

## SOFTWARE

# ArborSim: Articulated, branching, OpenSim routing for constructing models of multi-jointed appendages with complex muscle-tendon architecture

Xun Fu<sup>1</sup>, Jack Withers<sup>2</sup>, Juri A. Miyamae<sup>1</sup>, Talia Y. Moore<sup>1,3,4\*</sup>

**1** Robotics, University of Michigan, Ann Arbor, Michigan, United States of America, **2** Computer Science, University of Michigan, Ann Arbor, Michigan, United States of America, **3** Mechanical Engineering, University of Michigan, Ann Arbor, Michigan, United States of America, **4** Ecology and Evolutionary Biology, Museum of Zoology, University of Michigan, Ann Arbor, Michigan, United States of America

\* [taliaym@umich.edu](mailto:taliaym@umich.edu)



## OPEN ACCESS

**Citation:** Fu X, Withers J, Miyamae JA, Moore TY (2024) *ArborSim*: Articulated, branching, OpenSim routing for constructing models of multi-jointed appendages with complex muscle-tendon architecture. *PLoS Comput Biol* 20(7): e1012243. <https://doi.org/10.1371/journal.pcbi.1012243>

**Editor:** Kaushik Jayaram, University of Colorado Boulder, UNITED STATES OF AMERICA

**Received:** January 15, 2024

**Accepted:** June 10, 2024

**Published:** July 5, 2024

**Peer Review History:** PLOS recognizes the benefits of transparency in the peer review process; therefore, we enable the publication of all of the content of peer review and author responses alongside final, published articles. The editorial history of this article is available here: <https://doi.org/10.1371/journal.pcbi.1012243>

**Copyright:** © 2024 Fu et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

**Data Availability Statement:** The source code of ArborSim is freely and anonymously accessible on GitHub, under the permissive Apache License 2.0,

## Abstract

Computational models of musculoskeletal systems are essential tools for understanding how muscles, tendons, bones, and actuation signals generate motion. In particular, the OpenSim family of models has facilitated a wide range of studies on diverse human motions, clinical studies of gait, and even non-human locomotion. However, biological structures with many joints, such as fingers, necks, tails, and spines, have been a long-standing challenge to the OpenSim modeling community, especially because these structures comprise numerous bones and are frequently actuated by extrinsic muscles that span multiple joints—often more than three—and act through a complex network of branching tendons. Existing model building software, typically optimized for limb structures, makes it difficult to build OpenSim models that accurately reflect these intricacies. Here, we introduce *ArborSim*, customized software that efficiently creates musculoskeletal models of highly jointed structures and can build branched muscle-tendon architectures. We used *ArborSim* to construct toy models of articulated structures to determine which morphological features make a structure most sensitive to branching. By comparing the joint kinematics of models constructed with branched and parallel muscle-tendon units, we found that among various parameters—the number of tendon branches, the number of joints between branches, and the ratio of muscle fiber length to muscle tendon unit length—the number of tendon branches and the number of joints between branches are most sensitive to branching modeling method. Notably, the differences between these models showed no predictable pattern with increased complexity. As the proportion of muscle increased, the kinematic differences between branched and parallel models units also increased. Our findings suggest that stress and strain interactions between distal tendon branches and proximal tendon and muscle greatly affect the overall kinematics of a musculoskeletal system. By incorporating complex muscle-tendon branching into OpenSim models using *ArborSim*, we can gain deeper insight into the interactions between the axial and

at <https://github.com/EMBiRLab/ArborSim>. The GitHub repository is called “ArborSim.”

**Funding:** The author(s) received no specific funding for this work.

**Competing interests:** The authors have declared that no competing interests exist.

appendicular skeleton, model the evolution and function of diverse animal tails, and understand the mechanics of more complex motions and tasks.

## Author summary

OpenSim models of musculoskeletal systems are valuable tools for understanding the mechanisms underlying biological movement. However, highly jointed structures, like necks and tails, have been difficult to model, likely due to the presence of numerous bones, and their complex network of branched muscle-tendon units that span many joints. We introduce *ArborSim*, a modeling tool that facilitates construction of branched muscle-tendon units. Our comparisons of equivalent branched and unbranched models suggest that the modeling approach greatly affects the model predictions, which thereby indicates the importance of accurately modeling branching in multi-jointed structures. With *ArborSim*, we provide a unified framework that not only streamlines the creation of the complex musculoskeletal models of these structures but also paves the way for more in-depth investigations into the biomechanical behavior of branched muscle-tendon networks. This encourages more detailed study of the material properties and mechanics of complex muscle-tendon branching networks and highly jointed structures.

## Introduction

Animal movement results from intricate interactions among elements of musculoskeletal systems [1–3]. Investigating these interactions using experimental *in vivo* data alone can be challenging and often impractical due to measurement constraints and invasive methods. Computational musculoskeletal modeling and simulations offer a complementary approach. By integrating models of system elements, dynamic simulations utilizing the integrated model are generated, providing estimates of hard-to-measure variables [4, 5]. These simulations enable researchers to reproduce observed experimental results, comprehend system coordination, design innovative experiments, and predict treatment outcomes, making them valuable tools in biomechanics studies [4, 5].

In the last few decades, substantial advancements have been made in detailed musculoskeletal modeling of the human body [6–12], which has led to the discovery of fundamental human movement principles, advanced injury diagnosis, and informed prosthetic and robotic design [13–23]. OpenSim [24, 25], a widely-used open-source software for musculoskeletal modeling and simulation, has been a key contributor to these advancements, fostering the sharing and dissemination of musculoskeletal models among the biomechanics community through the SimTK database. Recently, there has been increasing interest in developing musculoskeletal models of other animals [26–31], such as ostrich [27], chimpanzee [26], and mouse [28, 31], with the aim of enriching the understanding of a wide range of forms and behaviors exhibited by other species, and inspiring the development of bioinspired robotic technologies [32–34].

In every vertebrate animal, highly jointed systems, such as phalanges and the axial skeleton (spine, including necks and tails), facilitate greater curvature and are crucial to performance and survival. For example, cats adjust the stiffness of their flexible backs to absorb kinetic energy upon landing, allowing their limbs to attenuate landing impulses more safely and effectively [35]. Certain monkeys use their prehensile tails as a fifth limb, providing essential grasp and balance while navigating through trees, facilitating climbing in their arboreal environment

[36]. As an extreme example, snakes are remarkable, highly jointed systems by themselves. They possess highly flexible and muscular bodies that allow for a range of movement adaptations, enabling them to effectively navigate various terrains and tight spaces [37–39].

Nevertheless, in most existing musculoskeletal models, highly jointed systems are often simplified or excluded to facilitate computationally challenging analyses [30, 40, 41]. For instance, a musculoskeletal model of the dog does not consider the motion of the spine as a series of individual vertebrae, but rather simplifies the spine into discrete regions such as thorax and tail which are granted three degrees of freedom (DOFs) each [30]. A mouse model simulating a trotting gait focuses only on the hindlimb, disregarding the spine and tail entirely [28].

The practice of simplifying highly jointed systems, while helpful in reducing computational complexity, may unintentionally mask their complete biomechanical importance. However, even within these limitations, recent studies using simplified, but articulated models of highly jointed systems have identified a significant role for them in animal locomotion. For example, using a musculoskeletal model of a dinosaur that includes a simplified two-link tail, researchers have found that the tail performs a critical role in enhancing locomotor efficiency [42]. Moreover, by incorporating tail compliance properties through a five-link tail model, researchers have used the natural frequency of a tail to estimate the preferred walking speed of a dinosaur [43]. These studies exemplify the unexpected and substantial contributions of highly jointed systems that await discovery through modeling, underscoring the potential for gaining even more profound insights from more detailed and comprehensive models of these systems.

Despite the development of musculoskeletal modeling software over the past few decades, constructing detailed models of highly jointed systems remains both time-consuming and challenging. The primary challenge results from the presence of numerous bones and intricate muscle and tendon architectures, in which muscles and tendons can span multiple joints; the very morphology that allows highly jointed structures to form smooth curves. To compound upon this complexity, tail tendons frequently show a branching morphology that manifests as multiple tendons arising from a shared muscle-tendon transition zone, tendons splitting distally into multiple branches before reaching their insertion sites on multiple vertebrae, and/or tendons forking near their insertion site to blend into the muscle-tendon transition zone of a small local intrinsic muscle slip [44–46]. This branching phenomenon greatly challenges musculoskeletal modeling processes that were developed for studying limb mechanics.

When designing software to digitally represent morphology, simplifying assumptions can inadvertently become restrictive constraints. This is evident in the modeling of complex muscle-tendon branching architectures. Due to the limitation of representing only one tendon per muscle, researchers typically adopt an approach utilizing multiple muscle-tendon units (MTUs) in parallel, each representing a unique insertion and a portion of the muscle associated with that insertion. For example, the extensor digitorum communis (EDC) muscle in humans has a tendon that divides into four distinct branches, each of which inserts into a different finger [47]. In existing musculoskeletal models of the human hand [8, 48, 49], the EDC is represented as multiple independent compartments, each associated with a specific tendon that extends to a particular finger. This modeling simplifies the representation of the muscle and tendon architecture. However, it may not fully capture the physical interactions that occur between the muscle and tendons, increasing the likelihood that such models may provide limited, or even misleading predictions.

Physical interactions between branched MTUs are likely to be particularly important in cases where the branches insert on different bones, the MTU spans many joints, or the MTUs form a complex branching network with more than two insertions. For example, consider an MTU that originates on one bone and then inserts on two different bones. If external forces bend the more distal joint, that branch of the tendon experiences stress and strain, which

increases stress on the tendon and muscle proximal to the branching site, but reduces stress on the parallel distal branch. This type of complex stress field has been essential to predicting tears in the human supraspinatus and infraspinatus [50, 51], but there are currently no examples of OpenSim models that reflect these empirical data. Here, we introduce a new modeling framework to explicitly model the mechanics of branched MTUs and use simplified case studies to estimate the effect of modeling strategy on motion predictions.

Current tools designed for musculoskeletal modeling are not well suited for handling the aforementioned modeling complexities for highly jointed systems. NMSBuilder, an open-source software to create subject-specific musculoskeletal models, was established to facilitate the model building process for lower limbs [52]. The resulting models can be easily exported to OpenSim. While NMSBuilder improves musculoskeletal modeling efficiency, the modeling flow still requires manual addition of every entity and is, therefore, not well-suited for modeling highly articulated systems, particularly those featuring complex muscle-tendon branching architectures. A similar challenge exists with OpenSim Creator [53], a newly introduced software still in its alpha stage, designed for creating musculoskeletal models. Furthermore, NMSBuilder only allows the use of a subset of modeling components of OpenSim; any elements absent in NMSBuilder, such as constraints and controllers, must be added manually post hoc in OpenSim if required. Other modeling software options, such as SIMM [54, 55], Anybody [56], and Visual 3D (Visual 3D Professional, C-motion, Germantown, MD, USA), are proprietary, expensive (\$4995–7950 per year, with academic discounts applied), and tailored for limb modeling. Because none of these software packages provide full access to source code, it is difficult to expand their functionality to accommodate the complex tendon branching phenomena. Here, we introduce *ArborSim* (Articulated, Branching, OpenSim Routing), an extensible open-source tool integrated with the application programming interface (API) of OpenSim for constructing detailed musculoskeletal models of highly jointed systems, while accounting for complex muscle-tendon architectures. We intend the toolkit to enable biomechanics researchers to efficiently construct musculoskeletal models of highly jointed structures from CT scans and dissection data. We provide templates to store all essential data for modeling as Comma-Separated Values (CSV) files that are then used to automatically construct the model. This highly customizable framework enables users to easily set or update variables based on empirical data for any musculoskeletal parameter, even in highly complex systems.

In addition, we propose a novel approach to modeling complex muscle-tendon architectures that explicitly considers the interactions between each component. The approach is seamlessly integrated into the tool. By changing one line of code, users can decide to construct musculoskeletal models using either the conventional (parallel MTU) approach to modeling branching or our proposed method, both using the same data. We employ our tool to investigate the impact of using different methods to model MTU branching by constructing simplified, multi-jointed systems, i.e. “toy models.” Large differences between the kinematic outputs of simulations using the conventional branching modeling method and the proposed method highlight the need for more accurate models of complex branching phenomena.

## Design and implementation

### Overview of *ArborSim*

A fundamental representation of a musculoskeletal system in OpenSim involves rigid bodies, joints, and MTUs. The rigid bodies represent bones and embody associated inertial characteristics including mass, center of mass, and moment of inertia that incorporate the mass of the muscles and tendons that move in tandem with each bone. These rigid bodies are interconnected through joints defined by Joint Coordinate Systems (JCSs). MTUs, functioning as

actuators, are represented by lines of action along with a set of parameters that govern muscle contraction dynamics. These lines of action are determined by MTU paths, which consist of user-selected path points. Additionally, wrap objects can be introduced to shape the line of action of an MTU along parametric surfaces, commonly known as wrapping surfaces.

The development of our tool centers around the procedural creation of these fundamental entities. To enhance the software's structure, reusability, and ease of maintenance, the tool is built using a four-layer architecture (see Fig 1). Among the four layers, the first and second layers (Fig 1A and 1B) serve as input interfaces, allowing users to input customized data for constructing an articulated musculoskeletal model of a highly jointed system. In these two layers, the first requires data in a "global" format, whereas the second needs "local" format data (More details on these formats are provided in subsequent sections). Users have the flexibility to input data in either format. If "global" data is provided, the "**Transformer**" feature will convert it into "local" format, ready for model construction.

The components of the musculoskeletal model, derived from the inputs of the first two layers, are then modified via the implementation of the proposed branching modeling approach, named "**Brancher**". These modified components reside in the third layer (Fig 1C). During this modification process, branched muscles and tendons—initially represented as multiple independent MTUs according to the conventional branching modeling method—are altered to conform with the proposed branching modeling method. Subsequently, depending on the desired branching representation, the tool assembles the stored components either in the second or third layer via the functionality named "**Builder**", thereby generating and exporting the resulting musculoskeletal model that can be directly loaded into OpenSim. This resulting model constitutes the fourth layer (Fig 1D).

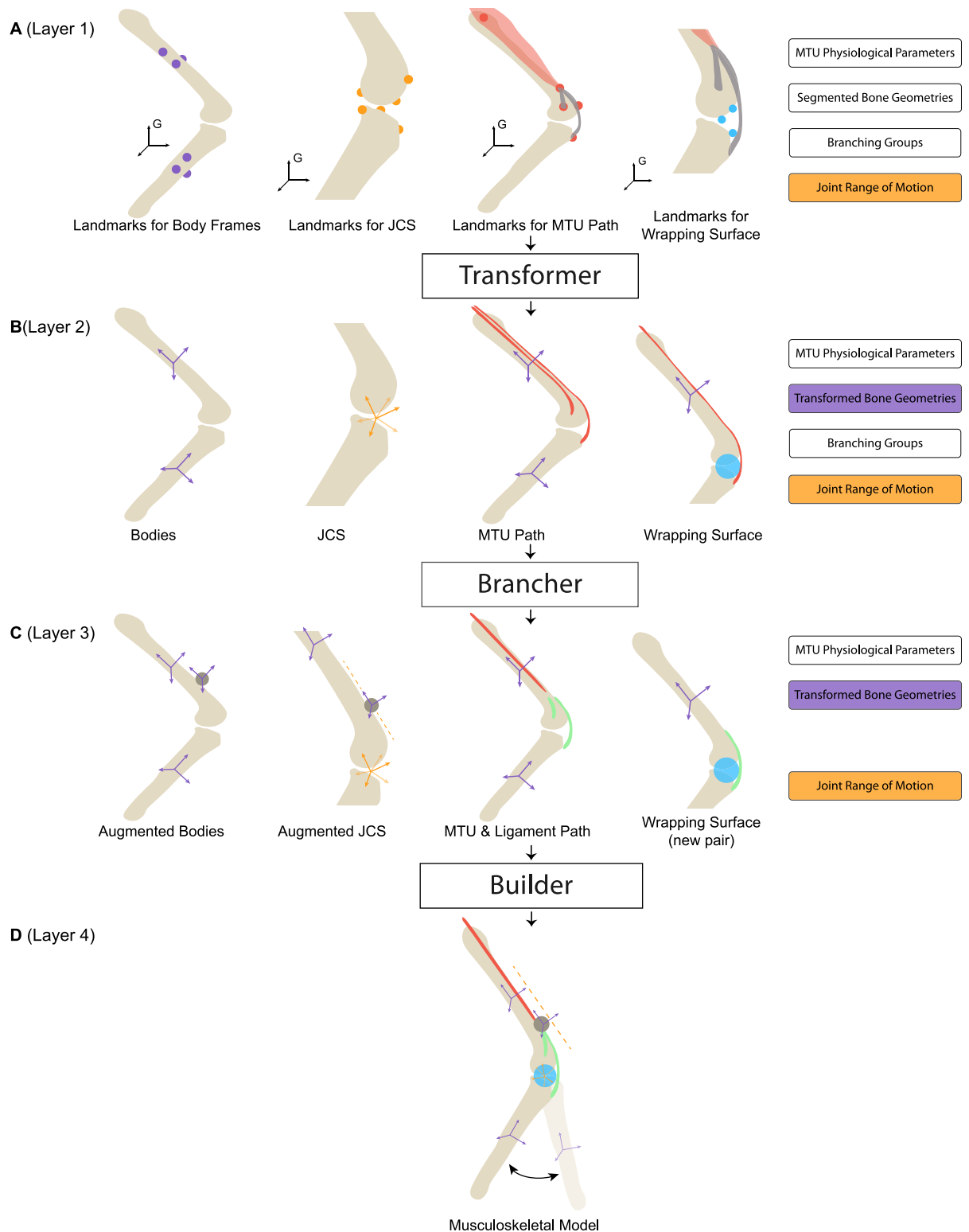
Each category of data within the first two layers is stored in a dedicated CSV file. The natural data structure and ubiquity of CSV files offer a clear and intuitive way to represent the data, and simplify data sharing and collaboration with researchers using diverse tools or platforms. Moreover, the tool can be easily extended, as one can create additional dedicated CSV files to accommodate extra components and functionalities for the tool.

## Layer 1

In the first layer, users provide the following categories of data:

- Coordinates of anatomical landmarks for establishing body frames.
- Coordinates of anatomical landmarks for defining JCSs.
- Coordinates of anatomical landmarks outlining paths for MTUs.
- Coordinates of anatomical landmarks for wrapping surfaces.
- Physiological parameters of MTUs.
- Segmented bone geometries.
- Branching groups.
- Joint range of motion.

Among the data mentioned above, the coordinates of anatomical landmarks are represented in a global reference frame. In the context of working with a CT scan or other biomedical imaging resources, these landmarks can be identified in segmentation and visualization software such as Mimics (Materialise NV, Leuven, Belgium), 3D Slicer [57], and ITK-SNAP [58]. In this context, the term "global" coordinates of these landmarks refers to the coordinates



**Fig 1. Overview of ArborSim.** (A) The input layer with global data, divided into eight categories. In the diagrams, purple dots indicate anatomical landmarks used for establishing body frames. Orange dots represent landmarks for defining JCSs, while red dots denote landmarks for MTU paths. Blue dots are used to mark landmarks for defining wrapping surfaces. Transparent red contours depict the muscle, grey slender contours outline the tendons, and the black frames labeled “G” represent the reference frame in which the landmark coordinates are collected. (B) The input layer with local data. Separate MTUs (red) are used to represent the muscle-tendon structure. (C) The middle

layer with modified components following the proposed branching modeling method. The dark grey circles denote the extra body introduced to connect the single MTU (red) and the passive elements (green). (D) The output layer with the constructed musculoskeletal model.

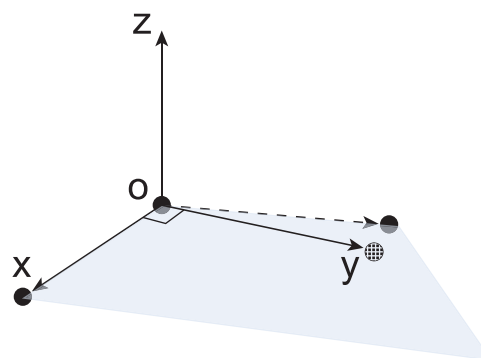
<https://doi.org/10.1371/journal.pcbi.1012243.g001>

within the default coordinate system in which the CT scan is saved and visualized. This default coordinate system provides a common and consistent spatial reference for all the data in the CT scan.

To establish a reference coordinate system within the current software setup, it is required to provide three landmark points: one for the origin of the coordinate system, one to denote the positive X-axis direction, and another to denote the positive Y-axis direction (Fig 2). Thus, for each body, users provide coordinates of three anatomical landmarks for establishing the corresponding body frame.

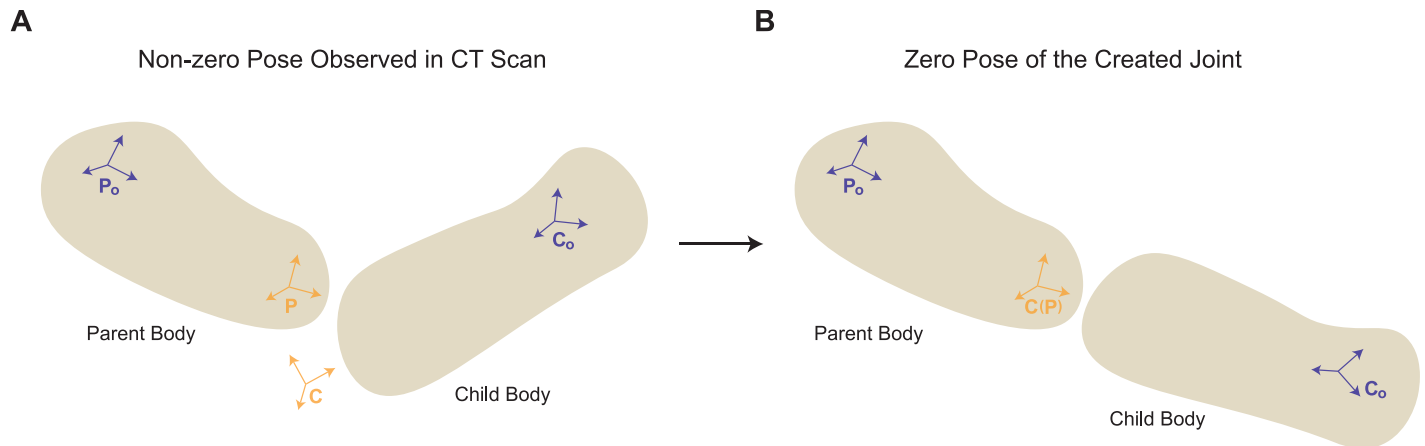
To create a joint that defines the kinematic relationship between two bodies, termed parent and child bodies, two joint frames are constructed, each affixed to the parent body and child body (see Fig 3), respectively. Therefore, six landmark points are required to construct the joint frames for each joint. Three of these landmarks specify the location and orientation of the joint frame on the parent body, and the other three specify the location and orientation of the joint frame on the child body. During the JCS landmark identification process, some recommended guidelines for standardized JCS definitions can be found in [59–61].

To maintain compatibility with the conventional method of modeling branched muscles and tendons in cases of complex branching phenomena, and to seamlessly incorporate the proposed branching modeling method, offering flexibility in creating models using either technique, the MTU-related data provided in the input are stored in a manner that presents branched muscles and tendons as distinct, separate, and independent muscle-tendon compartments. This adheres to the rule of a single tendon per muscle in OpenSim. For instance, when a muscle is attached to a tendon that branches into two insertion points (see Fig 1), two MTUs are used to represent this muscle-tendon structure. The anatomical landmarks for MTU path are provided for each MTU, featuring identical landmarks along the muscle pathway, and different landmarks along the tendon pathway. Correspondingly, two sets of physiological parameters are designated for these two MTUs, encompassing divided maximal isometric force, consistent pennation angle and optimal fiber length, and different tendon slack length, etc..



**Fig 2. Coordinate system construction.** Three solid black dots indicate the landmarks for the origin and the positive directions of the X and Y axes, respectively. The plane, depicted in light blue, is defined by these points. The dashed black dot represents the adjusted landmark for the positive Y-axis direction, ensuring orthogonality with the X-axis.

<https://doi.org/10.1371/journal.pcbi.1012243.g002>



**Fig 3. Joint construction.** A joint that defines the kinematic relationship between frames P and C is created, where frame P is affixed to the parent body, with its body frame denoted by  $P_o$ , and frame C is affixed to the child body, with its body frame denoted by  $C_o$ . By making frames P and C coincident, the non-zero pose observed when collecting landmarks to build the JCS from CT scans or other data sources, shown in (A), is transformed into the default zero-pose, shown in (B).

<https://doi.org/10.1371/journal.pcbi.1012243.g003>

To enable MTUs to interact with certain wrapping objects, mimicking the way muscles and tendons navigate around bones and other structures, wrapping surfaces can be established. These surfaces guide the paths of linked MTUs in a manner that cannot be achieved with fixed MTU path points alone. To create a wrapping surface, provide three anatomical landmarks that define the reference frame, as well as additional geometry parameters such as the radius and height of the wrapping surface.

To prepare for implementing the proposed branching method, i.e., the conversion from the second layer to the third layer, the fact that multiple MTUs together form a branching unit within the biological equivalent is preserved as branching group data.

For other essential data, the segmented bone geometries enable effective visualization within OpenSim. Notably, these segmented geometries are exported directly from the same global reference frame used for collecting landmark coordinate data, aligning with the default frame of the CT scan or other imaging data.

The joint range of motion data restricts the relative movement of the bodies. Such data might not be immediately accessible to users until the joints are precisely defined in the model as this information is more readily obtainable in local frames as opposed to a single global frame. Therefore, users may initially overlook this aspect and later assign joint range of motions after constructing a model devoid of embedded joint range of motion. In a model, joint range of motion can be determined by ensuring no penetration occurs between the parent and child body of a joint. Experimental data can also be instrumental in this process.

### Transformer creates layer 2

In contrast to data in the first layer, which is collected under the same global coordinate system, OpenSim models use distinct reference frames for each body, making the model more biomechanically relevant and intuitive. Consequently, it necessitates the transformation of each segmented bone geometry, initially exported from the segmented CT scan, so that each body in the resulting model has its own local coordinate system. Additionally, elements such as joints, MTU paths, wrapping surfaces, etc., are constructed based on the relationships between associated bodies. This construction requires transforming their “global” coordinates into the respective “local” coordinate systems of the associated bodies.

Within the tool, when the “global” data and other necessary data are input into the first layer, the tool seamlessly performs the required transformations from the “global” data to a “local” format, while keeping other necessary data unchanged. From there, users can obtain bodies, JCSs, MTUs, and wrapping surfaces that are directly compatible with OpenSim, all of which are stored in the second layer. This conversion is particularly advantageous as it eliminates the need for manual transformations by users. Consequently, users can directly integrate data obtained from other software tools, thus streamlining the model development process.

A detailed explanation of the automated transformation from “global” to “local” is provided as follows.

As stated in the previous subsection, to establish a reference coordinate system, three landmark points are provided. These three landmark points combine to form two vectors, representing the reference X and Y-axes (see Fig 2), respectively.

When identifying these landmarks, it is very likely that these two vectors are not perfectly perpendicular to each other due to various factors, such as measurement errors during the process. This does not conform to standard Cartesian geometry. To address this, the tool adjusts the location of the landmark indicating the positive direction of the Y-axis. This is achieved by locating a new point on the plane defined by these two vectors that is closest to the original landmark for the positive Y-axis direction, and ensuring that the new point creates a resulting angle of 90 degrees between the reference X-axis and corrected reference Y-axis. Subsequently, the reference Z-axis can be defined by taking the cross product of the X-axis vector and corrected Y-axis vector. Finally, we normalize these vectors by dividing each by their respective magnitudes to establish a well-defined coordinate system.

By following the procedure outlined above, body frames are established using the “global” coordinates of three anatomical landmarks for each body from the supplied data in the first layer of the tool. From the normalized vectors, which represent body frame axes, we can build transformation matrices. These transformation matrices can then transform the original segmented bone geometries from the “global” coordinate system to their individual “local” coordinate systems. As a result, with the transformed bone geometries and associated center of mass and inertia defined in the corresponding body frames, a set of rigid bodies—a primary building block of the model—can be readily derived using OpenSim’s API.

With six anatomical landmarks provided for each joint construction, as explained earlier, three represent the joint frame’s location and orientation on the parent body, and the other three represent those on the child body. Following the method used to transform bone geometries from “global” to “local”, a constant transformation that represents the offset between a joint frame and the corresponding body frame can be calculated. That establishes the JCSs.

Joints are modeled as CustomJoints, which is the most generic joint representation in OpenSim. It allows for motion in 6 DOFs, offering the flexibility in modeling traditional joints (e.g., slider joint, universal joint) as well as complex biomechanical joints.

Implementing the same transformation concept, the positions of MTU path points relative to the body frames through which the MTU traverses can be calculated. Similarly, the location and orientation of a wrapping object in relation to the associated body segments can also be determined.

The aforementioned section explains how the tool efficiently converts the provided “global” data into a “local” format, designed to integrate smoothly with OpenSim. Alternatively, if users opt to directly provide local data and construct a model from there, they can bypass the first layer and simply supply the local data through the second layer. This feature allows users to create artificial toy musculoskeletal models conveniently, as working with local data aligns more intuitively with the understanding of the desired local relationships.

### Brancher creates layer 3

As briefly touched on in previous sections, the traditional representation of a muscle-tendon branching structure involves multiple independent MTUs (see Fig 1B). In that representative scenario, two distinct MTUs are used to account for the two tendons with different insertions. Correspondingly, the muscle is represented as two compartments, each associated with a tendon. Both MTUs stem from the same muscle origin, following the same path up to the branching point, then diverging along their unique tendon paths.

The physiological parameters of the two MTUs depend on the respective muscle compartment and tendon. Usually, they share the same muscle pennation angle, and possess nearly identical optimal fiber lengths. The maximal isometric force of the muscle is allocated based on the physiological cross-sectional area (PCSA) of each muscle compartment, and the tendon slack length of each MTU aligns with the tendon's resting length. Collectively, these parameters characterize the active and passive force-length curve, the force-velocity curve of the muscle, and the force-length curve of the tendons, determining the force development of MTUs in a musculoskeletal model. To synchronize the functioning of these muscle compartments, two identical muscle excitation signals can be applied across the MTUs.

In contrast to the conventional method, our proposed branching modeling approach utilizes a single MTU to represent the segment of muscles and tendons that share the same pathway from their origin, prior to the emergence of any branching, within a group of original MTUs specified in the provided branching group data. Hence, in the representative scenario (Fig 1C), the muscle is no longer fragmented into two compartments. Instead, a single MTU is used to represent the muscle in the real physiological system. The maximal isometric force of this single MTU equals the sum of the maximal isometric forces from the two MTUs in the conventional branching modeling method, while its optimal fiber length and pennation angle remain consistent with those of the two MTUs.

There are two cases of tendon branching that must be modeled slightly differently. The muscle-tendon unit depicted in Fig 1C directly branches into two tendons at the muscle-tendon transition. In this case, the single MTU would ideally represent a muscle without tendons. However, because OpenSim does not support MTUs with zero tendon (slack) length, an exceedingly short tendon can be introduced into that single MTU. This ensures that the MTU predominantly represents the muscle. To prevent the stretching of this extra tendon from affecting muscle force distribution and coordination to the downstream tendons, the short tendon can be made rigid. Alternatively, when a tendon segment is connected to the muscle before branching occurs, the additional rigid short tendon becomes unnecessary. Instead, we can simply use an MTU to represent the muscle associated with that particular tendon.

To represent the tendons post-branching, ligaments are used. In OpenSim, as in biomechanics in general, a ligament represents passive connective tissue that connects bones to other bones. Both the tendon in an MTU and a ligament are defined by a normalized force-length curve that describes how the force responds to being stretched, a force scale that scales the normalized force-length curve, and the resting length. Therefore, by assigning the same properties, they are treated equivalently in OpenSim.

To emulate the branching architecture, extra bodies, modeled as spheres, are introduced for each branching point. As branching points in reality can slide relative to vertebrae, a CustomJoint is incorporated between each extra body and its corresponding vertebra to ensure this movement capability. Within the CustomJoint, the three rotational DOFs are restricted, rendering it solely as a sliding joint. To effectively negate the impact of inertia changes induced by the inclusion of the extra body to the model on the force development and the overall motion, the mass of the extra body is set to an exceptionally low value. For the representative

muscle-tendon structure, one extra body is employed at the muscle-tendon transition point, accounting for the muscle dividing into two tendons. By linking the extra body to the respective tendon insertion points using ligaments, the muscle and tendons in the musculoskeletal system are inherently interconnected in the model. Therefore, the intricate interactions and dependencies within the musculoskeletal system can be more faithfully represented.

In the proposed branching modeling method, the resting length of each ligament should match that of its corresponding tendon segment in the actual system. The force scale for each ligament should be allocated as a portion of the muscle's maximal isometric force, because the force in the normalized force-length curve is normalized by the muscle's maximal isometric force in OpenSim. If the tendons have uniform material properties, the ligament's force scale can be determined by distributing the maximal isometric force based on the tendons' cross-sectional areas.

In summary, based on the data from the second layer, where the elements of the musculoskeletal model align with the traditional branching modeling approach, the proposed branching modeling method demands several adaptations. Specifically, these include the addition of supplemental bodies to the original model, the designation of sliding joints for these newly added bodies, adjustments to MTU pathway, the integration of ligaments to denote the branched tendons, recalibration of muscle-tendon physiological parameters, and updates to the paired association between the wrapping surfaces and the MTUs/ligaments. This conversion is inherently integrated into the tool, facilitating the creation of models using the proposed branching modeling method. The detailed procedure for the implementation of this conversion is expounded in Algorithm 1.

#### **Algorithm 1: Brancher**

##### **Step 1. Construct Directed Tree Graph**

**foreach** branching group **do**

**foreach** MTU in the branching group **do**

**foreach** path point of the MTU (from its origin to its insertion) **do**

      Check for existing vertex by associated body, the position on that body, and the point type

**if** vertex exists **then**

        Append the path point information (MTU name, index, type) to the vertex

**else**

        Create a new vertex for the path point

        Associate the path point information to the vertex

**if** path point is not the origin **then**

          Create a directed edge from the previous path point vertex to the new vertex

##### **Step 2. Identify Special Vertices**

**foreach** vertex in the graph **do**

**if** vertex's indegree × vertex's outdegree ≠ 0 and vertex's indegree + vertex's outdegree ≥ 3 **then**

    Mark the vertex as a branching vertex

**if** vertex represents an MTU's origin **then**

    Mark the vertex as an origin vertex

**if** vertex represents an MTU's insertion **then**

    Mark the vertex as an insertion vertex

##### **Step 3. Add Extra Bodies and Sliding Joints**

**foreach** branching vertex **do**

  Construct an extra body modeled as a sphere

  Build a CustomJoint with translational DOFs between the associated body of the vertex and the sphere

##### **Step 4. Refactor MTU Paths and Build Ligament Paths**

**foreach** origin vertex **do**

Traverse the graph until reaching any branching vertex  
Model this segment as an MTU, originating from the origin vertex's associated body, inserting to the branching vertex's associated extra body

**foreach** branching vertex **do**

Traverse the graph until reaching a branching or insertion vertex  
Model this segment as a ligament, originating from the branching vertex's associated body, inserting to the branching or insertion vertex's associated (extra) body

In the algorithm, directed tree graphs are constructed to represent the muscle-tendon branching structure. By analyzing these tree graphs, the algorithm automatically makes the necessary adjustments and outputs the modified components into CSV files, except for the reassessment of muscle-tendon physiological parameters. Users manually input physiological parameters for the modified MTUs and newly introduced ligaments in the corresponding CSV file, based on anatomical data.

## Builder creates layer 4

Leveraging OpenSim's API, the modified components in the third layer that fits into OpenSim are integrated via the functionality named "**Builder**" to create a musculoskeletal model. In this model, the branched muscles and tendons are modeled using the proposed method. As mentioned earlier, the components in the second layer present the branched muscles and tendons using the conventional method, and these components are also directly compatible with OpenSim. Therefore, to create a musculoskeletal model using the conventional branching modeling method, users can simply bypass the third layer and construct a model from the second layer. Thus, the tool offers the capability of creating two musculoskeletal models, using two different branching modeling methods, both utilizing the same data.

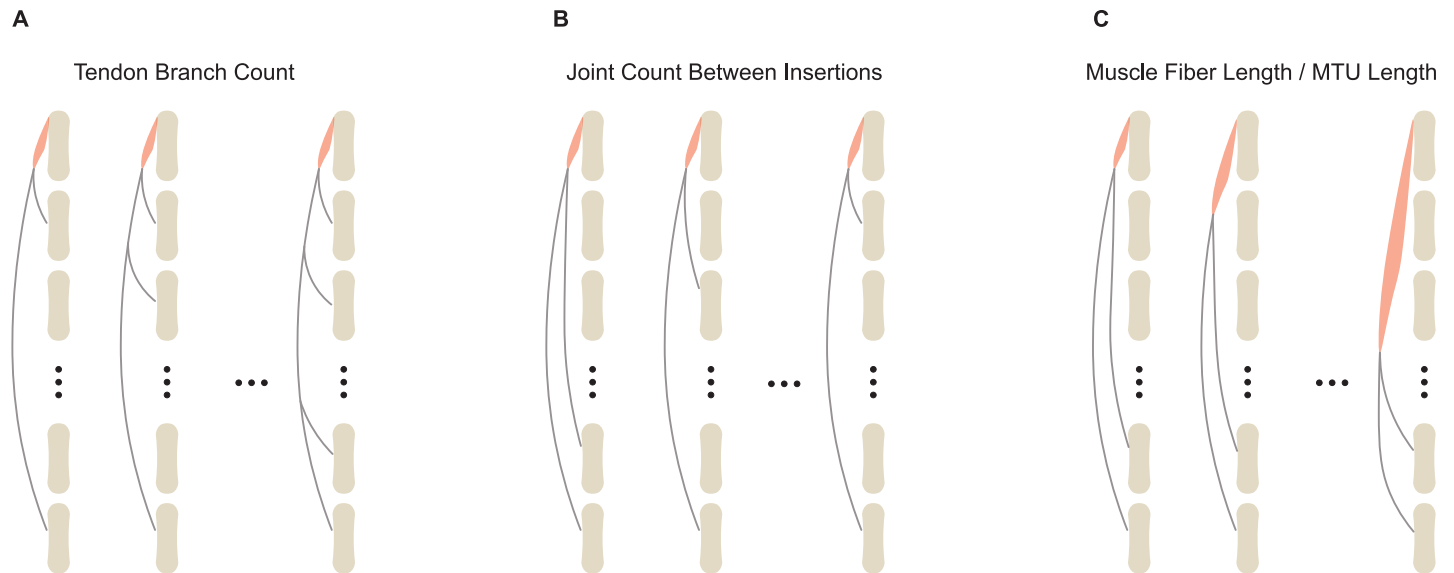
## Comparative simulation case study

Here, we employ *ArborSim* to investigate the tangible distinctions between the conventional parallel modeling method and the proposed branching modeling method for modeling complex muscle-tendon architectures. The investigation used comparative simulations conducted on artificial toy articulated musculoskeletal systems. Within these systems, we created muscle-tendon architectures with branching features. We focused on the branching of tendons rather than muscles, in line with our findings in mammalian tails [45, 46].

Drawing from observed variations in muscle-tendon branching architectures across a variety of mammalian tails [45, 46], we identified three relatively prevalent categories of variation, which serve as the main focus of our comparative simulations:

- Variations in the number of tendon branches into which a muscle divides.
- Variations in the number of joints between the terminals of the branched tendons.
- Variations in the ratio of muscle fiber length to MTU length.

These categories are illustrated in Fig 4. For each category, six musculoskeletal systems were designed. To simplify the comparison between the traditional branching modeling approach and our proposed method, we restricted these toy musculoskeletal models to a planar configuration. Each toy musculoskeletal model consisted of 8 vertebrae connected by planar joints, with each vertebra modeled as a uniformly dense cylinder in OpenSim. The cylinders had a mass of 0.1 kg, length of 0.2 m, and radius of 0.025 m. A gap of 0.025 m



**Fig 4. Comparative simulation categories.** The category of varying numbers of tendon branches is shown in (A). The category representing the varying numbers of joints spanned between insertions is depicted in (B). The category showing the varied muscle fiber length to the longest MTU ratio is presented in (C). In all three diagrams illustrating these categories, the red contours indicate the muscles, while the grey slender contours represent the branched tendons. To more clearly visualize the branching, the tendons and muscles in the diagrams are displayed away from the bones. However, in the models, they are all connected to the bones through the use of “via points” on the bone surfaces, allowing them to exert force on the bodies along the pathway.

<https://doi.org/10.1371/journal.pcbi.1012243.g004>

between each pair of consecutive vertebrae was introduced to prevent interpenetration during movement. The range of motion for each joint was calculated based on the cylinder dimensions and the gap length, with `CoordinateLimitForce` in OpenSim applied to enforce these constraints.

Each system was equipped with a muscle-tendon branching architecture, tailored to its respective category. Each muscle-tendon branching architecture consisted of a single muscle and multiple branched tendons. To ensure that all vertebrae actively respond to muscle activation, a long tendon spanning from the muscle-tendon transition area to the most distal vertebra (bottom in Fig 4) was present in all the systems. In the category where the ratio of muscle fiber length to MTU length varied (Fig 4C), the MTU length was denoted by the measurement of the longest MTU in a given model.

For each system, two musculoskeletal models were constructed: one using the conventional branching modeling method and the other using the proposed branching modeling method, following the steps presented in previous subsections (note that both branch modeling methods are incorporated within `ArborSim`). Given the physiological parameters of the muscle, for models using the conventional branching modeling method, the maximal isometric force of the muscle was evenly allocated among these MTUs. For models using the proposed branching modeling method—where a single MTU is used to represent the muscle—the maximal isometric force was directly assigned to that MTU. For fair comparison between equivalent models, we divided the total maximum isometric force of the muscle by the number of insertions to set the force scale for each distal ligament element. The force scale for the proximal ligaments was then set to the sum of the force scales of the connected distal ligaments.

Regarding other MTU and ligament modeling parameters, the optimal and default muscle fiber lengths were defined as the distance between the muscle’s origin and insertion in a zero-pose, where all the joint angles are zero, resulting in a straight, fully extended pose. Similarly,

the tendon slack lengths for MTUs in the conventional method, as well as the resting lengths for ligaments in the proposed method, were established based on the default lengths of their respective segments in this zero-pose configuration. For the branched model, an extremely short rigid tendon of 0.0001 m was included in the MTU that represents the muscle to circumvent OpenSim's limitation of not supporting MTUs with zero tendon length.

To ensure comparability between the conventional and proposed methods, the same normalized force-length curves and force-velocity curves were used for the muscle. Additionally, the normalized force-length curve of the tendon in the MTU was identical to that of the ligament. The muscle was modeled as a Millard2012EquilibriumMuscle [62] in OpenSim. The mass of the extra body introduced when using the proposed branching modeling method was set to  $5 \times 10^{-4}$  kg.

Within each category, an identical muscle excitation signal was applied to all systems with varying muscle-tendon architectures, and simulations were run under that excitation signal for a duration of  $T = 3$  seconds. To evaluate the difference between the traditional branching modeling method and proposed method on the resulting kinematics, we defined the following metric:

$$\Delta = \max_t \sum_{i=1}^n \frac{|\theta_i^{con}(t) - \theta_i^{pro}(t)|}{\bar{\theta}_i} \times 100\%, \quad t \in [0, T] \quad (1)$$

where  $\theta_i^{con}(t)$  denotes the joint angle of  $i$ -th joint at time  $t$  for models generated using the conventional branching modeling method,  $\theta_i^{pro}(t)$  denotes the joint angle of  $i$ -th joint at time  $t$  for models generated using the proposed branching modeling method,  $\bar{\theta}_i$  denotes the positive bound of  $i$ -th joint, and  $n$  is the number of joints, which is 8 in our study. In the metric, the numerator  $\sum_{i=1}^n |\theta_i^{con}(t) - \theta_i^{pro}(t)|$  captures the total absolute differences in joints angles across all eight joints at time  $t$ . The denominator  $\sum_{i=1}^n \bar{\theta}_i$  represents the sum of the positive bound of each joint. This reflects the pose where the toy musculoskeletal system is rotated to its limit.

Therefore, the metric quantifies the maximal percentage of the total absolute differences in joint angles across all eight joints over the entire motion trajectory, relative to the joint limits. This metric highlights the worst-case scenario in terms of the discrepancy between the two methods, providing a robust evaluation of their impact on system behavior.

## Sensitivity analysis

Performing a sensitivity analysis is valuable for assessing the impact of potential measurement errors on the parameters of a model, providing insights into the model's robustness. Results from previous extensive sensitivity analyses, particularly focusing on the effects of Hill-type muscle-tendon model parameters within musculoskeletal models [63–70] indicate that the tendon slack length and maximal isometric force are the most sensitive parameters. Therefore, in this study, we conducted a sensitivity analysis on our proposed branching modeling method, focusing on these two parameters, to investigate their influence on the joint kinematics of the toy musculoskeletal models introduced in the previous section. Because all the tendons in these toy musculoskeletal systems were represented by ligaments, the sensitivity analysis on the tendon slack length should be equivalent with a sensitivity analysis on the ligament resting length in this context.

To assess the influence of changes in maximal isometric force, we applied ten perturbations to their nominal values, varying from  $-10\%$  to  $+10\%$  in  $2\%$  increments. Initially, we planned to use this same range for the nominal ligament resting length in each toy model. However, in most toy musculoskeletal systems, tendons are considerably longer compared to muscle length. Consequently, substantial perturbations, particularly negative ones, to ligament resting

length could lead to simulation failures in OpenSim. This issue arises because negative perturbations cause the ligament to contract at the start of the simulation when the muscle is not fully activated. The larger the negative perturbation, the more the ligament contracts. If such contraction moves the extra body, representing tendon branching points outside the two consecutive via points at the ends of the cylinder body, the simulation halts. As a result, we reduced the perturbation range for ligament resting length to  $-5\%$  to  $+5\%$ , also in  $2\%$  steps. Notably, even this altered range is large due to the tendons' considerable length in these systems. While it was possible to address simulation crashes by manually adjusting parameters in the toy models, we avoided this due to the numerous parameters, their complex interrelations, and their complex effects on the resulting motions, especially in the context of branching. Therefore, we chose to simply reduce the perturbation range.

It is important to note that as there was more than one tendon in all of these toy musculoskeletal systems; we have chosen to apply the perturbation of the ligament resting length to all the ligaments simultaneously. For each perturbation, a new simulation was run under the same excitation signal used for the original experiments. Sensitivity analyses were conducted on both the conventional method and the proposed method.

To quantify the effect of the perturbation on the resulting motion, we re-applied the metric introduced:

$$\Delta = \max_t \sum_{i=1}^n \frac{|\tilde{\theta}_i^{pro}(t) - \theta_i^{pro}(t)|}{\bar{\theta}_i} \times 100\%, \quad t \in [0, T] \quad (2)$$

This time, instead of comparing the proposed method with the conventional method, we compared the perturbed with unperturbed model for both the proposed method and the conventional branching modeling methods. Here,  $\tilde{\theta}_i^{pro}(t)$  is the perturbed joint angle of  $i$ -th joint at time  $t$ .  $\theta_i^{pro}(t)$  is the unperturbed joint angle of  $i$ -th joint at time  $t$ .

## Results and discussion

### Model construction

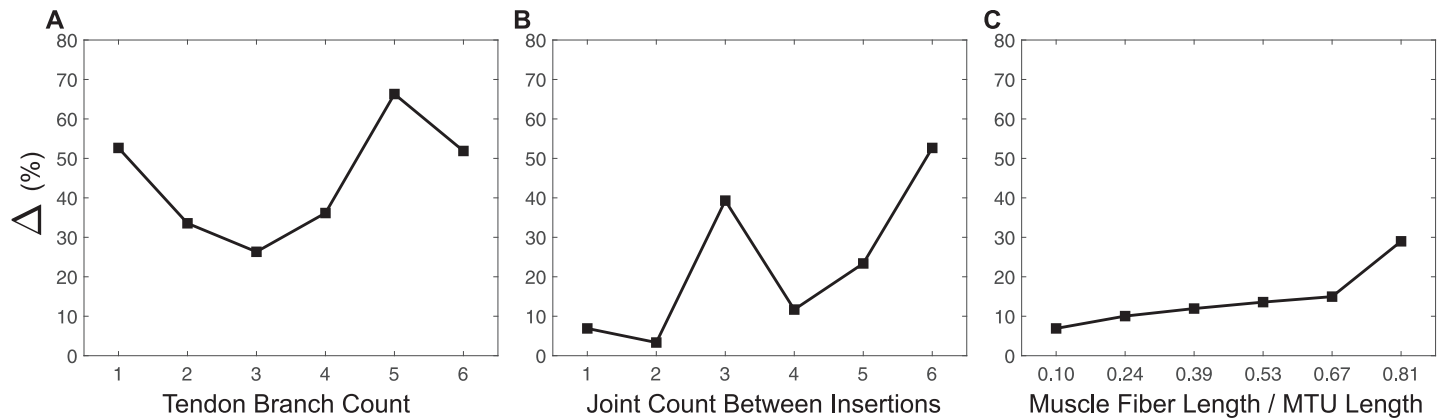
To demonstrate the use of the designed software for building articulated musculoskeletal models of highly jointed systems, the toy musculoskeletal systems introduced in the previous section were constructed by running through the entire software model construction pipeline. In addition, joint coordinate reporters and muscle controllers are assigned to each model using the OpenSim API. This setup allows for the recording and post-analysis of simulated motion in response to predefined muscle excitation signals.

The CSV data files and code for this model construction and the associated simulation results are available at <https://github.com/EMBiRLab/ArborSim>.

### Comparative simulations

The joint kinematics of the musculoskeletal models, which represent articulated systems built using the conventional and proposed methods, showed large differences. The quantitative differences between the derived motions, evaluated using metric (1), are presented in Fig 5.

Specifically, within the category of different numbers of tendon branches (Fig 5A), the range of  $\Delta$  across six simulations was [26.33%, 66.31%]. The peak value of this difference occurred when there were five tendon branches. Overall, as the number of tendon branches increased, the difference in joint kinematics first decreased and then increased. This trend suggested a complex, nonlinear relationship in between branched tendon systems.



**Fig 5. Quantitative motion comparison.** Quantitative differences between the proposed branching modeling method and the conventional branching modeling method, evaluated by metric (1), on the resulting motions of articulated systems with a varied number of tendon branches (A), a varied number of joints between insertions (B), and varied muscle fiber length to the MTU length ratios (C) are shown, respectively. Note that the ratios of muscle fiber length to MTU length presented in (C) correspond to those ratios at zero-pose.

<https://doi.org/10.1371/journal.pcbi.1012243.g005>

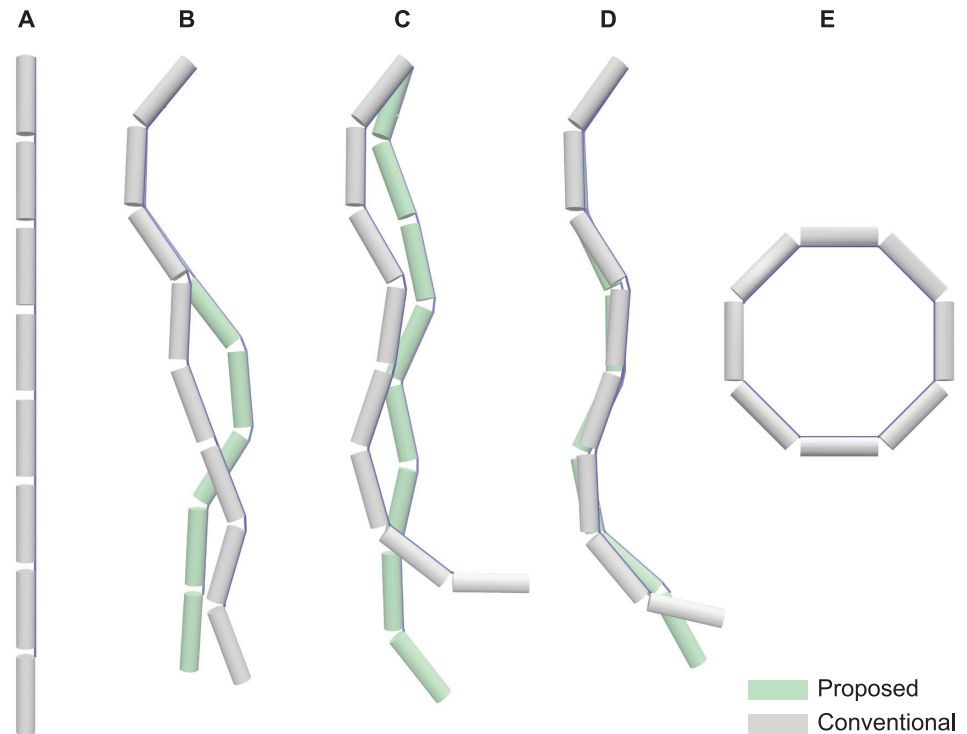
Within the category of different numbers of joints between insertions (Fig 5B), the range of  $\Delta$  across six simulations was [3.34%, 52.65%]. The largest  $\Delta$  occurred when the branched tendons in the models spanned six joints between insertions. Disregarding the abrupt increase in the percentage change at three joint counts between insertions, a general incremental trend was observed in the motion differences with respect to the joint count between insertions. However, this jump might once again indicate a more complex, possibly nonlinear relationship.

Within the category of different muscle fiber length to MTU length ratios (Fig 5C), the range of  $\Delta$  across six simulations was [6.91%, 28.99%]. The largest  $\Delta$  occurred when the muscle spanned five joints, with the corresponding ratio being 0.81. There was a consistent trend of increasing motion differences as the muscle fiber length to MTU length ratio increased. However, compared to the other two categories, variations in the muscle fiber length to MTU length ratios resulted in relatively smaller changes and variance in  $\Delta$  values.

To visually demonstrate the motion differences between the two modeling methods, we included a figure that highlights the comparison with the largest kinematic differences, determined by the metric (1), across each category (Fig 6). Animated motion comparisons throughout complete simulations are provided in S1–S3 Videos files.

Collectively, these results indicate that the conventional and proposed branching modeling methods exhibit notable differences in predicting joint kinematics. In particular, these differences become more pronounced as the complexity of muscle-tendon branching architectures increases, which involves numerous tendon branches, and an extensive number of joints spanned by branched tendon insertions. The resulting large differences can be concerning in fields demanding precise movement prediction, such as surgery outcomes, especially as the complexity of the muscle-tendon networks escalates in systems like human hands, spines, and elongated tails.

To delve further into the source of the differences between the proposed method and the conventional method, we examined changes in muscle fiber length over time for each method. Because muscle fiber length affects contraction velocity and force, it is important to correctly model changes in muscle fiber length through time. In a system with one muscle attaching to multiple tendons, an accurate MTU model should reflect the uniform muscle fiber length changes within that single muscle. Using the conventional method with independent MTUs



**Fig 6. Qualitative motion comparison.** (A) displays the zero-pose of the models. For (B), (C), and (D), each panel presents two comparative snapshots from the simulation that exhibited the highest  $\Delta$  out of six simulations per category. These snapshots represent the largest motion differences between the conventional branching modeling method (depicted in white grey) and the proposed branching modeling method (depicted in light green). Specifically, (B) compares models with varying numbers of tendon branches, (C) compares models with different joint counts between insertions, and (D) compares models with varied muscle fiber to MTU length ratios. (E) illustrates the pose where each joint is set to its maximum value within the range of motion.

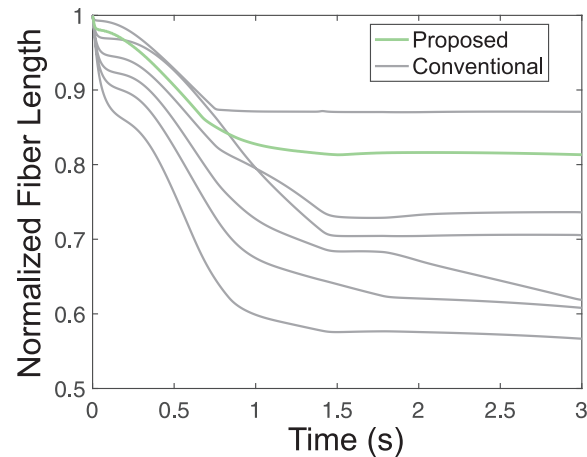
<https://doi.org/10.1371/journal.pcbi.1012243.g006>

reflecting each tendon insertion point, the normalized fiber length changes over time are not uniform (Fig 7, gray lines). This discrepancy indicates that the conventional branching modeling method fails to explicitly consider the physical interactions between the entities in a complex muscle-tendon network. In contrast, the proposed branched modeling method only includes one MTU, inherently constraining all muscle fiber length changes to be uniform, as in a real muscle (Fig 7, green line).

The results here highlight the role of the physical interactions between branched muscles and tendons in highly jointed systems. The study, though preliminary and focused on a kinematics comparison in toy musculoskeletal systems, underscores the necessity for continued in-depth exploration and advancement in the understanding of complex muscle-tendon branching phenomena and the branching modeling techniques in future works.

In this study, we employed toy musculoskeletal models to compare the conventional branching modeling method with our newly proposed approach. Future implementation of our methods will involve validation with real *in vivo* data.

Validation can be performed at two different hierarchical scales. One is solely on the scale of the branched muscle-tendon network. This can be pursued by collecting data on contraction, the distributed interaction forces of muscles and tendons, and other relevant factors when a load is applied to a branched muscle-tendon network. One can follow an approach similar to that outlined in [51], which examined the strain in branched tendons. To align



**Fig 7. Normalized muscle fiber length comparison.** The plot compares the normalized muscle fiber lengths of the independent MTUs modeled using the conventional method with those of a single MTU modeled using the proposed branching method. Here, “normalized” refers to the normalization of fiber length relative to the optimal fiber length. The comparison highlights the case with the largest  $\Delta$ , as shown in Fig 5, corresponding to a Tendon Branch Count of 5 in Fig 5A. In the plot, normalized muscle fiber lengths are represented in light gray for the conventional method and light green for the proposed method.

<https://doi.org/10.1371/journal.pcbi.1012243.g007>

experiments with simulations, we need detailed characteristics of each individual muscle and tendon within the network, including maximal muscle force, default fiber length, and tendon slack length for each branch. By measuring these specific parameters, we can replicate the experiments under identical loading conditions, thus allowing a direct comparison of experimental and simulation data to validate our branching modeling method.

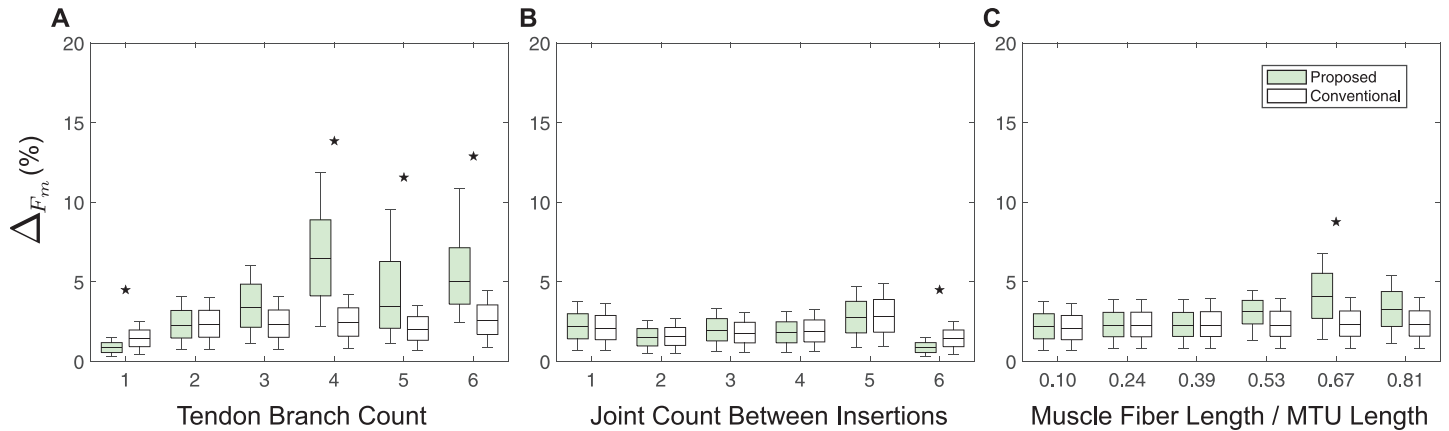
Another hierarchy of validation would involve the integration of the branched muscle-tendon network with skeletal bodies, and the comparison of the resulting motions with both the traditional and proposed branching modeling methods. Such validation would require the construction of detailed musculoskeletal models with branched tendons, such as mammal tails [45, 46]. To achieve this, in addition to the muscle-tendon-wise parameters, capturing muscle excitation and joint kinematic data would be helpful.

### Sensitivity analysis

The resulting deviations from the nominal movement, evaluated by (2), showed large variability across different categorized muscle-tendon branching structures. The effects of the perturbations on maximal isometric force, and ligament resting length on the joint kinematics are shown in Figs 8 and 9, respectively.

Regardless of the muscle-tendon branching structure category, the models’ joint kinematics was more sensitive to ligament resting length than to muscle maximal isometric force. This is consistent with previous studies [63–68], emphasizing the crucial role of the tendon slack length plays in the joint kinematics of the musculoskeletal systems.

Regarding the sensitivity to muscle maximal isometric force, specifically within the categories of varying numbers of joints between insertions (Fig 8B) and varying muscle fiber length to MTU length ratios (Fig 8C), the observed  $\Delta$  values were consistently low and stable. This suggests a low impact of these categories and variations within these categories on joint movement sensitivity. Within the category of varying numbers of tendon branches (Fig 8A), systems with fewer than 4 tendon branches exhibited a similar low sensitivity. However, for systems

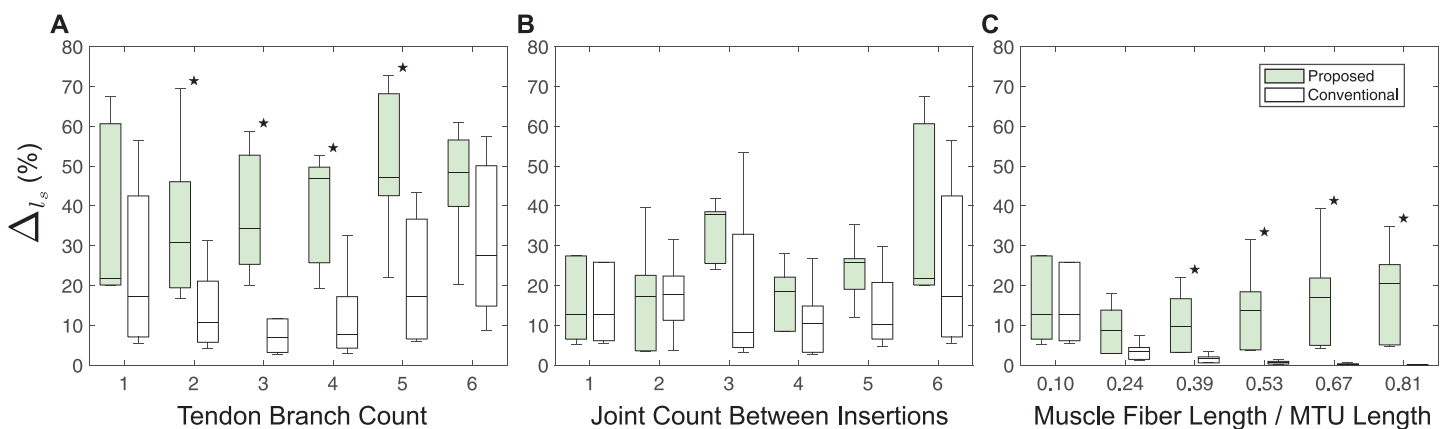


**Fig 8. Maximal isometric force sensitivity analysis.** Impact of maximal isometric force changes in the proposed versus conventional method on articulated system movement, measured by metric (2) across variations in tendon branch count (A), joint count between insertions (B), and muscle fiber length to the MTU length ratio (C). Solid black stars, if present, above the box plots indicate that the difference in  $\Delta$  values between two groups, modeled by the conventional and proposed methods, is significant, determined by a two-tailed T-test ( $P$ -value  $< 0.05$ ). The sample size for each box in the box plot was 10, corresponding to ten perturbations of nominal values adjusted from  $-10\%$  to  $+10\%$  in increments of  $2\%$ , as described in the main text.

<https://doi.org/10.1371/journal.pcbi.1012243.g008>

with 4 or more tendon branches, the joint kinematics showed a notably increased sensitivity for the proposed branching modeling methods. Moreover, the models built using the proposed method were significantly more sensitive than those built using the conventional method.

Regarding the sensitivity to tendon/ligament resting length, within the category of varying numbers of tendon branches (Fig 9A), the proposed method yielded higher  $\Delta$  values compared to the conventional method, suggesting that the proposed method is more responsive to changes in number of tendon branches. In the category of varying numbers of joints between insertions (Fig 9B), both methods demonstrated similar levels of  $\Delta$  values, indicating a comparable sensitivity to the number of joints between insertions. In the category of varying muscle fiber length to MTU length ratios (Fig 9C), an increase in the ratio of muscle fiber length to MTU length resulted in a decrease in  $\Delta$  values for the conventional method. Conversely, the



**Fig 9. Ligament resting length sensitivity analysis.** Impact of ligament resting length changes in the proposed versus conventional method on articulated system movement, measured by (2) across variations in tendon branch count (A), joint count between insertions (B), and muscle fiber length to the MTU length ratio (C). Solid black stars, if present, above the box plots indicate that the difference in  $\Delta$  values between two groups, modeled by the conventional and proposed methods, is significant, determined by a two-tailed T-test ( $P$ -value  $< 0.05$ ). The sample size for each box in the box plot was 6, corresponding to six perturbations of nominal values adjusted from  $-5\%$  to  $+5\%$  in increments of  $2\%$ , as described in the main text.

<https://doi.org/10.1371/journal.pcbi.1012243.g009>

Table 1. Computational time comparison.

Architecture Category	Conventional (sec)		Proposed (sec)	
	Mean	Variance	Mean	Variance
Tendon Branch Count	4.44	3.24	4.92	5.87
Joint Count Between Insertions	4.06	0.90	4.29	1.46
Muscle Fiber Length / MTU Length	4.06	0.38	3.57	0.17

<https://doi.org/10.1371/journal.pcbi.1012243.t001>

proposed method exhibited a relative consistent level of  $\Delta$  values across different ratios. Additionally, at each ratio, the proposed method showed higher  $\Delta$  values than the conventional method, highlighting its higher sensitivity to the muscle fiber length to MTU length ratio.

These findings collectively underscore the importance of considering tendon slack length and the complexity of tendon branching when modeling muscle-tendon systems. They also highlight the enhanced responsiveness of the proposed method to key biomechanical parameters, suggesting its potential for facilitating more realistic simulations in biomechanics research.

### Computational time

In the proposed branching modeling method, introducing extra bodies of extremely small mass could potentially extend computational time due to potentially stiffer dynamics. To evaluate computational performance, we compared the computational times of the simulations using the proposed branching modeling method with those using the conventional branching modeling method. The comparison results are documented in Table 1.

In the table, for each muscle-tendon architecture category, we considered all the simulations presented in Figs 5, 8 and 9 to calculate the average computational time required to complete a simulation, each having a duration of  $T = 3$  (sec).

Overall, the average computational times were similar between the two branching modeling methods.

Statistically, a significant difference was observed between the methods in the category of the ratio of muscle fiber length to MTU length, determined by a two-tailed T-test ( $P$ -value  $< 0.05$ ). However, there were no significant differences observed between the methods in categories with varying numbers of tendon branches or joint counts between insertions.

In the category of varying numbers of tendon branches, the simulations using the proposed methods took longer than those using the conventional method. This likely suggests that the more extra bodies of extremely small mass were introduced to the systems, the simulation time increased. It is important to note that the effect of these extra bodies on computational time is influenced by their mass. Here, the mass of the extra bodies used was  $5 \times 10^{-4}$  kg, which was 0.5% of a single vertebra (i.e., a single cylinder) in the models. Increasing the mass to a larger value, while still keeping it reasonably small relative to the body mass, can mitigate the slowdown effect.

### Availability and future directions

The source code of ArborSim is freely and anonymously accessible on GitHub, under the permissive Apache License 2.0, at <https://github.com/EMBiRLab/ArborSim>.

While the current version of the tool has proven effective by going through the procedure to create the toy models, we recognize opportunities for advancement and refinement. Several functionalities and improvements can be envisioned to extend its applicability and ease of use in the future. For example, in the current implementation of the “Transformer”, global

landmarks are required to establish joint coordinate systems, representing the axes of these systems. Identifying landmarks that can depict the axes of coordinate systems is indeed a crucial goal. However, acquiring these landmark points often necessitates identifying other biological references as a preliminary step. At present, we assume that this process is completed prior to using the designed tool. Nevertheless, integrating this preliminary step into the tool could further streamline the building process.

Moreover, the current implementation of the “**Brancher**” requires users to input anatomical and physiological parameters. Automatic parameter allocation could be realized by incorporating some predefined assumptions in future iterations of the software. Besides these, other elements such as complex coupled joints in which the spatial transformation might be dependent on other joint coordinates may also be included in future iterations of the tool to represent more complex joint movements observed in biological systems.

In the accompanying study, we conducted comparative simulations using the proposed branching modeling method and the conventional method on toy models. The results revealed large differences in the resulting movement between the models created using these two approaches. We believe that these findings will convince researchers the importance of modeling the branched muscle-tendon networks, an aspect largely neglected in existing research, and motivate further research into branching modeling. Looking forward, we aim to collect necessary material data of real muscle-tendon architectures to validate and refine the proposed branching method. We hope that our work will empower researchers to construct detailed, articulated musculoskeletal models of various systems, particularly those with multiple joints and complex muscle-tendon architectures, and study their contributions to animal motion.

It is possible to revise components, add components, or employ the proposed branching modeling method with existing OpenSim musculoskeletal models using *ArborSim*. Because one can directly access the “local” data used to build these models, one can format these local data into CSV files, which will then be inputted into the tool for model development. By revising the original parameters or appending additional data to the CSV files, one can choose to convert any existing branched muscles and tendons modeled by the conventional branching modeling method to the proposed branching modeling method.

## Supporting information

**S1 Video. Animated motion comparison: Branch number variation.** Demonstrates the differences in motion using the proposed branching modeling method versus the traditional method across models with varying numbers of tendon branches.

(MP4)

**S2 Video. Animated motion comparison: Joint count variation between insertions.** Compares motion in models with varying joint count between tendon insertions using the proposed method versus the conventional method.

(MP4)

**S3 Video. Animated motion comparison: Muscle fiber length / MTU length variation.** Compares motion in models with varying ratios of muscle fiber length to MTU length using the proposed method versus the conventional method.

(MP4)

## Acknowledgments

The authors would like to thank Brian Umberger, Swithin Razu, Pranav Embar, and Yassar Abdelrahman for helpful discussions regarding this topic.

## Author Contributions

**Conceptualization:** Xun Fu, Jack Withers, Talia Y. Moore.

**Funding acquisition:** Talia Y. Moore.

**Methodology:** Xun Fu.

**Project administration:** Talia Y. Moore.

**Resources:** Juri A. Miyamae, Talia Y. Moore.

**Software:** Xun Fu, Jack Withers.

**Validation:** Xun Fu.

**Visualization:** Xun Fu.

**Writing – original draft:** Xun Fu.

**Writing – review & editing:** Xun Fu, Jack Withers, Juri A. Miyamae, Talia Y. Moore.

## References

1. Dickinson MH, Farley CT, Full RJ, Koehl MA, Kram R, Lehman S. How animals move: an integrative view. *science*. 2000 Apr 7; 288(5463):100–6. <https://doi.org/10.1126/science.288.5463.100> PMID: [10753108](https://pubmed.ncbi.nlm.nih.gov/10753108/)
2. Grillner S, El Manira A. Current principles of motor control, with special reference to vertebrate locomotion. *Physiological reviews*. 2019 Nov 18. PMID: [31512990](https://pubmed.ncbi.nlm.nih.gov/31512990/)
3. Merel J, Botvinick M, Wayne G. Hierarchical motor control in mammals and machines. *Nature communications*. 2019 Dec 2; 10(1):5489. <https://doi.org/10.1038/s41467-019-13239-6> PMID: [31792198](https://pubmed.ncbi.nlm.nih.gov/31792198/)
4. De Groot F, Falisse A. Perspective on musculoskeletal modelling and predictive simulations of human movement to assess the neuromechanics of gait. *Proceedings of the Royal Society B*. 2021 Mar 10; 288(1946):20202432. <https://doi.org/10.1098/rspb.2020.2432> PMID: [33653141](https://pubmed.ncbi.nlm.nih.gov/33653141/)
5. Uhlrich SD, Uchida TK, Lee MR, Delp SL. Ten steps to becoming a musculoskeletal simulation expert: A half-century of progress and outlook for the future. *Journal of Biomechanics*. 2023 May 10; 111623. <https://doi.org/10.1016/j.jbiomech.2023.111623> PMID: [37210923](https://pubmed.ncbi.nlm.nih.gov/37210923/)
6. Delp SL, Loan JP, Hoy MG, Zajac FE, Topp EL, Rosen JM. An interactive graphics-based model of the lower extremity to study orthopaedic surgical procedures. *IEEE Transactions on Biomedical Engineering*. 1990 Aug; 37(8):757–67. <https://doi.org/10.1109/10.102791> PMID: [2210784](https://pubmed.ncbi.nlm.nih.gov/2210784/)
7. Gonzalez RV, Buchanan TS, Delp SL. How muscle architecture and moment arms affect wrist flexion-extension moments. *Journal of biomechanics*. 1997 Jul 1; 30(7):705–12. [https://doi.org/10.1016/S0021-9290\(97\)00015-8](https://doi.org/10.1016/S0021-9290(97)00015-8) PMID: [9239550](https://pubmed.ncbi.nlm.nih.gov/9239550/)
8. Holzbaur KR, Murray WM, Delp SL. A model of the upper extremity for simulating musculoskeletal surgery and analyzing neuromuscular control. *Annals of biomedical engineering*. 2005 Jun; 33:829–40. <https://doi.org/10.1007/s10439-005-3320-7> PMID: [16078622](https://pubmed.ncbi.nlm.nih.gov/16078622/)
9. Arnold EM, Ward SR, Lieber RL, Delp SL. A model of the lower limb for analysis of human movement. *Annals of biomedical engineering*. 2010 Feb; 38:269–79. <https://doi.org/10.1007/s10439-009-9852-5> PMID: [19957039](https://pubmed.ncbi.nlm.nih.gov/19957039/)
10. Rajagopal A, Dembia CL, DeMers MS, Delp DD, Hicks JL, Delp SL. Full-body musculoskeletal model for muscle-driven simulation of human gait. *IEEE transactions on biomedical engineering*. 2016 Jul 7; 63(10):2068–79. <https://doi.org/10.1109/TBME.2016.2586891> PMID: [27392337](https://pubmed.ncbi.nlm.nih.gov/27392337/)
11. Roos PE, Vasavada A, Zheng L, Zhou X. Neck musculoskeletal model generation through anthropometric scaling. *Plos one*. 2020 Jan 28; 15(1):e0219954. <https://doi.org/10.1371/journal.pone.0219954> PMID: [31990914](https://pubmed.ncbi.nlm.nih.gov/31990914/)
12. Schmid S, Burkhart KA, Allaire BT, Grindle D, Anderson DE. Musculoskeletal full-body models including a detailed thoracolumbar spine for children and adolescents aged 6–18 years. *Journal of biomechanics*. 2020 Mar 26; 102:109305. <https://doi.org/10.1016/j.jbiomech.2019.07.049> PMID: [31471110](https://pubmed.ncbi.nlm.nih.gov/31471110/)
13. Neptune RR, Kautz SA, Zajac FE. Contributions of the individual ankle plantar flexors to support, forward progression and swing initiation during walking. *Journal of biomechanics*. 2001 Nov 1; 34(11):1387–98. [https://doi.org/10.1016/S0021-9290\(01\)00105-1](https://doi.org/10.1016/S0021-9290(01)00105-1) PMID: [11672713](https://pubmed.ncbi.nlm.nih.gov/11672713/)

14. Fregly BJ, Reinbolt JA, Rooney KL, Mitchell KH, Chmielewski TL. Design of patient-specific gait modifications for knee osteoarthritis rehabilitation. *IEEE Transactions on Biomedical Engineering*. 2007 Aug 20; 54(9):1687–95. <https://doi.org/10.1109/TBME.2007.891934> PMID: 17867361
15. Liu MQ, Anderson FC, Schwartz MH, Delp SL. Muscle contributions to support and progression over a range of walking speeds. *Journal of biomechanics*. 2008 Nov 14; 41(15):3243–52. <https://doi.org/10.1016/j.jbiomech.2008.07.031> PMID: 18822415
16. Kuo AD, Donelan JM. Dynamic principles of gait and their clinical implications. *Physical therapy*. 2010 Feb 1; 90(2):157–74. <https://doi.org/10.2522/ptj.20090125> PMID: 20023002
17. Hoang HX, Reinbolt JA. Crouched posture maximizes ground reaction forces generated by muscles. *Gait posture*. 2012 Jul 1; 36(3):405–8. <https://doi.org/10.1016/j.gaitpost.2012.03.020> PMID: 22542242
18. Steele KM, Rozumalski A, Schwartz MH. Muscle synergies and complexity of neuromuscular control during gait in cerebral palsy. *Developmental Medicine & Child Neurology*. 2015 Dec; 57(12):1176–82. <https://doi.org/10.1111/dmnc.12826> PMID: 26084733
19. Moraud EM, Capogrosso M, Formento E, Wenger N, DiGiovanna J, Courtine G, Micera S. Mechanisms underlying the neuromodulation of spinal circuits for correcting gait and balance deficits after spinal cord injury. *Neuron*. 2016 Feb 17; 89(4):814–28. <https://doi.org/10.1016/j.neuron.2016.01.009> PMID: 26853304
20. Sharbafi MA, Rode C, Kurowski S, Scholz D, Möckel R, Radkhah K, Zhao G, Rashty AM, von Stryk O, Seyfarth A. A new biarticular actuator design facilitates control of leg function in BioBiped3. *Bioinspiration & biomimetics*. 2016 Jul 1; 11(4):046003. <https://doi.org/10.1088/1748-3190/11/4/046003> PMID: 27367459
21. Asano Y, Kozuki T, Ookubo S, Kawamura M, Nakashima S, Katayama T, Yanokura I, Hirose T, Kawaharazuka K, Makino S, Kakiuchi Y. Human mimetic musculoskeletal humanoid Kengoro toward real world physically interactive actions. In 2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids) 2016 Nov 15 (pp. 876-883). IEEE.
22. Falisse A, Serrancolí G, Dembia CL, Gillis J, Jonkers I, De Groote F. Rapid predictive simulations with complex musculoskeletal models suggest that diverse healthy and pathological human gaits can emerge from similar control strategies. *Journal of The Royal Society Interface*. 2019 Aug 30; 16(157):20190402. <https://doi.org/10.1098/rsif.2019.0402> PMID: 31431186
23. Wang K, Ren L, Qian Z, Liu J, Geng T, Ren L. Development of a 3D printed bipedal robot: towards humanoid research platform to study human musculoskeletal biomechanics. *Journal of Bionic Engineering*. 2021 Jan; 18:150–70. <https://doi.org/10.1007/s42235-021-0010-6>
24. Delp SL, Anderson FC, Arnold AS, Loan P, Habib A, John CT, Guendelman E, Thelen DG. OpenSim: open-source software to create and analyze dynamic simulations of movement. *IEEE transactions on biomedical engineering*. 2007 Oct 22; 54(11):1940–50. <https://doi.org/10.1109/TBME.2007.901024> PMID: 18018689
25. Seth A, Hicks JL, Uchida TK, Habib A, Dembia CL, Dunne JJ, Ong CF, DeMers MS, Rajagopal A, Millard M, Hamner SR. OpenSim: Simulating musculoskeletal dynamics and neuromuscular control to study human and animal movement. *PLoS computational biology*. 2018 Jul 26; 14(7):e1006223. <https://doi.org/10.1371/journal.pcbi.1006223> PMID: 30048444
26. O'Neill Matthew C., et al. "A three-dimensional musculoskeletal model of the chimpanzee (Pan troglodytes) pelvis and hind limb." *Journal of Experimental Biology* 216.19 (2013): 3709–3723. <https://doi.org/10.1242/jeb.079665> PMID: 24006347
27. Hutchinson JR, Rankin JW, Rubenson J, Rosenbluth KH, Siston RA, Delp SL. Musculoskeletal modeling of an ostrich (*Struthio camelus*) pelvic limb: influence of limb orientation on muscular capacity during locomotion. *PeerJ*. 2015 Jun 11; 3:e1001. <https://doi.org/10.7717/peerj.1001> PMID: 26082859
28. Charles JP, Cappellari O, Hutchinson JR. A dynamic simulation of musculoskeletal function in the mouse hindlimb during trotting locomotion. *Frontiers in Bioengineering and Biotechnology*. 2018 May 16; 6:61. <https://doi.org/10.3389/fbioe.2018.00061> PMID: 29868576
29. Cox SM, Easton KL, Lear MC, Marsh RL, Delp SL, Rubenson J. The interaction of compliance and activation on the force-length operating range and force generating capacity of skeletal muscle: a computational study using a guinea fowl musculoskeletal model. *Integrative Organismal Biology*. 2019; 1(1):obz022. <https://doi.org/10.1093/iob/obz022> PMID: 32510037
30. Stark H, Fischer MS, Hunt A, Young F, Quinn R, Andrada E. A three-dimensional musculoskeletal model of the dog. *Scientific reports*. 2021 May 31; 11(1):1–3. <https://doi.org/10.1038/s41598-021-90058-0> PMID: 34059703
31. Ramalingasetty ST, Danner SM, Arreguit J, Markin SN, Rodarie D, Kathe C, Courtine G, Rybak IA, Ijspeert AJ. A whole-body musculoskeletal model of the mouse. *IEEE Access*. 2021 Dec 6; 9:163861–81. <https://doi.org/10.1109/access.2021.3133078> PMID: 35211364

32. Yamada Y, Nishikawa S, Shida K, Niiyama R, Kuniyoshi Y. Neural-body coupling for emergent locomotion: A musculoskeletal quadruped robot with spinobulbar model. In 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems 2011 Sep 25 (pp. 1499-1506). IEEE.
33. Rosendo A, Nakatsu S, Narioka K, Hosoda K. Pneupard: A biomimetic musculoskeletal approach for a feline-inspired quadruped robot. In 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems 2013 Nov 3 (pp. 1452-1457). IEEE.
34. Ijspeert AJ. Amphibious and sprawling locomotion: from biology to robotics and back. *Annual Review of Control, Robotics, and Autonomous Systems*. 2020 May 3; 3:173–93. <https://doi.org/10.1146/annurev-control-091919-095731>
35. Zhang Z, Yang J, Yu H. Effect of flexible back on energy absorption during landing in cats: a biomechanical investigation. *Journal of Bionic Engineering*. 2014 Oct 1; 11(4):506–16. [https://doi.org/10.1016/S1672-6529\(14\)60063-9](https://doi.org/10.1016/S1672-6529(14)60063-9)
36. Hickman GC. The mammalian tail: a review of functions. *Mammal review*. 1979 Dec; 9(4):143–57. <https://doi.org/10.1111/j.1365-2907.1979.tb00252.x>
37. Moon BR, Gans C. Kinematics, muscular activity and propulsion in gopher snakes. *Journal of Experimental Biology*. 1998 Oct 1; 201(19):2669–84. <https://doi.org/10.1242/jeb.201.19.2669> PMID: 9732322
38. Cohn MJ, Tickle C. Developmental basis of limblessness and axial patterning in snakes. *Nature*. 1999 Jun 3; 399(6735):474–9. <https://doi.org/10.1038/20944> PMID: 10365960
39. Astley HC. The biomechanics of multi-articular muscle–tendon systems in snakes. *Integrative and Comparative Biology*. 2020 Jul 1; 60(1):140–55. <https://doi.org/10.1093/icb/icaa012> PMID: 32211841
40. Young F, Rode C, Hunt A, Quinn R. Analyzing moment arm profiles in a full-muscle rat hindlimb model. *Biomimetics*. 2019 Jan 25; 4(1):10. <https://doi.org/10.3390/biomimetics4010010> PMID: 31105196
41. Bishop PJ, Michel KB, Falisse A, Cuff AR, Allen VR, De Groot F, Hutchinson JR. Computational modelling of muscle fibre operating ranges in the hindlimb of a small ground bird (*Eudromia elegans*), with implications for modelling locomotion in extinct species. *PLoS Computational Biology*. 2021 Apr 1; 17(4):e1008843. <https://doi.org/10.1371/journal.pcbi.1008843> PMID: 33793558
42. Bishop PJ, Falisse A, De Groot F, Hutchinson JR. Predictive simulations of running gait reveal a critical dynamic role for the tail in bipedal dinosaur locomotion. *Science Advances*. 2021 Sep 22; 7(39):eabi7348. <https://doi.org/10.1126/sciadv.abi7348> PMID: 34550734
43. Van Bijlert PA, van Soest AK, Schulp AS. Natural Frequency Method: estimating the preferred walking speed of *Tyrannosaurus rex* based on tail natural frequency. *Royal Society Open Science*. 2021 Apr 21; 8(4):201441. <https://doi.org/10.1098/rsos.201441> PMID: 33996115
44. Rigby BJ, Hirai N, Spikes JD, Eyring H. The mechanical properties of rat tail tendon. *The Journal of general physiology*. 1959 Nov 1; 43(2):265–83. <https://doi.org/10.1085/jgp.43.2.265> PMID: 19873525
45. Miyamae J.A., Moore T.Y. Entails a Closer Look: Comparative muscular morphology and function of the mammalian tail. *Society for Integrative and Comparative Biology*. 2023.
46. Miyamae J.A., Moore T.Y. Telling Tails: Comparative muscular morphology and function of mammalian tails. *International Congress of Vertebrate Morphology*. 2023.
47. Lieber RL, Jacobson MD, Fazeli BM, Abrams RA, Botte MJ. Architecture of selected muscles of the arm and forearm: anatomy and implications for tendon transfer. *The Journal of hand surgery*. 1992 Sep 1; 17(5):787–98. [https://doi.org/10.1016/0363-5023\(92\)90444-T](https://doi.org/10.1016/0363-5023(92)90444-T) PMID: 1401782
48. MacIntosh AR, Keir PJ. An open-source model and solution method to predict co-contraction in the finger. *Computer Methods in Biomechanics and Biomedical Engineering*. 2017 Oct 3; 20(13):1373–81. <https://doi.org/10.1080/10255842.2017.1364732> PMID: 28817976
49. McFarland DC, Binder-Markey BI, Nichols JA, Wohlman SJ, de Bruin M, Murray WM. A Musculoskeletal Model of the Hand and Wrist Capable of Simulating Functional Tasks. *IEEE Transactions on Biomedical Engineering*. 2022 Oct 27; 70(5):1424–35. <https://doi.org/10.1109/TBME.2022.3217722>
50. Miller RM, Thunes J, Musahl V, Maiti S, Debski RE. Effects of tear size and location on predictions of supraspinatus tear propagation. *Journal of biomechanics*. 2018 Feb 8; 68:51–7. <https://doi.org/10.1016/j.jbiomech.2017.12.017> PMID: 29306551
51. Andarawis-Puri N, Ricchetti ET, Soslowsky LJ. Interaction between the supraspinatus and infraspinatus tendons: effect of anterior supraspinatus tendon full-thickness tears on infraspinatus tendon strain. *The American journal of sports medicine*. 2009 Sep; 37(9):1831–9. <https://doi.org/10.1177/0363546509334222> PMID: 19483078
52. Valente G, Crimi G, Vanella N, Schileo E, Taddei F. nmsBuilder: Freeware to create subject-specific musculoskeletal models for OpenSim. *Computer methods and programs in biomedicine*. 2017 Dec 1; 152:85–92. <https://doi.org/10.1016/j.cmpb.2017.09.012> PMID: 29054263
53. Kewley A, Beesel J, Seth A. OpenSim creator [Internet]. Zenodo; 2023. Available from: <https://zenodo.org/records/10182789>

54. Delp SL, Loan JP. A graphics-based software system to develop and analyze models of musculoskeletal structures. *Computers in biology and medicine*. 1995 Jan 1; 25(1):21–34. [https://doi.org/10.1016/0010-4825\(95\)98882-E](https://doi.org/10.1016/0010-4825(95)98882-E) PMID: 7600758
55. Delp SL, Loan JP. A computational framework for simulating and analyzing human and animal movement. *Computing in Science Engineering*. 2000 Sep; 2(5):46–55. <https://doi.org/10.1109/5992.877394>
56. Rasmussen J, Damsgaard M, Surma E, Christensen ST, De Zee M, Vondrak V. Anybody—a software system for ergonomic optimization. In *Fifth world congress on structural and multidisciplinary optimization* 2003 May 19 (Vol. 4, No. 6).
57. Fedorov A, Beichel R, Kalpathy-Cramer J, Finet J, Fillion-Robin JC, Pujol S, Bauer C, Jennings D, Fennessy F, Sonka M, Buatti J. 3D Slicer as an image computing platform for the Quantitative Imaging Network. *Magnetic resonance imaging*. 2012 Nov 1; 30(9):1323–41. <https://doi.org/10.1016/j.mri.2012.05.001> PMID: 22770690
58. Yushkevich Paul A., et al. “User-guided 3D active contour segmentation of anatomical structures: significantly improved efficiency and reliability.” *Neuroimage* 31.3 (2006): 1116–1128. <https://doi.org/10.1016/j.neuroimage.2006.01.015> PMID: 16545965
59. Cappozzo Aurelio, et al. “Position and orientation in space of bones during movement: anatomical frame definition and determination.” *Clinical biomechanics* 10.4 (1995): 171–178. [https://doi.org/10.1016/0268-0033\(95\)91394-T](https://doi.org/10.1016/0268-0033(95)91394-T) PMID: 11415549
60. Wu Ge, 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.” *Journal of biomechanics* 35.4 (2002): 543–548. [https://doi.org/10.1016/S0021-9290\(01\)00222-6](https://doi.org/10.1016/S0021-9290(01)00222-6) PMID: 11934426
61. Wu Ge, et al. “ISB recommendation on definitions of joint coordinate systems of various joints for the reporting of human joint motion—Part II: shoulder, elbow, wrist and hand.” *Journal of biomechanics* 38.5 (2005): 981–992. <https://doi.org/10.1016/j.jbiomech.2004.05.042> PMID: 15844264
62. Millard M, Uchida T, Seth A, Delp SL. Flexing computational muscle: modeling and simulation of musculotendon dynamics. *Journal of biomechanical engineering*. 2013 Feb 1; 135(2):021005. <https://doi.org/10.1115/1.4023390> PMID: 23445050
63. Scovil CY, Ronsky JL. Sensitivity of a Hill-based muscle model to perturbations in model parameters. *Journal of biomechanics*. 2006 Jan 1; 39(11):2055–63. <https://doi.org/10.1016/j.jbiomech.2005.06.005> PMID: 16084520
64. Redl C, Gfoehler M, Pandy MG. Sensitivity of muscle force estimates to variations in muscle–tendon properties. *Human movement science*. 2007 Apr 1; 26(2):306–19. <https://doi.org/10.1016/j.humov.2007.01.008> PMID: 17343945
65. Xiao M, Higginson J. Sensitivity of estimated muscle force in forward simulation of normal walking. *Journal of applied biomechanics*. 2010 May 1; 26(2):142–9. <https://doi.org/10.1123/jab.26.2.142> PMID: 20498485
66. De Groote F, Van Campen A, Jonkers I, De Schutter J. Sensitivity of dynamic simulations of gait and dynamometer experiments to hill muscle model parameters of knee flexors and extensors. *Journal of biomechanics*. 2010 Jul 20; 43(10):1876–83. <https://doi.org/10.1016/j.jbiomech.2010.03.022> PMID: 20392450
67. Ackland DC, Lin YC, Pandy MG. Sensitivity of model predictions of muscle function to changes in moment arms and muscle–tendon properties: a Monte-Carlo analysis. *Journal of biomechanics*. 2012 May 11; 45(8):1463–71. <https://doi.org/10.1016/j.jbiomech.2012.02.023> PMID: 22507351
68. Carbone V, Van der Krogt MM, Koopman HF, Verdonchot N. Sensitivity of subject-specific models to Hill muscle–tendon model parameters in simulations of gait. *Journal of biomechanics*. 2016 Jun 14; 49(9):1953–60. <https://doi.org/10.1016/j.jbiomech.2016.04.008> PMID: 27131851
69. Blache Y, Michaud B, Rogowski I, Monteil K, Begon M. Sensitivity of shoulder musculoskeletal model predictions to muscle–tendon properties. *IEEE Transactions on Biomedical Engineering*. 2018 Sep 30; 66(5):1309–17. <https://doi.org/10.1109/TBME.2018.2872833> PMID: 30281425
70. Bujalski P, Martins J, Stirling L. A Monte Carlo analysis of muscle force estimation sensitivity to muscle–tendon properties using a Hill-based muscle model. *Journal of biomechanics*. 2018 Oct 5; 79:67–77. <https://doi.org/10.1016/j.jbiomech.2018.07.045> PMID: 30146173