xml
results\output.inp
```

with the condition that the text file is in the same directory of the folders in which there are the VMEmesh, the VMEvolume and the calibration file. In the example above, the output will be saved in a folder located in the same directory of the text file.

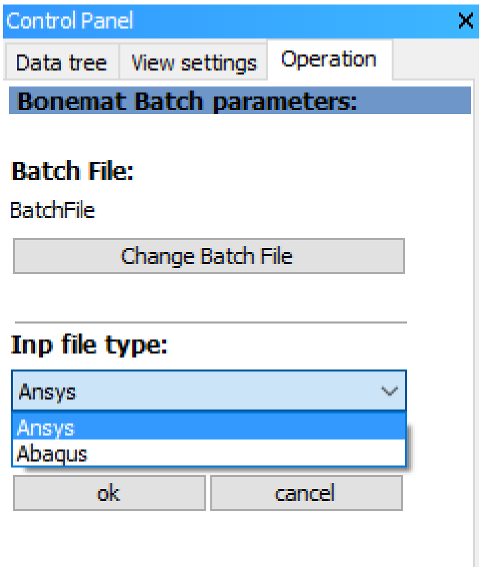
It is not necessary to specify the paths when the text file is in the same folder of the VMEmesh, the VMEvolume and the calibration file. The output will be automatically located in the same directory of the text file, if its path is not specified.

Comments can be added to the text file by adding "#" before them. Please observe that a calibration file in .xml format is recommended because it contains the "advanced options", explained in more detail in the section 6.3.1.3. It is anytime possible to update the configuration file in the Bonemat operation, by saving it in .xml format.

The "Bonemat Batch parameters" can be modified in the Operation Panel (Figure 38):

- Batch File: the user can change the text file for launching the Bonemat batch Operation
- Inp file type: the user can select the type of Input file for VMEmesh, by choosing between Ansys or Abaqus formats

The Bonemat batch Operation can also handle an input VMEmesh composed by N components, so that the operation will be executed on all of the N components. Please note it will produce as oputput N output files (output1.inp, output2.inp, ... ,outputN.inp, if "output.inp" is the name for the output FE model in the text file). If the user needs to have the N output FE models in a single VMEmesh, he can import the N output files and use the Merge Mesh Operation to merge them (for more details please look at the section 6.3.5).



Bonemat Batch parameters:

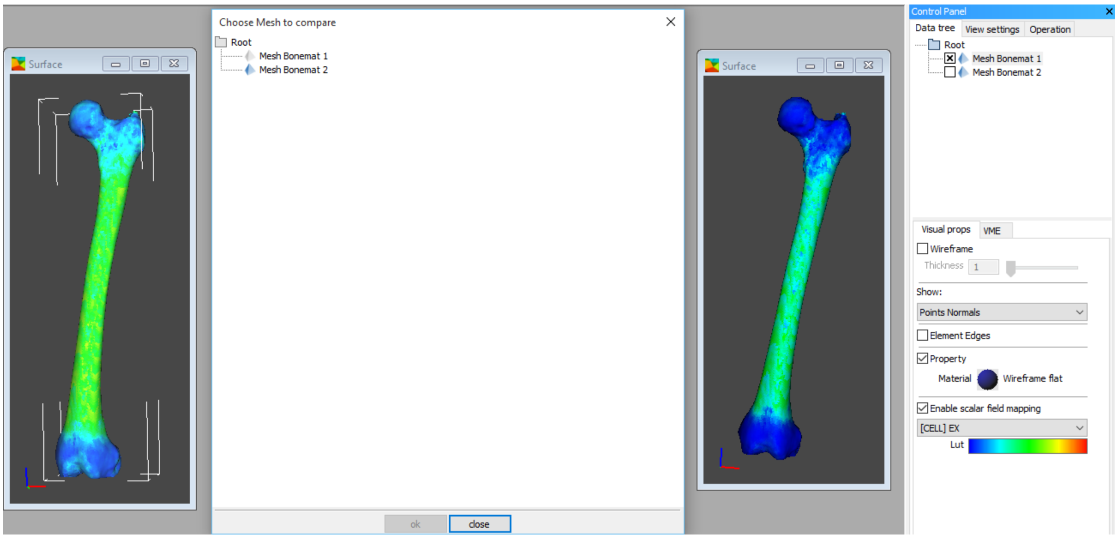
Batch File:
BatchFile

Inp file type:
Ansys

Figure 38

6.3.3 Compare Mesh Data

This Operation allows the user to compare the scalars associated to two meshes that share the same topology (i.e. connectivity). The Operation can be run when a VMEmesh (primary input) is activated. Another VMEmesh (secondary input) has to be chosen by the user for comparison (Figure 39).

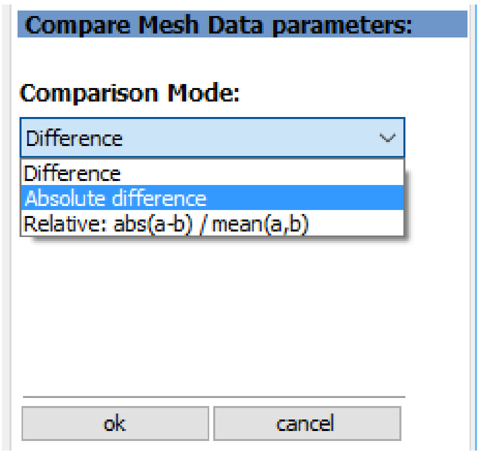


☐ Wireframe
Thickness 1
Show: Points Normals
☐ Element Edges
☑ Property
☑ Enable scalar field mapping
[CELL] EX

Figure 39

The user can choose in the “Comparison mode” menu of the Operation panel which kind of comparison to perform, among “difference”, “absolute difference”, “relative difference” (Figure 40).

The output is a new VMEmesh, whose scalars are the result of the comparison between the two input VMEmesh (Figure 41).



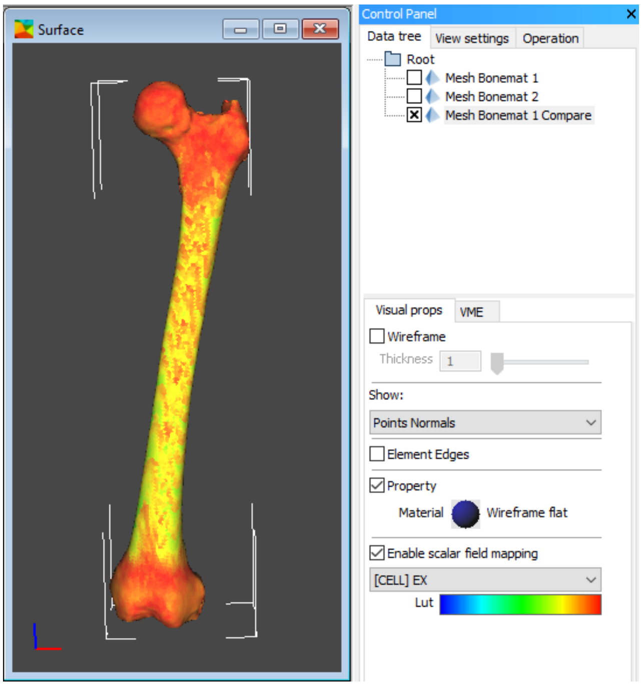
Compare Mesh Data parameters:

Comparison Mode:

Difference

Difference
Absolute difference
Relative: abs(a-b) / mean(a,b)

Figure 40



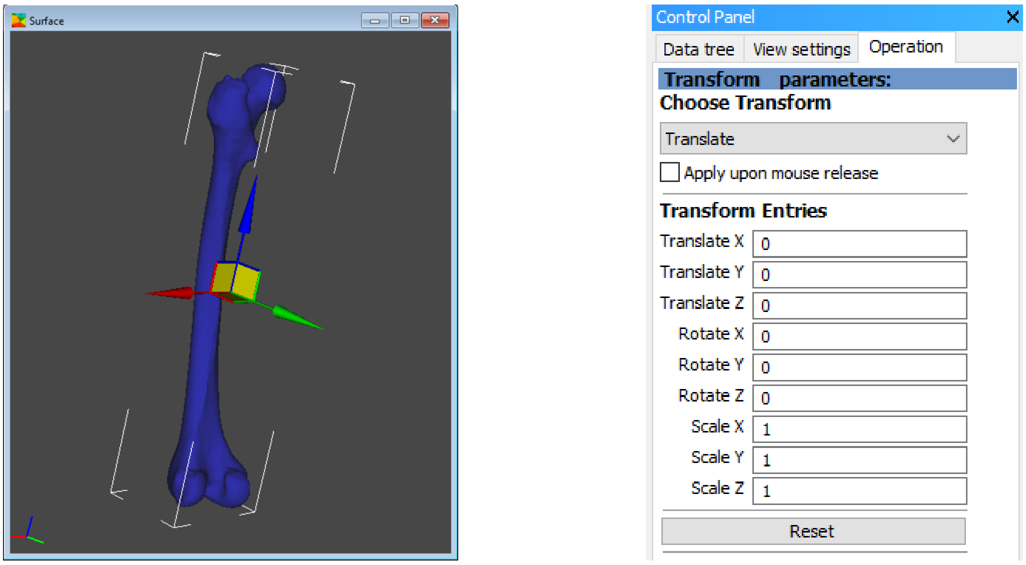
Root
☐ Mesh Bonemat 1
☐ Mesh Bonemat 2
☒ Mesh Bonemat 1 Compare

☐ Wireframe
Thickness 1
Show: Points Normals
☐ Element Edges
☑ Property
Material Wireframe flat
☑ Enable scalar field mapping
[CELL] EX

Figure 41

6.3.4 Transform

This Operation allows the user to perform affine transformation (roto-translation and non-uniform scaling) on the VMEs. The input is any VME in the data tree. The output is a modified VME, with a matrix pose applied to it (please note the original dataset, stored in .vtk format, is not changed, just a pose matrix is applied to it whenever visualizing or interacting with it). The user interface (Operation panel in the data tree) lets the user apply the transformation either by inserting text or interacting with Gizmos (Figure 42).



Transform parameters:

Choose Transform

Translate

☐ Apply upon mouse release

Transform Entries

Translate X 0
Translate Y 0
Translate Z 0
Rotate X 0
Rotate Y 0
Rotate Z 0
Scale X 1
Scale Y 1
Scale Z 1

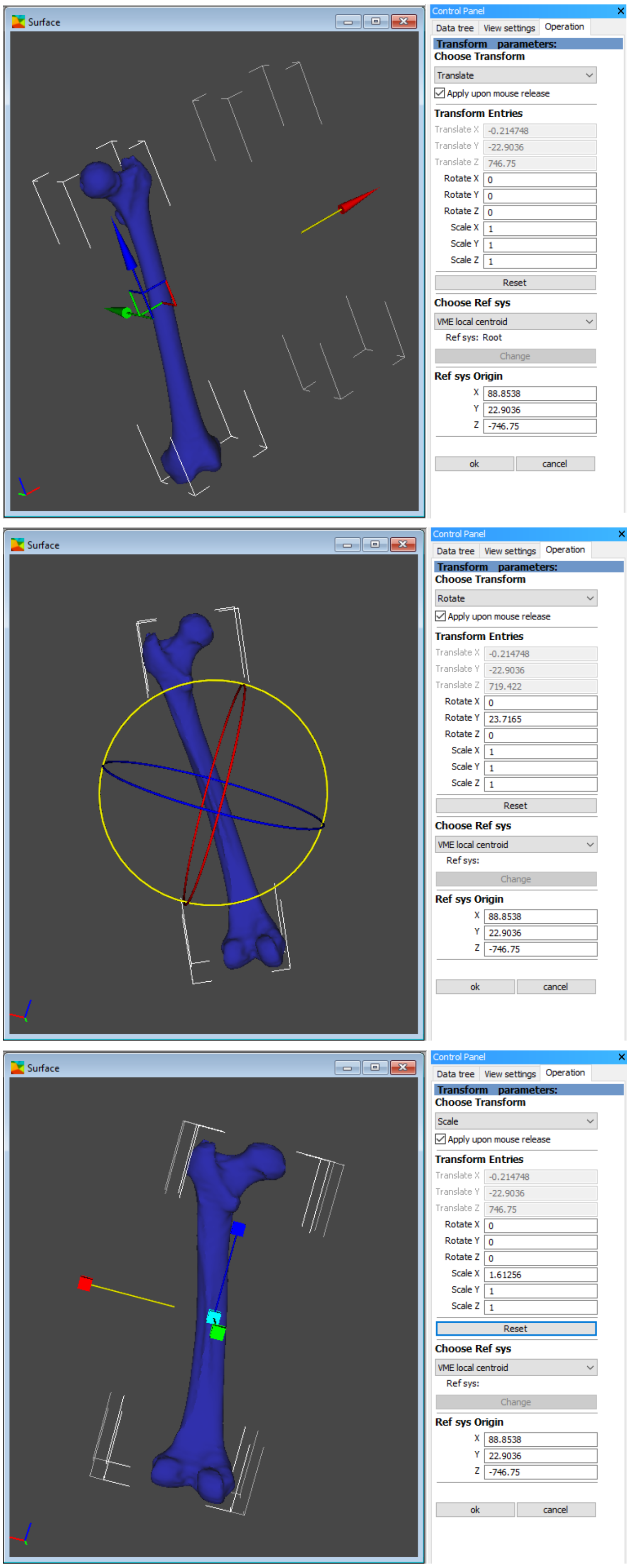
Figure 42

If interacting with Gizmos, the user has first to select the transformation mode (translate, rotate, scale) (Figure 43).

The user can activate the "Apply upon mouse release" option (Figure 44) that allows him to visualize the transformation only after the mouse release and not interactively following mouse movement.

The pose parameters shown in the text are interactively updated when moving Gizmos. The pose parameters and the Gizmos can be shown in the following reference systems (Figure 45):

- Absolute, i.e. the global reference system
- VME base ref sys, i.e. the local reference system of the selected VME at the beginning of the operation, so that the VME base reference system is fixed and the user can control at each instant the transformation entity on the text entries.
- VME local centroid, i.e. a reference system centered at the centroid of the selected VME; the user can modify the coordinates of the reference system origin to control the transform entity

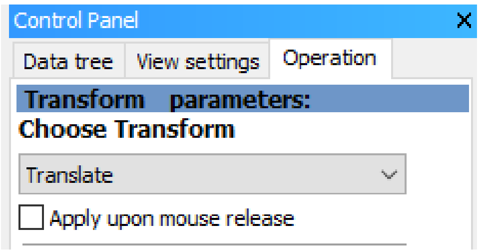


Choose Transform: Translate
☑ Apply upon mouse release
Transform Entries
Translate X -0.214748
Translate Y -22.9036
Translate Z 746.75
Rotate X 0
Rotate Y 0
Rotate Z 0
Scale X 1
Scale Y 1
Scale Z 1
Choose Ref sys: VME local centroid
Ref sys: Root
Ref sys Origin
X 88.8538
Y 22.9036
Z -746.75

Choose Transform: Rotate
☑ Apply upon mouse release
Transform Entries
Translate X -0.214748
Translate Y -22.9036
Translate Z 719.422
Rotate X 0
Rotate Y 23.7165
Rotate Z 0
Scale X 1
Scale Y 1
Scale Z 1
Choose Ref sys: VME local centroid
Ref sys:
Ref sys Origin
X 88.8538
Y 22.9036
Z -746.75

Choose Transform: Scale
☑ Apply upon mouse release
Transform Entries
Translate X -0.214748
Translate Y -22.9036
Translate Z 746.75
Rotate X 0
Rotate Y 0
Rotate Z 0
Scale X 1.61256
Scale Y 1
Scale Z 1
Choose Ref sys: VME local centroid
Ref sys:
Ref sys Origin
X 88.8538
Y 22.9036
Z -746.75

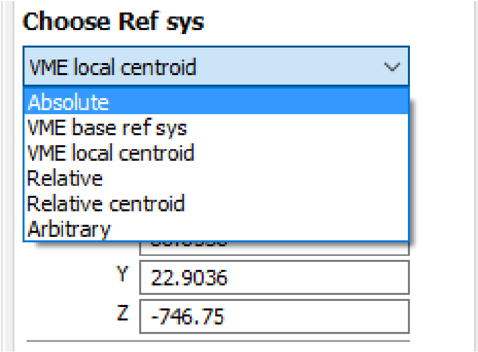
Figure 43



Choose Transform: Translate

☐ Apply upon mouse release

Figure 44



Y 22.9036

Z -746.75

Figure 45

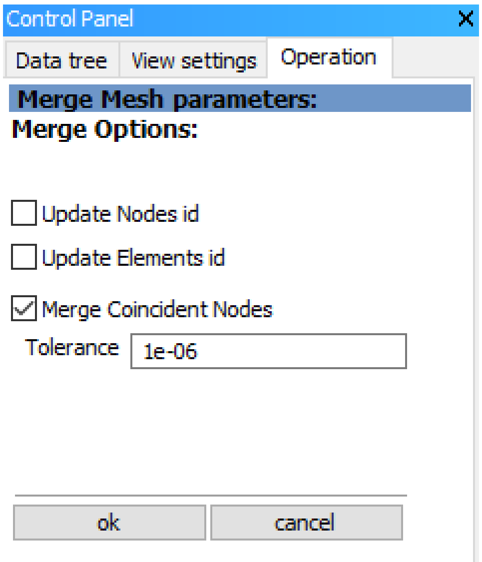
- Relative, i.e. a reference system specified by the user among the VMEs present in the data tree
- Relative centroid, i.e. a reference system centered at the centroid of a VME specified by the user among the ones present in the data tree
- Arbitrary, i.e. an arbitrary reference system that the user can define from the global reference system, by modifying the coordinates of the reference system origin.

6.3.5 Merge Mesh

This Operation allows the user to update nodes id, and/or to update elements id, and/or to merge coincident nodes of two meshes according to a distance tolerance. The Operation can start if a VMEmesh (primary input) is selected in the tree, while a second VMEmesh is chosen by the user in the VME tree. The user can choose among three different options in the “Merge options” menu of the Operation Panel (Figure 46):

- Update Nodes id: the highest node id of the primary input is added to each node id of the secondary input, so that the minimum id of the secondary input is greater than the highest id of the primary input, while nodes id of the primary input remain unchanged.
- Update Elements id: it works in the same way of the “Update Nodes id” option but it acts on the Elements id, i.e. the highest element id of the primary input is added to each element id of the secondary input, while elements id of the primary input remain unchanged.

- Merge Coincident Nodes: the user can set a distance tolerance (default value: 1e-006), below which nodes are considered coincident, and so nodes of the secondary input assume the same id of the corresponding nodes in the primary input, while nodes id of the primary input remain unchanged. Ids of all nodes whose distance is greater than the tolerance meshes remain unchanged.



Merge Mesh parameters:

Merge Options:

☐ Update Nodes id

☐ Update Elements id

☑ Merge Coincident Nodes

Tolerance 1e-06

Figure 46

The output is a new VMEmesh that cointains the two input meshes (Figure 47) with updated nodes id, or updated elements id, or merged coincident nodes, depending on the option chosen by the user. The user can also simultaneously activate more Merge options: if all of the three different options are activated, then the Operation works by updating nodes id at first, than updating elements id and finally merging coincident nodes.

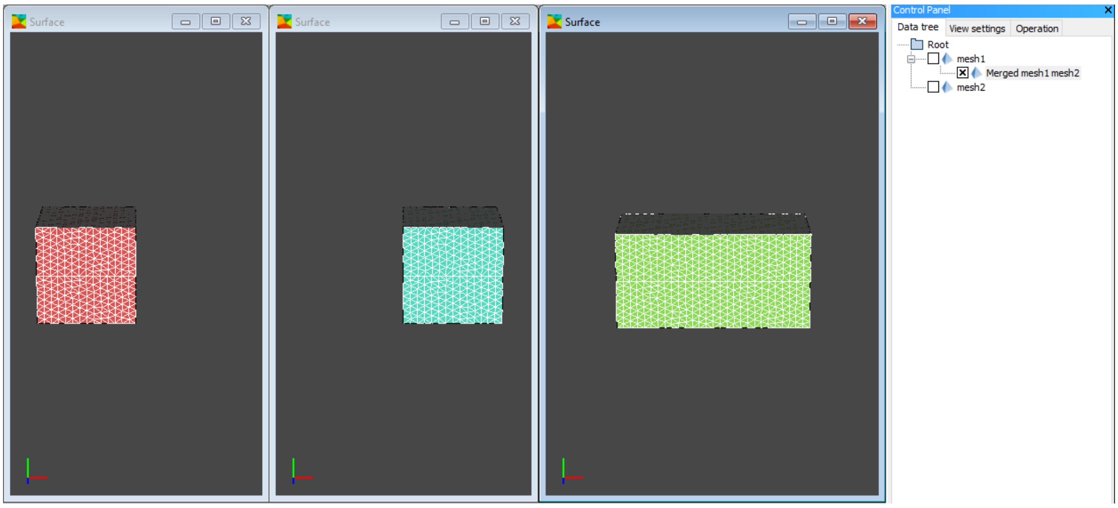


Figure 47

References

[1] B. Helgason, E. Perilli, E. Schileo, F. Taddei, S. Brynjòlfsson, and Viceconti M. Mathematical relationships between bone density and mechanical properties: a literature review. *Clinical Biomechanics*, (23):135–46, Feb 2008.

[2] E. Schileo, F. Taddei, A. Malandrino, L. Cristofolini, and M. Viceconti. Subject-specific finite element models can accurately predict strain levels in long bones. *Journal of Biomechanics*, (40):2982–9, 2007.

[3] F. Taddei, A. Pancanti, and M. Viceconti. An improved method for the automatic mapping of computed tomography numbers onto finite element models. *Medical Engineering and Physics*, (26), Jan 2004.

[4] F. Taddei, E. Schileo, Helgason B., L. Cristofolini, and Viceconti M. The material mapping strategy influences the accuracy of ct-based finite element models of bones: An evaluation against experimental measurements. *Medical Engineering and Physics*, (29):973–9, Nov 2007.

[5] C. Zannoni, R. Mantovani, and M. Viceconti. Material properties assignment to finite element models of bone structures: a new method. *Medical Engineering and Physics*, (20):735–40, Dec 1998.