ORIGINAL ARTICLE



Automated personalization of biomechanical knee model

Alexandra Yurova¹ · Alexey Lychagin² · Eugene Kalinsky² · Yuri Vassilevski^{1,2,3} · Mikhail Elizarov² · Andrey Garkavi²

Received: 13 September 2023 / Accepted: 9 February 2024 © CARS 2024

Abstract

Purpose Patient-specific biomechanical models of the knee joint can effectively aid in understanding the reasons for pathologies and improve diagnostic methods and treatment procedures. For deeper research of knee diseases, the development of biomechanical models with appropriate configurations is essential. In this study, we mainly focus on the development of a personalized biomechanical model for the investigation of knee joint pathologies related to patellar motion using automated methods.

Methods This study presents a biomechanical model created for patellar motion pathologies research and some techniques for automating the generation of the biomechanical model. To generate geometric models of bones, the U-Net neural network was adapted for 3D input datasets. The method uses the same neural network for segmentation of femur, tibia, patella and fibula. The total size of the train/validation (75/25%) dataset is 18,183 3D volumes of size $512 \times 512 \times 4$ voxels. The configuration of the biomechanical knee model proposed in the paper includes six degrees of freedom for the tibiofemoral and patellofemoral joints, lateral and medial contact surfaces for femur and tibia, and ligaments, representing, among other things, the medial and lateral stabilizers of the knee cap. The development of the personalized biomechanical model was carried out using the OpenSim software system. The automated model generation was implemented using OpenSim Python scripting commands.

Results The neural network for bones segmentation achieves mean DICE 0.9838. A biomechanical model for realistic simulation of patellar movement within the trochlear groove was proposed. Generation of personalized biomechanical models was automated.

Conclusions In this paper, we have implemented a neural network for the segmentation of 3D CT scans of the knee joint to produce a biomechanical model for the study of knee cap motion pathologies. Most stages of the generation process have been automated and can be used to generate patient-specific models.

Keywords Lateral patellar compression syndrome \cdot Knee joint \cdot Knee CT segmentation \cdot U-Net \cdot Machine learning \cdot Biomechanical knee model \cdot Patellofemoral joint

Alexandra Yurova alexandra.yurova@gmail.com

Alexey Lychagin dr.lychagin@mail.ru

Eugene Kalinsky eugene_kalinsky@mail.ru

Yuri Vassilevski yuri.vassilevski@gmail.com

Mikhail Elizarov Elizarovm07@gmail.com Andrey Garkavi garkavi_a_v@staff.sechenov.ru

- ¹ Marchuk Institute of Numerical Mathematics, Russian Academy of Sciences, 8 Gubkin Str., Moscow 119333, Russia
- ² Sechenov University, 8-2 Trubetskaya str., Moscow 119991, Russia
- ³ Center for IT&AI, Sirius University, 1 Olympiyskii pr., Sochi 354340, Russia

The knee joint is a complex joint with several articular surfaces that allow flexion and extension movements (around the frontal axis) as well as rotation [1]. It consists of two articulations: tibia and femur form the tibiofemoral joint, and patella and femur form the patellofemoral joint (Fig. 1a). The motion of the patella relative to the trochlear groove during knee flexion and extension is an important research task, as lateral displacement of the patella and increased contact pressure over the lateral patellar facet in flexion lead to patellofemoral disorders, such as lateral patellar compression syndrome. This accounts for up to 7-15% of all knee joint disorders [2]. Clinical manifestations of impaired patellofemoral biomechanics lead to significant decrease in the quality of life of patients [3]. During each step the distal edge of the patellar articular surface begins to contact the articular end of the trochlear groove at knee flexion angle $10 - 15^{\circ}$. In the seated position with knee flexion angle 90°, the proximal edge of the patellar articular surface contacts the trochlear groove as well. The contact of the patella with the femoral articular surfaces causes pain and restricts movements. The incidence rate of anterior knee pain is high, with a yearly occurrence of 22/1000 individuals [4, 5]. Females are affected approximately twice as often as males [4-6]. The causes of anterior knee pain are multifactorial and include overuse (stress) of the extensor mechanism of the shin (tendinitis, insertional tendinosis), patellar instability, chondral and osteochondral damage [7]. Patellofemoral pain syndrome (PFPS) is a common cause of anterior knee pain and predominantly affects young females without any structural changes, such as an increased Q angle (quadriceps angle) or significant pathological changes in the articular cartilage [Boling, Robinson, Fulkerson, Petersen, Alhakim]. Therefore, PFPS is a diagnosis of exclusion [8]. Other associated manifestations of anterior knee pain include crepitus and functional deficit [8]. The symptoms of PFPS force many athletes to limit their sports activities [9]. According to some authors, PFPS will ultimately lead to osteoarthritis [10–12]. The pathogenesis of PFPS is multifactorial and involves various functional impairments of the lower extremities [13]. The role of patellar tracking abnormalities in the development of PFPS has long been a controversial issue, but recent studies show that improper patellar tracking likely plays a key role. For example, Draper et al. [14] demonstrated using dynamic MRI that patients with anterior knee pain during squatting have increased lateralization and lateral deviation of the patella. Witvrouw et al. [15] showed that hypermobility of the patella has a significant correlation with the frequency of pain in the patellofemoral region. Wilson et al. [16] used a skin marker and an optoelectronic motion capture system to investigate patellar tracking in patients with PFPS in standing and squatting positions. In this study, the patella of patients with PFPS had significantly increased lateral translation (maltracking), lateral rotation and tendency toward increased lateral tilt compared to healthy subjects [16].

The patellar ligaments, including the medial patellofemoral ligament (MPFL), medial patellotibial ligament (MPTL) and lateral retinaculum, serve as essential stabilizers for the patella within the knee joint. The MPFL plays a crucial role in preventing lateral dislocation of the patella, ensuring its proper tracking during movements. The MPTL contributes to the stability of the patella by connecting it to the tibia. On the lateral side, the lateral retinaculum provides additional support by maintaining proper alignment and preventing excessive lateral movement. Dysfunction or injuries to these ligaments can lead to issues like patellar instability and affect the knee's biomechanics and overall stability.

By now, no general approach to diagnostics of patellar disorders has been proposed. Patellar kinematics is complicated and study of muscles and ligaments impact to the movement in the healthy case is necessary for its deeper understanding. Patient-specific biomechanical models of the knee joint can be used for this purpose. Biomechanical knee models [17-19] developed on the basis of OpenSim platform [20] are focused on the tibiofemoral contact and use standard bones surface meshes provided by OpenSim. Combination of standard bone geometries is often impossible in simulations, as the geometries may originate from different patients. For example, the commonly used generic musculoskeletal model Gait 2392 [21–23] contains bones surface data collected from various sources [21, 24]. Moreover, most of the existing models use a simplified representation of the knee joint and are therefore unsuitable for the study of patellofemoral kinematics (Fig. 1b). In the Gait 2392 and Gait 2354 models, the patella has been removed to avoid kinematic constraints and insertions of the quadriceps on the tibia are modeled as moving points in the tibia frame (Fig. 1c).

Nevertheless, individual surface features of bones can be crucial for the analysis of bone motion [26, 27]. For correct consideration of patellar kinematics, one has to take into account individual geometry of the bones, particularly the articular surfaces. To the best of our knowledge, open-source models developed for patellar motion and patellofemoral contact studying do not exist. In [28] we presented a biomechanical model of patellofemoral joint for analysis of knee pathologies. However, the model contains standard geometries [29]. Therefore, the personalization of geometric knee models is an extremely important step for orthopedic research. As manual bone segmentation is a time-consuming process, an automated method for segmentation of knee images is required.

Magnetic resonance imaging (MRI) and computed tomography (CT) are the most commonly used visualization tools used in diagnostics [30]. Most of the modern methods are still being developed for the segmentation of the knee anatomical



Fig. 1 Lateral view of the knee joint [25] (a), simplified representation of patellar kinematics: only saggittal plane is considered, femoral condyles are ellipses [21] (b), knee joint with removed patella in Gait 2392 model (c)

structures from MRI images [25]. The outer layer of bone (cortical bone) has an excellent representation on the CT scans, which are therefore more suitable for reconstruction of bones surfaces. This paper presents a method for automatic bone segmentation from the CT scans.

In [31], we proposed a threshold-based method for bone segmentation. Its main drawback is its high sensitivity to cases with a thin cortical layer. The results of our previous study were used for the development of a new segmentation method based on the U-Net neural network [32]. Advantages of the U-Net architecture for different segmentation problems have been demonstrated in many studies. However, it has been shown in [33], that not all U-Net-based solutions are efficient, as typical U-Net can produce high level results with thorough design of adaptive preprocessing, training scheme and inference. For the purpose of this study, the U-Net architecture was, on the one hand, simplified by reduction of the contracting path. On the other hand, it was adapted for 3D input data, batch normalization and dropout steps layers were added, the input dataset was expanded and augmented taking into account the peculiarities of knee CT scans.

The main contributions of this work are as follows: a neural network for the segmentation of femur, tibia, fibula and patella from CT scans is implemented; design of a biomechanical model for the investigation of patellar motion pathologies is proposed; automated pipeline for the generation of patient-specific biomechanical knee model is developed.

The paper is organized as follows: Sect. 2 describes the segmentation method and analyzes its results, and Sect. 3 presents the design of the biomechanical model and demonstrates the pipeline for the automated generation of patient-specific models. Finally, Sect. 4 presents the conclusions.

Geometric model of knee joint

Datasets

In this study to train, validate and test a neural network we used datasets prepared for the development of a thresholdbased bone segmentation method [31]. Fifteen knee CT datasets acquired on a Toshiba Aquillion One computed tomography scanner were provided by Sechenov University. The dimensions of the datasets were $512 \times 512 \times z$, where $z \approx 300$, and the size of each voxel is less than 1 mm. All datasets were divided into two parts: training/validation and test datasets (7 and 8 datasets, respectively). In addition, a CT dataset, acquired on the Canon CT scanner provided by Sechenov University, was used for testing.

Data preprocessing

From each CT dataset, 3D volumes with dimensions of $512 \times 512 \times 4$ voxels were extracted with a 1-voxel step size along the z-axis. For each pair of neighboring 3D volumes, there is an overlap over three slices. Specifically, for each volume with a size of $512 \times 512 \times z$ voxels z - 4 3D volumes with a size of $512 \times 512 \times 4$ were obtained.

A total of 1653 volumes were obtained. The training/validation dataset was expanded through correction of contrast (1) and gamma correction (2) (Fig. 2). It is assumed that such expansion in the training/validation datasets represents variability of real clinical data due to the use of different protocols on different scanners.

The new intensity values for CT data voxels were calculated as:

$$I_{\text{aug}} = (I - I_{\text{mean}}) * c + I_{\text{mean}}, \text{ or }$$
(1)

$$I_{\text{aug}} = \alpha * I^{\gamma}, \tag{2}$$



Fig. 2 Examples of contrast correction (top) with parameters $c \in \{1.0, 1.1, 1.2, 1.3, 1.4, 1.5\}$ and gamma correction with parameters $\gamma \in \{1.0, 1.02, 1.04, 1.06, 1.08, 1.1\}$

where *I* is the initial voxel intensity, I_{mean} is the mean intensity value of the image, α is a constant typically set to 1, $c \in \{1.1, 1.2, 1.3, 1.4, 1.5\}$, and $\gamma \in \{1.02, 1.04, 1.06, 1.08, 1.1\}$ are contrast and gamma correction factors, respectively. Increasing the contrast factor *c* results in a histogram with a wider spread of pixel values and greater separation between peaks. For gamma correction factor $\gamma > 1$, the histogram is shifted to the left and the output image is darker than the original. The parameter values for *c* and γ were chosen so that the new images keep the properties of real CT scans in terms of the distribution of intensity values between different tissues. For each image of size $512 \times 512 \times 4$, contrast adjustment and gamma correction were performed both for five factor values, i.e., 16,530 new images were generated. The total size of the train/validation (75/25%) dataset is 18183.

Network architecture

The presented network was implemented in Python 3.8 using TensorFlow. The architecture of the U-Net proposed in [32] for medical image segmentation has been modified for the segmentation of 3D knee CT scans (Fig. 3). The main modifications to the U-Net network are as follows: the incorporation of 3D datasets, integration of dropout and implementation of batch normalization.

The size of the input image, $512 \times 512 \times 4$ voxels, was chosen as the optimal size which fits to the graphics processing unit (GPU) memory and at the same time allows the neural network to perceive information in a spatial context. Batch normalization [34] and dropout [35] steps were added after each convolutional layer with a $3 \times 3 \times 3$ kernel followed by ReLU activation function. Batch normalization is a necessary step, because intensities of images after the correction of contrast and the gamma correction can differ significantly within one randomly generated batch. Dropout is used to avoid overfitting and to improve model generalization by reducing the dependence on specific neurons during training. It prevents the network from becoming specialized to the training data and makes it more robust to different inputs.

Since the max pooling operation was performed with a window size of $2 \times 2 \times 2$, the contraction path contains only two downsampling steps. The sigmoid activation was used in the output layer.

Training procedure

At the beginning of the network training procedure all input images were first normalized to zero mean and unit variance. The mean and variance required for normalization were computed using the corresponding TensorFlow function. To avoid overfitting, we randomly applied to the images basic augmentation techniques: horizontal and vertical flipping and translation. Other augmentation methods, such as cropping or scaling, were not applied, because their output is different from the standard knee CT scan. Computations were run on a GPU Tesla V100 32GB. The training was performed in 30 epochs with the batch size 2. This number of epochs was chosen because the training error stopped improving after this point. A batch size of 2 in the training was chosen due to GPU memory constraints and a consideration of computational efficiency. The Adam optimizer with a learning rate 10^{-4} was used. Dice loss was used as the loss function. The absence of significant gaps between the training and validation error curves indicated that the model generalized well to unseen data. The combination of dropout, normalization, data augmentation and appropriate model complexity effectively prevented overfitting during training.



Fig. 3 Network architecture



Fig. 4 Target areas of post-processing procedures: tomograph voxels removal (a), vessel voxels removal (b)

Inference procedure

For the inference phase, we used 8 CT datasets generated on Toshiba CT scanner and 1 CT dataset generated on Canon CT scanner provided by Sechenov University. The datasets were first divided into subsets with the size of $512 \times 512 \times 4$ voxels. After segmentation performed by the neural network, segmentations for all subvolumes were combined into segmented data corresponding to the full volume dataset. The computation time varied from 20 and 30 s depending on the *z*-size of the full volume dataset.

General post-processing techniques

Based on our experience from the development of the threshold-based method [31], we applied two simple post-

processing procedures in order to avoid labeling CT voxels with high intensity values as bone.

The voxel from a lower CT scan part belonging to the tomograph (red area in Fig. 4a) is labeled 0. Voxels corresponding to blood vessels (Fig. 4b) that were incorrectly segmented as bone can also be labeled as the background (label 0) after consistent application of two morphological operations, erosion and dilation. Erosion is a morphological operation that diminishes the boundaries of foreground objects in an image by removing voxels at the object's periphery. In contrast, dilation is an operation that expands the boundaries. The operations involve sliding a structuring element over the image and replacing each pixel with the minimum pixel value within the corresponding neighborhood defined by the structuring element. These operations

 Table 1
 DICE coefficient values for the segmentation results obtained by the network presented in Section "Network architecture" on the test dataset

Knee	Femur	Tibia	Fibula	Patella	All bones
1	0.9924	0.9921	0.9748	0.9873	0.9913
2	0.9919	0.9915	0.9775	0.9871	0.9909
3	0.9887	0.9850	0.9656	0.9808	0.9862
4	0.9934	0.9904	0.9725	0.9900	0.9913
5	0.9245	0.8423	0.8663	0.9332	0.8931
6	0.9660	0.8518	0.6813	0.8939	0.9142
7	0.9937	0.9882	0.9838	0.9893	0.9907
8	0.9911	0.9899	0.9844	0.9877	0.9901
9	0.9210	0.8826	0.8787	0.9391	0.9053
10	0.9503	0.8630	0.9045	0.8943	0.9146
11	0.9662	0.9632	0.9589	0.9682	0.9649
12	0.9624	0.9550	0.9527	0.9494	0.9586
13	0.9935	0.9900	0.9846	0.9901	0.9915
14	0.9901	0.9887	0.9843	0.9886	0.9892
15	0.9899	0.9868	0.9744	0.9879	0.9879
16	0.9887	0.9914	0.9815	0.9884	0.9895
17	0.9947	0.9902	0.9866	0.9885	0.9923
18	0.9921	0.9901	0.9824	0.9890	0.9908

can be carried out automatically, regardless of the presence of incorrectly segmented voxels.

Segmentation results

Ground truth segmentations based on the previously developed algorithm [31] indicate that bone voxels make up only 1–3% of all knee CT scan voxels. Therefore, knee bone segmentation problem is a class imbalance problem for which accuracy metric calculated as percentage of correctly segmented voxels is not informative. We used the Dice coefficient (overlap index) [36] to compare the segmentation result (S_r) generated by our model and the ground truth (S_g):

$$\text{Dice} = \frac{2 \times |S_r \cap S_g|}{|S_g| + |S_r|}.$$

The proposed algorithms were first evaluated on the test dataset consisting of 1969 volumes with the size of $512 \times 512 \times 4$ voxels. The mean value of the DICE coefficient is 0.9838.

Further, segmentations for whole CT scans were obtained and evaluated. Table 1 presents the values of the DICE coefficient measured for 9 CT datasets with both knees. Each knee joint is considered separately. The DICE coefficient values were calculated for each bone and for all bones in total. The lines corresponding to data from different knees of the same patient are placed next to each other. The first two rows of Table 1 correspond to the dataset generated on the Canon Aquillion One scanner. The segmentation quality for this dataset is comparable to the results for the datasets on which the model was trained. The worst DICE coefficient values were obtained for datasets with a thinned cortical layer. To improve the segmentation quality for such cases, one has to add corresponding examples to the training dataset.

Figure 5 demonstrates several segmentation results for the right knee. One observes differences in the individual anatomy, both in the relative size and in the location of the patella and the trochlear groove. These characteristics, together with the geometric configuration of the articular surfaces and the distance between patella and the femoral trochlear groove, may be crucial for patellar kinematics.

Biomechanical model of knee joint

In this section, we describe the process of creating a personalized biomechanical model using individual geometries. The bone surface meshes were extracted from CT data of a patient with gonarthrosis, provided by Sechenov University. The biomechanical model of the knee was created in the OpenSim software system. Some additional procedures were performed using Paraview [37], Gmesh [38] and PyMeshLab [39].

Methodology for model generation

The model contains four right leg bones: pelvis, femur, tibia, and patella (Fig. 6a). These bones are connected with joints. The model tree is presented in Fig. 6b, where rectangles correspond to bodies and arrows correspond to joints. To realistically reproduce the patellar movement within the trochlear groove, the patellofemoral joint is defined as a joint with 6 degrees of freedom. The tibiofemoral joint is also defined with 6 degrees of freedom.

The knee model contains ten ligaments, seven of them are presented in model [17] as well: patellar tendon (PT), lateral collateral ligament (LCL), medial collateral ligament (MCL), posterior cruciate ligament (PCL), posterior cruciate ligament (PCL), anterior cruciate ligament (ACL) and oblique popliteal ligament (OPL) (Fig. 7).

Since the trajectory of the patella is strongly determined by its stabilizers, the stabilizers were included in the model. The medial stabilizers of the patella are presented as medial patellofemoral ligament (MPFL, Fig. 8a) and medial patellotibial ligament (MPTL, Fig. 8b). Lateral stabilizer, lateral retinaculum (LR), is included in the model with two components: femoral and tibial. Each component contains two ligaments (Fig. 8c).

To avoid intersection of lateral and medial stabilizers and bones during knee flexion, wrapping objects were defined



Fig. 5 Examples of segmentation results for the right knee demonstrate differences in individual anatomy



Fig.6 Bones included in the model: pelvis, femur, patella and tibia (**a**), topology of biomechanical model (**b**)

(Fig. 9). Two ellipsoids were inserted in the medial and lateral condyles of the femur and one ellipsoid in the tibia condyle.

The choice of ligament parameters, such as the resting length of the ligament and the force magnitude for scaling the force–length curve, significantly impacts the trajectory of the patella. However, determining the correct patella trajectory for real patients lacks reliable methods. Furthermore, data for knee movement within the normal range show considerable variability. Figures 10 from study [40] illustrate the differences in patellar trajectories between healthy knees of different patients. The parameters in this study were chosen so that the main pattern of lateral-medial displacement during knee flexion was preserved: during flexion of up to approximately 30°, there is medial displacement, and beyond that point, there is lateral displacement.

To prevent the patella and femur from intersecting during knee flexion and to measure contact forces, a contact surface mesh must be generated and added to the model. We assume that the articulation surface has thickness of 1–2 mm. The main steps of the contact surface generation are the selection of the articulation triangles on the patellar surface mesh in the Paraview interface (Fig. 11a–c) and the construction of an additional surface with the same topology, shifted along the normal at each mesh point of the initial surface using Gmsh (Fig. 12a,b). The final step is required to make the contact mesh closed and simulate cartilage thickness. To ensure coherent orientation of all faces, we used the PyMeshab filter. Using this pipeline, we generated contact meshes for the patella, femur and tibia (Fig. 13).

Results of automated biomechanical model generation

The biomechanical model was generated using OpenSim scripting commands in Python, automating the process. The primary functions of the code are:

```
addBody(mesh)
addJoint(parent, child, joint_data)
addJointCoordinatesRot(coordinate_data)
addJointCoordinatesTrans(coordinate_data)
addLigaments(ligaments_data)
addContact(contact_data)
addWrapObjects(WrapObj_data).
```

The function addBody (mesh) takes the body's associated surface mesh as input, uses an OpenSim Python command to add the body and computes the center of mass and inertia





Fig. 8 a Medial patellofemoral ligament (MPFL), **b** medial patellotibial ligament (MPTL) and **c** Lateral retinaculum (LR)

tensor from the surface mesh using PyMeshLab [39]. This function is used to define the pelvis, femur, tibia and patella bodies. Geometries corresponding to the tibia and fibula are attached to the tibia body.

The function addJoint(parent, child, joint_ data) creates OpenSim Custom joint with 6 predefined transform axes, addJointCoordinatesRot(coordi nate_data) and addJointCoordinatesTrans (coordinate_data) set parameters for rotational and translational coordinates, respectively. All parameters for the joints are defined by model design, and additional input data are not required.

The function addLigaments (ligaments_data) adds ligaments to the model, getting coordinates of the attachment points, resting length and force magnitude for scaling the force-length curve. Some attachment points may be found using methods proposed in [41]. Attachment points of stabilizers can be chosen manually, e.g., in OpenSim interface. For adjustment of the resting length parameter, we assume that in the extended knee position all ligaments are either relaxed or slightly strained, i.e., the resting length makes up 0.95 - 1.0of the length in the extended position. The force magnitudes for the force-length curve scaling for ligaments presented in model [17] were retained. The values for the stabilizers have



Fig. 9 Lateral stabilizers, ellipsoids wrapping during knee flexion: **a** wrapping object in femur and **b** wrapping object in tibia

been chosen taking into account anatomical similarities with other ligaments and considerations of proper patellar motion. The function addContact(contact_data) adds contact geometries to the model. This function is used for including contact surfaces to the patella, medial and lateral components of the femur and tibia. The elastic foundation contact model is used in all cases. Parameters of stiffness, dissipation and friction were chosen experimentally. A contact mesh must be provided by the user. The mesh generation



Fig. 10 Patellofemoral joint movements along different axes [40]: a lateral translation, b lateral rotation and c lateral tilt



Fig. 11 a Patella surface mesh, b selected polygons of the articulation surface, c triangles of patella articulation surface



Fig. 12 a Triangles of patella articulation surface, b patella articulation surface with some thickness



shapes of the femur condyles and tibia condyle. For now, the most accurate choice of ellipsoid parameters can be achieved

using OpenSim interface.

extrude the mesh.

The pseudocode for these functions can be found in "Appendix A".

process was automated by using the Gmsh [38] procedure to

The function addWrapObjects(WrapObj_data) is used to generate three ellipsoids, which approximate the

Fig. 13 Patella articulation surface in OpenSim model

Conclusion

In this study, we implemented a neural network for segmentation of 3D CT scans of the knee joint. Reduction of the region of interest is not necessary. The described algorithms were evaluated on the CT scans of nine patients in two different ways. We applied the biomechanical model design proposed in [28] for the personalized study of knee pathologies based on patient data. The novelties of the model are six degrees of freedom of the patella movement, the presence of medial and lateral patellar stabilizers and contact surfaces for the patellofemoral and tibiofemoral joints, allowing contact forces to be measured during knee flexion. In this work, we have presented automated procedures enabling the automated generation of the biomechanical models using personalized data from arbitrary patients.

The use of personalized geometries in the biomechanical model constitutes the primary value for clinical practice. Acquiring the necessary data for personalized biomechanical model generation is a challenging task. The biomechanical model generation process can be accelerated by employing automated algorithms for detecting ligament attachment points, constructing contact surfaces and wrapping objects. For clinical use, the forward simulations need to be accelerated.

To validate the model, one can compare the simulated joint range of motion with clinical measurements obtained from motion capture systems. The adjustment of ligament properties to better align with observed joint movements can be performed.

The development and implementation of a mathematical model for personalized diagnosis of anterior knee pain hold promise for creating a universal diagnostic method. The model can be utilized for patients at risk of developing patellofemoral pain syndrome by enabling dynamic monitoring in order to take preventive measures. Additionally, the proposed model can serve as a tool for preoperative planning and assist the surgeon in determining the extent of intervention and postoperative rehabilitation regimens. By taking into account individual patient parameters, the mathematical model can be used to develop a rehabilitation protocol that emphasizes restoration of the most damaged dynamic stabilizers, thereby restoring knee joint balance for each patient. The effectiveness and relevance of this model provide hope for the development of similar systems for other mobile elements of the human musculoskeletal system.

Acknowledgements The authors would like to acknowledge Fedor Loginov for the assistance in processing personalized data.

Funding The research was funded by Russian Science Foundation Grant 21-71-30023.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Ethics approval The data were obtained retrospectively from anonymized databases and not generated intentionally for the study. For this type of study, formal consent is not required.

Informed consent There is no informed consent required for the work reported in this manuscript.

Appendix A Pseudocode for automated model generation functions

```
1. addBody(model, BodyMesh, BodyName):
      body = createBody();
      body.setName(BodyName);
      mass = computeMass(BodyMesh);
      body.setMass(mass);
      massCenter = computeMassCenter
      (BodyMesh);
      body.setMassCenter(massCenter);
      inertia = computeMassCenter
      (BodyMesh);
      body.setInertia(inertia);
      body.attachGeometry(BodyMesh);
      model.addBody(body);
2. addJoint(model, JointName, parent,
      child):
      SpatialTransform[0] =
      setRotationX();
      SpatialTransform[1] =
      setRotationY();
      SpatialTransform[2] =
      setRotationZ();
      SpatialTransform[3] =
      setTranslationX();
      SpatialTransform[4] =
      setTranslationY();
      SpatialTransform[5] =
      setTranslationZ();
      joint = createJoint(JointName,
      SpatialTransform, parent, child)
      model.addJoint(joint);
addJointCoordinatesRot(joint,
```

RangeMin, ... RangeMax, DefaultValue, DefaultSpeedValue) for i in range(0,3): coordRot[i].setRangeMin(); coordRot[i].setRangeMax(); coordRot[i].setDefaultValue(); coordRot[i].setDefault

```
SpeedValue();
4. addJointCoordinatesTrans(joint,
             RangeMin, ...
             RangeMax, DefaultValue,
             DefaultSpeedValue):
      for i in range(0,3):
          coordTrans[i].setRangeMin();
          coordTrans[i].setRangeMax();
          coordTrans[i].setDefault
           Value();
          coordTrans[i].setDefault
          SpeedValue();
5. addLigaments(model,
      LigamentsDataFileName):
      LigamentName, RestingLength,
      pcsa, origin, insertion = read
      (LigamentsDataFileName);
      ligament = createLigament();
      ligament.setName(LigamentName);
      ligament.setRestingLength
      (RestingLength);
      ligament.setPcsaForce(pcsa);
      ligament.setOrigin(origin);
      ligament.setInsertion(origin);
      model.addForce(ligament);
6. addContact(model,
   ContactDataFileName):
      ContactName, ContactMeshes,
       ContactPatemeters = ...
           read (ContactDataFileName);
      contact = createElastic
       FoundationForce();
      contact.setName(ContactName);
      contact.addGeometry
      (ContactMeshes);
      contact.setContactPatemeters
      (ContactPatemeters);
      model.addForce(contact);
7. addWrapObject(model,
       ObjectParameters):
      obj = createWrapEllipsoid();
      obj.setName(WrapObjectName);
      obj.setParameters
       (ObjectParameters);
      wrapObjectSet=model.getWrap
      ObjectSet();
      wrapObjectSet.add(obj);
```

References

1. Loudon J (2016) Biomechanics and pathomechanics of the patellofemoral joint. Int J Sports Phys Ther 11:820–830

- Tarabarko IN, Lychagin AV, Bobrov DS (2018) Status Quo: diagnostics and surgical treatment of excessive lateral pressure syndrome. Orthop Surg. https://doi.org/10.17238/ISSN2226-2016. 2018.1.46-51
- Coburn SL, Barton CJ, Filbay SR, Hart HF, Rathleff MS, Crossley KM (2018) Quality of life in individuals with patellofemoral pain: a systematic review including meta-analysis. Phys Ther Sport 33:96– 108
- Boling M, Padua D, Marshall S, Guskiewicz K, Pyne S, Beutler A (2010) Gender differences in the incidence and prevalence of patellofemoral pain syndrome. Scand J Med Sci Sports 20(5):725– 730
- Robinson RL, Nee RJ (2007) Analysis of hip strength in females seeking physical therapy treatment for unilateral patellofemoral pain syndrome. J Orthop Sports Phys Ther 37:232–238
- Fulkerson JP, Arendt EA (2000) Anterior knee pain in females. Clin Orthop Relat Res 431:69–73
- Petersen W, Ellermann A, Gösele-Koppenburg A, Best R, Rembitzki IV, Brüggemann GP, Liebau C (2014) Patellofemoral pain syndrome. Knee Surg Sports Traumatol Arthrosc 22(10):2264–74. https://doi.org/10.1007/s00167-013-2759-6
- Al-Hakim W, Kumar Jaiswal P, Khan W, Johnstone D (2012) The non-operative treatment of anterior knee pain. Open Orthop J 6(Suppl 2: M10):320–326
- Blond L, Hansen L (1998) Patellofemoral pain syndrome in athletes: a 5.7-year retrospective follow-up study of 250 athletes. Acta Orthop Belg 64:393–400
- Myer GD, Ford KR, Barber Foss KD, Goodman A, Ceasar A, Rauh MJ, Divine JG, Hewett TE (2010) The incidence and potential pathomechanics of patellofemoral pain in female athletes. Clin Biomech (Bristol, Avon) 25(7):700–707
- Thomas MJ, Wood L, Selfe J, Peat G (2010) Anterior knee pain in younger adults as a precursor to subsequent patellofemoral osteoarthritis: a systematic review. BMC Musculoskelet Disord 11:201–208
- Utting MR, Davies G, Newman JH (2005) Is anterior knee pain a predisposing factor to patellofemoral osteoarthritis? Knee 12:362– 365
- Barton CJ, Menz HB, Levinger P, Webster KE, Crossley KM (2011) Greater peak rearfoot eversion predicts foot orthoses efficacy in individuals with patellofemoral pain syndrome. Br J Sports Med 45(9):697–701
- 14. Draper CE, Besier TF, Santos JM, Jennings F, Fredericson M, Gold GE, Beaupre GS, Delp SL (2009) Using real-time MRI to quantify altered joint kinematics in subjects with patellofemoral pain and to evaluate the effects of a patellar brace or sleeve on joint motion. J Orthop Res 27(5):571–577
- Witvrouw E, Lysens R, Bellemans J, Cambier D, Vanderstraeten G (2000) Intrinsic risk factors for the development of anterior knee pain in an athletic population. A two-year prospective study. Am J Sports Med 28:480–489
- Wilson NA, Press JM, Koh JL, Hendrix RW, Zhang LQ (2009) In vivo noninvasive evaluation of abnormal patellar tracking during squatting in patients with patellofemoral pain. J Bone Joint Surg Am 91(3):558–566
- Schmitz A, Piovesan D (2016) Development of an open-source, discrete element knee model. IEEE Trans Biomed Eng 63(10):2056–67. https://doi.org/10.1109/TBME.2016.2585926
- Xu H, Bloswick D, Merryweather A (2014) An improved OpenSim gait model with multiple degrees of freedom knee joint and knee ligaments. Comput Methods Biomech Biomed Eng. https://doi. org/10.1080/10255842.2014.889689
- Bedo BLS, Catelli DS, Lamontagne M, Santiago PRP (2020) A custom musculoskeletal model for estimation of medial and lateral tibiofemoral contact forces during tasks with high knee and hip

flexions. Comput Methods Biomech Biomed Engin 23(10):658–663. https://doi.org/10.1080/10255842.2020.1757662

- Delp SL, Anderson FC, Arnold AS, Loan P, Habib A, John CT, Guendelman E, Thelen DG (2007) OpenSim: open-source software to create and analyze dynamic simulations of movement. IEEE Trans Biomed Eng 54(11):1940–50. https://doi.org/10.1109/ TBME.2007.901024
- Delp SL, Loan JP, Hoy MG, Zajac FE, Topp EL, Rosen JM (1990) An interactive graphics-based model of the lower extremity to study orthopaedic surgical procedures. IEEE Trans Biomed Eng 37:757– 767
- 22. Yamaguchi GT, Zajac FE (1989) A planar model of the knee joint to characterize the knee extensor mechanism. J Biomech 21:1–10
- Anderson FC, Pandy MG (1999) A dynamic optimization solution for vertical jumping in three dimensions. Comput Methods Biomech Biomed Eng 2:201–231
- 24. Stredney DL (1982) The representation of anatomical structures through computer animation for scientific, educational and artistic applications. Master Thesis, The Ohio State University
- Ridhma K, Manvjeet K, Sanjeev S, Devendra Ch (2021) Review of automated segmentation approaches for knee images. IET Image Process. https://doi.org/10.1049/ipr2.12045
- Powers CM (2000) Patellar kinematics, part II: the influence of the depth of the trochlear groove in subjects with and without patellofemoral pain. Phys Ther 80(10):965–78
- Dong C, Zhao C, Kong L, Piao K, Hao K, Wang F (2022) Medialization of trochlear groove was correlated with extended lateral trochlear in trochlear dysplasia: a transverse CT analysis. J Orthop Surg Res 17(1):276. https://doi.org/10.1186/s13018-022-03166-6
- Yurova AS et al An advanced patellofemoral joint biomechanical model for knee pathologies analysis. Sechenov Medical J 15 (to appear)
- Patellofemoral joint model for knee pathologies analysis. https:// simtk.org/projects/kneepatfemjoint
- Ciliberti FK, Guerrini L, Gunnarsson AE, Recenti M, Jacob D, Cangiano V, Tesfahunegn YA, Islind AS, Tortorella F, Tsirilaki M, Jónsson H Jr, Gargiulo P, Aubonnet R (2022) CT- and MRI-Based 3D Reconstruction of Knee Joint to Assess Cartilage and Bone. Diagnostics 12:279. https://doi.org/10.3390/diagnostics12020279
- Salamatova VY, Yurova AS, Vassilevski YV, Wang L (2019) Automatic segmentation algorithms and personalized geometric modelling for a human knee. Russ J Numer Anal Math Model 34(6):361–367. https://doi.org/10.1515/rnam-2019-0031
- Ronneberger O, Fischer Ph, Brox Th (2015) U-net: convolutional networks for biomedical image segmentation, medical image computing and computer-assisted intervention (MICCAI), LNCS, vol 9351. Springer, Berlin, pp 234–241

- 33. Isensee F, Jaeger PF, Kohl SAA, Petersen J, Maier-Hein KH (2021) nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation. Nat Methods 18(2):203–211. https://doi.org/10.1038/s41592-020-01008-z
- 34. Ioffe S, Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. In: Proceedings of the 32nd international conference on international conference on machine learning—Volume 37 (ICML'15). JMLR.org, pp 448–456
- 35. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 15:1929–1958
- Dice LR (1945) Measures of the amount of ecologic association between species. Ecology 26:297–302
- 37. An open-source multiple-platform application for interactive, scientific visualization. https://www.paraview.org/
- GMSH Open Source Mesh Generator. https://gmsh.info/doc/ texinfo/gmsh.html
- Muntoni A, Cignoni P (2021) PyMeshLab, Zenodo. https://doi. org/10.5281/zenodo.4438750
- Amis AA, Senavongse W, Bull AM (2006) Patellofemoral kinematics during knee flexion-extension: an in vitro study. J Orthop Res 24(12):2201–11. https://doi.org/10.1002/jor.20268
- Yurova A, Salamatova V, Lychagin A, Vassilevski Y (2022) Automatic detection of attachment sites for knee ligaments and tendons on CT images. Int J Comput Assist Radiol Surg 17(2):393–402. https://doi.org/10.1007/s11548-021-02527-6

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.