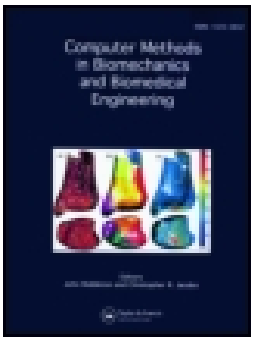


Taylor & Francis
Taylor & Francis Group

Computer Methods in Biomechanics and Biomedical Engineering

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/gcmb20

# BOPS: a Matlab toolbox to batch musculoskeletal data processing for OpenSim

**Bruno L. S. Bedo , Alice Mantoan , Danilo S. Catelli , Willian Cruaud , Monica Reggiani & Mario Lamontagne**

**To cite this article:** Bruno L. S. Bedo , Alice Mantoan , Danilo S. Catelli , Willian Cruaud , Monica Reggiani & Mario Lamontagne (2021): BOPS: a Matlab toolbox to batch musculoskeletal data processing for OpenSim, Computer Methods in Biomechanics and Biomedical Engineering

**To link to this article:** https://doi.org/10.1080/10255842.2020.1867978 https://doi.org/10.1080/10255842.2020.1867978

# BOPS: a Matlab toolbox to batch musculoskeletal data processing for OpenSim

Bruno L. S. Bedo[a,b], Alice Mantoan[c], Danilo S. Catelli[b], Willian Cruaud[b], Monica Reggiani[c] and Mario Lamontagne[b]

[a]Ribeirao Preto Medical School, University of Sao Paulo, Ribeirao Preto, SP, Brazil; [b]School of Human Kinetics, University of Ottawa, Ottawa, ON, Canada; [c]Department of Management and Engineering, University of Padova, Vicenza, Italy

**ABSTRACT**

This paper presents Batch OpenSim Processing Scripts (BOPS), a Matlab toolbox for batch processing common OpenSim procedures: Inverse Kinematics, Inverse Dynamics, Muscle Analysis, Static Optimization, and Joint Reaction Analysis. BOPS is an easy-to-use and highly configurable tool that aims to reduce the time required to process large datasets, thus fostering the adoption of musculoskeletal modeling and simulations in daily practice. Its graphical user interface includes pre-defined setup files and has been designed to fulfill the needs of different research projects by simplifying the customization of the procedures, facilitating the analysis, and boosting research group collaborations. BOPS is released under Apache License 2.0, and its source code is freely available on SimTK and GitHub.

**ARTICLE HISTORY**
Received 18 June 2020
Accepted 20 December 2020

**KEYWORDS**
Musculoskeletal simulations; OpenSim; data processing; Matlab toolbox

## Introduction

Musculoskeletal modeling provides a non-invasive approach to estimate muscle forces and joint contact loading, and it is an essential tool to better understand the musculoskeletal system's actions during human movement (Delp et al. 2007; Seth et al. 2018). However, it usually requires to collect motion data and to process and export them to obtain input data for the simulations. Besides, applications may require personalized musculoskeletal models, which thus must be generated. Commercial solutions and free tools are available to the biomechanics community to simplify each step within the workflow of neuro- and musculoskeletal modeling (Figure 1). For instance, BTK (Barre and Armand 2014) allows the reading and writing of C3D files, the standard file format in the field (C3D—Coordinate 3 D, Motion Lab Systems), while Mokka, a standalone application by the same authors, was conceived for their visual inspection; the MoCap (Burger and Toiviainen 2013) and biomechZoo (Dixon et al. 2017) are two toolboxes enabling motion data analysis and visualization; ADAT toolbox (James and Wixted 2011) is intended to handle sports movement data; pyomeca (Martinez et al. 2020) enables data extraction, processing and visualization for use in research and education. MOtoNMS (Mantoan et al. 2015) processes experimental data from different motion capture devices and generates inputs for neuromusculoskeletal modeling, while CEINMS (Pizzolato et al. 2015) is the first open-source toolbox that allows individual neural drive from EMG to be used as input into simulations. NMS Builder (Valente et al. 2017) was conceived as a free software application to generate subject-specific musculoskeletal models for OpenSim (Stanford University, Stanford, USA), as well as the open-source Musculoskeletal Atlas Project (MAP) Client software (Zhang et al. 2014), that provides workflows to generate musculoskeletal models from clinical data using statistical models.

Most of the listed tools have been developed in Matlab (Mathworks Inc., Natick, USA), indeed only those requiring high computational performances (CEINMS) or conceived as a standalone application (i.e. NMS Builder and OpenSim) are based on C++ programming language (Table 1). Therefore, Matlab turns out to be the preferred development tool for data processing and analysis in biomechanics, likely due to its general availability within research groups and the basic programming skills required.

**CONTACT** Monica Reggiani monica.reggiani@unipd.it
B.L.S.B., A.M. and D.S.C. contributed equally as first authors.

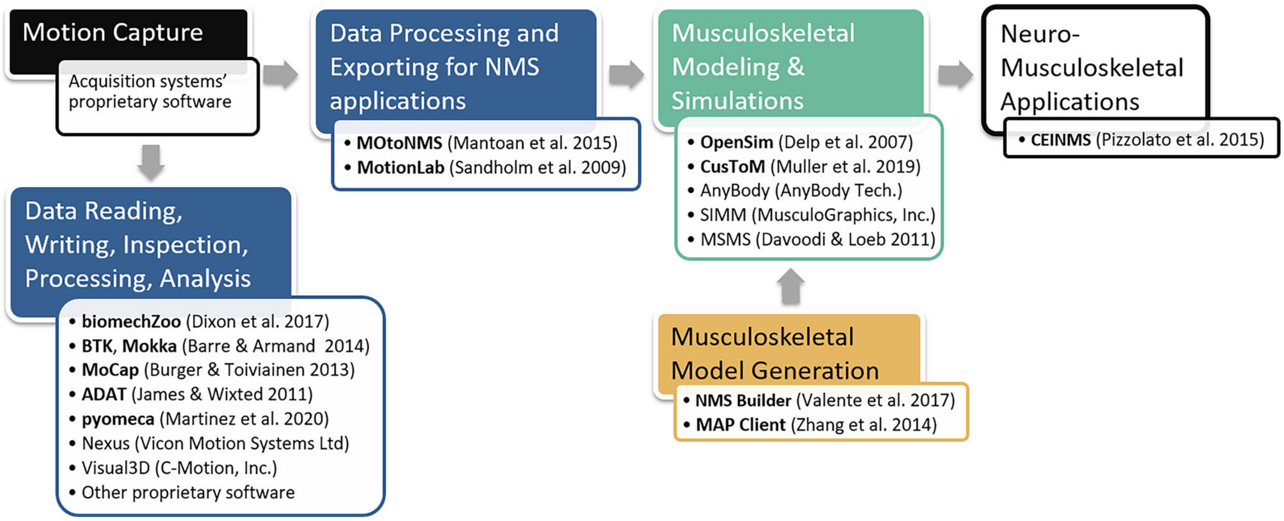


**Figure 1.** Workflow for neuro- and musculoskeletal modeling applications and list of toolboxes available to accomplish each step. Open-source toolboxes are highlighted in bold.

Although various commercial software tools to analyze human motion through computational modeling and simulation of musculoskeletal systems are available (Davoodi and Loeb, 2011; Damsgaard et al. 2006, Figure 1—red box), researchers often choose OpenSim for its open-source availability and the large community of users (Delp et al. 2007; Seth et al. 2018). OpenSim enables the estimation of joint angles, moments, muscle and joint contact forces for different movement tasks (Lerner et al. 2015; Rajagopal et al. 2016; Catelli et al. 2019b; Bedo et al. 2020). However, it often requires an extensive number of steps to process even a single trial,[1] which increases the risk of blunders when considering multiple trials and also results in long processing time.

Although Mansouri and Reinbolt (2012) proposed a Matlab/Simulink tool based on OpenSim to perform forward dynamics simulation of OpenSim models with feedback control, to the authors' knowledge, there is yet no open-access tool available to time-optimize data processing and outcome analysis in OpenSim.

Therefore, this paper presents Batch OpenSim Processing Scripts[2] (BOPS), an easy-to-use Matlab toolbox provided with a graphical user interface (GUI) to batch process complex biomechanics dataset with the following OpenSim procedures: *Inverse Kinematics*, *Inverse Dynamics*, *Muscle Analysis*, *Static Optimization*, and *Joint Reaction Analysis*.

This toolbox was conceived to foster musculoskeletal simulations for a large amount of data: it has the goal to decrease both processing and analysis time, and to move a step forward towards a standardization of the methodological approach, with the potential to encourage research collaborations and boost translation of applications into clinical practice. BOPS, indeed, allows to: (i) batch process a whole dataset using a single pipeline; (ii) track changes in processing parameters and compare results obtained with different setup combinations; and, (iii) organize results in structured folders to facilitate the retrieve and comparisons with previous or future analysis. BOPS was conceived and initially developed by the Rehabilitation Engineering Group (University of Padua, Italy) and made freely available to the research community in 2015 as a beta version (Mantoan and Reggiani 2015). The subsequent continuous process of extensive testing and refinement at the Human Movement Biomechanics Laboratory (University of Ottawa, Canada) led to the release of the current improved and stable version (2.0).

This paper describes the toolbox structure and then introduces a testing procedure with detailed results. Finally, the paper points out BOPS 2.0 key features and main advantages. Test data and results, freely available on the SimTK platform, show that BOPS is a user-friendly method that can efficaciously support musculoskeletal research teams in using OpenSim.

## Methods

### *System overview*

BOPS is designed to batch process data from musculoskeletal simulation using OpenSim. It was outlined in OpenSim 3.3 and it currently supports its most recent version (i.e. 4.1). The tool targets a broad spectrum of researchers as it does not require any computer programming skills after the one-time setup to interface Matlab with OpenSim APIs[3].

**Table 1.** Details on open-source toolboxes in Figure 1 (colors on the left column refer to the corresponding step and toolbox main goal depicted in Figure 1). The used platform, the availability of Test Data, and the License are compared. Links to the project pages, software repositories, and available documentation are included.

| Toolbox | Platform | Project page | Repository | Documentation | Test Data | License |
|---|---|---|---|---|---|---|
| biomechZoo | Matlab | biomechzoo.com/ | github.com/PhilD001/biomechZoo | x | x | Apache v2.0 |
| BTK, Mokka | C++ | code.google.com/archive/p/b-tk/ | code.google.com/archive/p/b-tk/github.com/Biomechanical-ToolKit | biomechanical-toolkit.github.io/ | ✓ | New BSD license |
| MoCap | Matlab | jyu.fi/music/coe/materials/mocaptoolbox | x | x | x | GPL |
| ADAT | Matlab | adat.qsportstechnology.com/ | x | x | x | Creative Commons |
| pyomeca | Python | github.com/pyomeca/ | github.com/pyomeca/pyomeca/ | github.com/pyomeca/Documentation | ✓ | Creative Commons |
| MOtoNMS | Matlab | simtk.org/home/motonms/ | github.com/RehabEngGroup/MOtoNMS | rehabenggroup.github.io/MOtoNMS/ | ✓ | GNU General Public License v3 |
| NMS Builder | C++ | nmsbuilder.org/ | x | nmsbuilder.org/downloads/nmsBuilder_Manual_102.pdf | ✓ | ALBA and OpenSim Library Licence |
| MAP Client | Python | map-client.readthedocs.io/en/latest/ | github.com/MusculoskeletalAtlasProject/mapclient | map-client.readthedocs.io/en/latest/developer/index.html | ✓ | GNU General Public License v3 |
| OpenSim | C++ | simtk.org/projects/opensim | github.com/opensim-org | simtk-confluence.stanford.edu/display/OpenSim/OpenSim + Documentation | ✓ | Apache v 2.0 |
| CusToM | Matlab | github.com/anmuller/CusToM | github.com/anmuller/CusToM | github.com/anmuller/CusToM/blob/master/Docs/CusToMDocumentation.pdf | ✓ | GPL-3.0 |
| CEINMS | C++ | simtk.org/projects/ceinms | github.com/CEINMS/CEINMS | simtk.org/frs/?group_id=840 | ✓ | Apache v 2.0 |

A GUI has been carefully designed to simplify the use of BOPS procedures. Six main panels provide the user with a full overview of the available functionalities and setup parameters: (i) *Settings*; (ii) *Inverse Kinematics* (IK); (iii) *Muscle Analysis* (MA); (iv) *Inverse Dynamic* (ID); (v) *Static Optimization* (SO); and (vi) *Joint Reaction Analysis* (JRA) (Figure 2 and Table 2). The GUI allows selecting multiple or single analyses (e.g. IK or JRA, only), and the processing of multiple or individual trials of a single subject at a time. In addition, the users can customize processing parameters (i.e. cutoff frequencies for filtering) and select the desired output results (i.e. the procedures to run and the plotting setups, such as variables of interest and axes labels) directly from the GUI.

### Data management

The toolbox integrates perfectly with OpenSim API and promotes the organization of data in a predefined folder structure (Mantoan et al. 2015) (Figure 3—left). This design choice has several advantages: (i) it provides a guideline to researchers in the setup of a new dataset; (ii) it makes it easier to collaborate and share data among research groups; (iii) it offers a method to keep a detailed track of the previously performed analyses. Indeed, in each trial folder, the setup files listing the parameter choices are included for future reference (System configuration section).

### Input data loading

The input data are the same required by OpenSim, i.e. markers trajectories and force plate data (*.trc* and *.mot* file formats, respectively). MOtoNMS tool (Mantoan et al. 2015), freely available, allows the transformation of data collected in a motion analysis laboratory into the inputs required by both BOPS and OpenSim and automatically organizes them in the recommended folder structure (Figure 3—left). However, the input data can also be extracted directly using the proprietary motion capture system software and manually stored as required.

### Output data generation and storage structure

The output data consist of the results from OpenSim procedures, i.e. joint angles (IK), joint moments (ID), muscle length and moment arms (MA), muscle activations and forces (SO), and joint reaction forces (JRA). All outputs are automatically generated and stored mirroring the input folder structure (*dynamicElaboration*, Figure 3—left), relieving the

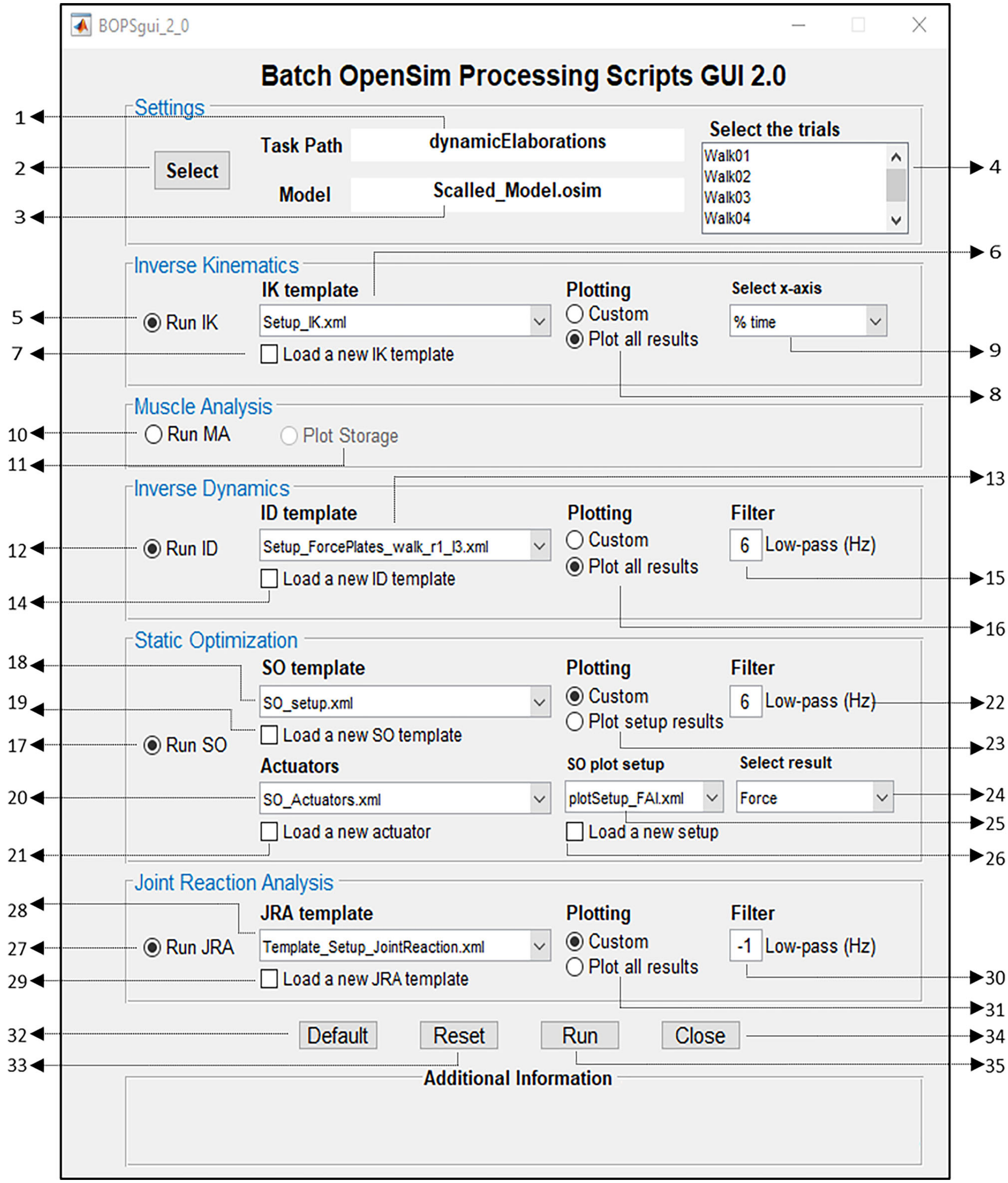


**Figure 2.** BOPS GUI. Layout of the user-friendly graphical Matlab interface available in BOPS. It allows to select which OpenSim procedures to perform sequentially, among *Inverse Kinematics*, *Inverse Dynamic*, *Muscle Analysis*, *Static Optimization*, and *Joint Reaction Analysis*, and to input both processing parameters and corresponding setup files. Please, refer to Table 2 for a detailed description of the GUI functionalities.

users from having to organize the storing of many output files manually and resulting in a consistent data organization. BOPS also gives the user a high level of output customization, allowing to export additional file formats such as *.sto* file (OpenSim storage), setups, logs and text formats files (Table 2), as well as to generate output figures based on the selected variables of interest (Figure 2).

**Table 2.** Summary of BOPS GUI functionalities. Detailed description of the GUI's main panels and functionalities available through the GUI graphical elements. The Matlab function styles are specified, as well as options available for the processing and any additional related information.

| Main panel | | Function style | Operation | Additional information |
|---|---|---|---|---|
| Settings | 1 | *text* | Folder where input data are located | Input data are .trc and .mot files |
| | 2 | *pushbutton* | Select the input data and OpenSim model folders | – |
| | 3 | *text* | OpenSim model's name | – |
| | 4 | *listbox* | Show all trials in the input data folder and allow to select those to process | Input data must be organized as shown in Figure 3 |
| Inverse Kinematic (IK) | 5 | *radiobutton* | Select IK procedure | – |
| | 6 | *popupmenu* | Show IK templates available and select the one to use | Recommended: locate IK templates in *Templates/ IK procedure* |
| | 7 | *checkbox* | Load an IK template located in a folder different from that recommended | – |
| | 8 | *radiobutton* | Plot IK results | *Custom* option: allow to select which results to plot *Plot all results* option: plot all IK angles based on IK template |
| | 9 | *popupmenu* | Select the *x-axis* variable for all plots | – |
| Muscle Analysis (MA) | 10 | *radiobutton* | Select MA procedure | – |
| | 11 | *radiobutton* | Plot MA results | – |
| Inverse Dynamic (ID) | 12 | *radiobutton* | Select ID procedure | – |
| | 13 | *popupmenu* | Show ID templates available and select the one to use | Recommended: locate ID templates in *Templates/ ID procedure* |
| | 14 | *checkbox* | Load an ID template located in a folder different from that recommended | – |
| | 15 | *edit* | Customize the low-pass cut frequency | |
| | 16 | *radiobutton* | Plot ID results | *Custom option*: allow to select which results to plot *Plot all results option*: plot all moments based on ID template |
| Static Optimization (SO) | 17 | *radiobutton* | Select SO procedure | – |
| | 18 | *popupmenu* | Show SO templates and select the one to use | Recommended: locate SO templates in *Templates/ StaticOptimization* |
| | 19 | *checkbox* | Load a SO template located in a folder different from that recommended | – |
| | 20 | *popupmenu* | Select actuators | – |
| | 21 | *checkbox* | Load an Actuators file located in a folder different from that recommended | Recommended: locate Actuators files in *Templates/ StaticOptimization* |
| | 22 | *edit* | Customize the low-pass cut frequency to SO analysis | |
| | 23 | *radiobutton* | Plot SO results | *Custom option*: allow to select which results to plot *Plot all results option*: plot all output based on SO template |
| | 24 | *popupmenu* | Select the results to plot | Options: Force and Activation |
| | 25 | *popupmenu* | Load a SO plot template | Recommended: locate SO plot templates in *Templates/ PlotStorage* |
| | 26 | *checkbox* | Load a SO plot setup located in a folder different from that recommended | – |
| Joint Reaction Analysis (JRA) | 27 | *radiobutton* | Select JRA procedure | – |
| | 28 | *popupmenu* | Show JRA templates available and select the one to use | Recommended: locate JRA templates in *Templates/JRA* |
| | 29 | *checkbox* | Load a JRA template located in a folder different from that recommended | – |
| | 30 | *edit* | Customize the low-pass cut frequency to JRA analysis | |
| | 31 | *radiobutton* | Plot JRA results | *Custom* option: allow to select which results to plot *Plot all results* option: plot all output based on JRA template |
| Final options | 32 | *pushbutton* | Set default configurations | – |
| | 33 | *pushbutton* | Reset all configurations | – |
| | 34 | *pushbutton* | Perform BOPS | – |
| | 35 | *pushbutton* | Close BOPS | – |

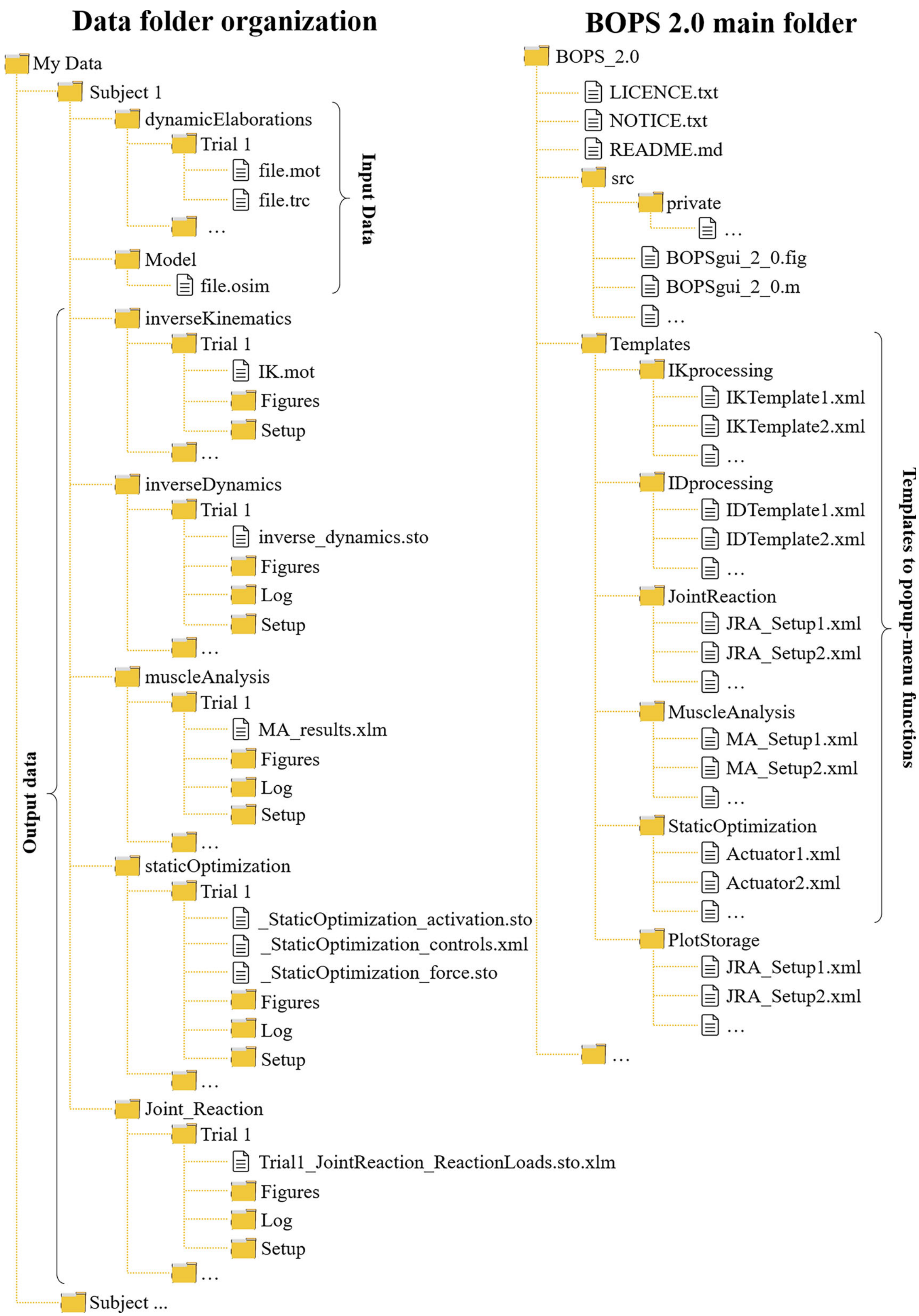


**Figure 3.** BOPS folders organization. The structure on the left presents the data organization suggested by the authors. All input data must be grouped in a *dynamicElaborations* folder, and subdivided in trials' folders that include both *.mot* and *.trc* files. This file organization can be easily obtained using MOtoNMS software. The OpenSim model file (*.osim*) can be automatically loaded on the BOPS GUI if located in a dedicated folder as recommended. The output data folders including processing results, figures, logs, and setup files are automatically created mirroring the structure of the *dynamicElaborations* folder. On the right, BOPS main folder organization is presented. The *.src* folder has the main source code while the *Templates* folder stores the setup files necessary to run the scripts.

### *System configuration*

BOPS has been designed to be a flexible and highly configurable toolbox that can fulfill the needs of different research projects without the necessity of accessing and changing the underlying source code. The user can indeed select directly from the intuitive and user-friendly interface: (i) the OpenSim procedures to run, (ii) the trials to process, (iii) the OpenSim model, scaled on the subject, to use on the simulations, (iv) the cutoff frequency for the filters, (v) the residual actuators, and (vi) the output variables to plot. When additional information is required for the processing, pre-defined setup files can be used to input: markers' names and weights, external loads (force plates), muscles and moment arms of interest, static optimization conditions and muscle reserve actuators loads, and joints of interest for contact force processing. The file format used for the setup files is the *.xml* (eXtensible Markup Language) as it is extremely suitable for encoding parameter information that is both human and machine-readable. BOPS already provides several template files in dedicated subfolders (one per each supported OpenSim procedure) distributed with BOPS 2.0 source code (Figure 3—right). Researchers that need to define new setups based on their laboratory configuration and analysis of interest can easily create new setup files customizing the templates already available. Similar to the processing parameters that can be specified through the GUI, also the setup files are automatically loaded and can be selected on a list box menu in BOPS 2.0. However, if the user prefers to store the setup files in a different location, a popup menu will allow the selection of the custom files.

### *Data processing*

BOPS has already been used to process different datasets for multiple applications (Ceseracciu et al. 2015; Pizzolato et al. 2015; Catelli et al. 2019a, 2020). However, to illustrate its potentialities, we analyzed data from a healthy participant (35.0 years, 78.2 kg, and 176 cm) performing five gait trials. A 10-camera capture system (200 Hz, MX-13 Vicon, Oxford, UK) recorded 3-dimensional kinematics of 45 full-body marker trajectories (Mantovani and Lamontagne 2017) and two force plates (1000 Hz, Bertec Corporation, Columbus, USA) recorded the ground reaction forces. The motion capture data were filtered and exported into *.mot* and *.trc* files (Nexus 2.8, Vicon, Oxford, UK). The simulation was performed using a generic musculoskeletal model (Catelli et al. 2019b) and included computation of joint angles (IK), joint moments (ID), and muscle activation (SO) during the stance phase. The JRA was also performed using resultant forces and moments acting on each articulating joint, the muscle forces, and the internal and external load applied to the model. Two skilled users of OpenSim were asked to perform the analysis on the same data, first using the OpenSim 3.3 GUI directly, and then batch processing the data using BOPS 2.0.

Processing times were recorded starting when the software tools (OpenSim and BOPS, respectively) were opened and ending with the storing of the final results.

## Results

The time required to process five gait trials using BOPS 2.0 was approximately 65% less than when processed in OpenSim GUI (435 ± 4 s using BOPS, and 1205 ± 27 s using OpenSim GUI). Figure 4 shows the results obtained through the OpenSim GUI and exploiting BOPS 2.0. The test data used for this experiment are available at the SimTK platform (*TestData-v2.0*), together with BOPS 2.0 source code, to allow reproducibility of results and any additional testing.

## Discussion

The purpose of this paper was to present BOPS, a Matlab toolbox to batch process the most common OpenSim procedures. Although OpenSim is an extremely efficient tool for musculoskeletal modeling and simulation investigations, it requires user expertise to execute repetitive steps that sometimes can be boring, laborious, or high-intensive work for the user. To the authors' knowledge, software tools to speed-up and automatize this process have not been proposed yet; nevertheless, their availability may result crucial when the datasets to analyze are composed of multiple trials, different tasks, and multiple subjects, as it is usually the case. BOPS addresses the need and aims to significantly reduce data processing time and facilitate the analysis in OpenSim, thus reducing the risk of missteps, and fostering to process larger datasets in musculoskeletal modeling and simulation. Mirroring the OpenSim approach and similarly to MOtoNMS, BOPS was conceived to operate on data coming from a single subject at a time. It is provided with the plotting functionality for visual inspection of outcomes at

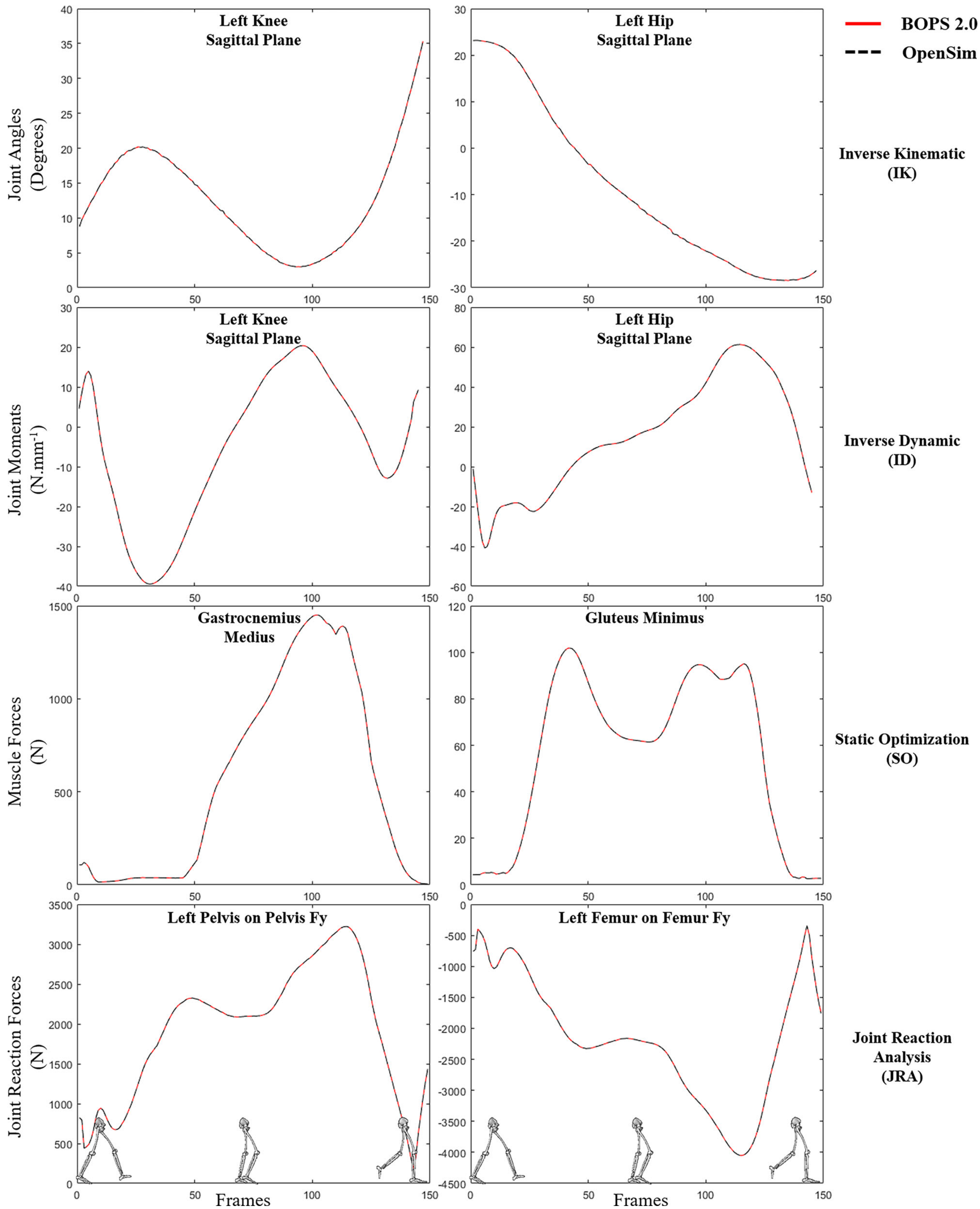


**Figure 4.** Testing procedure results. Examples of inverse kinematic, inverse dynamic, static optimization and joint reaction results from BOPS (red) and OpenSim (black) during the stance phase of a gait trial. The results obtained with the two approaches fully overlap.

all the intermediate stages (i.e. for all selected OpenSim procedures), combining the intent to speed up a process that would have to be done manually from OpenSim GUI, with the need to always keep a critical view on the results to assure the quality of simulations.

BOPS 2.0 is provided with a GUI (Figure 2) that allows users to: (i) load and select the trials to process and the pre-defined setup files, (ii) decide the sequence of OpenSim procedures to run, (iii) input cutoff frequencies for the filters of each OpenSim procedure, and (iv) select variables to be plotted, directly from the intuitive interface. This makes batch data processing more straightforward and faster than processing the trials individually through the OpenSim GUI, as also confirmed by the time required to elaborate the Test Data (approx. 65% less). It can reasonably be expected that the time gain would be much more appreciable as the dataset dimension and the number of trials increase.

The customizable selection of all parameters of interest on the single GUI layout offered by BOPS 2.0 has the potential not only to speed up the configuration process but also to avoid incorrect selection or inaccurate manual inputs on the numerous panels when executing multiple times the same analyses using the OpenSim GUI.

Since BOPS relies on OpenSim APIs, fully matching results were expected among the two different approaches, which were indeed obtained (Figure 4), confirming that BOPS can be considered a reliable tool.

Moreover, it is always possible to load the setup or output files generated with BOPS into OpenSim, for results reproduction or visualization. BOPS, indeed, stores output files and figures, and processing logs and .xml setups files into output directories automatically generated and organized with a uniquely defined structure. This standard approach to dataset organization facilitates data exchange among research groups, which can customize their tools and applications relying on a common data structure.

Results show that BOPS 2.0 succeeds in processing musculoskeletal data consistently and repeatably based on the user-defined system configuration. The Test Data used in this study are included as a supplementary file for results replicability, while additional test data are available on the SimTK project platform (*TestData-v0.9*) to allow the testing of the tool on different datasets. A tutorial video is also available in the BOPS project on the SimTK platform to illustrate how to use the toolbox and to provide an overview of its potentialities.

BOPS is released under Apache License 2.0 to encourage its use and it is freely available on the SimTK platform (https://simtk.org/projects/bops). The latest updates can be found on the GitHub repository (https://github.com/RehabEngGroup/BOPS), where the OpenSim research community is strongly encouraged to collaborate on its improvement. Authors' plans for future developments include the addition of other OpenSim procedures (i.e. Computed Muscle Control) and the improvement of real-time feedback during the processing, logging files, data visualization and output plotting features. This could allow to evaluate the implementation of an additional layer to process multiple subjects as a major update in the future, thus further speeding up the processing of large datasets while limiting the risk of compromising accuracy and reliability of results.

BOPS has been developed in Matlab for its widespread use in the biomechanics field (Table 1). Other freely available development platforms, such as Python (Python Software Foundation) or GNU Octave were considered five years ago, during the initial stages of this project, but discarded as potentially less comfortable for the users working on this field. Indeed, when directly compared with Python, the PubMed database show a higher incidence of "biomechanics toolbox" and "human movement toolbox" wording associated with Matlab (13 and 41, respectively) instead of Python (two and seven, respectively). However, Python scripting and extensions of the OpenSim API is available since OpenSim 3.2 and other biomechanics software are being provided with a Python package and/or binding (e.g. BTK, since 2014). Therefore, for its increasing use also in this field, for the higher computing performance, and to encourage adoption also from users who could prefer a free option, a Python BOPS version will be implemented in the near future.

The authors expect that BOPS will be useful to the research community by significantly reducing the data processing time, facilitating OpenSim analysis, and boosting research group collaborations. The availability of such an easy-to-use and highly configurable toolbox will potentially facilitate researchers to use musculoskeletal modeling and simulations in research, training, and clinical applications.

## Notes

1. OpenSim User's Guide: simtk-confluence.stanford.edu:8443/display/OpenSim/User's+Guide
2. Batch OpenSim Processing Scripts: github.com/RehabEngGroup/BOPS or simtk.org/projects/bops
3. Scripting with MATLAB: simtk-confluence.stanford.edu/display/OpenSim/Scripting+with+Matlab

## Availability and requirements

**Project name**: BOPS

**Project links:**
SimTK (https://simtk.org/projects/bops)
GitHub (https://github.com/RehabEngGroup/BOPS)
**Programming language**: Matlab
**License:** Batch OpenSim Processing Scripts (BOPS) Licensed under the Apache License, Version 2.0 (the "License")
**Any restrictions to use by non-academics**: None
**Software requirements:** OpenSim 3.3 and 4.x.
**Additional files:**
Test Data v2.0
Tutorial video exemplifying how to use BOPS.

## Disclosure statement

The authors do not have any conflicts to declare that could inappropriately influence this article.

## Funding

This study was partially financed by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior—Brasil (CAPES) [Finance Code 001]; Canadian Institutes of Health Research (CIHR) [PJT-162333].

## ORCID

Bruno L. S. Bedo http://orcid.org/0000-0003-3821-2327
Alice Mantoan http://orcid.org/0000-0001-5538-1291
Danilo S. Catelli http://orcid.org/0000-0002-8546-0836
Willian Cruaud http://orcid.org/0000-0001-8186-9311
Monica Reggiani http://orcid.org/0000-0002-5029-1892
Mario Lamontagne http://orcid.org/0000-0001-5635-7162

## References

Barre A, Armand S. 2014. Biomechanical toolkit: Open-source framework to visualize and process biomechanical data. Comput Methods Programs Biomed. 114(1): 80–87.

Bedo BLS, Catelli DS, Lamontagne M, Santiago PRP. 2020. A custom musculoskeletal model for estimation of medial and lateral tibiofemoral contact forces during tasks with high knee and hip flexions. Comput Methods Biomech Biomed Engin. 23(10):658–663.

Burger B, Toiviainen P. 2013. MoCap Toolbox - A Matlab toolbox for computational analysis of movement data. In: Sound and Music Computing Conference 2013.

Catelli DS, Ng KCG, Kowalski E, Beaulé PE, Lamontagne M. 2019a. Modified gait patterns due to cam FAI syndrome remain unchanged after surgery. Gait Posture. 72:135–141.

Catelli DS, Wesseling M, Jonkers I, Lamontagne M. 2019b. A musculoskeletal model customized for squatting task. Comput Methods Biomech Biomed Engin. 22(1):21–24.

Catelli DS, Ng KCG, Wesseling M, Kowalski E, Jonkers I, Beaulé PE, Lamontagne M. 2020. Hip muscle forces and contact loading during squatting after cam-type FAI surgery. J Bone Joint Surg Am. 102(Issue Suppl 2):34–42.

Ceseracciu E, Mantoan A, Bassa M, Moreno JC, Pons JL, Prieto GA, Del Ama AJ, Marquez-Sanchez E, Gil-Agudo A, Pizzolato C, et al. 2015. A flexible architecture to enhance wearable robots: Integration of EMG-informed models. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. Hamburg; p. 4368–4374.

Damsgaard M, Rasmussen J, Christensen ST, Surma E, de Zee M. 2006. Analysis of musculoskeletal systems in the AnyBody Modeling System. Simul Model Pract Theory. 14(8):1100–1111.

Davoodi R, Loeb G E. 2011. MSMS software for VR simulations of neural prostheses and patient training and rehabilitation. Stud Health Technol Inform. 163:156–162.

Delp SL, Anderson FC, Arnold AS, Loan P, Habib A, John CT, Guendelman E, Thelen DG. 2007. OpenSim: open-source software to create and analyze dynamic simulations of movement. IEEE Trans Biomed Eng. 54(11):1940–1950.

Dixon P C, Loh J J, Michaud-Paquette Y, Pearsall D J. 2017. biomechZoo: An open-source toolbox for the processing, analysis, and visualization of biomechanical movement data. Comput. Methods Programs Biomed. 140:1–10.

James DA, Wixted A. 2011. ADAT: a Matlab toolbox for handling time series athlete performance data. Procedia Eng. 13:451–456.

Lerner ZF, DeMers MS, Delp SL, Browning RC. 2015. How tibiofemoral alignment and contact locations affect predictions of medial and lateral tibiofemoral contact forces. J Biomech. 48(4):644–650.

Mansouri M, Reinbolt J A. 2012. A platform for dynamic simulation and control of movement based on OpenSim and MATLAB. J. Biomech. 45(8):1517–1521.

Mantoan A, Reggiani M. 2015. BOPS v0.9 (Version v0.9). Zenodo.

Mantoan A, Pizzolato C, Sartori M, Sawacha Z, Cobelli C, Reggiani M. 2015. MOtoNMS: a MATLAB toolbox to process motion data for neuromusculoskeletal modeling and simulation. Source Code Biol Med. 10(1):12.

Mantovani G, Lamontagne M. 2017. How different marker sets affect joint angles in inverse kinematics framework. J Biomech Eng. 139(4):044503.

Martinez R, Michaud B, Begon M. 2020. Pyomeca: an open-source framework for biomechanical analysis. JOSS. 5(53):2431.

Muller A, Pontonnier C, Puchaud P, Dumont G. 2019. CusToM: a Matlab toolbox for musculoskeletal simulation. JOSS. 4(33):927.

Pizzolato C, Lloyd DG, Sartori M, Ceseracciu E, Besier TF, Fregly BJ, Reggiani M. 2015. CEINMS: a toolbox to investigate the influence of different neural control solutions on the prediction of muscle excitation and joint moments during dynamic motor tasks. J Biomech. 48(14):3929–3936.

Rajagopal A, Dembia CL, DeMers MS, Delp DD, Hicks JL, Delp SL. 2016. Full-body musculoskeletal model for muscle-driven simulation of human gait. IEEE Trans Biomed Eng. 63(10):2068–2079.

Sandholm, A, Pronost, N, Thalmann, D. 2009. MotionLab: A Matlab Toolbox for extracting and processing experimental motion capture data for neuromuscular simulations, In: Second 3D Physiological Human Workshop (3DPH).

Seth A, Hicks JL, Uchida TK, Habib A, Dembia CL, Dunne JJ, Ong CF, DeMers MS, Rajagopal A, Millard M, et al. 2018. OpenSim: Simulating musculoskeletal dynamics and neuromuscular control to study human and animal movement. PLoS Comput Biol. 14(7):e1006223.

Valente G, Crimi G, Vanella N, Schileo E, Taddei F. 2017. nmsBuilder : Freeware to create subject-specific musculoskeletal models for OpenSim. Comput. Methods Programs Biomed. 152:85–92.

Zhang J, Sorby H, Clement J, Thomas CDL, Hunter P, Nielsen P, Lloyd D, Taylor M, Besier t. 2014. The MAP client: User-friendly musculoskeletal modelling workflows. In: Bello F, Cotin S, editors. Biomedical simulation. Vol. 8789. Strasbourg: Springer; p. 182–192.