SGABU Computational Platform as a Tool for Improved Education and Research in Multiscale Modelling

Tijana Geroski[1,2,*[0000-0003-1417-0521]], Jelena Živković[1,2], Christian Hellmich[3[0000-0003-0153-4859]], Themis Exarchos[4,5[0000-0001-9420-9419]], Hans Van Oosterwyck[6[0000-0002-2142-9717]], Djordje Jakovljević[7[0000-0003-2686-6542]], Miloš Ivanović[2,8[0000-0002-8974-2267]], Nenad Filipović[1,2[0000-0001-9964-5615]]

[1] Faculty of Engineering, University of Kragujevac, Kragujevac, Serbia
[2] Bioengineering Research and Development Center (BioIRC), Kragujevac, Serbia
[3] Vienna University of Technology, Vienna, Austria
[4] University of Ioannina, Ioannina, Greece
[5] Ionian University, Corfu, Greece
[6] Katholieke Universiteit Leuven, Leuven, Belgium
[7] Coventry University, Coventry, United Kingdom
[8] Faculty of Science, University of Kragujevac
*tijanas@kg.ac.rs

Abstract. There is a need to develop an integrated computational platform that will contain both datasets and multiscale models related to bone (modelling), cancer, cardiovascular diseases, and tissue engineering. The SGABU platform is a robust information system capable of data integration, information extraction, and knowledge exchange, with the goal of designing and developing suitable computing pipelines to give accurate and adequate biological information from the patient's molecular to organ level. Datasets integrated into the platform are directly obtained from experimental and/or clinical studies and are mostly in tabular or image file format. Multiscale models range from models that can be described using partial or ordinary differential equations, to complex models that use finite element modelling. The majority of the SGABU platform's simulation modules are built as Common Workflow Language workflows. This implies creating a CWL implementation on the Functional Engine Service backend and creating an acceptable User Interface. The key advantage of SGABU platform is the utilization of new, contemporary, modular, and unique technology for various levels of architecture.

Keywords: computational platform, multiscale modelling, user-friendly interface, open science

1 Introduction

The goal of multiscale computational modeling is to link complicated networks of effects at multiple spatial and/or temporal scales. These networks, for example, frequently include intracellular molecular signaling, crosstalk, and other interactions between neighboring cell populations, as well as higher levels of emergent phenomena across

different tissues and collections of tissues or organs interacting with each other throughout the body [1]. Powerful computers and effective numerical tools make it possible to solve complex biological issues in actual geometries and in remarkable detail within short time. Globally, education and research in this field are of considerable interest. All of this motivates the development of a rapid and resilient system that integrates multiscale models from different fields to aid medical professionals in decision making, while maintaining accuracy and precision.

1.1 Literature review

Certain useful biomedical research platforms have arisen. For example, PANBioRA is a modular platform that standardizes biomaterial evaluation and enables pre-implantation, personalized diagnostics for biomaterial-based applications [2]. The SILICOFCM platform is a revolutionary cloud-based in silico clinical trial solution for the design and functional optimization of total cardiac performance, as well as monitoring the efficacy of pharmaceutical therapy [3]. The Bioengineering and Technology (BET) platform fosters innovation, advances research, and connects the greater interdisciplinary community participating in translational bioengineering [4]. It is primarily concerned with the needs of the cancer research community.

However, to the best of our knowledge, there is no platform that incorporates multiple domains and mixes models from various modeling fields. Furthermore, there are no platforms that are dedicated to both research and education at the same time. The goal of the project "Increasing Serbia's Scientific, Technological, and Innovation Capacity in the Domain of Multiscale Modeling and Medical Informatics in Biomedical Engineering (SGABU)" [5] is to create a platform that will include examples from four different fields: cardiovascular, cancer, tissue, and bone modeling. Motivated by this latest idea and aforementioned demand for having a user friendly and adaptable platform that suits the needs of the students, researchers, but also doctors and specialists, SGABU platform is proposed to integrate multiscale models and datasets from four different fields (cardiovascular, bone, tissue and cancer modelling).

2 Materials and Methods

The integration of multiscale models is performed according to standardized procedures, which enables secure delivery of all applications to anyone with an Internet-connected device and a browser. All data, analytical tools, and techniques generated under the SGABU project framework should be findable, accessible, interoperable, and reusable (FAIR principles), not only for practical reasons but also to encourage an open-science attitude. The FAIR principles serve as a guideline for data producers and researchers to ensure that their data is as interoperable as feasible. Individual tools should be standardized in their organization and interconnection. This entails packaging software using Linux container technologies like Docker or Singularity, and then coordinating workflows and pipelines with domain-specific workflow languages like WDL (Workflow Description Language) and CWL (Common Workflow Language).

We used Common Work-flow Language[1] as a specification pathway for all of our processes when developing the SGABU platform backend. It makes use of Docker containers as fundamental building pieces to give a concise explanation of any scientific procedure. In the following sections, we will demonstrate how to construct a typical CWL process using pre-existing software components.

In general, creating a CWL process consists of three separate steps:

1. Containerization,
2. Creating a CWL tool out of Docker image, and
3. Merging the tools into a CWL workflow.

Existing computer models for bone modelling, cancer modelling, cardiovascular diseases modelling and tissue engineering with patient specific databases are integrated. The main aim is to: (i) develop robust information system capable for data integration, information extraction and knowledge sharing, (ii) design and develop proper computation pipelines to provide valid and sufficient biomedical information from the molecular to organ level of the patient, (iii) produce sophisticated mathematical models to predict the progression of the diseases, their relation with the collected biological markers and, desirably, tolerance/resistance to various drug families and the existing risks to the patient. The integration process can be divided into two main subtasks:

1. Integration of the datasets,
2. Integration of the multiscale models.

Integration of the datasets. Depending on the requirements, datasets can be given in a simple form, represented as tabular data, while most of the datasets required further tuning carried out by front-end developers employing technologies such as Angular, Plotly.js, Paraview Glance, etc. Angular [6] is one of the most popular forntend development tools for creating single page dynamic apps. The SGABU platform may be utilized on any device thanks to automatic screen adjustment (responsiveness). Plotly.js [7], which is based on JavaScript, may be used to visualize certain data. The advantage over other visualization tools is the variety of graphs available, such as maps, pie charts, bar charts, bubble charts, and so on. Plotly.js has several features such as plot download as a PNG, zoom, pan etc. Datasets can also be downloaded in their original form, which is in accordance with Open Science policy [8].

Integration of the multiscale models. The majority of the SGABU platform's simulation modules are built as Common Workflow Language (CWL) workflows. This solution is an obvious choice since it makes use of Docker containerization and a standardized manner of describing inputs, outputs, and intermediate outcomes, resulting in intrinsic findability, accessibility, inter-operability, and reusability (FAIR principles). The effort involved in offering CWL-type processes is divided into two main actions: creating a CWL implementation on the FES (Functional Engine Service) backend and creating an acceptable user interface (UI). The second activity consists of developing the UI elements as proper workflow input forms with validation of numeric values,

[1] https://www.commonwl.org/user_guide/

filetypes, etc. as well as output visualization tabs with tabular views, interactive diagrams, 3D views, and animations.

3 Results and Discussion

Homepage of SGABU platform is depicted in **Fig. 1**. Each investigated field section is divided into subsections - datasets and multiscale modes. After accessing any of the modules, the help page opens automatically to point the user towards the guidelines of how to use a specific model/dataset, together with the theoretical background and references for further reading. The access to the Biomaterial corrosion module in the Bone Modelling section on the SGABU platform is provided through the main dashboard. This module will serve as an example of integration of one model on the platform.

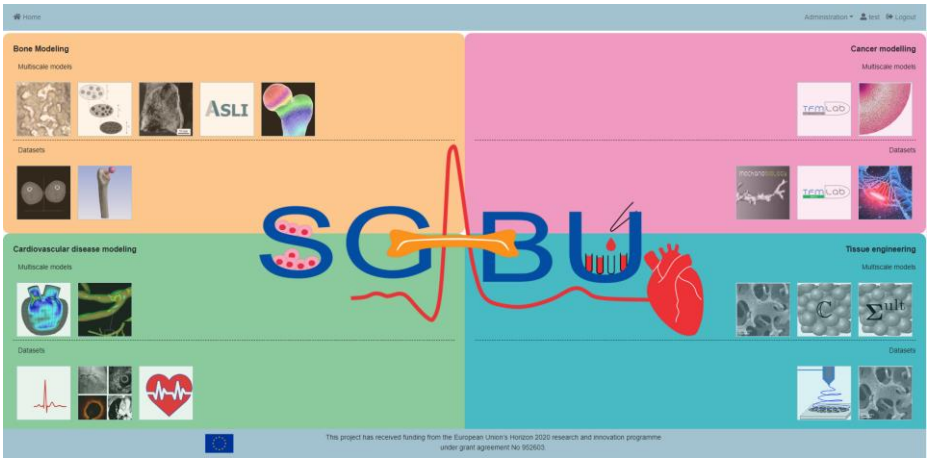


Fig. 1. SGABU platform homepage

3.1 Use case of a model – biomaterial corrosion

The model of corrosion has been developed based on Cellular Automata (CA) theory and the methodology of the model and main research results are published in [9]. In this paper we focus on the methodology for integration of one such model within SGABU platform.

The computational modelling of multi-pit corrosion in medical implants based on cellular automata (CA) is divided into two sub-models - pit initiation and growth models, where the evolution of cell CA each occurs through a series of synchronous updates of all cells, governed by a set of rules. Since the visualization of corrosion is presented in image, the state of each cell will be in an interval 0 – 255, meaning an 8-bit image (un-corroded cell has the value of 0 and totally corroded cell the value of 255). This means that we look at the surface and look at where the corrosion pit will appear (corrosion pit initiation) and how it will develop (corrosion growth model). User Interface for CorrosionPit example is shown in **Fig. 2**.

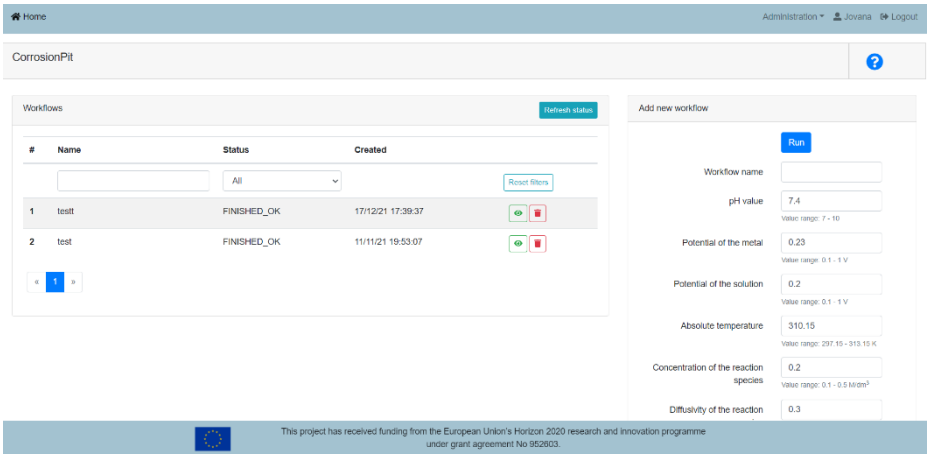


Fig. 2. User Interface for CorrosionPit module

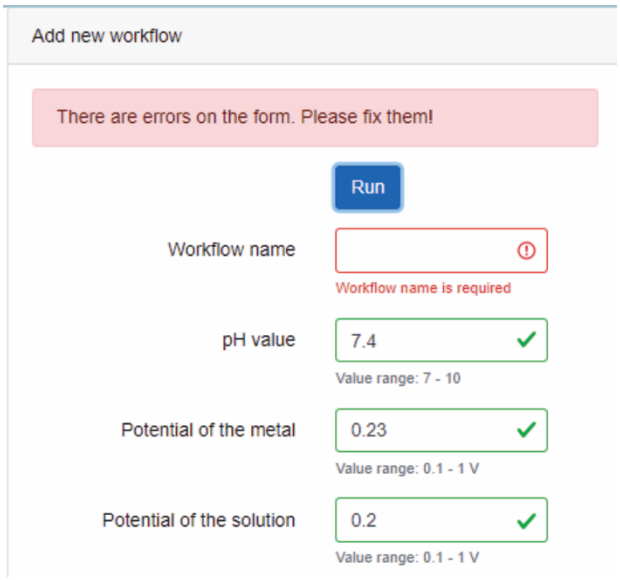
The window is divided into 2 sections. The basic execution unit in SGABU is a workflow. In the left section of the window, users are able to see names and status of the workflows. Possible statuses of the workflow are:

- Not yet executed
- Terminated
- Running
- Finished OK
- Finished Error

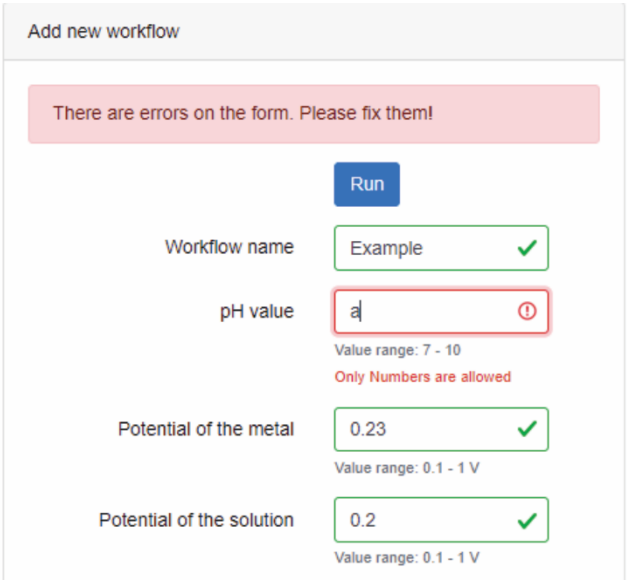
In the right section of the window, users can create new workflows for this submodule. Each of the forms needs to be filled out in order for simulation to run. Users are expected to fill out the following forms:

- Workflow name
- pH value
- Potential of the metal
- Potential of the solution
- Absolute temperature
- Concentration of the reaction species
- Diffusivity of the reaction species
- Charge of the reaction species
- Time of simulation

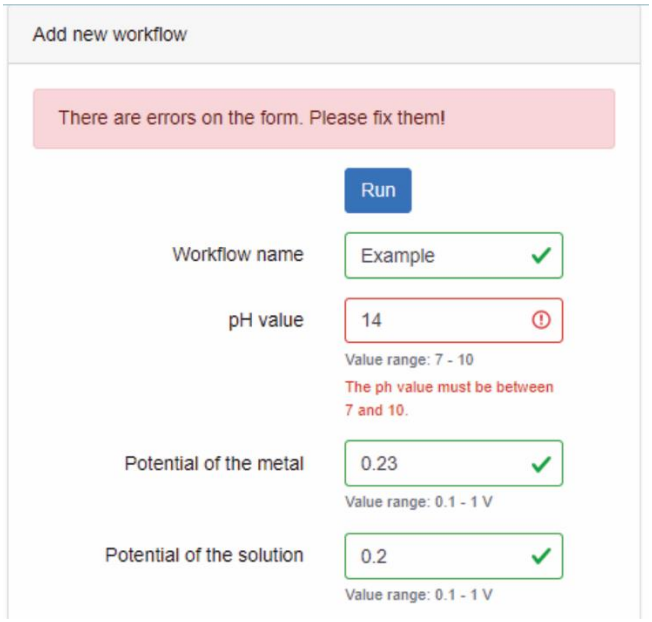
All forms except for Workflow name are numerical and value ranges are provided for the users. The exception handling is integrated into the user interface (UI) of the SGABU platform. Specific examples can be seen in **Fig. 3** (a. Empty forms, b. Non-numerical forms, c. Out of range values).



a. Exception handling – empty forms



b. Exception handling – non-numerical forms



c. Exception handling – out of range values
Fig. 3. Exception handling

Once everything is correctly filled, the workflow can be started. The user can monitor the current status of the workflow in the left section of the window (**Fig. 4**).

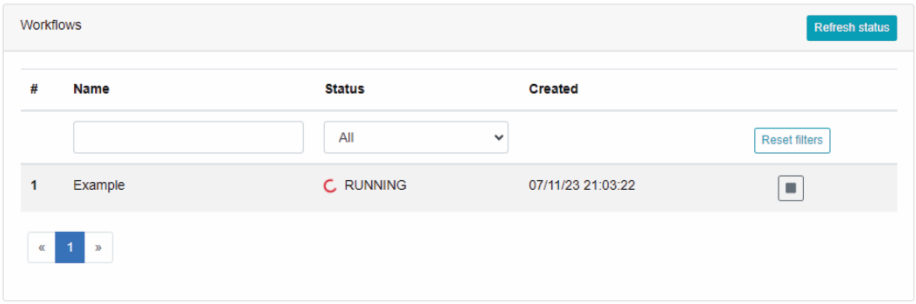


Fig. 4. Running the workflow

At the beginning we assume no corrosion, meaning the first step is the black image (all zeros). After that, in pit initiation model, each cell x is associated with an initial potential state $I(u,t) = I(u,t-1) + \alpha$ at time t. For each un-corroded cell u, initiation potential state values in its Von Neumann neighbourhood together with its own is considered $I(u,t) + \sum I(u + \delta_i, t)$ and if that sum divided by 3 overcomes the threshold value, then a corrosion pit is initiated at cell u and its corrosion state $S(u,t)$ is set to a small positive number between 3 and 5. For the next time steps, pit initiation model is

applied on the uncorroded cells again, and at the same time pit growth model is applied on the cells where corrosion has been initiated with formula:

$$S(t+1,x) = S(t,x) + k_1 f[S(t,x)] + k_2 \sum_i f[S(t,x+c_i)] + k_3 \sum_j f[S(t,x+d_j)] + k_4 \Delta \quad (1)$$

where $c_i = (0,-1),(1,0),(0,1),(-1,0)$, $d_j = (1,1),(1,-1),(-1,-1),(-1,1)$ for $i,j = 1,2,3,4$. This means that we use Moore neighborhood to describe how the surrounding cells influence the cell of interest. Coefficient k_1 is described as:

$$k_1 = \lambda \times (pH-7)^2 \times step(4,\ 8.5) \times e^{\varphi_M - \varphi_S} \times (1/T) \times C \times D \times z \quad (2)$$

In these eq. λ is a discount factor ranges from 1 to 3; pH is the pH value of the solution; step (4, 8.5) is a function with value 0 between 4 and 8.5, and 1 otherwise; φ_M and φ_S are the potentials of the metal and solution, respectively; T is the absolute temperature; C is the concentration of the reaction species; D is the diffusivity of the reaction species; z is the charge of the reaction species. The parameters k_2, k_3 and k_4 are in similar forms as k_1, but with different discount factors.

Different environmental factors are included in the model to describe their effect on the corrosion. Inputs to the corrosion pit model with allowed ranges and default values, as well as adequate unites are given in **Table 1**.

Table 1. Inputs for the corrosion pit model

Name of the parameter	Label	Description	Unit	Range	Default value
pH value	pH	pH value of the solution	/	7-10	7.4
Potential of the metal	φ_M	change in a corrosion system of the metal	V	0.1 - 1	0.23
Potential of the solution	φ_S	change in a corrosion system of the solution	V	0.1 - 1	0.2
Absolute temperature	T	Absolute temperature of the environment	K	297.15 -313.15 (24°C – 40°C)	310.15
Concentration of the reaction species	C	Concentration of one of the species participating in a corrosion reaction	M/dm^3	0.1-0.5	0.2
Diffusivity of the reaction species	D	The rate of diffusion-controlled corrosion of reaction species	m^2 /sec	0.1-0.5	0.3
Charge of the reaction species	z	Charge resulting from the reaction of species	Faraday	0.1-0.5	0.2
Time of simulation	t	Number of time steps to run the simulation	/	1000	100

Results are displayed in the form of images and analysis of the resulted images that represent the corrosion in time. The model describes adequately the multi-pit corrosion pit initiation and growth. **Fig. 5** shows the corrosion states at four different time steps, where black indicates uncorroded cells and white indicates fully corroded cells.

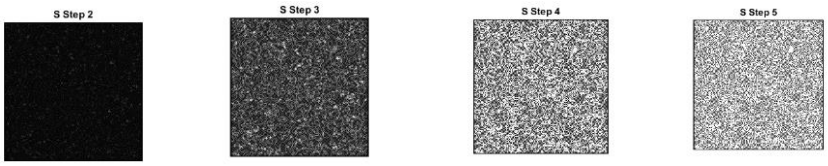


Fig. 5. Corrosion states (every 20 iterations are plotted)

After the workflow has been successfully executed, a user can visualize the results on the platform itself. The results window, includes the following sections:

- Overview of the input parameters (**Fig. 6**)
- Results in the table form (**Fig. 7**),
- Results in the form of figures, following corrosion steps in time (**Fig. 8**).

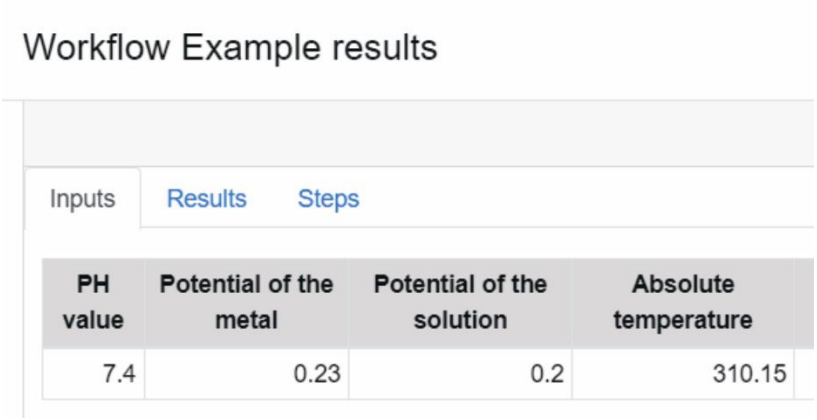


Fig. 6. Inputs section – table form

Workflow Example results

Inputs | Results | Steps

results.txt

	Step 0	Step 1	Step 2	Step 3
Mean	0	0	3.76215	10.9426
Standard deviation	0	0	3.77803	9.81423
Skew	NaN	NaN	0.132853	0.0342759
Corroded area (%)	0	0	51.55	59.305
Kurtosis	NaN	NaN	1.25303	1.54321
Energy	1	1	0.292049	0.194695
Entropy	-0	-0	2.28213	3.39195
Power	0	0	4.54831e+10	3.45691e+11
Contrast	0	0	0	0.0101508
Wavelet features (S1)	0	0	2.98982	8.70658

Fig. 7. Results section – table form

Fig. 8. Steps section - figures

The results can also be downloaded in the form of figures and txt files. In order to estimate quantitatively how the corrosion has progressed in time, we calculate statistical measures presented in textual file:

- Mean corrosion (calculated based on histogram probability as the average grey value
- Standard deviation (describes the spread in the data, which is related to the contrast)
- Skew (measures the asymmetry about the mean value in the distribution)
- Percentage of corroded material (sum of the corroded pixels (all non-zero values) divided by number of pixels and multiplied by 100)
- Kurtosis (shows whether the data are peaked or flat relative to a normal distribution)
- Energy (in the case of corrosion image, this feature indicates the degree of corrosion at the pit level)
- Entropy (entropy of each sub-band provides a measure of the image characteristics in that sub-band)
- Power (indicates the texture property in an image and in this case, the level of corrosion itself)
- Contrast (difference between maximum and minimum pixel intensity in an image)
- Wavelet features (calculated through the processes of singular values decomposition (SVD) and first two eigenvalues are reported).

4 Conclusions

There is a need to create an integrated computational platform that would include datasets as well as multiscale models relating to bone (modeling), cancer, cardiovascular disorders, and tissue engineering. SGABU platform represents the first platform of its kind that integrates both multiscale datasets and models. The main impact of the proposed methodology is the established e-infrastructure capacity and designed protocols for integrating novel multiscale solutions into a platform that will be scalable and capable of adopting new ideas and solutions, and capable of accepting research challenges. Future development would be directed towards faster calculations of the models on the platform, as well as even better user-platform interactivity.

Acknowledgement.

This research is funded by Serbian Ministry of Education, Science, and Technological Development [451-03-68/2022-14/200107 (Faculty of Engineering, University of Kragujevac)]. This research is also supported by the project that has received funding from the European Union's Horizon 2020 research and innovation programmes under grant agreement No 952603 (SGABU project). This article reflects only the author's view. The Commission is not responsible for any use that may be made of the information it contains.

References

1. Versypt A. N. F.: Multiscale modeling in disease. Current Opinion in Systems Biology 27, 100340 (2021).
2. Personalised and generalized integrated biomaterial risk assessment (PANBioRa), https://www.panbiora.eu/, last accessed 10 August 2022.
3. In Silico trials for drug tracing the effects of sarcomeric protein mutations leading to familial cardiomyopathy (SILICOFCM), https://silicofcm.eu/. last accessed 15 November 2022.
4. Bioengineering and Technology platform (BET), https://www.epfl.ch/research/facilities/ptbet/, last accessed 07 December 2022.
5. Increasing scientific, technological and innovation capacity of Serbia as a Widening country in the domain of multiscale modelling and medical informatics in biomedical engineering (SGABU), http://sgabu.eu/, last accessed 05 January 2023.
6. Seshadri, S. Angular: Up and running: Learning angular, step by step, O'Reilly Media (2018).
7. Plotly homepage, https://plotly.com/, last accessed 15 December 2022.
8. European Comission, The EU's open science policy, https://research-and-innovation.ec.europa.eu/strategy/strategy-2020-2024/our-digital-future/open-science_en, last accessed 23 January 2023.
9. Šušteršič, T., Simsek, G. M., Yapici, G. G., Nikolić, M., Vulović, R., Filipovic N. and Vrana, N. E.: An In-Silico Corrosion Model for Biomedical Applications for Coupling With In-Vitro Biocompatibility Tests for Estimation of Long-Term Effects, Frontiers in Bioengineering and Biotechnology 9, (2021).