

Final Report: BAA-12-00118

Title: The Musculoskeletal Atlas Project (MAP): an anatomical and functional population model of the musculoskeletal system to facilitate virtual clinical trials

Contact:

Thor Besier
Associate Professor
Auckland Bioengineering Institute
The University of Auckland
70 Symonds St, Auckland 1010
New Zealand

Ph: +64 9 373 7599 ext 86952
Cell: +64 220 733 926
Email: t.besier@auckland.ac.nz



MAP
Musculoskeletal
Atlas Project

Summary

Computational models have tremendous potential to complement clinical trials, reducing the cost of bringing a new product to market through *in silico* testing across a virtual population. However, the predictive power of computational models is dependent on the ability to accurately capture the complex geometry of the musculoskeletal system and the ability to describe appropriate loads and boundary conditions (i.e. muscle and joint forces). By addressing these challenges, the Musculoskeletal Atlas Project (MAP) will make significant steps toward achieving our long-term goal of performing virtual clinical trials for medical devices.

The purpose of this work was to develop a population-based anatomical and functional atlas of the human musculoskeletal system for orthopaedic research. The key deliverable from this proposal is an open-source software platform to rapidly generate accurate, volumetric meshes of the muscles and bones of the lower limb, using statistical shape models extracted from a large imaging dataset (<https://simtk.org/home/map>). We have leveraged existing software tools developed for the *IUPS Physiome Project* to facilitate image segmentation, meshing, and data query. To facilitate exchange of data and models, accelerate validation and provide robust peer review, the MAP database has been implemented within the *Physiome Model Repository*. Using these software tools, we are actively engaging the biomechanics community to build the MAP database and provide a unique resource for population-based orthopaedic modeling and *in silico* testing and validation.

Our proposed software architecture leverages several international efforts to develop mathematical models of the human body, for clinical, research and educational purposes. This unified approach will maximize the potential end user benefits and promises to make a significant and lasting impact on the orthopaedic research community, by: enabling rapid generation of accurate musculoskeletal models; facilitating exchange of models and modeling methods; and providing access to unique, comprehensive datasets for validation. Long-term, this database will accelerate the predictive capability of computational models for orthopaedic applications. We anticipate this framework will become integral in developing a standard for virtual clinical trials.

Statement of Work Completed

We have developed an open-source modeling platform, consisting of three integrated components:

1. **MAP Client:** a desktop modeling client for 3D visualization, segmentation, and meshing. The MAP client has been developed using Python/Qt and a 'plug-in' architecture to maximize community input and flexibility. Users create their own workflow, consisting of steps to perform specific tasks. The MAP Client is also the portal for accessing the MAP Database and performing queries.
2. **MAP Database:** Models and links to raw imaging and functional data are stored within the Physiome Model Repository (*PMR*⁸), a database that has been developed in conjunction with the IUPS Physiome Project (<http://physiomeproject.org>; www.cellml.org). This database includes access and version control and facilitates the exchange of models, workflows, and steps.
3. **MAP Query Tool:** We have the capability of interrogating population models and data via the MAP Client, with a MAP Query tool. We have built search capabilities using standard anatomical ontologies.

Summary of Technical Tasks Completed

MAP Client

A1. Complete Python/Qt front end

We completed a graphical user interface for the MAP client, which is a cross-platform desktop application. Documentation for the client is available at <https://map-client.readthedocs.org/en/latest/>.

A2. Implement cmgui segmentation

We implemented segmentation features, including spline interpolation and an image re-slicing feature to enable segmentation from arbitrary imaging planes.

A3. Implement anatomical landmark ID tool

This tool enables users to identify landmarks, or key anatomical features. These features can be used to describe anatomical coordinate systems, or as reference markers for fitting and registration.

A4. Implement rigid body registration

We implemented tools to perform rigid body registration.

A5. Implement coordinate system tool

This tool enables users to describe anatomical coordinate systems from a series of points, axes, or planes.

A6. Implement face fitting via CMISS

We implemented surface or face fitting of higher order elements to customise mesh surfaces to new geometry.

A7. Implement host mesh fitting

Host-mesh fitting is a unique customization procedure that warps a template mesh to match a new set of digitized points or landmarks. This method was implemented in the MAP Client.

A8. Implement mesh export tool

Users can export surface meshes in several formats, including .stl and .inp.

A9. OpenSim export

Meshes from the MAP Client can be exported for use in OpenSim. We are working with the OpenSim developers to directly access OpenSim via their API and generate osim files.

Milestone 1. MAP Client v1.0

Source code for the MAP Client is available for download at <https://github.com/MusculoskeletalAtlasProject/mapclient>.

MAP Database

B1. Create parametric mesh of the lower limb bones

We have parametric finite element meshes of the pelvis, femur, patella, tibia and fibula.

B2. Segment lower limb bones from CT images

We segmented lower limb bones from 15 female and 15 male CT data sets.

B3. Create statistical shape model of the lower limb bones

Using the segmented images from task B2, we performed a Principal Component Analysis and created statistical shape models of the pelvis, patella, tibia, and fibula.

B4. Create an Active Shape Model of lower limb bones

Active shape models of the lower limb bones were created for the purpose of automatically segmenting further CT data.

B5. Segment and mesh remaining CT dataset

Using CT data from the Melbourne Femur Collection, we created shape models to represent a larger population of male and female bones.

Milestone 2. Population database of the lower limb bones

Our second milestone was a population database of lower limb bones, including the pelvis, femur, tibia, fibula and patella.

MAP Query

C1. Create Python bindings to Physiome Model Repository

Python bindings between the MAP Client and the Physiome Model Repository have been completed, enabling upload of models into the MAP Database.

C2. Implement web-based search tools

Basic web-services have been implemented to enable users to search the MAP Database for relevant models or data.

Milestone 3. Searchable, Web-based Query Tool v1.0

Searching via MAP Client enabled.

Milestone 4. Draft grant submission

We will use this development work and framework as a platform for further grant funding to extend the MAP database to include functional data from modeling estimates of muscle forces and joint contact loads. Realistic loading profiles across a population will be essential for testing medical devices across a range of realistic conditions. Our initial intention was to apply for NIH funds and we anticipate submitting a grant in 2015 now that we have reasonable preliminary data and a useable platform. We have also applied for further funds from the Australian Research Council through our collaborator on this project, Prof Mark Taylor at Flinders University in Adelaide.

Deliverables

1. A stand-alone, cross-platform, desktop software application (The MAP Client), capable of viewing medical image data, performing segmentation, meshing, and registration.
2. A population database of lower limb bones, including the pelvis, femur, tibia, fibula and patella.
3. A MAP Query Tool to probe the MAP Database for anthropometric measures and/or functional metrics.
4. A grant proposal to the Australian Research Council to extend the work of the MAP. Submitted a proposal to the Victorian Institute of Forensic Medicine to extend the population database.
5. Documentation on the MAP Client, MAP Database, and MAP Query Tool.
6. Peer reviewed journal paper accepted at ISBMS in Strasbourg. Second manuscript in preparation to Journal of Orthopaedic Research.
7. Conference paper presented at the 7th World Congress of Biomechanics in 2014 to raise the awareness of the Musculoskeletal Atlas Project. Please see the following link for a copy of the presentation: http://prezi.com/li9tkfp5hapn/?utm_campaign=share&utm_medium=copy
8. Following the release of MAP Client v1.0, we will host a webinar via SimTK.org to promote the use of the MAP Client to the OpenSim community.
9. Invited to present at the FDA Medical Devices Workshop in May 2015.

The MAP Client: User-friendly Musculoskeletal Modelling Workflows

Ju Zhang¹, Hugh Sorby¹, John Clement², C David L Thomas², Peter Hunter¹, Poul Nielsen¹, David Lloyd³, Mark Taylor⁴, and Thor Besier¹

¹ Auckland Bioengineering Institute, University of Auckland, Auckland, New Zealand

² Melbourne Dental School, University of Melbourne, Melbourne, Australia

³ Griffith Health Institute, Griffith University, Queensland, Australia

⁴ School of Computer Science, Engineering and Mathematics, Flinders University, Adelaide, Australia

Abstract. Subject-specific models of the musculoskeletal system are capable of accurately estimating function and loads and show promise for clinical use. However, creating subject-specific models is time-consuming and requires high levels of expertise. To address these issues, we have developed the open source Musculoskeletal Atlas Project (MAP) Client software. The MAP Client provides a user-friendly interface for creating musculoskeletal modelling workflows using community-created plug-ins. In this paper, we discuss the design of the MAP Client, its plug-in architecture and its integration with the Physiome Model Repository. We demonstrate the use of MAP Client with a subject-specific femur modelling workflow using a set of modular open source plug-ins for image segmentation, landmark prediction, model registration and customisation. Our long-term goal is to foster a community of MAP users and plug-in developers to accelerate the clinical use of computational models.

1 Introduction

Computational models play a vital role in understanding structural form-function relationships and mechanisms of injury and disease of the musculoskeletal system. This knowledge is also critical in designing, testing, and validating orthopaedic devices and implants. However, the predictive power of computational models is dependent on the ability to accurately capture the complex geometry of the musculoskeletal structures[1]. Musculoskeletal model generation is a wide and highly active field of research. Topics range from image analysis to computational geometry to statistical modelling. Implementing and using methods of such breadth and depth required a high level of expertise not to mention investment of time. In other fields, a number of projects have been successful in consolidating computational methods into standardised software libraries, e.g. InsightToolkit[6] and CGAL⁵. At a higher workflow-based level, GIMIAS[4]

⁵ Computational Geometry Algorithms Library, <http://www.cgal.org>

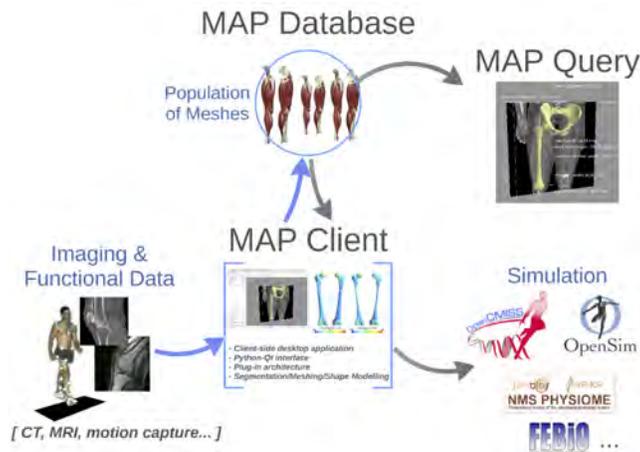


Fig. 1: MAP Framework. The MAP Client imports images and functional data and facilitates segmentation and meshing using the MAP Database population. A MAP Query tool can determine anatomical features across the population. Meshes exported from the MAP Client are compatible with various simulation environments.

focuses on cardiac modelling workflows, while MedINRIA⁶, MeVisLab⁷, and Slicer3D⁸ provide medical image visualisation and processing frameworks.

The Musculoskeletal Atlas Project (MAP) aims to establish a unified database of musculoskeletal models as well as computational methods for research and clinical application in the field of orthopaedic biomechanics. Three software components make up the MAP (Fig.1): the MAP Database, an online repository for modelling data and workflows; the MAP Query Tool, for extracting population-based metrics from the MAP Database; and the MAP Client, a model generation plug-ins based workflow software with access to the MAP Database.

This paper focuses on the open source MAP Client which enables researchers to rapidly generate accurate models of the musculoskeletal system from imaging and other data, using a toolbox of community-developed plug-ins. We outline the design of the MAP Client in the next section, explaining its handling of plug-ins and workflows, and integration with the MAP Database. Section 3 illustrates the use of the MAP Client in a real-world problem: creating patient-specific femur models from MRI and motion-capture (mo-cap) data. We explain the construction of the workflow and provide details on the key plug-ins used. Finally, the key features of the MAP Client are summarised in section 4.

⁶ <http://med.inria.fr>

⁷ <http://www.mevislab.de>

⁸ <http://www.slice.org>

2 The MAP Client

The MAP Client is a cross-platform desktop application written in Python based on the Qt widget library⁹. The application is fundamentally a framework for managing workflows. A MAP Client *Workflow* is a collection of *Steps* connected together to define a particular process. Each Step is implemented in software as a plug-in. Therefore, a MAP Client plug-in will be referred to as a Step. The MAP Client is able to describe any process using appropriate Steps, but is focused on providing those useful for processes used in the musculoskeletal and bioengineering communities.

The design of the MAP Client is set around two main objectives: ease-of-use and community engagement. These two objectives were taken into account when determining which technologies to base the MAP Client on. To meet these objectives the MAP Client is written in the Python language which has been widely adopted by the scientific community, and employs a simple plug-in architecture (see below).

The MAP Client is currently released under the GNU General Public Licence version 3 and it is available for download from <https://launchpad.net/mapclient> with documentation available at <http://map-client.readthedocs.org/en/latest/>.

2.1 Plug-in Architecture

Plug-ins lie at the heart of the MAP framework by providing Workflow Steps. The plug-in framework is lightweight, requires no external libraries, and compliant with both Python 2 and Python 3. These design choices allow Steps to be implemented as simply as possible to encourage community involvement. The plug-in interface is defined in documentation with the expectation of compliance placed on developers.

The Plug-in Wizard incorporated in the MAP Client guides the developer in defining a Step and generates a skeleton Step that conforms to the plug-in interface. It is then up to the developer to implement a small number of methods, e.g. `setPortData`, `getPortData`, and `execute`, which handle respectively the input, output, and processing of data by the Step. Any Python or Python-bound libraries can be used by the Step, while the graphical user interface (GUI) is designed using Qt widgets. Published Steps can be downloaded from <https://github.com/mapclient-plugins>.

2.2 MAP Database Integration

The MAP Database is built on the Physiome Model Repository (PMR)¹⁰ and stores musculoskeletal models, images, experimental data, and MAP Client Workflows. The PMR Tool in the MAP Client connects the application to the the

⁹ <http://qt-project.org/>

¹⁰ <http://models.physiomeproject.org>

MAP Database via PMR webservice. Using this tool one can search for and find suitable resources. The PMR Tool can also be accessed through the plug-in interface, so that Steps can retrieve or store data within the MAP Database if required.

2.3 User Interface

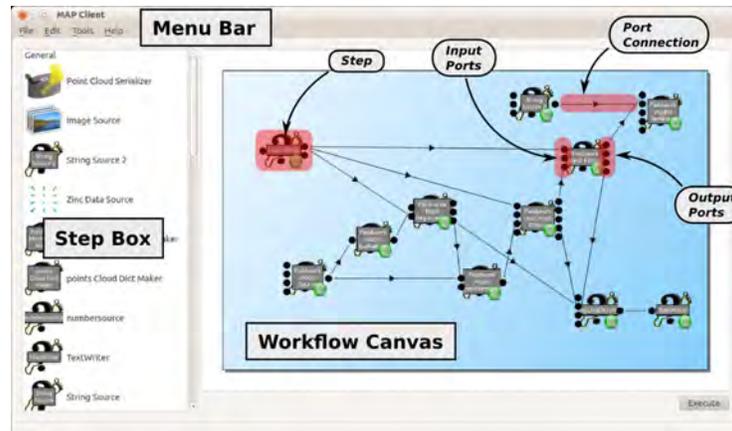


Fig. 2: MAP Client GUI. Installed Steps are listed in the *Step Box*. These can be dragged and dropped onto the *Workflow Canvas* to assemble Workflows. Steps are connected by clicking and dragging between their input and output ports.

The MAP Client GUI is composed of three areas: the *Menu Bar*, the *Step Box*, and the *Workflow Canvas* (Fig.2). The Menu Bar provides a number of drop-down menus for accessing the applications functions such as saving/loading Workflows, undo/redo, the PMR Tool, Step management tools, and accessing documentation. The Step Box shows the selection of Steps that are currently installed. The Workflow Canvas is where Workflows are constructed and edited, and where Steps can be configured.

2.4 Workflows

There are two types of Workflows that can be created: local Workflows and PMR Workflows. A local Workflow will create a project on the host computer while a PMR Workflow will create a project on the MAP Database and make it available to the host computer by creating a local copy. PMR Workflows can be downloaded by other MAP Database users.

Workflows are created by dragging and dropping Steps from the Step Box onto the Workflow Canvas. Dragging between Step ports then creates connections to define the complete Workflow. Saving a Workflow records the arrangement of its Steps and their states. Once the Workflow is saved, the Workflow

is executed by clicking the *Execute* button. The intuitive user interface enables complex Workflows to be constructed and executed by non-programmer users.

3 Example Workflow: Patient-specific Femur Modelling

To demonstrate the use of the MAP Client for a real-world problem, we present a MAP Client Workflow for generating femur models. Creating patient-specific femur models to predict *in vivo* knee joint loading is a common task in musculoskeletal modelling[2]. Femur position and general size can be reconstructed from sparse anatomical landmarks in mo-cap data. Detailed distal femur geometry can be segmented from high resolution knee magnetic resonance images (MRI). By using a statistical shape model of the femur, an atlas femur mesh can be customised to the landmarks and segmentation to reconstruct an accurate patient-specific femur model. A MAP Client Workflow has been created for the tasks above and is presented here with associated key Steps (Fig.3).

The Workflow begins by the TRC Source Step reading and outputting a set of landmarks from mo-cap data in the TRC format. The Hip Joint Centre Prediction Step then estimates the positions of the centres of the femoral heads based on the other landmarks present. The landmarks are then passed to the Landmark Registration Step where an atlas femur mesh is morphed to the femoral landmarks, according to a femur statistical shape model. On a parallel path, the Image Source Step loads an MRI stack and sends it to the Manual Segmentation Step for segmentation. The Point Cloud Registration Step aligns the segmented point cloud to the femur mesh before they are passed to the Principal Components Fitting Step which fits the femur mesh to the point cloud according to the femur shape model. Finally, the Parametric Mesh Fitting Step refines the fit by adjusting distal femur mesh control points without the constraint of the shape model. The mesh output from the Workflow is of sufficient accuracy (typically < 1.0 mm RMSE) for patient-specific modelling of knee kinematics and kinetics. The mesh can be output to in a variety of common formats using appropriate output Steps. The key Steps above are explained below. Of course, these Steps can be assembled in many other ways to accomplish other tasks.

Manual Segmentation Step This is a tool to manually create a point cloud from a stack of images. Step input is a Python object that describes the image stack, location on disk, and image format; Step output is a Python list of point cloud coordinates (x, y, z) . The Step GUI allows the user to interact with the image stack through a 3D scene and a 2D scene. The main segmentation tools are the point tool which adds single segmentation points; and the curve tool which uses splines to add a connected sequence of segmentation points. Plane orientation and position tools allow the user to translate and rotate the image plane so that segmentation can be carried out in any position.

Hip Joint Centre Prediction Step This Step implements well-known methods [5] for predicting the hip joint centre from anatomical landmarks on the

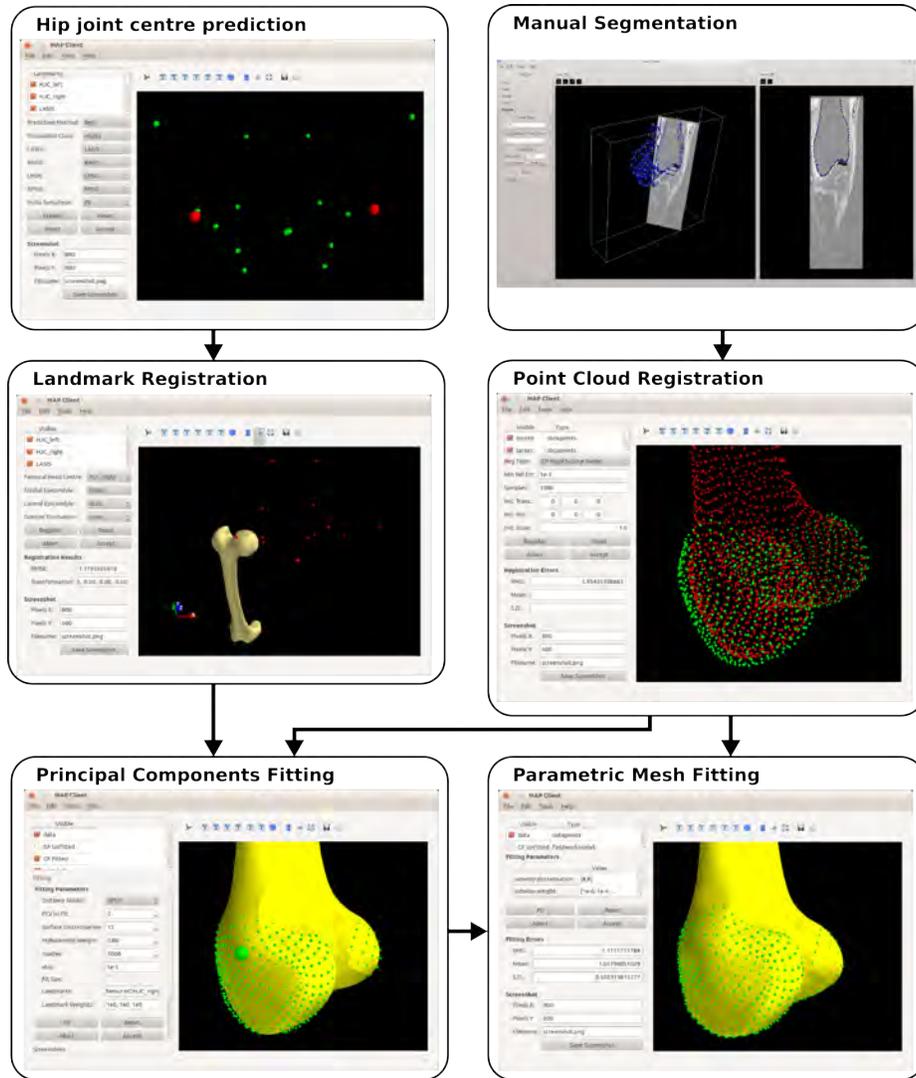


Fig. 3: Major Steps of the patient-specific femur modelling workflow. Anatomical landmarks from mo-cap data and the Hip Joint Centre Prediction Step are used to register a femur atlas mesh in the Landmark Registration Step. Point clouds from the Manual Segmentation Step are registered to the femur mesh in the Point Cloud Registration Step. The registered point clouds and atlas mesh are passed to the Principal Components Fitting Step and the Parametric Mesh Fitting Step where the mesh is fitted to the segmented data. The end product is a patient-specific femur reconstructed from mo-cap data and knee MRI.

pelvis. The Step takes a Python dictionary containing landmark name - landmark coordinates key-value pairs as input, and outputs the same with additional hip joint centre terms. The Step GUI displays the landmarks in 3-D, allows the user to select the prediction method, patient population (male, female, or adult), and the landmarks to use for prediction.

Landmark Registration Step This Step registers a 3-D atlas femur mesh to femoral landmarks. The Step uses rigid-body registration followed by non-rigid registration based on a femur shape model[7] to minimise the least-squares distance between known landmarks on the mesh and those given as input. A set of landmarks, a principal components model object (output by the PC Source Step), and a femur parametric mesh (output by the Model Source Step) are required as inputs; the registered mesh, the final transformation, and the registration error are given as outputs. The GUI displays the landmarks and femur mesh in 3-D, allows the selection of input landmarks that correspond to standard femoral landmarks, and displays the registration errors.

Point Cloud Registration Step This Step provides rigid-body and affine registration between two non-correspondent or correspondent point clouds. The Step takes as input two point clouds (source and target) and outputs the registered source point cloud along with the final transformation and the registration error. The GUI allows selection of the registration method, initial transformations, optimisation options, and displays the point clouds in 3-D.

Principal Components Fitting Step This Step fits an atlas surface mesh to a point cloud (and any provided landmarks) using rigid-body translation, rotation and deformation along its principal components[7]. The principal components ensures that the fitting is robust against noisy or incomplete point clouds. Step inputs are a point cloud, a mesh to be fitted, principal components, and optional initial transforms, data weights, and landmarks. Step outputs are the fitted mesh, final transformation, and fitting errors. The GUI displays the mesh, point cloud, and landmarks in 3-D, and allows the user to modify options regarding the fitting objective function and regularisation terms.

Parametric Mesh Fitting Step This Step provides fine-scale fitting of parametric meshes to point clouds with Sobolev regularisation[3]. The Step inputs are a parametric mesh to be fitted, a point cloud, and optional data weights. Step outputs are the fitted mesh and fitting errors. As above, the GUI allows the user to view the mesh and point cloud in 3-D, and to modify fitting options.

4 Summary

The MAP Client is an open-source workflow framework targeted toward musculoskeletal modelling. Workflows can be quickly assembled in a graphical user

interface from community created Steps (plug-ins). Once created, Workflows can be shared through the MAP Database along with data and run with minimal expert knowledge, enabling more researchers to participate in musculoskeletal modelling. As demonstrated in this paper, currently available Steps enable the creation of patient-specific lower-limb bone models from imaging and mo-cap data. Development is under way to improve core MAP Client functionalities such as pipeline flow control, the user-interface, and integration with the MAP Database. Also in development are Steps for automatic segmentation and integration with popular finite-element and kinematic software. Our long-term goal is to foster a community of MAP users and Step developers to accelerate the clinical use of computational models in musculoskeletal modelling.

Acknowledgments

This work is funded by the US Food & Drug Administration (grant HHSF22320 1310119C).

References

1. Cleather, D.J., Bull, A.M.: The development of lower limb musculoskeletal models with clinical relevance is dependent upon the fidelity of the mathematical description of the lower limb. Part 2: patient-specific geometry. *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine* 226, 133–145 (2012)
2. Fregly, B.J., Besier, T.F., Lloyd, D.G., Delp, S.L., Banks, S.A., Pandy, M.G., D’Lima, D.D.: Grand challenge competition to predict in vivo knee loads. *Journal of Orthopaedic Research* 30, 503–513 (2012)
3. Hunter, P.J., Nielsen, P.M., Smaill, B.H., LeGrice, I.J., Hunter, I.W., Others: An anatomical heart model with applications to myocardial activation and ventricular mechanics. *Critical reviews in biomedical engineering* 20(5-6), 403 (1992)
4. Larrabide, I., Omedas, P., Martelli, Y., Planes, X., Nieber, M., Moya, J.A., Butakoff, C., Sebastian, R., Camara, O., De Craene, M., Bijmens, B.H., Frangi, A.F.: GIMIAS: An open source framework for efficient development of research tools and clinical prototypes. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. vol. 5528, pp. 417–426 (2009)
5. Seidel, G.K., Marchinda, D.M., Dijkers, M., Soutas-Little, R.W.: Hip joint center location from palpable bony landmarks - A cadaver study. *Journal of Biomechanics* 28(8), 995–998 (1995)
6. Yoo, T.S., Ackerman, M.J., Lorensen, W.E., Schroeder, W., Chalana, V., Aylward, S., Metaxas, D., Whitaker, R.: Engineering and algorithm design for an image processing api: a technical report on itk-the insight toolkit. *Studies in health technology and informatics* pp. 586–592 (2002)
7. Zhang, J., Malcolm, D., Hislop-Jambrich, J., Thomas, C.D.L., Nielsen, P.M.: An anatomical region-based statistical shape model of the human femur. *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization* pp. 1–10 (Feb 2014)