

biomechZoo: An open-source toolbox for the processing, analysis, and visualization of biomechanical movement data

Philippe C. Dixon^{a,b,*}, Jonathan J. Loh^c, Yannick Michaud-Paquette^d, David J. Pearsall^d

^a*Department of Environmental Health, Harvard T.H. Chan School of Public Health*

^b*Liberty Mutual Research Institute for Safety, Hopkinton, MA, USA*

^c*Medicus Corda, Montreal, Canada*

^d*Department of Kinesiology and Physical Education and McGill Research Centre for Physical Activity and Health, McGill University, Montreal, Quebec, Canada*

Abstract

It is common for biomechanics data sets to contain numerous dependent variables recorded over time, for many subjects, groups, and/or conditions. These data often require standard sorting, processing, and analysis operations to be performed in order to answer research questions. Visualization of these data is also crucial. This manuscript presents biomechZoo, an open-source toolbox that provides tools and graphical user interfaces to help users achieve these goals. The aims of this manuscript are to (1) introduce the main features of the toolbox, including a virtual three-dimensional environment to animate motion data (Director), a data plotting suite (Ensembler), and functions for the computation of three-dimensional lower-limb joint angles, moments, and power and (2) compare these computations to those of an existing validated system. To these ends, the steps required to process and analyze a sample data set via the toolbox are outlined. The data set comprises three-dimensional marker, ground reaction force (GRF), joint kinematic, and joint kinetic data of subjects performing straight walking and 90° turning manoeuvres. Joint kinematics and kinetics processed within the toolbox were found to be similar to outputs from a commercial system. The biomechZoo toolbox represents the work of several years and multiple contributors to provide a flexible platform to examine time-series data sets typical in the movement sciences. The toolbox has previously been used to process and analyse walking, running, and ice hockey data sets, and can integrate existing routines, such as the KineMat toolbox, for additional analyses. The toolbox can help researchers and clinicians new to programming or biomechanics to process and analyze their data through a customizable workflow, while advanced users are encouraged to contribute additional functionality to the project. Students may benefit from using biomechZoo as a learning and research tool. It is hoped that the toolbox can play a role in advancing research in the movement sciences. The biomechZoo m-files, sample data, and help repositories are available online (www.biomechzoo.com) under the Apache 2.0 License. The toolbox is supported for Matlab (r2014b or newer, The Mathworks Inc., Natick, USA) for Windows (Microsoft Corp., Redmond, USA) and Mac OS (Apple Inc., Cupertino, USA).

Keywords: Biomechanics, Gait, Kinematics, Kinetics, Programming, Visualization, Analysis, Processing

*Corresponding author

Email address: pdixon@hsph.harvard.edu or philippe.dixon@gmail.com (Philippe C. Dixon)

URL: <http://www.biomechzoo.com> (Philippe C. Dixon)

1. Introduction

The field of movement science has a long history [2], with the advent of the modern computer arguably providing the latest impetus for advances in our understanding of movement. Today, commercial motion capture data collection systems abound and are often packaged with software capable of computing standard biomechanical variables such as joint kinematics and kinetics. Popular examples are the Vicon (Vicon Motion Systems Ltd., Oxford, UK), Qualisys (Qualisys AB, Göteborg, Sweden), and Motion Analysis (Motion Analysis Corp., Santa Rose, USA) systems. Unfortunately, these software are inflexible, costly, and opaque, making it difficult to adapt the software to the needs of a particular research project or to examine the underlying code. Visual3D (C-Motion, Inc., Germantown, USA) is a popular alternative, allowing users to develop bespoke pipelines; however, the platform is costly and advanced processing can be difficult to implement.

Many scientists instead choose to use platforms such as Matlab (The Mathworks Inc., Natick, USA) to develop and share custom toolboxes given the simplicity, flexibility, and robustness of the platform. There are many such toolboxes available to address specific needs of the movement science community. An early example is the KineMat toolbox [24] which provides routines for three-dimensional kinematic analyses. Other notable examples are the Mo-Cap [5], MOtoNMS [22], and ADAT [16] toolboxes which provide specialized analyses for music related movement, neuromuscular simulation, or sport, respectively. It is also important to highlight the BTK project [4] which contains functions to read, edit, and visualize motion capture data. This project is active and aims to expand to include greater analysis capabilities [3]. Finally, the International Society of Biomechanics maintains a repository of numerous user-generated software applications on their website (<http://isbweb.org/software/>). These toolboxes play crucial roles in the advancement of the movement sciences; however, what is lacking is a comprehensive framework from which to implement these various projects.

Although the research interests of the movement science community are varied, data sets are often remarkably similar. In many cases, data sets comprise numerous variables recorded over time from one or more data acquisition systems for many subjects, groups, and/or conditions. These data sets can therefore rely on a similar framework to perform sorting, processing, analysis, and visualization operations. The biomechZoo toolbox is an open-source project that provides such a framework. We strive to highlight the importance of systematic data processing and analysis procedures in improving research. Thus, the analogy of a Zoo, where the animal kingdom phyla are neatly organized, is the inspiration and guiding principle behind biomechZoo.

The aims of this manuscript are to present biomechZoo through the processing, analysis, and visualization of a sample data set and to compare biomechZoo's kinematic and kinetic computations against existing software. The toolbox can help researchers, clinicians, and students new to programming or movement science to process their data through a customizable work-flow, while advanced users are encouraged to contribute additional functionality.

2. Program description

The biomechZoo framework and features, including the web resources and graphical user interfaces (GUIs) are now described. The biomechZoo project contains two separate GitHub repositories: “biomechZoo” contains the toolbox code (m-files), while “biomechZoo-help” contains the sample data described in this manuscript, example scripts demonstrating the toolbox functions, and supplemental help materials. These repositories are all forkable and can be accessed through GitHub or the project website (www.biomechzoo.com).

In this manuscript, file names are typeset in typewriter font while commands, GUI headings, extensions, and file components are *italicized*.

2.1. General framework

The biomechZoo framework is set-up to allow users to process, analyze, and visualize motion capture data or data from any other field in which multiple time-dependant variables are collected. Reliance on this framework itself, rather

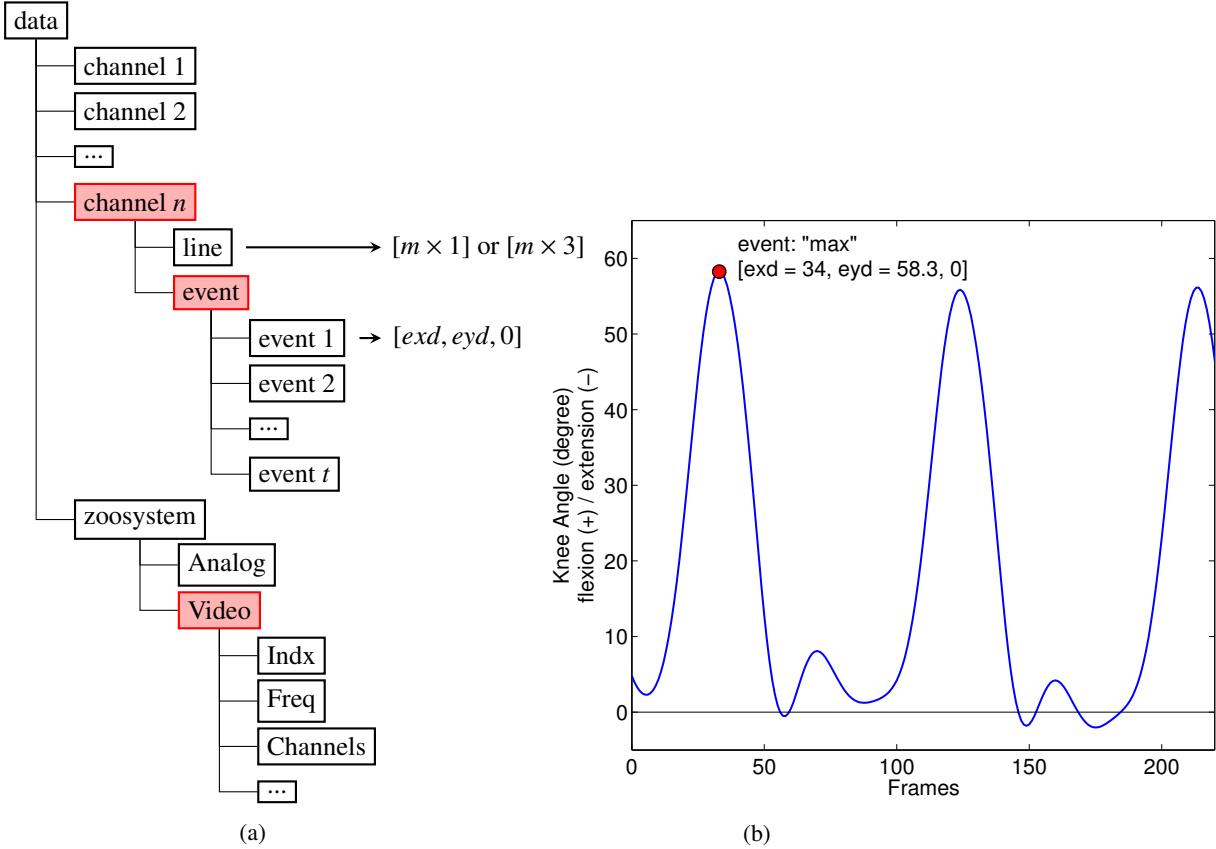


Figure 1: The biomechZoo file structure: (a) Diagrammatic representation of a zoo file with key fields expanded (n th data channel shows *line* and *event* sub-branches, *event* sub-branch reveals t events, and zoosystem channel makes visible various meta-information fields) and (b) plot of an exemplar zoo channel where *line* data is a $[220 \times 1]$ vector and the coordinates of an event “max” are shown.

than particular functions, allows biomechZoo to incorporate functionality from existing software and toolboxes (see Section 3.3 for how processes from the KineMat toolbox [24] and Vicon (Vicon Motion Systems Ltd., Oxford, UK) modeller were incorporated into biomechZoo). For each research project, a processing “pipeline” can be designed using blocks of biomechZoo functions, i.e., code cells defined by double comment characters (%%). This allows the user to focus on a particular process and to iteratively update the pipeline in preparation for the final analysis.

2.2. File structure and format

BiomechZoo files (zoo files) are standard Matlab MAT-files (.mat) saved with the extension .zoo. The use of this novel extension helps ensure that the appropriate files in a folder are loaded by biomechZoo functions. Each zoo file is a structured array with $n + 1$ branches, where n represents the total number of video and analog variables (channels) recorded by the data acquisition system (Fig. 1). An additional *zoosystem* branch, contains meta-information related to the trial (e.g. sampling frequency). Each video or analog data channel branch is further divided into *line* and *event* sub branches. The *line* sub branch contains the time series data as a $m \times 1$ or $m \times 3$ matrix for one-dimensional or three-dimensional data, respectively, where m represents the number of data frames. The *event* sub branch is a 1×3 vector ([frame number, line value, 0]) that stores discrete information computed for a given channel. For example, if the maximum value y_{max} of a channel occurs at frame x , then the event will read $[x, y_{max}, 0]$. The third entry is currently unused and is set to 0 by default. Many t events can be added to a given channel’s event branch.

Table 1: Main functions available in the biomechZoo toolbox

Function	Description
<code>bmech_addevent(fld,ch,<i>ename</i>,<i>type</i>,<i>nfld</i>)</code>	Adds events <i>ename</i> defined by <i>type</i> to channel <i>ch</i>
<code>bmech_explode(fld,ch)</code>	Splits $m \times 3$ channel(s) <i>ch</i> into three $m \times 1$ channels
<code>bmech_filter(fld,<i>ch</i>,<i>filt</i>)</code>	Filters channel(s) <i>ch</i>
<code>bmech_jointcentrePiG(fld,<i>joints</i>)</code>	Computes ankle, knee, and hip joint centres (c.f. [29])
<code>bmech_kinetics(fld,<i>settings</i>)</code>	Computes joint kinetics based on the PiG approach (c.f. [29])
<code>bmech_kinematics(fld,<i>settings</i>)</code>	Computes joint kinematics based on the PiG approach (c.f. [29])
<code>bmech_kinematicsRvdB(fld,<i>pelvis</i>,<i>thigh</i>,<i>shank</i>,<i>seq</i>)</code>	Computes joint kinematics via the KineMat toolbox (c.f. [24])
<code>bmech_normalize(fld,<i>ch</i>,<i>nlength</i>,<i>method</i>)</code>	Time normalization of data to a given number of frames
<code>bmech_processGRF(fld,<i>filt</i>)</code>	Performs basic processing of ground reaction force data
<code>bmech_partition(fld,<i>evt1</i>,<i>evt2</i>,<i>nfld</i>)</code>	Partitions data between frames defined by events <i>evt1</i> and <i>evt2</i>
<code>bmech_removechannel(fld,<i>ch</i>,<i>action</i>)</code>	Removes or keeps (<i>action</i>) channel(s) <i>ch</i>
<code>bmech_removefolder(fld,<i>sfld</i>)</code>	Removes subfolder(s) <i>sfld</i> from data set
<code>c3d2zoo(fld,<i>del</i>)</code>	Converts files in <i>fld</i> from .c3d to .zoo format
<code>director(fld,<i>del</i>)</code>	Three-dimensional environment for motion data animation
<code>engine(<i>varargin</i>)</code>	Recursively searches and returns files based on <i>varargin</i>
<code>ensemblер(fld,<i>del</i>)</code>	Data plotting suite for figure preparation
<code>eventval(<i>varargin</i>)</code>	Exports event data to a spreadsheet based on <i>varargin</i>
<code>nrmse(<i>a</i>,<i>b</i>)</code>	Computes normalised root mean squared error between <i>a</i> and <i>b</i>
<code>outlier(fl,<i>ch</i>,<i>event</i>)</code>	Tags channels <i>ch</i> in file <i>fl</i> as an outlier
<code>grab</code>	Loads individual zoo files into the Matlab workspace
<code>zsave</code>	Saves a zoo file to disk

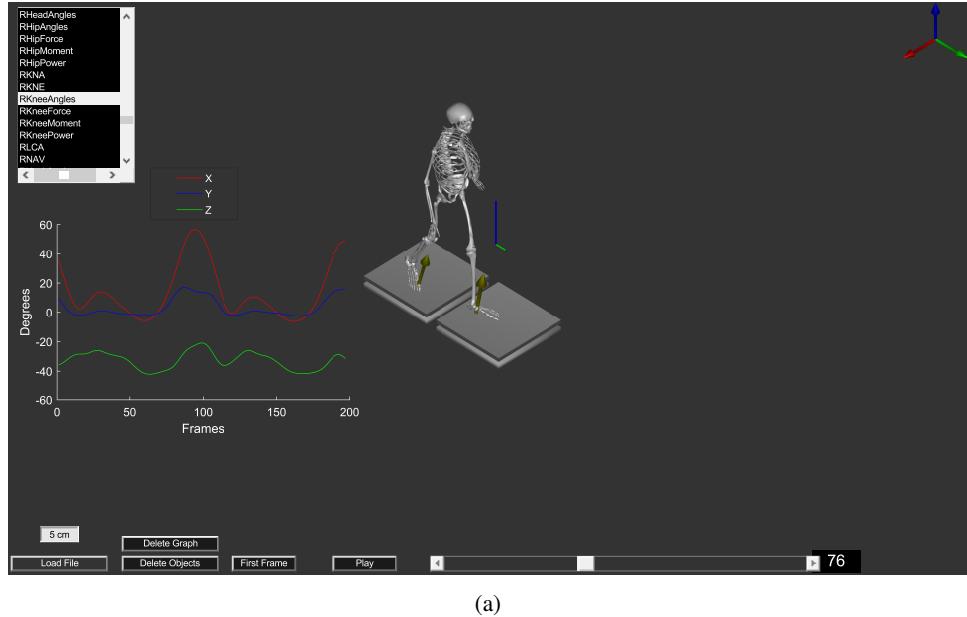
Main biomechZoo functions used in the current study. Batch processing functions acting on folder *fld* contain the `bmech_` prefix. **Bold** arguments are required inputs. Additional arguments (normal font) are described in each function's help section. Related functions acting on zoo files or zoo channels with suffixes _data or _line, respectively, are not shown. Example scripts, with the suffix _example, associated to these functions can be found in the help repository. See Toolbox subfolders for additional functions.

2.3. Batch processing functions

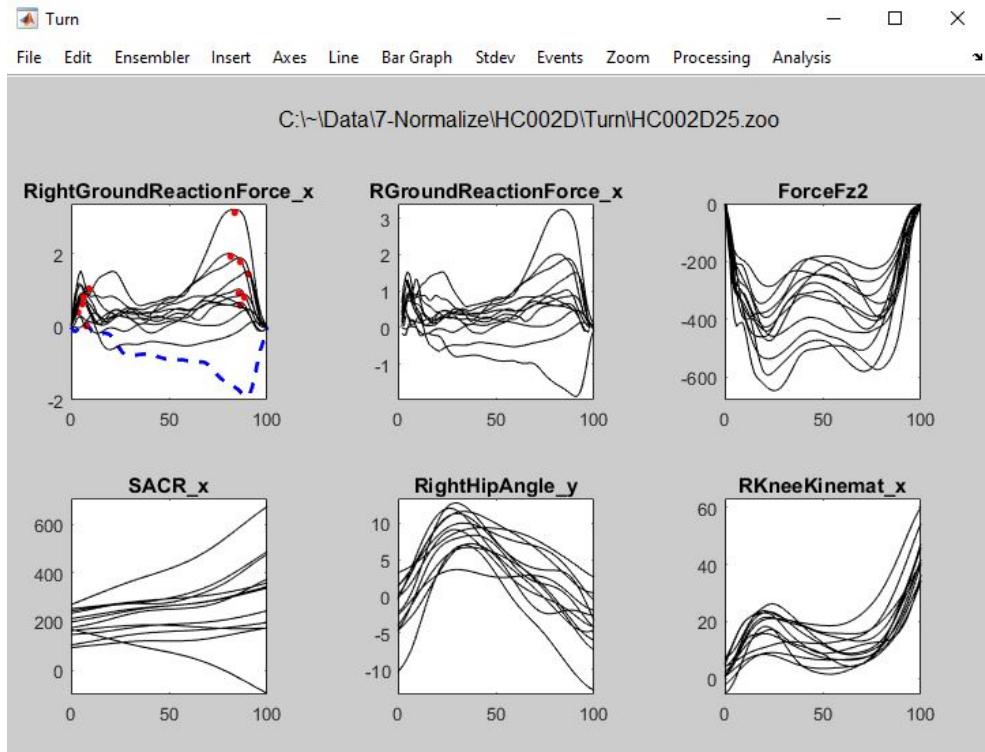
The biomechZoo toolbox is powered by functions that perform specific roles. Functions are stored within ten sub folders (Biomech Ops, Gait, KineMat, Mac Fixes, PiG Ops, Statistics, Support Functions, Visualization, Zoo Conversions, and Zoo Processing). The biomechZoo framework allows processes to be repeated automatically across the entire data set using batch processing algorithms. These functions are identifiable via the prefix `bmech_`. Related functions acting on a zoo file or line data contain the suffix _data or _line , respectively. For example, `bmech_normalize`, `normalize_data`, and `normalize_line`, time normalizes a set of files in a folder, a single loaded zoo file, or a single channel from a loaded zoo file, respectively. Main functions used in the current study are described in Table 1. See Appendix A for sample code to build a generic batch processing function. A log of the processing steps performed can be found in the *data.zoosystem.Processing* field of each zoo file.

2.4. Graphical user interfaces

The biomechZoo toolbox contains two GUIs: Director and Ensembler (Fig. 2), accessed by typing `director` or `ensemblер`, respectively, in the Matlab command window. Director is a virtual three-dimensional environment used to animate motion capture data. Quantities such as joint angles can be graphed alongside the animations for enhanced data visualization. Force plates and accompanying ground reaction forces (GRFs) as well as custom objects (props) can also be added by the user (see `sample_prop_example`). The Ensembler GUI plots average time series data or bar graphs sorted by group or condition. Additionally, Ensembler can help users visually identify outliers and can perform data processing and analysis procedures via the *Processing* and *Analysis* menu options, respectively. Users can add functionality by editing the code that controls the elements of the menu bar or apply bespoke processes to a data set by typing a function name into the *custom* section of the *Processing* menu item.



(a)



(b)

Figure 2: The biomechZoo toolbox graphical user interfaces: (a) Director, a three-dimensional environment for motion data visualization with embedded graphing capabilities and (b) Ensembler, a data sorting and figure preparation suite. The “max” events for the *RightGroundReactionForce_x* channel display as red circles. Clicking on a given line highlights the trace (shown as a dashed blue line) and returns the associated file information at the top of the figure. Here, the selected trace is the outlier described in Section 3.3.

3. Sample data set

This section demonstrates how to process, analyse, and visualize data to answer typical research questions in the study of gait. These processes were tested on Mac (OSX 10.11.6, Matlab v2016a) and PC (Windows 7 Enterprise, Matlab 2014a, 2016b; Windows 10 Enterprise, Matlab 2015a, 2016a, 2016b) platforms.

3.1. Description of the sample data set

Gait data from twelve typically developing children (11.3 ± 3.1 years) performing straight and 90° turning gait tasks (dynamic trials) were extracted in *.c3d* format from the Oxford Gait Laboratory database. Static trials, in which subjects remained in the anatomical position, were also collected. Informed consent/assent for inclusion of data in further studies was obtained for all children. Positions in space of skin-mounted retro-reflective markers placed according to the Plug-in Gait (PiG) set [17] were collected at 100 Hz using a motion capture system (Vicon Motion Systems Ltd., Oxford, UK), while force plates (Advanced Mechanical Technology Inc., Watertown, USA) were used to record GRFs at 1000 Hz. Pre-processing of data within the Vicon Nexus software environment (v1.8.4, Vicon Motion Systems Ltd., Oxford, UK) used in this study were marker gap filling and filtering [30]. The PiG modeller was run to obtain processed GRFs, joint kinematics and kinetics for comparisons with biomechZoo outputs.

Previously, we showed how to process and analyse two PiG modeller outputs [11]. Here, we demonstrate how “raw” PiG marker data and GRFs can be used as inputs to generate joint kinematic and kinetic quantities via a standalone biomechZoo post-processing pipeline.

3.2. Research question

For demonstration purposes, we hypothesized that straight (Straight) and turning (Turn) differ in (1) maximum medio-lateral GRF (GRF_{ML}), (2) maximum stance phase hip adduction angle (Hip_{ADD}), (3) knee flexion moment at foot-off ($Knee_{FLX}$), and (4) maximum ankle power generation ($Ankle_{PWR}$), extracted from the *RightGroundReactionForce_x*, *RightHipAngle_y*, *RightKneeMoment_x*, and *RightAnklePower* channels, respectively.

3.3. Data processing and analysis

Data processing steps are also summarized in `samplestudy_process` (see biomechZoo-help repository). This script contains nine cells, each representing a step in the procedure. Here, the variable `fld` refers to the root folder path (string). The processing can be run in a single step or in a stepwise fashion by setting the variable `mode` to ‘auto’ (default) or ‘manual’, respectively. Outputs of each step are provided in the data set download.

Step 1: Conversion to the biomechZoo format

The *.c3d* files were converted to biomechZoo format via `c3d2zoo`. This function calls on biomechZoo’s *c3d* reader (`readc3d`). Alternatively, users may wish to rely on the *c3d* reading capabilities of the BTK toolkit [4] (separate download) accessible within biomechZoo through the `c3d2zooBTK` and `readc3dBTK` functions. Conversion speed of *.c3d* files was comparable across readers for this data set (13.4 vs 13.9 sec for the biomechZoo and BTK toolkit readers, respectively, on our test PC computer). Note, the `csv2zooVicon` function can be used to convert Vicon generated *.csv* files to the biomechZoo format; however, these files lack meta information required for force plate processing. Conversion of a *.csv* file is demonstrated in the example script `csv2zooVicon_example`.

Step 2: Cleaning the data

Channels not required for analysis were removed via `bmech_removechannel(fld, ch, 'keep')`. The third argument represents an action (“keep” or “remove”) to be applied to the channels *ch*. This process reduces the size of the zoo files folder from 31.4 to 8.5 MB, thereby increasing processing speed for the following steps.

Step 3: Processing force plate data

Although the files in this data set contain PiG processed GRFs (*RGroundReactionForce*, *LGroundReactionForce*), the raw force plate channels (*ForceFx1*, *ForceFy1*, ..., *ForceFz2*) were used to demonstrate that the toolbox could accurately reproduce these outputs using the function `bmech_processGRF`. This function contains multiple processing steps. First, the GRF channels were filtered by internally calling `filter_data(data, ch, filt)`. The third argument `filt` is a structured array defining the filter properties, a forth-order zero-lag 20 Hz low-pass Butterworth filter. Next, these data were down-sampled to 100 Hz and converted from N to N/kg via `resample_data` and `massnormalize_data`, respectively. The limb that made contact with the force plate was identified and used to replace the GRF channel suffixes (1 or 2) with prefixes (Right or Left) using `forceplate2limbside_data` and `fprename_data`, respectively. The former relies on a marker-based algorithm [31], implemented in `ZeniEventDetect`. Finally, the GRFs were expressed in the body frame using `grfref_data`. Centre of pressure data are also computed here in anticipation of joint kinetic computations (see `centrofpressure_data`).

Step 4: Partitioning the data

Trials in this data set are of indeterminate length, primarily based on the availability of marker data (occlusions can occur near the edges of the motion capture volume). In preparation for partitioning, right foot-strike (RFS) and foot-off (RFO) events were identified as the first and last non-zero frames of vertical GRF data (*RightGroundReactionForce*) by calling `bmech_addevent(fld, ch, evt, type, subfld)` where the second, third, and fourth arguments represent the channel to search, the name of the event to append to the channel's event branch ('RFS' or 'RFO'), and the type of event to compute ('FS_FP' or 'FO_FP'), respectively. The fifth argument excludes the files within subfolder 'Static' from the batch process since static trial do not have any gait events. These gait events were then used to partition data to a single right-limb stance phase using `bmech_partition(fld, 'RFS', 'RFO', subfld)`.

Step 5: Computing joint kinematics and kinetics

In preparation for kinematic analyses, virtual hip (HipJC), knee (KneeJC), and ankle (AnkleJC) joint centre positions were determined by `bmech_jointcentrePiG` based on regression equations [7] (hip) and the PiG "chord" function [29] (knee, ankle). Lower-limb joint kinematics were computed using two implementations of the Grood and Sunday approach [15]. First, the angles were computed using `bmech_kinematicsRvdB` which relies on the function `cardan` contained within the KineMat toolbox of Reinschmidt and van den Bogert [24] (included in the biomechZoo download). This function requires static and dynamic trial marker positions for at least three markers of the proximal and distal segments for each joint. The second method calls `bmech_kinematics` and follows the approach taken by the PiG modeller [17]. This function internally calls `makebones` to create virtual markers defining segment-embedded axes for the computation of joint kinematics and to display scaled bones in Director. These virtual marker are also created by the PiG modeller and therefore follow the same nomenclature (e.g. PELO, PELA, PELL, PELP define the origin as well as the anterior, lateral, and posterior axes for the pelvis segment, respectively).

Lower limb joint kinetics were computed using `bmech_kinetics` which implements a standard inverse dynamics analysis approach [28] to compute net internal joint forces and moments of the hip, knee, and ankle. Net internal joint power is computed according to Kwon (eq. 7 in [19]).

Step 6: Adding events

Events required to test our hypotheses were then computed. First, `bmech_explode(fld)` was called to split the $m \times 3$ channels into three $m \times 1$ channels. Next, `bmech_addevent(fld, 'RightGroundReactionForce_x', 'max', 'max')` identified GRF_{ML} by searching for the maximum value in the `line` data for the *RightGroundReactionForce_x* channel (maximum lateral force) and appended the event *max* to the *event* branch. The second argument refers to the name of the new event in the *event* branch (string), while the third argument points to a specific case (min) within the function. Similarly, *Hip_{ADD}* and *Ankle_{PWR}* were obtained from the maximum value of the *RightHipAngle_y* and *RightAnklePower* channel, respectively. An event was not added for *Knee_{FLX}* because this event coincides with the RFO event ("global" event). The other events are considered "local".

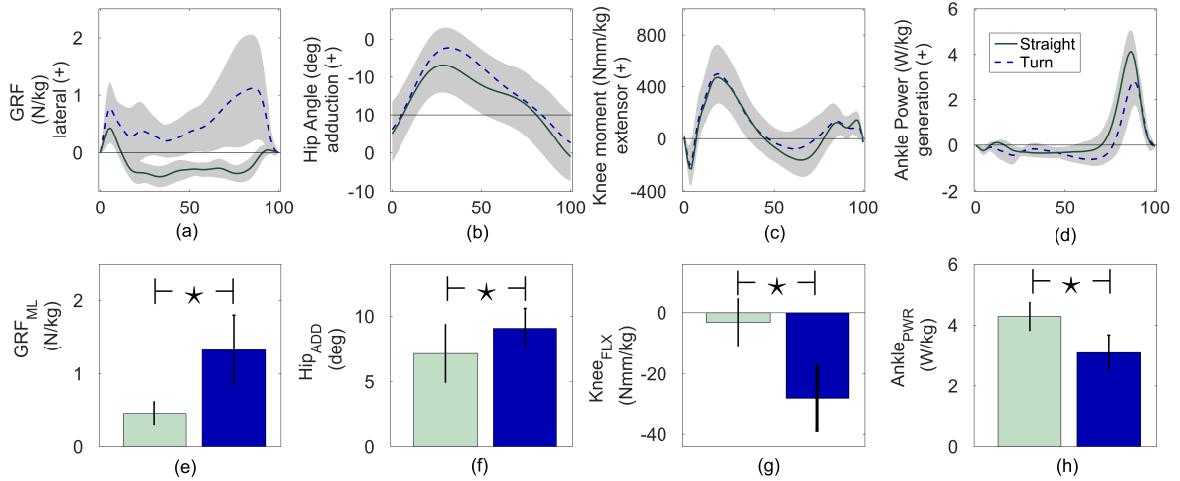


Figure 3: Time series graphs of (a) *RightGroundReactionForce_x*, (b) *RightHipAngle_y*, (c) *RightKneeMoment_x*, and (d) *RightAnklePower* (0–100% of the stance phase) as well as bar graphs for the (e) maximum lateral ground reaction force (*GRF_{ML}*), (f) hip adduction angle (*Hip_{ADD}*), (g) knee flexor moment (*Knee_{FLX}*), and (h) ankle power generation (*Ankle_{PWR}*) across Straight and Turn conditions. Time-series graphs show mean and standard deviation band while bar graphs display mean and 95% confidence interval. Significant differences across conditions denoted by ★.

Step 7: Normalizing the data

This final processing step normalized all channels of every file to 101 frames (0–100% of the right limb stance phase) via `bmech_normalize(fld, 'all', 100, method)`. Normalization relies on Matlab's `interp1` to linearly interpolate data (default setting of `method`); however, other methods could be implemented (e.g. cubic spline).

Step 8: Data visualization

Ensembler identified that the polarity is reversed for the *RightGroundReactionForce_x* channel of file HC002D25.zoo (see Fig. 2 b). Director revealed that this trial was performed in the opposite direction to the others, explaining this polarity change. For the purposes of this demonstration, the file was tagged as an outlier. Outliers can be removed using Ensembler. Deleting a single channel for a given file rewrites all *line* data to 999 and *event* data to [1, 999, 0]. This coding is used to ignore a single channel entry during statistical analysis. Instructions to reproduce the processes in this step using Ensembler and Director are found in Appendix B and Appendix C, respectively.

Step 9: Statistical analysis

Two statistical analysis approaches are possible in biomechZoo. For this simple study, a script was written to extract event data to the workspace (requires Matlab's `ttest` function, see `extractevents`). Alternatively, `eventval` can be used to export data to a spreadsheet for analysis by third-party software. This approach is recommended for data sets requiring an analysis of variance. The code snippet to create the event spreadsheet follows:

```

fld = uigetfolder('select ''7-normalize'''); % folder to search
levts = {'max'}; % local events
gevts = {'RFO'}; % global events
ch = {'RightGroundReactionForce_x', 'RightHipAngle_y', ... % channels to search
       'RightKneeMoment_x', 'RightAnklePower'};
dim1 = {'Straight', 'Turn'}; % condition dimension
dim2 = {'HC002D', 'HC030A', 'HC031A', 'HC032A', 'HC033A', 'HC036A', ... % subject dimension
        'HC038A', 'HC039A', 'HC040A', 'HC044A', 'HC050A', 'HC055A'};
eventval('fld', fld, 'dim1', dim1, 'dim2', dim2, 'localevts', levts, ... % function call
        'globalevts', gevts, 'ch', ch)

```

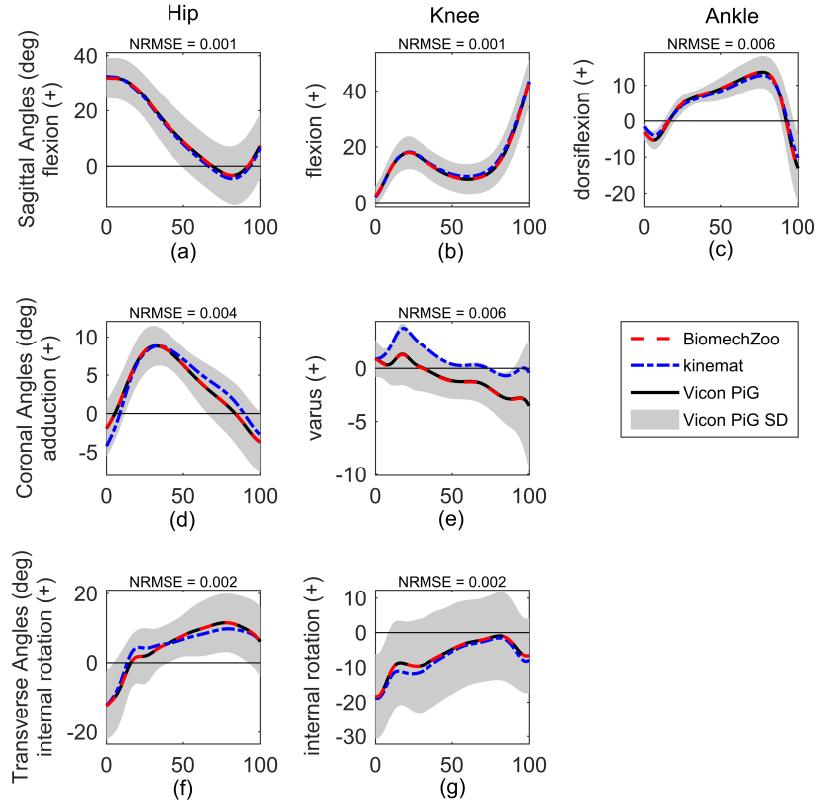


Figure 4: Hip, knee, and ankle joint kinematic estimations via biomechZoo, kinemat, and Plug-in Gait (PiG) in the sagittal (a–c), coronal (d and e), and transverse (f and g) planes for the right limb during the Turn condition (normalized to 100% of the stance phase). Standard deviation (SD) for the PiG model also shown. Ankle kinematics only shown in the sagittal plane given the limitations of the single-segment PiG model (c.f. [6]).

The spreadsheet (see `eventval.xls`) comprises sheets for each channel *ch* which contain event data for each file. An additional *info* sheet contains processing information.

Paired t-tests in SPSS (v23, IBM Corp., Armonk, USA) revealed that all outcome measures were significantly different across conditions (Fig. 3). For GRF_{ML} , the Turn condition revealed increased lateral force compared to Straight ($p = 0.008$, $df = 9$). For Hip_{ADD} , the Turn condition led to larger adduction angles compared to the Straight condition ($p = 0.007$, $df = 11$). $Knee_{FLX}$ and $Ankle_{PWR}$ were decreased during the Turn condition, compared to Straight ($p = 0.001$ and $p = 0.002$, respectively, $df = 11$).

No correction for multiple comparisons or verification of parametric assumptions were performed for simplicity. Inspection of the indices of GRF_{ML} (*exd*) reveals that the event lies in early stance for the Straight condition and in late stance for the Turn condition (see Fig. 3 a). In a real study, it might be important to limit analysis to a specific zone to ensure that meaningful events are being analyzed. The `eventval.xls` spreadsheet and statistical test outputs are available in the help repository (`~\biomechZoo-help\sample_study\Statistics\`)

4. Assessment of biomechZoo computations

The biomechZoo processed GRFs, joint kinematic, and joint kinetic estimations were compared against those output by the Vicon (Vicon Motion Systems Ltd., Oxford, UK) PiG model. Similarity was assessed using signal range normalized root mean squared differences (NRMSE) (see `nrmse`). Results were compared across conditions for the right side only. Turn condition joint kinematic and kinetic graphs are presented in the manuscript. Supplemental figures referenced herein are found in the help repository (`~\biomechZoo-help\sample_study\Figures\Manuscript\`)

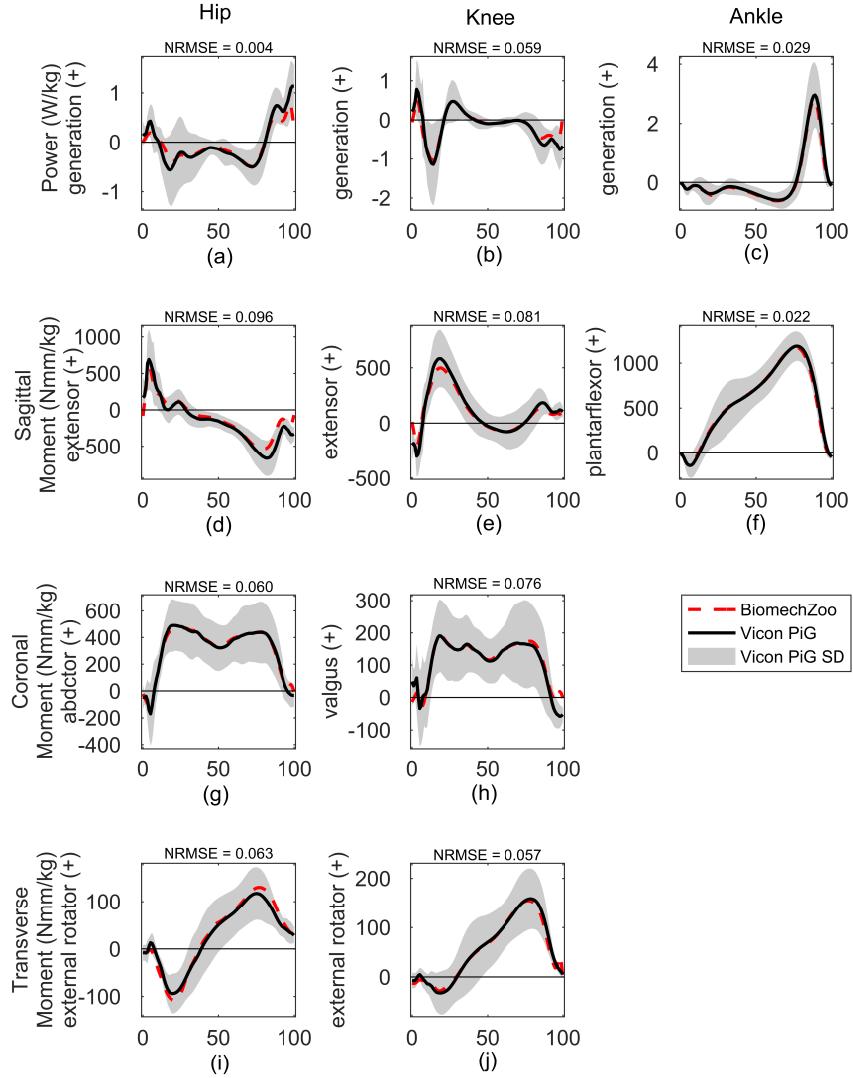


Figure 5: Hip, knee, and ankle joint power (a–c) and moment (d–j) estimations via biomechZoo and Plug-in Gait (PiG) for the right limb during the Turn condition (normalized to 100% of the stance phase). Standard deviation (SD) for the PiG model also shown. Ankle joint moments only shown in the sagittal plane given the limitations of the single-segment PiG model (c.f. [6]).

For the Straight condition, the NRMSE for the medio-lateral, antero-posterior, and vertical GRF components were 0.030, 0.019, and 0.021, respectively (Supplemental Fig.1). For the joint kinematics, differences between biomechZoo and Vicon PiG were small with $NRMSE \leq 0.012$ (Supplemental Fig.2). For the joint power the $NRMSE$ were less than or equal to 0.121, while for the joint moments the $NRMSE$ were less than or equal to 0.084 with the transverse hip moments showing the largest difference (Supplemental Fig.3).

For the Turn condition, the NRMSE for the medio-lateral, antero-posterior, and vertical GRFs were 0.044, 0.024, and 0.024, respectively (Supplemental Fig.4). The NRMSE for the joint kinematics were all less than 0.006 (Fig. 4). Joint kinematics from the KineMat toolbox are graphed, but are not compared. For the joint moments and power, NRMSE were less than 0.1 and 0.06, respectively (Fig. 5).

5. Discussion

5.1. Summary

The biomechZoo toolbox represents the work of several years and multiple contributors to provide a generic yet flexible interface to examine time series data sets typical in the movement sciences. This manuscript demonstrates the features of biomechZoo via a sample data set. We show that existing code from the KineMat toolbox [24] can be integrated into our toolbox and that the PiG joint kinematics and kinetics could be reproduced at a high level of accuracy. For typical gait data, at least, biomechZoo can be used as a stand-alone post-processing tool. Moreover, the sample data can be used in a teaching context or by researchers interested in gaining further familiarity with biomechZoo features before implementing the system in their own work.

5.2. Comparison to state of the art

Many data analysis and processing options are available to the movement scientist. We will compare a few alternatives to biomechZoo for motion capture studies. Starting with the commercial options, Vicon Nexus, and other motion capture software, are invaluable tools for data collection and basic processing tasks such as marker identification. Nonetheless, exporting data to Matlab for further post-processing is preferred by many users, despite native integration of Matlab code in newer Nexus versions. We have shown that trajectory filtering and computation of joint kinematics and kinetics according to the PiG model can be reproduced at a high level of accuracy by biomechZoo. Visual3D (C-Motion, Inc., Germantown, USA) is a three-dimensional analysis software that allows users to process data within an intuitive interface. These features may be attractive (especially rich graphic capabilities that outperform Director's); however, processes remain “black box” and the interface makes specialized coding (e.g. work-flows requiring logic statements) difficult. Furthermore, its high cost (currently 12,995 USD) represents a financial barrier to adoption. Open-source projects, such as BTK [4] (.c3d reading, writing, and visualization) and OpenSim [9] (musculo-skeletal modelling) focus their efforts on specific research areas and are successful platforms. Moreover, the open-source Python (Python Software Foundation, python.org) project pyCGM (github.com/cadop/pyCGM) can compute full-body PiG joint kinematics, but is still in a pilot stage. The Biomechanics Toolbar [27] is an Excel (Microsoft Corp., Redmond, USA) add-on that can help students understand how to process motion data in a familiar environment. biomechZoo does not attempt to reproduce or replace these efforts, but rather can be used in conjunction with, or in addition to, these and many other software packages as a general framework for analysis.

5.3. Past projects

The biomechZoo toolbox has been in use in our laboratories since 2006. Here, we list some past projects to provide the scope of the toolbox’s capabilities. BiomechZoo was used to analyze force profiles of skating manoeuvres captured via a portable logging unit [26] as well as three-dimensional motion capture data of ice hockey shooting [21, 23] and skating [25] tasks, walking [12, 13, 18], and running [14]. Recent work [10] has relied on BTK [4] and the Lichtwark toolbox [20] to generate files for musculo-skeletal modelling in OpenSim (v3.1, Stanford, USA).

5.4. Limitations

The code for biomechZoo is free and open-source (editable m-files); however, the toolbox itself must be run within Matlab which hides its own source code from the user. Versions of the toolbox running in Python (open-source) are planned for the future. Director displays “bone” objects to form skeletons (see Fig. 2 a) for PiG marker data only. Other marker configurations will display as multi-colored spheres. Moreover, only AMTI force plates will display GRF correctly. Updated are planned to allow generic marker sets and other force plates to display similarly.

5.5. Dependencies

BiomechZoo takes advantage of the Apache POI java library via de Zegher's `xlwrite` [8] in order to export event data to spreadsheets on Mac platforms and systems where Excel is not installed. Exporting figures to vector graphic format in Ensembl is supported by `export_fig` of Woodford and Altman [1]. BiomechZoo requires Matlab's Signal Processing (`bmech_filter`) and Statistics (e.g. `ttest,range, nansum`) toolboxes.

5.6. Conclusion

We have demonstrated that biomechZoo can process, analyse, and visualize gait data, including the accurate reproduction of kinematic and kinetic outputs from a commercial biomechanical model. The open-source biomechZoo project welcomes contributions from the movement science community at large; the Git Hub repositories can be accessed from www.biomechzoo.com and allows users to suggest updates, report bugs, and fork the repositories for independent development efforts. Moreover, biomechZoo has been designed with the learning needs of novice programmers in mind. These features distinguish biomechZoo from other offerings. Finally, It is hoped that biomechZoo can also help standardize and advance research in biomechanics and the movement sciences.

6. Acknowledgements

PCD was supported by the McGill University Department of Kinesiology Graduate Studies Fellowship (2007), the Natural Sciences and Engineering Research Council of Canada (NSERC) Doctoral Scholarship (2011-2014), the Fonds de Recherche du Québec-Santé (FRQS) doctoral (2014-2015) and post-doctoral (2016) awards, while DJP was supported by NSERC grants (CRDJP 311213-04, 36358607; 453725-13), during the development of biomechZoo. These agencies had no role in the design, collection, analysis, and interpretation of data; in the writing of the manuscript; nor in the decision to submit the manuscript for publication. All authors acknowledge Andrew Lewis and the Oxford Gait Laboratory staff for help with the collection and initial processing of the sample data set as well as Jacob Banks, Jesse Jacobs, and Philippe Renaud, for review of manuscript drafts. PCD thanks Mathew Schwartz from the pyCGM project for help with computation of ankle offset angles.

7. Conflict of interest statement

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

8. Availability and requirements

Project name: biomechZoo

Project site: <http://www.biomechzoo.com>

Operating systems: Mac and Windows

Programming language: MATLAB v2014b or newer

Other requirements: Matlab Signal Processing and Statistics toolboxes.

License: Apache Version 2.0

Restrictions to use by non-academics: None

Referencing: The present paper should be referenced if the toolbox is used in the preparation of a manuscript.

9. References

- [1] Altman Y.M. and Woodford O.J. `export_fig`. 2014, <https://github.com/altmany>. Accessed Jan 29th, 2015.
- [2] Baker R. The history of gait analysis before the advent of modern computers. *Gait Posture*, 26(3):331–342, 2007.
- [3] Barre A. Biomechanical toolkit: A step towards independency between motion capture data and biomechanical analysis. In *Proceedings of the International Conference on Biomechanics in Sport*, 2015.
- [4] Barre A. and Armand S. Biomechanical toolkit: Open-source framework to visualize and process biomechanical data. *Comput Meth Prog Bio*, 114(1):80–7, 2014.
- [5] Burger B. and Toivainen P. Mocap toolbox – a matlab toolbox for computational analysis of movement data. In *Proceedings of the 10th Sound and Music Computing Conference*, 2013.
- [6] Carson M.C., Harrington M.E., Thompson N., O'connor J.J., and Theologis T. Kinematic analysis of a multi-segment foot model for research and clinical applications: A repeatability analysis. *J Biomech*, 34(10):1299–307, 2001.
- [7] Davis R.B., Ounpuu S., Tyburski D., and Gage J.R. A gait analysis data-collection and reduction technique. *Hum Mov Sci*, 10(5):575–87, 1991.
- [8] De Zegher A. `xlwrite`. 2012, <http://www.mathworks.com/matlabcentral/fileexchange>. Accessed Jan 8th, 2016.
- [9] Delp S.L., Anderson F.C., Arnold A.S., Loan P., Habib A., John C.T., Guendelman E., and Thelen D.G. Opensim: Open-source software to create and analyze dynamic simulations of movement. *IEEE Trans Biomed Eng*, 54(11):1940–50, 2007.
- [10] Dixon P.C., Jansen K., Jonkers I., Stebbins J., Theologis T., and Zavatsky A.B. Muscle contributions to centre of mass acceleration during turning gait in typically developing children: A simulation study. *J Biomech*, 48(16):4238–4245, 2015.
- [11] Dixon P.C., Loh J.J., Michaud-Paquette Y., and Pearsall D.J. The Zoosystem: An open-source movement analysis Matlab toolbox. In *Proceedings of the 23rd European Society for Movement Analysis in Adults and Children conference*, 2014.
- [12] Dixon P.C., Stebbins J., Theologis T., and Zavatsky A.B. Spatio-temporal parameters and lower-limb kinematics of turning gait in typically developing children. *Gait Posture*, 38(4):870–5, 2013.
- [13] Dixon P.C., Stebbins J., Theologis T., and Zavatsky A.B. Ground reaction forces and lower-limb joint kinetics of turning gait in typically developing children. *J Biomech*, 47(15):3726–33, 2014.
- [14] Dixon P.C., Tisseyre M., Damavandi M., and Pearsall D.J. Inter-segment foot kinematics during cross-slope running. *Gait Posture*, 33(4):640–644, 2011.
- [15] Grood E.S. and Suntay W.J. A joint coordinate system for the clinical description of three-dimensional motions: Application to the knee. *J Biomed Eng*, 105(2):136–44, 1983.
- [16] James D.A. and Wixted A. ADAT: A Matlab toolbox for handling time series athlete performance data. In *Proceedings of the 5th Asia-Pacific Congress on Sports Technology*, 2011.
- [17] Kadaba M.P., Ramakrishnan H.K., and Wootten M.E. Measurement of lower extremity kinematics during level walking. *J Orthop Res*, 8(3):383–92, 1990.
- [18] Kothari A., Dixon P.C., Stebbins J., Zavatsky A.B., and Theologis T. The relationship between quality of life and foot function in children with flexible flatfeet. *Gait Posture*, 41(3):786–790, 2015.
- [19] Kwon Y.H. Joint energetics. 1998, <http://www.kwon3d.com>. Accessed, Jan 12th, 2015.
- [20] Lichtwark G. Matlab OpenSim PipelineTools v2. 2011, <http://simtk.org>. Accessed Jan 3rd, 2014.
- [21] Lomond K.V., Turcotte R.A., and Pearsall D.J. Three-dimensional analysis of blade contact in an ice hockey slap shot, in relation to player skill. *Sports Eng*, 10(2):87–100, 2007.
- [22] Mantoan A., Pizzolato C., Sartori M., Sawacha Z., Cobelli C., and Reggiani M. MOtoNMS: A MATLAB toolbox to process motion data for neuromusculoskeletal modeling and simulation. *Source Code Biol Med*, 10:12, 2015.
- [23] Michaud-Paquette Y., Magee P., Pearsall D.J., and Turcotte R. Whole-body predictors of wrist shot accuracy in ice hockey: a kinematic analysis. *Sports Biomech*, 10(1):12–21, 2011.
- [24] Reinschmidt C. and Van Den Bogert T. KineMat: A MATLAB toolbox for three-dimensional kinematic analyses. 1997, <http://isbweb.org/software/movanal/kinemat>. Accessed Jan 29th, 2015.
- [25] Robert-Lachaine X., Turcotte R., Dixon P.C., and Pearsall D.J. Impact of hockey skate design on ankle motion and force production. *Sports Eng*, 15(4):197–206, 2012.
- [26] Stidwill T.J., Turcotte R., Dixon P.C., and Pearsall D.J. Force transducer system for measurement of ice hockey skating force. *Sports Eng*, 12(12):63–68, 2010.
- [27] Vanrenterghem J. Biomechanics toolbar. 2016, <http://www.biomechanicstoolbar.org>. Accessed Feb 15th, 2016.
- [28] Vaughan C.L., Davis B.L., and O'connor J.C. *Dynamics of human gait*. Kiboho Publishers, Cape Town, South Africa, 2nd edition, 1999.
- [29] Vicon Motion Systems Ltd . Plug-In Gait manual version 1.
- [30] Woltring H.J. A FORTRAN package for generalized, cross-validatory spline smoothing and differentiation. *Adv Eng Softw UK*, 8(2):104–13, 1986.
- [31] Zeni J.A., Jr., Richards J.G., and Higginson J.S. Two simple methods for determining gait events during treadmill and overground walking using kinematic data. *Gait Posture*, 27(4):710–4, 2008.

Appendix A. Batch processing within biomechZoo

The following code demonstrates how batch processing procedures are implemented within biomechZoo. The function runs a sub function `process_data` that performs some operation directly on the structured array `data`. An additional function `process_line` might act on the matrix or column vector contained within a given channel.

```
function bmech_process(fld,ch)

fl = engine('path',fld,'extension','zoo'); % returns all zoo files in folder fld as fl

for i = 1:length(fl) % runs the process for each zoo file in fl
    data = zload(fl{i}); % loads the ith zoo file
    batchdisp(fl{i}, 'some process'); % displays process information
    data = process_data(data,ch); % performs some process on 'ch' within 'data'
    zsave(fl{i},data); % saves process and records process information
end
```

Appendix B. Ensembler instructions

Generating time-series graphs for `RightGroundReactionForce_x`, `RightHipAngle_y`, `RightKneeMoment_x`, and `RightAnklePower`:

1. Type `ensembler` in the Matlab command window. A preliminary settings window opens.
2. Change the `name` field to *Straight Turn*, rows to *1*, and columns to *4* and click *OK*. Two generic Ensembler figures are created, each with four empty axes. The main figure (Turn) contains a menu bar at the top. To resize figure windows and axes, select *File, restart* from the menu bar and edit the sizing options.
3. Select *Axes, re-tag* and choose any fully processed zoo file in the final normalized folder. A channel selection window opens. Associate `RightGroundReactionForce_x`, `RightHipAngle_y`, `RightKneeMoment_x`, and `RightAnklePower` to the generic *1 1*, *1 2*, *1 3*, and *1 4* axes, respectively, by clicking on the arrows. Each channel should now be listed on the right of its corresponding axis number in the center of the selection window. Select *OK* to update the figure axis titles with the selected channel names.
4. In the menu bar, choose *File, load data*, and select the folder containing normalized data. Ensembler populates the axes with the corresponding channel data for each condition. Trials from the Straight and Turn conditions will be sorted into the Straight and Turn figures, respectively.
5. Clear all events by selecting *Events, clear all events* from the menu bar.
6. One line for `RightGroundReactionForce_x` for each condition appears separate from the others. Left click on the trace to identify the trials (`HC002D25.zoo` and `HC036A10.zoo` for the Straight and Turn conditions, respectively.). Press *delete* on the keyboard and select *Delete Channel* to erase the traces. The line and event values are replaced with the value 999 (see Section 3.3 for justification).
7. Select *Ensembler, Ensemble (SD)* then *Ensembler, combine data* to graph the average of both conditions in a single figure. Line styles and colors can be updated via the *Line* menu to differentiate the conditions.
8. Add a legend by selecting *Insert, legend* and associating each condition with a number indicating the order in which the legend entries are displayed (vertically).
9. Finalize graphs by exploring the menu bar options or by selecting *Edit, property editor on*.
10. Save the figure by selecting *File, save fig* or export to vector graphics format (.pdf) by selecting *File, export*.

Generating a bar graph for the maximum hip adduction angle (`HipADD`):

1. Repeat steps 1–4 from the time-series instructions, modified to load the `RightHipAngle_y` channel data only.
2. Clear the event `NRMSE` by selecting *Events* then *clear events by type*.

3. Select *Ensembler, ensemble (CI)* then *Ensembler, combine data*.
4. Select *Bar Graph* from the menu, then *bar graph* to display discrete event data as a bar graph.
5. Finalize, and save graphs using steps 8–10 from the time-series graphing instructions.

Appendix C. Director instructions

Visualizing three-dimensional motion capture data:

1. Type *director* in the command window. A blank three-dimensional canvas opens.
2. Select *Load File* and choose a file from the “Step 1” or “raw c3d files” folder.
3. Choose *lower-limbs* to load a lower-body skeleton for the selected trial.
4. Choose a few markers to display in the animation and select *OK*
5. Choose a channel to plot by clicking on the channel list in the upper left corner.
6. Press *Play* to start animation (*Stop* to stop).
7. Repeat this process for a number of trials (including HC002D25.zoo) to reveal the reason for removing the specific trial from the analysis (see Section 3.3).