ELECANS – An Integrated Model Development Environment for Multiscale Cancer Systems Biology

Safee Ullah Chaudhary, Sung-Young Shin, Daewon Lee, Je-Hoon Song and Kwang-Hyun Cho

Department of Bio and Brain Engineering, Korea Advanced Institute of Science and Technology (KAIST), 291 Daehak-ro, Yuseong-gu, Daejeon 305-701, Republic of Korea.

ABSTRACT

Motivation: Computational multiscale models help cancer biologists to study the spatiotemporal dynamics of complex biological systems and to reveal the underlying mechanism of emergent properties. To facilitate the construction of such models, we have developed a next generation modeling platform for cancer systems biology, termed ‘ELECANS’ (electronic cancer system). It is equipped with a graphical user interface (GUI)-based development environment for multiscale modeling along with a software development kit (SDK) such that hierarchically complex biological systems can be conveniently modeled and simulated by using the GUI/SDK combination. Associated software accessories can also help users to perform post-processing of the simulation data for visualization and further analysis. In summary, ELECANS is a new modeling platform for cancer systems biology and provides a convenient and flexible modeling and simulation environment that is particularly useful for those without an intensive programming background.

Availability and Implementation: ELECANS, its associated software accessories, demo examples, documentation and issues database are freely available at [http://sbie.kaist.ac.kr/sub_0204.php](http://sbie.kaist.ac.kr/sub_0204.php).

Contact: ckh@kaist.ac.kr

Supplementary Information: Supplementary information is available at Bioinformatics online.

1 INTRODUCTION

The development and integration of complex multiscale biological models have been a great challenge in the area of cancer systems biology (Deisboeck et al., 2011). However, the lack of a user-friendly and integrative software environment has made it difficult to develop such multiscale models and derive biological insights from them. This has been a problem particularly for experimental scientists and modelers who have a limited software development capability. So, we have developed ‘ELECANS’ (electronic cancer system), a next generation modeling and simulation platform for cancer systems biology, to overcome such hurdles in multiscale modeling and simulation. ELECANS is equipped with a rich graphical user interface (GUI) studio coupled with a powerful software development kit (SDK). This GUI/SDK combination allows users to conveniently design the extra-cellular environment, cellular phenotypes and tissue models. Cell cycle and sub-cellular components can be modeled, executed and hot-switched during simulations. The two and three dimensional discrete or hybrid simulation environments can also be constructed in ELECANS.

2 METHODS & IMPLEMENTATION

Redundant n-tier architecture was employed for the software design of ELECANS (Fig. 1A). The fully object oriented Visual C# 4.0 with Microsoft .NET Framework 4.0 was used to implement the multi-agent system core of ELECANS and the front-end GUIs. OpenTK .NET ([http://www.opentk.com](http://www.opentk.com)), Steven Fortune’s Voronoi (Fortune et al., 1987), and Lundin’s Mathparser ([http://www.lundin.com/mathparser.aspx](http://www.lundin.com/mathparser.aspx)) libraries were employed to implement 2D/3D visualization, off-lattice cell distribution, and cell geometry vertex definition, respectively. ALGLIB ([http://www.alglib.net](http://www.alglib.net)) was used for matrix functions, and an in-house partial differential equation (PDE) solver was developed using the Alternating Directions Implicit (ADI) (Chang et al., 1991) method. This PDE solver does not support mesh partitioning for utilization with multi-core systems, but users can implement their own mesh partitioning algorithms and concomitant PDE solvers using the ELECANS SDK for computational scalability.

The resulting software platform embeds a Studio with three editors: Agents Editor, Environment Editor and Simulation Editor (see section 3 for details). The core engine object is shared and synchronized among these editors. The software was extensively tested by using black and white box testing approaches and the issues list was made available online (Table S1 in Supplementary information (SI)). Release notes and an updated bug database repository are also available for download (Table S1 in SI). Simulation data can be visualized and analyzed by the accompanying Matlab toolboxes or by user developed software (Table S1 in SI).

3 KEY FEATURES OF ELECANS

ELECANS provides a modeling and simulation studio (Fig. 1B) with a user friendly GUI (Table S1 in SI) for cancer systems biology. The salient features and concepts of ELECANS are summarized below.

*To whom correspondence should be addressed.

© The Author (2013). Published by Oxford University Press. All rights reserved. For Permissions, please email: journals.permissions@oup.com
Fig. 1. ELECANS software design and simulation examples, (A) ELECANS architectural layout, (B) ELECANS Studio and Editors, (C) Illustration of various simulation scenarios and the cell environmental bio-molecular profiles that can be developed using ELECANS.

(1) **ELECANS SDK** - At the core of ELECANS platform, there is a multi-agent simulation engine for processing discretely defined cellular agents. To help construct finely detailed models, ELECANS accommodates user customization of its core ‘Engine’ via SDK API (“Elecans_base.dll”). Modelers can selectively re-program ELECANS engine by writing wrapper classes over ELECANS SDK.

(2) **Rules Editors** - The ELECANS Rules Editor helps user to define two types of rules DLLs: ‘Cell Rules’ and ‘Simulation Rules’. Cell properties including cell cycle, signaling pathways and sub-cellular organelles can be programmed in cell rules DLLs. Simulation parameters and logic can be implemented as a separate simulation rules DLL. These rules DLLs are called during simulations where the cell rules DLLs can be hot-switched seamlessly to exhibit cellular phenotypic evolution.

(3) **Tissue Geometry Editor** - ELECANS helps users to rapidly assemble tissue level models by creating cell assemblies and geometries in ‘Cell Geometry Editor’. By defining mathematical equations, users can easily create template tissue shapes. Selected cells can then be placed at the coordinates defined by the equations. The template tissues can then be implanted in the environment, at the user defined location.

(4) **Third Party .NET Bridges (C, Matlab, Python .NET, and R)** - Numerous systems biology models have already been implemented in C, Matlab, and Python. ELECANS can seamlessly access and execute these models via third party .net DLL bridges. The DLL references to these third party models can be added into the Rules Editors. Simulation parameters can also be passed and computed results can be returned (see Table S5 in SI for exemplars).

4 **RESULTS & CONCLUSION**

ELECANS can be utilized to model diverse multi-scale biological phenomena. Figure 1C provides an illustration of several simulation examples developed and deployed in ELECANS. These include oncogenesis by mitochondrial incapacitation during the cell death process (Chaudhary et al., 2011) (see Supplementary Data ‘A’ for model details and performance analysis), off-lattice cellular tissue development, various cell geometries assembled in ELECANS, hypoxic stress in epithelial tissue, visualization of a continuous description of environmental bio-molecules, and a homeostatic cross-section of colon crypt (see Table S3 in SI for a step by step implementation). Several other examples were also provided (Supplementary Table S4 in SI) to highlight specific modeling capabilities and advantages of ELECANS. We also carried out comparative analysis between ELECANS and CompuCell3D (Izaguirre et al., 2004) by assembling and simulating example models (see Supplementary Data ‘B’ for details). In conclusion, ELECANS provides a comprehensive model development environment for multi-scale cancer systems biology, and offers a novel and powerful set of conveniently exercisable features, which empower the computational systems biologists as well as the experimental biologists.

**Funding**: This work was supported by the National Research Foundation of Korea (NRF) grants funded by the Korea Ministry of Education, Science and Technology (MEST) (2009-0086964 and 2010-0017662).

**REFERENCES**


