Computer Methods and Programs in Biomedicine 152 (2017) 85–92

Contents lists available at ScienceDirect

Computer Methods and Programs in Biomedicine

journal homepage: www.elsevier.com/locate/cmpb

NMSBUILDER: Freeware to create subject-specific musculoskeletal models for OpenSim

Giordano Valente[a,*], Gianluigi Crimi[a], Nicola Vanella[a], Enrico Schileo[b], Fulvia Taddei[a]

[a] Medical Technology Laboratory, Rizzoli Orthopaedic Institute, via di Barbiano 1/10, 40136 Bologna, Italy
[b] Computational Bioengineering Laboratory, Rizzoli Orthopaedic Institute, via di Barbiano 1/10, 40136 Bologna, Italy

ARTICLE INFO

Article history:
Received 5 April 2017
Revised 5 August 2017
Accepted 15 September 2017

Keywords:
Musculoskeletal model
Subject-specific model
Image-based model
Imaging data processing
Freeware
OpenSim

ABSTRACT

Background and objective: Musculoskeletal modeling and simulations of movement have been increasingly used in orthopedic and neurological scenarios, with increased attention to subject-specific applications. In general, musculoskeletal modeling applications have been facilitated by the development of dedicated software tools; however, subject-specific studies have been limited also by time-consuming modeling workflows and high skilled expertise required. In addition, no reference tools exist to standardize the process of musculoskeletal model creation and make it more efficient. Here we present a freely available software application, NMSBUILDER 2.0, to create musculoskeletal models in the file format of OpenSim, a widely-used open-source platform for musculoskeletal modeling and simulation. NMSBUILDER 2.0 is the result of a major refactoring of a previous implementation that moved a first step toward an efficient workflow for subject-specific model creation.
Methods: NMSBUILDER includes a graphical user interface that provides access to all functionalities, based on a framework for computer-aided medicine written in C++. The operations implemented can be used in a workflow to create OpenSim musculoskeletal models from 3D surfaces. A first step includes data processing to create supporting objects necessary to create models, e.g. surfaces, anatomical landmarks, reference systems; and a second step includes the creation of OpenSim objects, e.g. bodies, joints, muscles, and the corresponding model.
Results: We present a case study using NMSBUILDER 2.0: the creation of an MRI-based musculoskeletal model of the lower limb. The model included four rigid bodies, five degrees of freedom and 43 musculotendon actuators, and was created from 3D surfaces of the segmented images of a healthy subject through the modeling workflow implemented in the software application.
Conclusions: We have presented NMSBUILDER 2.0 for the creation of musculoskeletal OpenSim models from image-based data, and made it freely available via nmsbuilder.org. This application provides an efficient workflow for model creation and helps standardize the process. We hope this would help promote personalized applications in musculoskeletal biomechanics, including larger sample size studies, and might also represent a basis for future developments for specific applications.

1. Introduction

Musculoskeletal modeling and simulation of movement, a branch of computational modeling and simulation of biological systems, has grown rapidly in the past two decades, as demonstrated by the exponential increase in peer-reviewed publications [1]. Musculoskeletal modeling shows great potential in orthopedic and neurological scenarios to improve diagnosis and treatment of musculoskeletal disorders, and has been used, for example, in a variety of applications involving calculation of muscle and joint loads (e.g. [2–9]). In this context, software tools developed for musculoskeletal modeling have facilitated the expansion of such applications. In particular, OpenSim [10], an open-source software plat-

Abbreviations: API, application programming interface; GUI, graphical user interface; ALBA, agile library for biomedical applications; MAF, multimod application framework; VME, virtual medical entity; STL, stereolithography; VTK, visualization toolkit; VTP, VTK polygonal data; MSF, multimod storage file; ASCII, American standard code for information interchange; ISB, international society of biomechanics; MRI, magnetic resonance imaging; DOF, degree of freedom; Landmarks: RASIS, right anterior superior iliac spine; LASIS, left anterior superior iliac spine; RPSIS, right posterior superior iliac spine; LPSIS, left posterior superior iliac spine; RHC, right hip center; RME, right medial epicondyle; RLE, right lateral epicondyle; RMC, right medial condyle; RLC, right lateral condyle; RMM, right medial malleolus; RLM, right lateral malleolus; RPA_II, right plantar aspect on second metatarsal; RPA_CA, right plantar aspect on calcaneus..

* Corresponding author.
E-mail address: valente@tecno.ior.it (G. Valente).

http://dx.doi.org/10.1016/j.cmpb.2017.09.012
0169-2607/

form, has been increasingly used as a reference tool for musculoskeletal simulations of movement, also allowing sharing and distribution of musculoskeletal models. As personalized medicine has been practiced more widely, there has been an increase in subject-specific musculoskeletal modeling with different levels of personalization (e.g. [11–16]). Subject-specific modeling applications provide more accurate results compared to generic models based on cadaver data, particularly in pathological conditions [13,17].

However, personalized applications are limited by the process of model creation that requires specialized expertise and can be largely time-consuming, when experimental data coming from different sources, especially imaging data, need to be processed. In general, research groups that have developed image-based models and simulations of movement have used in-house software tools to the purpose (e.g. [12,13,17,18]). Therefore, no reference tools exist to standardize the process of musculoskeletal model creation and make it more efficient, which would promote applications of personalized modeling.

We have recently analyzed the robustness of image-based multibody models of the musculoskeletal system, and found that models are not markedly sensitive to the uncertainties in parameter identification [19]. Model creation was performed by using a software application that we developed and released, named NMSBuilder, which allows processing of image-based data and creation of subject-specific musculoskeletal models for OpenSim, and allowed us to reduce the uncertainty in the identification of some model parameters. This software application has been increasingly used for research (e.g. [15,19,20–23]), however it presented limitations in usability and efficiency. First, the creation of OpenSim models relied on the creation of specialized components in NMSBuilder, which allowed to represent an OpenSim model as C++ commands of the OpenSim application programming interface (API), and needed to be compiled to generate the model. This implied that additional software had to be installed (i.e. Visual Studio, CMake, Python, OpenSim), to create and modify C++ commands of the OpenSim API and call the compiler to create models in the OpenSim file format. This also implied that any modification to a model, e.g. to modify a musculotendon geometry path, required editing of the C++ commands and re-compiling of the script without software interaction. In addition, the working principle of some library objects produced slowing down of the modeling workflow (e.g. manipulation of landmark clouds) or instability of the program.

The development process of the software has been carried on and has reached a mature point, such that we have released NMSBuilder 2.0 as a freeware via nmsbuilder.org. NMSBuilder 2.0 is a result of a major software refactoring that leads to a novel modeling workflow and improves usability and efficiency in creating musculoskeletal models for OpenSim from 3D geometries, by overcoming previous limitations and including new features. Therefore, here we aim to provide an overview of NMSBuilder 2.0, and present a case study to create an image-based OpenSim musculoskeletal model.

2. Software implementation

2. 1. What is NMSBuilder?

NMSBuilder 2.0 is a freely available software to create subject-specific musculoskeletal models for OpenSim from 3D geometries such as segmented imaging data. It includes an end-user application with a graphical user interface (GUI) that provides access to all the functionalities. NMSBuilder 2.0 is developed at the Rizzoli Orthopedic Institute (Bologna, Italy), and it is distributed as freeware via nmsbuilder.org, under a custom license that allows use for non-commercial research and education. The application is based on the agile library for biomedical applications (ALBA), an open-source rapid application development framework for computer-aided medicine written in C++. The ALBA architecture is made of open source external libraries and components necessary to build applications such as NMSBuilder (Fig. 1). Among the libraries included, the visualization toolkit (VTK) is used for data representation, storage and filtering, on top of which we built the vtkALBA module that contains a set of filters that extend VTK functionalities. ALBA and related applications are developed through the Agile Software Development methodology, based on the approach of delivering software quickly and continuously by promoting customer involvement, continuous testing and planning, and close teamwork. ALBA represents a new project started from the multimod application framework (MAF) [24], now discontinued, at the Rizzoli Orthopedic Institute.

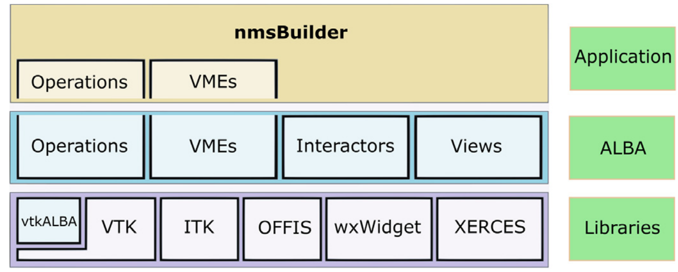

Fig. 1. Block diagram of the NMSBuilder software architecture. NMSBuilder is built as one application on top of the open source agile library for biomedical applications (ALBA). Some components of ALBA, i.e. operations and virtual medical entities (VMEs), are customized according to the application. In particular, VMEs are library objects to show, edit and store data, among which OpenSim library objects are included. ALBA is built on top of external libraries and the vtkALBA module that extends the visualization toolkit (VTK) functionalities of data representation, storage and filtering.

2.2. Overview of major functionalities

The NMSBuilder GUI features a layout made of the following areas (Fig. 2):

- Menu Bar: area containing the drop menus (*File, Edit, View, Operations, Window, Help*)
- Toolbar: area containing the shortcuts to main functionalities and camera options
- Views area: area containing all current views open
- Log Panel: area containing log information used for debugging and other advanced uses
- Control Panel: area containing three main tabs: data tree, view settings and operations. Data tree contains the information of the data objects, and it is divided in a top panel, which contains the visual representation of the data tree, and a bottom panel, where it is possible to edit visual properties (*visual props*) and data objects properties *(VME,*) i.e. Virtual Medical Entity). *View settings*) contain the settings of the selected view. *Operation* contains the GUI to execute the current *operation*) when launched.

The fundamental components of the NMSBuilder application, which are provided by the low-abstraction layer in the software architecture (Fig. 1), are:

- Virtual medical entities (VMEs): library objects used to show, edit ad store data. Each VME type (e.g. volume, surface, landmark) represents a specific data type and contains specific information. VMEs are organized in a hierarchical tree (*data tree*) and they all contain a dataset, a pose matrix that defines position and orientation, and metadata for the attributes.

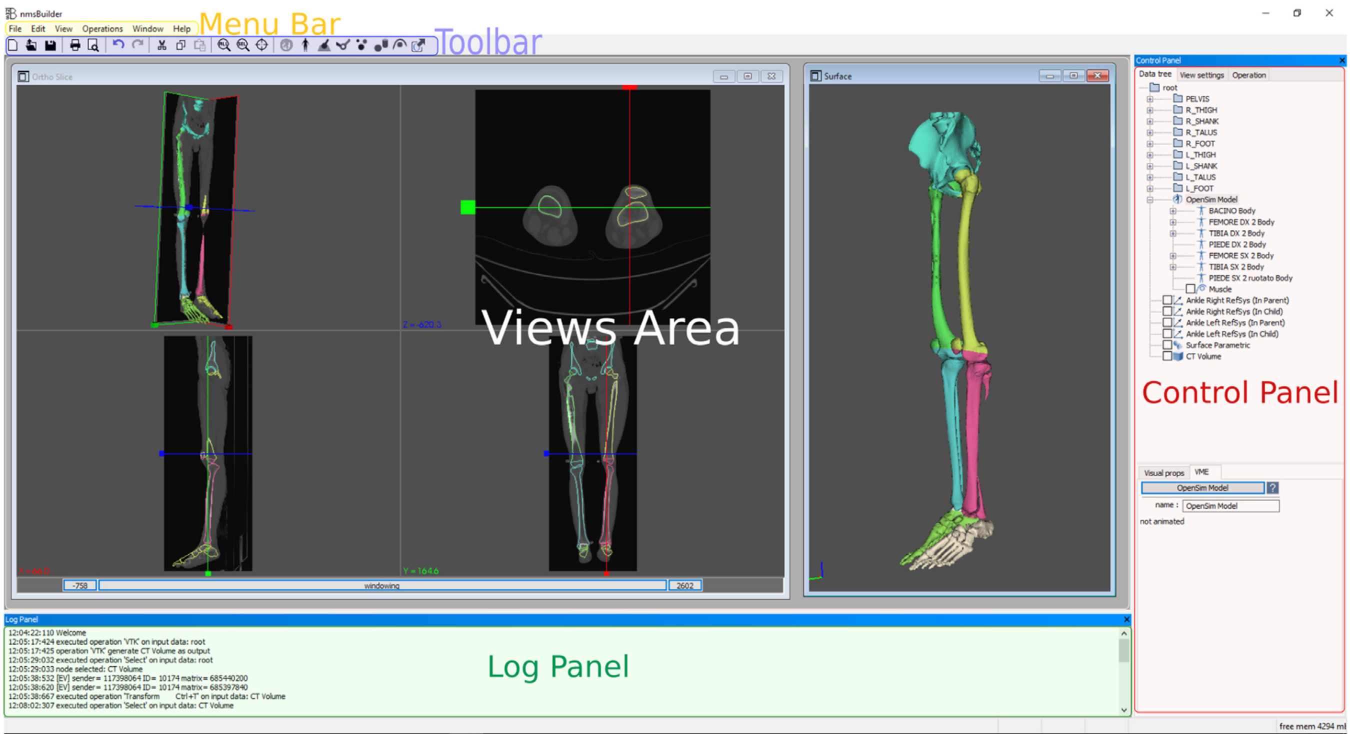

Fig. 2. Layout of the NMSBUILDER GUI. All areas are shown, and some example data are visualized in the views area by using two views from those available: *Orthoslice* in the left panel (volume and surfaces visualized) and *surface* in the right panel (surfaces visualized). The four-panel *orthoslice* view includes a perspective panel that visualize a volume intersected by three selectable orthogonal planes, and other three panels that show a parallel view of each other slice, while one or more surfaces can be visualized at the same time in the perspective panel, with the sliced contours in the orthogonal panels.

- Operations: procedures that manipulate VMEs. Operations include importers, exporters and standard operations.
- Views: visual representations of the data. Views allow specific representations of the VMEs. They can be standard, i.e. single render window, or compound, i.e. more complex views including multiple standard views and GUI components.
- Interactors: components for graphical interactions. They can be used in the views or operations to interact with the data.

All operations are finalized to the creation of subject-specific musculoskeletal models. All data objects, hierarchical structure and properties are saved in the multimod storage format (MSF), the NMSBUILDER native file format.

NMSBUILDER allows users to import several biomedical data into the hierarchical data tree, particularly images, surfaces and landmark clouds (Table 1).

Once the data are imported and organized into the data tree, they can be visualized by using an appropriate View from the seven available. For example, the *Surface* view (Fig. 2, right panel in the views area) is particularly suitable for the visualization of surfaces, which are totally radio-opaque objects, while the *Orthoslice* view (Fig. 2, left panel in the views area) is suitable for both volumes and surfaces.

Standard operations manipulate the data objects to create new VMEs or to modify existing VMEs (Table 1). *Create* operations particularly allow to create landmarks and reference systems. Landmarks can be created as generic points (*Add Landmark*), by interactively picking on selected VMEs or by manually entering coordinates of points, or they can be created from a predefined set of anatomical landmarks (*Anatomical Landmark Cloud*) according to recommendations of the International society of biomechanics (ISB) [25]. Similarly, reference frames can be created as generic reference systems (*Reference System*), or they can be created automatically as joint reference frames (*Anatomical Reference System*) from anatomical landmark clouds according to the ISB recommendations [25]. The creation of anatomical landmarks and joint reference systems allow model creation efficiently, and standardization in the definition of multibody musculoskeletal systems.

OpenSim objects (Table 1) can be directly created from surfaces, reference frames and landmarks defined in the data tree, or from other OpenSim objects. The preliminary step is the creation of the *OpenSim Model* VME, which contains all the current information of all the OpenSim objects necessary to create an OpenSim model. This has been made possible by the integration of the libraries and the API of OpenSim into NMSBUILDER 2.0. OpenSim objects include bodies, joints, marker sets and muscles. In particular, muscle geometry paths can be automatically created through an operation that recognizes origin points, intermediate via-points and insertion points of each muscle, and the body they belong to. Notably, *OpenSim Model* VMEs can be exported in the OpenSim file format (.osim) any time, as they contain current information of the OpenSim objects created. This implies that any modification to OpenSim objects will be automatically effective in the *OpenSim Model* VME. Therefore, the exported model can be modified any time by interactively editing the corresponding *OpenSim Model* created in NMSBUILDER, and then export it again.

Modify operations particularly allow translation, rotation and scaling of any VME, and registration of landmark clouds and surfaces (Table 1). Notably, the *Transform* operation applies an affine transformation to perform rotation, translation and non-uniform scaling of a VME, with respect of selectable reference systems and by choosing either the transform type or the VME pose. *Register Landmark Clouds* and *Register Surfaces* allow respectively a registration of a source landmark cloud onto a target landmark cloud and a source surface onto a target surface through rigid or affine transformations. They respectively use the Singular Value Decomposition and the Iterative Closest Point methods. These transformations allow an efficient data processing for the creation of muscu-

Table 1
Operations implemented in NMSBUILDER 2.0.

Importers/Exporters	**Standard operations** Create	Modify	Measure
Images	Group	Transform	2D Measure
■ DICOM	Add landmark	Register landmark cloud	VOI Density
■ Raw volume	Reference system	Register surface	
Surfaces	Anatomical landmark cloud	Filter surface	
■ STL	Anatomical reference system	Deform surface	
■ VTP	Parametric surface	Surface mirror	
VTK	Freeze VME	Crop volume	
MSF	Distance meter	Volume resample	
Landmark	Average landmark		
	Extract isosurface		
	OpenSim objects		
	■ Create OpenSim model		
	■ Create body		
	■ Create free joint with ground		
	■ Create joint		
	■ Create marker set		
	■ Create wrap object		
	■ Create muscle		
	■ Auto create muscles		

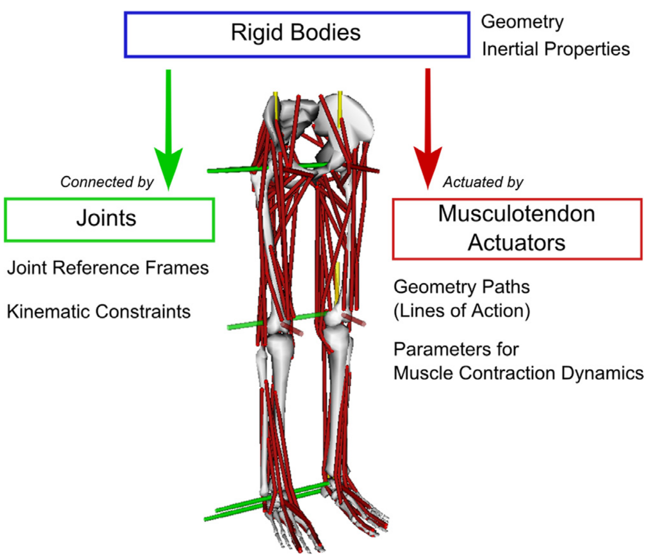

Fig. 3. Basic multibody model of the lower-limb musculoskeletal system in OpenSim.

loskeletal multibody models, for example when surfaces need to be opportunely moved, and points defining muscle paths need to be registered from an imported atlas.

Complementary operations include measurements of distances and angles on 2D representations of VMEs (*2D Measure*), and measurements of volume and average volume density from a volume of interest (*VOI Density*) (Table 1).

NMSBUILDER 2.0 includes a hypertext markup language (HTML) help that links to the user manual automatically in the default web browser, using the help menu bar or the help button corresponding to each VME type in the VME tab of the Control Panel.

2.3. Workflow for creating subject-specific OpenSim models from 3D geometries

A basic multibody model of the musculoskeletal system in the OpenSim file format includes rigid bodies, joints and musculotendon actuators [26] (Fig. 3). Rigid bodies are represented by geometries (bone and soft tissues) and inertial properties (mass, mass center, moments of inertia). Joints are represented by joint reference frames and kinematic constraints that specify permissible-motion manifolds parameterized by generalized coordinates. Musculotendon actuators are massless linear actuators represented by lines of action and a set of parameters involved in the equations of muscle contraction dynamics that relates muscle force to activation, muscle-fiber length and contractile velocity.

The creation of subject-specific models from 3D geometries in NMSBUILDER 2.0 relies on a workflow that includes the following operations, which are divided in a first step of data processing to create supporting objects necessary for musculoskeletal models, and a second step of creation of OpenSim objects and corresponding model (Fig. 4):

- Import/Create Surfaces

  Surfaces are used to define the bodies of a model. Surfaces (STL or VTP file format) can be either imported and/or created as parametric surfaces. Parametric surfaces support sphere, cone, cylinder, cube, plane and ellipsoid.

- Create Skeletal Landmark Clouds

  Skeletal landmark clouds are used to define joint reference systems and marker sets in a model. They can be created independently (*Add Landmark*), or from a predefined set of landmarks according to the different lower-limb joint regions, following the ISB recommendations on definitions of joint coordinate systems [25] (*Anatomical Landmark Cloud*).

- Create Joint Reference Systems

  Joint reference systems are used to define the joints in a model. Generic reference systems can be created by defining an origin and a plane from landmarks defined in the data tree (*Reference System*). Alternatively, anatomical reference systems can be created that define joint reference systems according to the ISB recommendations [25] from a predefined set of landmarks belonging to lower-limb regions (*Anatomical Reference System*). For each joint, two reference systems are automatically created that can be subsequently used for the definition of OpenSim joints, one with respect to the parent body and the other to the child body in the kinematic chain.

- Create Muscular Landmark Clouds

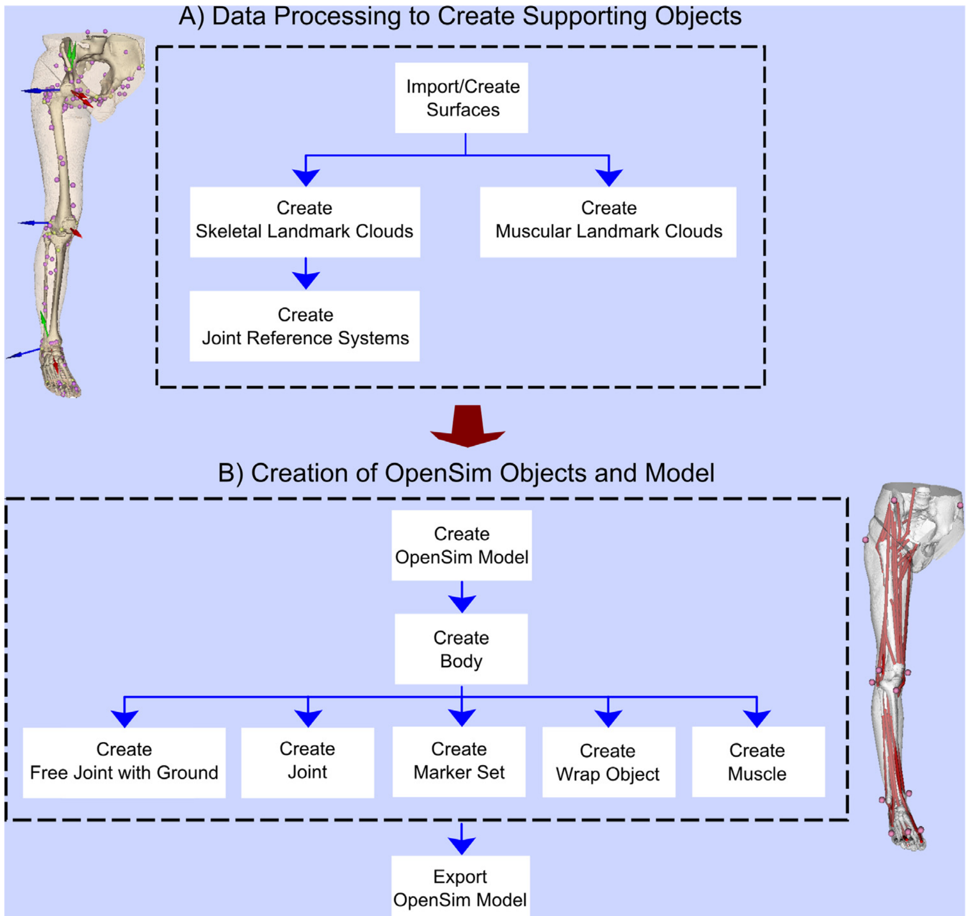

Fig. 4. Workflow for creating subject-specific OpenSim models from 3D geometries. (A) Data processing to create supporting objects necessary for the musculoskeletal model. (B) Creation of OpenSim objects and corresponding model.

Muscular landmark clouds are used to define muscle lines of action in a model. They constitute body-fixed points defining muscle geometry. Landmarks can be added as described previously (*Add Landmark*), or by importing an atlas made of muscular points, and registering it based on predefined sets of skeletal landmarks that allow scaling onto the specific geometry (*Register Landmark Cloud*).

- Create OpenSim Model

The *OpenSim Model* is made of the following objects:

- Bodies

The definition of a body requires a surface or a group of surfaces to be selected and the assignment of the values for density of each surface. The software will automatically calculate the inertial properties.

- Free joint with Ground

Which body is connected to the ground in the kinematic chain must be defined in the OpenSim model.

- Joints

The definition of a joint requires the choice of the kinematic constrains (i.e. joint type: pin, ball or custom), the parent and child bodies connected by the joint, and the joint reference system in parent and child (i.e. with respect to the parent body reference system and child body reference system, respectively). The software will automatically specify the generalized coordinates that parameterize the joint, calculate locations and orientations of the joint in the parent body (specified in the parent reference system) and child body (specified in the child reference system), define how the child body moves with respect to the parent body as a function of the generalized coordinates (especially rotation and translation axes).

- Marker Sets

The definition of a marker set requires the selection of a landmark cloud and the body it belongs to.

- Muscles

The definition of a muscle requires the choice of the path points defining the line of action and the assignment of the values for the parameters of muscle contraction dynamics. In addition, a *wrap* object can be defined that allows the line of action to wrap on a parametric surface.

- Export OpenSim Model

The OpenSim objects are transformed into an OpenSim model file, which includes the current information contained in the *OpenSim Model* VME. The model is saved as .osim file, the OpenSim native file format, and the geometry of the bodies are saved in the VTK polygonal data (VTP) format and automatically linked to the model.

3. Case study: MRI-based musculoskeletal model of the lower limb

We applied the described workflow for subject-specific OpenSim models (Fig. 4) to create a magnetic resonance imaging (MRI)-based multibody model of the lower limb that included an articulated linkage with:

- 4 rigid bodies: pelvis, thigh, shank and foot
- 5 degrees of freedom (DOF): spherical joint (3 DOF) at the hip, hinge joint (1 DOF) at the knee, hinge joint (1 DOF) at the ankle
- 43 musculotendon actuators according to the widely-used *Gait2392* model distributed in OpenSim [10]: gluteus maximus (3 lines of action), medius (3 lines of action), minimum (3 lines of action), piriformis, sartorius, tensor fascia latae, adductor magnus (3 lines of action), brevis, longus, gracilis, iliacus, psoas, pectineus, rectus femoris, gemellus, quadratus femoris, biceps femoris long head, semimembranosus, semitendinosus, biceps femoris short head, medial gastrocnemius, lateral gastrocnemius, vastus medialis, lateralis, intermedius, soleus, peroneus brevis, longus, tibialis posterior, flexor digitorum, flexor hallucis, extensor digitorum, extensor hallucis, peroneus tertium, tibialis anterior.

The model was created from lower-body MRI of a healthy subject (male; age: 31 years; height: 183 cm; weight: 70.5 kg) [19,23], publicly available via simtk.org for reproducibility. Bone and soft tissues were segmented semi-automatically using Amira (Visage Imaging, Berlin, Germany), to finally obtain the corresponding surfaces. We segmented both bone and relevant soft tissue through fixed thresholding followed by manual closing of the contours, according to the judgment of an experienced operator (bone, especially close to articular surfaces) or with reference to anatomical atlases (thin muscles and attachment of soft tissue to bone). We assumed that each body segment was made of two closed surfaces: bone and soft tissue. Each surface defining the body segments was imported as STL file format and hierarchically organized into the data tree (Fig. 5A).

Anatomical landmark clouds were created that included the following landmarks palpated on the bone surfaces (see Abbreviations for acronyms): RASIS, LASIS, RPSIS, LPSIS, on the pelvis; RHC, RME, RLE, on the femur; RMC, RLC, RMM, RLM, on the tibia; RPA_II, RPA_CA, on the foot (Fig. 5B). These landmarks are available in NMSBUILDER as sets of landmarks to create joint reference systems according to the ISB recommendations [25].

Hip, knee and ankle joint reference systems were created as anatomical reference systems from the defined sets of landmark clouds. Two reference systems per joint are automatically created, whose location and orientation are calculated with respect to the reference systems of the parent body and the child body respectively, according to the ISB recommendations [25] (Fig. 5C).

To define the geometry path of the muscle units, i.e. origin, insertion and intermediate via-points, the coordinates of the points defining the *Gait2392* model were imported (*Import Landmark*).

The imported landmark clouds were registered through affine registration onto the respective subject-specific target landmark clouds that included the skeletal landmarks palpated. The obtained muscular landmarks were then superimposed to the muscle volumes for visual check, and some landmarks were manually adjusted to lie in the respective volumes (Fig. 6A). The muscle landmarks were given coherent names, i.e. *muscle name orig, muscle name via 1, muscle name via 2, muscle name via n, muscle name ins*, to be subsequently used to create OpenSim muscle objects automatically.

The OpenSim model objects were then created as follows:

- Bodies

All bodies were assigned the same density values, i.e. 1.42 g/cm^3 and 1.03 g/cm^3 for bone and soft tissue surfaces, respectively. The inertial properties are automatically calculated by the software and stored in the *OpenSim Model* VME.

- Free joint with Ground

The pelvis body was connected to the ground via free joint.

- Joints

The hip joint was defined as a ball joint type, which articulates the previously defined hip joint reference system in parent with respect to the pelvis body and the hip joint reference system in child with respect to the thigh body. The knee joint was defined as a pin joint type, which articulates the previously defined knee joint reference system in parent with respect to the thigh body and the knee joint reference system in child with respect to the shank body. Finally, the ankle joint was defined as a pin joint type, which articulates the previously defined ankle joint reference system in parent with respect to the shank body and the ankle joint reference system in child with respect to the foot body.

- Marker Sets

All skeletal landmark clouds were attributed as marker sets to the corresponding bodies.

- Wrap Object

A *wrap* object was created to prevent soleus, medial and lateral gastrocnemius muscles from penetrating the tibia during ankle flexion. The *wrap* object was defined as a cylinder surface rigidly connected to the shank body (Fig. 6B).

- Auto Create Muscles

OpenSim muscle objects were automatically created from the landmarks that were found in the data tree and that defined each muscle path. For each muscle, the following parameters can be assigned through text entry: maximum isometric force, optimal fiber length, tendon slack length, pennation angle, activation and deactivation time. With the assumption of neglecting the force-length-velocity relationships of muscle, only the maximum isometric forces were inserted [27]. The maximum isometric force of the muscles were calculated previously [19,23], as the product of the muscle physiological cross-sectional area and the maximum muscle tension, and assuming muscle fiber length proportional to musculotendon length.

Finally, the created *OpenSim Model* VME was exported in the OpenSim file format (.osim) (Fig. 4).

4. Discussion

We provided an overview of NMSBUILDER 2.0, and presented a case study to create an OpenSim musculoskeletal model of the lower limb from segmented MRI data. The workflow for model creation that we implemented, led to develop a full state-of-the-art model of the lower limb within a few hours from 3D surfaces obtained by image segmentation. This result of efficiency and usability makes NMSBUILDER a powerful tool in the progress of personalized modeling and the consequent potential of musculoskeletal simulations of movement. The workflow includes standardization in the process of model creation, when defining OpenSim objects

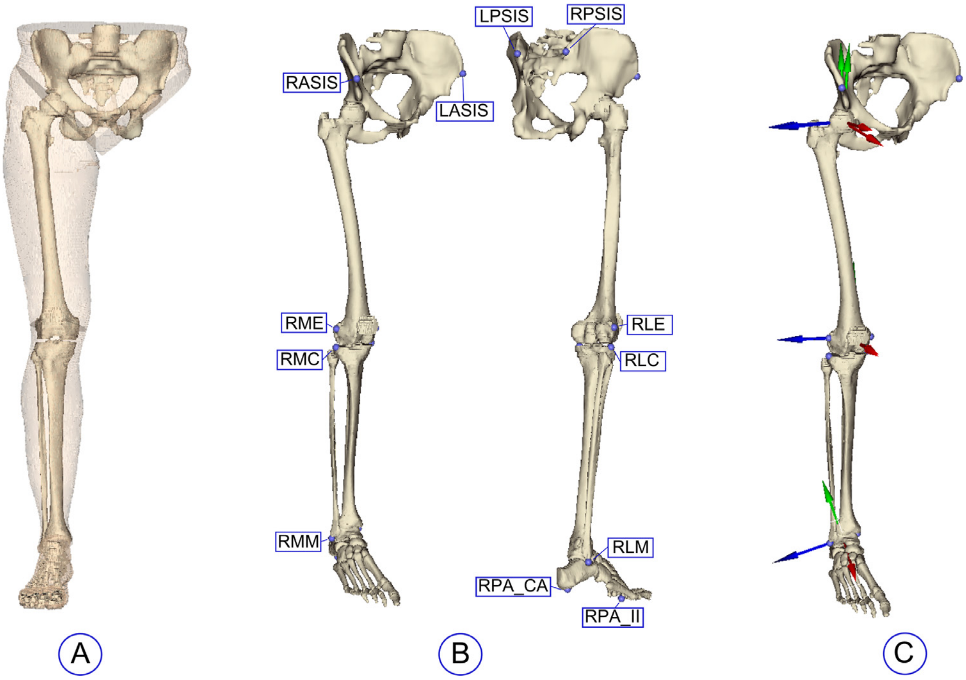

Fig. 5. Bodies, skeletal landmark clouds and joint reference systems of the model. (A) Four bodies made of two closed surfaces each, imported in NMSBUILDER as STL files. (B) Skeletal anatomical landmarks palpated on the bone surfaces. RASIS, right anterior superior iliac spine; LASIS, left anterior superior iliac spine; RPSIS, right posterior superior iliac spine; LPSIS, left posterior superior iliac spine; RHC, right hip center; RME, right medial epicondyle; RLE, right lateral epicondyle; RMC, right medial condyle RLC, right lateral condyle; RMM, right medial malleolus; RLM, right lateral malleolus; RPA_II, right plantar aspect on second metatarsal; RPA_CA, right plantar aspect on calcaneus. (C) Hip, knee and ankle joint reference systems defined from the skeletal landmarks.

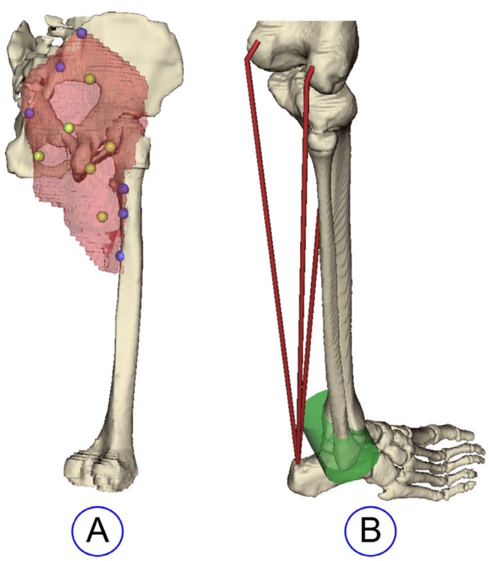

Fig. 6. Objects related to the creation of muscles. (A) Registration of muscular landmark clouds from the generic *Gait2392*) OpenSim model onto the specific anatomy: example for the three lines of action of the gluteus maximus. Purple landmarks represent origin and insertion points; green landmarks represent intermediate via-points. (B) Definition of a *wrap*) object, rigidly connected to the shank body, to prevent the lines of action of soleus, medial and lateral gastrocnemius from penetrating the tibia during ankle flexion. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

and models. In addition, there is the possibility to create anatomical reference frames that define the joint kinematics of the lower limbs according to international standards [25].

The current implementation of NMSBUILDER is to be considered as a tool to efficiently build subject-specific OpenSim models with essential features, saving time and effort and without requiring high skilled expertise. With this approach, we intend the software open to expansion and customization according to specific applications. Therefore, several limitations are present in the current implementation. The workflow is optimized for models of the lower limb, being the anatomical district with a larger interest within the musculoskeletal biomechanics community, included our research group. Models of the upper limb can be efficiently created, although more generic components need to be used and adapted to the purpose (e.g. definition of joint reference systems and constraints). To model joints, we implemented the pin, ball and custom joint classes, which are the most used in lower-limb models. In our case study, we used a pin joint for both the knee and the ankle: this assumption is widely adopted in the literature, and presents rather small differences compared to a hinge joint with coupled translations in terms of kinematics and muscle and joint forces during common motor activities when dealing with image-based models [23]. To implement more complex joints, the user currently needs to manually edit the custom joint object. Regarding the types of attachment points that define muscle paths, we implemented via points that are fixed to a body, and wrap points that allow wrapping over the surface of a wrap object. However, we did not implement moving muscle points, which would allow the muscle path to move in function of selected joint DOFs. This can be implemented directly in OpenSim, once via points have been defined in NMSBUILDER. In processing the case study, we chose a rather simple and overall time consuming segmentation technique. This choice was guided by the relatively coarse resolution and the signal limitations of the MR images: the whole lower body scan was taken with a 1.5 T MR scanner with a standard T1-weighted sequence, at 5 mm slice thickness and 5.5 mm slice spacing. We used a commercial software because available in our lab at the time the images were acquired, but we do not envisage serious hurdles to replicate our segmentation technique, as many alternative tools exist, including several instances of free software (e.g. ITK-SNAP [28] (www.itksnap.org), MITK (www.mitk.org) or Invesalius (www.cti.gov.br/invesalius)). Conversely, we see room for enhancement in the accuracy and automation of the segmentation technique, which may benefit from different MRI acquisitions or specific segmentation algorithms and software. In fact, interesting sequences to enhance bone contrast have been recently presented [29], and shape-based algorithms to improve segmentation of both bone [30] and soft tissues [31] are under development. We may consider including segmentation algorithms in future versions of NMSBUILDER.

5. Conclusions

We developed NMSBUILDER 2.0, a more stable, user-friendly and advanced version of the original software for biomedical data processing and creation of OpenSim musculoskeletal models dedicated to simulations of movement. The software application is freely available via the dedicated web site nmsbuilder.org. This tool presents an improved usability and efficiency following a major refactoring, which allow users to create complex multibody models in the OpenSim file format from 3D geometries with no additional software needed. The workflow for subject-specific model creation introduces standardization in the process, without requiring high skilled expertise. This allows saving time and effort, and then can contribute to promote personalized applications in musculoskeletal biomechanics, particularly to promote studies with larger sample size and clinical applications. In addition, as the software architecture allows for expansion of features for data processing and modeling, NMSBUILDER can be considered as a basis for future collaborations, where additional tools can be developed and implemented, according to specific applications in the field.

Acknowledgments

This study was partly funded by the Emilia Romagna Region—University Research Program 2010–2012 (Project name: "Applicability and efficacy of an innovative diagnostic-instrumental pathway for total hip arthroplasty in hip dysplasia patients to optimize functional recovery"). We would like to acknowledge additional people that contributed to the design and development of NMSBUILDER: Marco Viceconti, Stefano Perticoni, Debora Testi and Saulo Martelli, involved in the original design and development, as well as Luca Modenese, Erica Montefiori, Lorenzo Pitto, Daniele Ascani and Joe Prinold, involved in the current or past bug-fixing process.

Conflict of interest

All authors declare no financial and personal relationships with other people or organizations that could inappropriately influence (bias) their work.

References

[1] J.L. Hicks, T.K. Uchida, A. Seth, A. Rajagopal, S.L. Delp, Is my model good enough? Best practices for verification and validation of musculoskeletal models and simulations of movement, J. Biomech. Eng. 137 (2015) 020905.
[2] L.A. Fok, A.G. Schache, K.M. Crossley, Y.-C. Lin, M.G. Pandy, Patellofemoral joint loading during stair ambulation in people with patellofemoral osteoarthritis: joint loading in people with patellofemoral arthritis, Arthritis Rheum. 65 (2013) 2059–2069.
[3] G. Giarmatzis, I. Jonkers, M. Wesseling, S. Van Rossom, S. Verschueren, Loading of hip measured by hip contact forces at different speeds of walking and running, J. Bone Miner. Res. 30 (2015) 1431–1440.
[4] B.A. Knarr, D.S. Reisman, S.A. Binder-Macleod, J.S. Higginson, Understanding compensatory strategies for muscle weakness during gait by simulating activation deficits seen post-stroke, Gait Posture 38 (2013) 270–275.
[5] Z.F. Lerner, M.S. DeMers, S.L. Delp, R.C. Browning, How tibiofemoral alignment and contact locations affect predictions of medial and lateral tibiofemoral contact forces, J. Biomech. 48 (2015) 644–650.
[6] S.A. Schloemer, J.A. Thompson, A. Silder, D.G. Thelen, R.A. Siston, Age-related differences in gait kinematics, kinetics, and muscle function: a principal component analysis, Ann. Biomed. Eng. 45 (2017) 695–710.
[7] P. Sritharan, Y.-C. Lin, S.E. Richardson, K.M. Crossley, T.B. Birmingham, M.G. Pandy, Musculoskeletal loading in the symptomatic and asymptomatic knees of middle-aged osteoarthritis patients, J. Orthop. Res. 35 (2017) 321–330.
[8] G. Valente, F. Taddei, I. Jonkers, Influence of weak hip abductor muscles on joint contact forces during normal walking: probabilistic modeling analysis, J. Biomech. 46 (2013) 2186–2193.
[9] M. Wesseling, F. de Groote, C. Meyer, K. Corten, J.-P. Simon, K. Desloovere, I. Jonkers, Gait alterations to effectively reduce hip contact forces, J. Orthop. Res. 33 (2015) 1094–1102.
[10] S.L. Delp, F.C. Anderson, A.S. Arnold, P. Loan, A. Habib, C.T. John, E. Guendelman, D.G. Thelen, OpenSim: open-source software to create and analyze dynamic simulations of movement, IEEE Trans. Biomed. Eng. 54 (2007) 1940–1950.
[11] S.S. Blemker, D.S. Asakawa, G.E. Gold, S.L. Delp, Image-based musculoskeletal modeling: applications, advances, and future opportunities, J. Magn. Reson. Imaging 25 (2007) 441–451.
[12] L. Bosmans, M. Wesseling, K. Desloovere, G. Molenaers, L. Scheys, I. Jonkers, Hip contact force in presence of aberrant bone geometry during normal and pathological gait, J. Orthop. Res. 32 (2014) 1406–1415.
[13] T.A. Correa, R. Baker, H. Kerr Graham, M.G. Pandy, Accuracy of generic musculoskeletal models in predicting the functional roles of muscles in human gait, J. Biomech. 44 (2011) 2096–2105.
[14] F. Taddei, S. Martelli, G. Valente, A. Leardini, M.G. Benedetti, M. Manfrini, M. Viceconti, Femoral loads during gait in a patient with massive skeletal reconstruction, Clin. Biomech. 27 (2012) 273–280.
[15] G. Valente, L. Pitto, E. Schileo, S. Piroddi, A. Leardini, M. Manfrini, F. Taddei, Relationship between bone adaptation and in-vivo mechanical stimulus in biological reconstructions after bone tumor: a biomechanical modeling analysis, Clin. Biomech. 42 (2017) 99–107.
[16] M. Wesseling, F. De Groote, C. Meyer, K. Corten, J.-P. Simon, K. Desloovere, I. Jonkers, Subject-specific musculoskeletal modelling in patients before and after total hip arthroplasty*, Comput. Methods Biomech. Biomed. Eng. 19 (2016) 1683–1691.
[17] L. Scheys, A. Van Campenhout, A. Spaepen, P. Suetens, I. Jonkers, Personalized MR-based musculoskeletal models compared to rescaled generic models in the presence of increased femoral anteversion: effect on hip moment arm lengths, Gait Posture 28 (2008) 358–365.
[18] Z.F. Lerner, B.C. Gadomski, A.K. Ipson, K.K. Haussler, C.M. Puttlitz, R.C. Browning, Modulating tibiofemoral contact force in the sheep hind limb via treadmill walking: predictions from an OpenSim musculoskeletal model, J. Orthop. Res. 33 (2015) 1128–1133.
[19] G. Valente, L. Pitto, D. Testi, A. Seth, S.L. Delp, R. Stagni, M. Viceconti, F. Taddei, Are subject-specific musculoskeletal models robust to the uncertainties in parameter identification? PLoS ONE 9 (2014) e112625.
[20] J.A.I. Prinold, C. Mazzà, R. Di Marco, I. Hannah, C. Malattia, S. Magni-Manzoni, M. Petrarca, A.B. Ronchetti, L. Tanturri de Horatio, E.H.P. van Dijkhuizen, S. Wesarg, M. Viceconti, MD-PAEDIGREE consortium, a patient-specific foot model for the estimate of ankle joint forces in patients with juvenile idiopathic arthritis, Ann. Biomed. Eng. 44 (2016) 247–257.
[21] L. Modenese, E. Ceseracciu, M. Reggiani, D.G. Lloyd, Estimation of musculotendon parameters for scaled and subject specific musculoskeletal models using an optimization technique, J. Biomech. 49 (2016) 141–148.
[22] H. Kainz, L. Modenese, D.G. Lloyd, S. Maine, H.P.J. Walsh, C.P. Carty, Joint kinematic calculation based on clinical direct kinematic versus inverse kinematic gait models, J. Biomech. 49 (2016) 1658–1669.
[23] G. Valente, L. Pitto, R. Stagni, F. Taddei, Effect of lower-limb joint models on subject-specific musculoskeletal models and simulations of daily motor activities, J. Biomech. 48 (2015) 4198–4205.
[24] M. Viceconti, C. Zannoni, D. Testi, M. Petrone, S. Perticoni, P. Quadrani, F. Taddei, S. Imboden, G. Clapworthy, The multimod application framework: a rapid application development tool for computer aided medicine, Comput. Methods Programs Biomed. 85 (2007) 138–151.
[25] G. Wu, S. Siegler, P. Allard, C. Kirtley, A. Leardini, D. Rosenbaum, M. Whittle, D.D. D'Lima, L. Cristofolini, H. Witte, et al., ISB recommendation on definitions of joint coordinate system of various joints for the reporting of human joint motion—part I: ankle, hip, and spine, J. Biomech. 35 (2002) 543–548.
[26] A. Seth, M. Sherman, J.A. Reinbolt, S.L. Delp, OpenSim: a musculoskeletal modeling and simulation framework for in silico investigations and exchange, Proc. IUTAM. 2 (2011) 212–232.
[27] F.C. Anderson, M.G. Pandy, Static and dynamic optimization solutions for gait are practically equivalent, J. Biomech. 34 (2001) 153–161.
[28] P.A. Yushkevich, J. Piven, H.C. Hazlett, R.G. Smith, S. Ho, J.C. Gee, G. Gerig, User-guided 3D active contour segmentation of anatomical structures: significantly improved efficiency and reliability, Neuroimage 31 (2006) 1116–1128.
[29] K.A. Eley, A.G. Mcintyre, S.R. Watt-Smith, S.J. Golding, "Black bone" MRI: a partial flip angle technique for radiation reduction in craniofacial imaging, Br. J. Radiol 85 (2012) 272–278.
[30] S.S. Chandra, Y. Xia, C. Engstrom, S. Crozier, R. Schwarz, J. Fripp, Focused shape models for hip joint segmentation in 3D magnetic resonance images, Med. Image Anal. 18 (2014) 567–578.
[31] J. Kemnitz, F. Eckstein, A.G. Culvenor, A. Ruhdorfer, T. Dannhauer, S. Ring-Dimitriou, A.M. Sänger, W. Wirth, Validation of an active shape model-based semi-automated segmentation algorithm for the analysis of thigh muscle and adipose tissue cross-sectional areas, Magn. Reson. Mater. Phys. (2017) Epub ahead of print.