A Modelling Framework for Vascular Tumour Growth: An Extension to the Chaste Open Source C++ Library

Anthony J Connor[1,2], James A Grogan[1], Philip K Maini[1], Helen M Byrne[1,2], and Joe Pitt-Francis[2]

[1]Wolfson Centre for Mathematical Biology, Mathematical Institute, University of Oxford, Oxford, OX2 6GG, UK.
[2]Department of Computer Science, University of Oxford, Oxford, OX1 3QD, UK.

Abstract

Chaste – Cancer, Heart And Soft Tissue Environment – is an open source C++ library for the computational simulation of mathematical models developed for physiology and biology. Code development has been driven by three applications: cardiac electrophysiology, cancer and developmental biology. The latter two applications have lead to the previous development of discrete cell modelling functionality, including cell cycling described by ordinary differential equations and chemical transport described by partial differential equations. An extension for simulating vascular tumour growth has been developed, including models of off- and on-lattice angiogenesis, blood flow and structural adaptation and solute transport in vessels. These new developments extend the ability of Chaste to multi-scale modelling of the growth and treatment of vascularized tissues, with particular focus on tumours. The software is available as an open source (Berkeley Software Distribution license) Chaste extension project, written in C++, but with Python bindings for rapid model building and exploration. Code functionality and design is over-viewed and example applications, including simulation of a well known 'snail-trail' problem and a three-dimensional, off-lattice corneal micro-pocket assay problem, are demonstrated.

Keywords: Chaste - agent-based simulation - multi-scale model - vascular tumour growth - on-lattice model - off-lattice model

AUTHOR SUMMARY:

- Insert a few bullets in here explaining major contributions of paper.

Contents

1 Introduction

There now exists a range of open-source frameworks for agent based and multi-scale modelling of soft tissues, including Chaste [], OpenCMISS [] and CompuCell3D []. These software do not yet have functionality for modelling tissue with growing vasculature, as is of interest when modelling many tumours. Although there are many mathematical models of vascular tumour growth [][][], with an increasing number being released in the form of bespoke, open source software [][], a well documented and tested framework in which a range of models can be quickly built and predictions cross-compared does not yet exist.

The benefits of flexible and extensible software development in discrete soft tissue modelling are well known []. Modularity is of particular consequence for multi-scale models of vascular tumour growth. These models are usually comprised of several interacting sub-models, which include the effects of blood flow, cell cycling, nutrient transport and angiogenesis []. There are many ways to formulate the sub-models, incorporating different biophysical details [] and many questions remain regarding the sensitivity of overall model predictions to the chosen combination of sub-models []. Indeed, it has been noted that cross-comparing the predictions of the many different models of vascular tumour growth is an important, and challenging, step for the field of tumour modelling as a whole [].

The aim of the current project is to develop a modular framework for constructing and cross-comparing models of vascular tumour growth. A balance of performance, necessary when solving coupled problems over multiple size-scales, and the ability to readily construct models when exploring large model spaces, is achieved by using C++ code with Python bindings. Python binding of the code has three further, significant, advantages: i) users can quickly modify much of the functionality of a model in Python, without the need for recompilation, through Python-C++ inheritance and 'modifier' patterns, ii) external finite element solvers such as FEniCS and Abaqus can be readily incorporated, allowing challenging multi-physics and mechanics problems to be solved as sub-models of the tumour growth model, iii) use of different mark-up languages to describe models, for example SBML for the cell cycle, is straight-forward thanks to the availability of many Python tools []. The project is closely integrated with the Chaste C++ library for discrete cell modelling, sharing a new CMAKE based build-system, testing and documentation practices [] and has a subset of the dependencies, namely VTK for input, output and geometry calculations, PETSc and Boost uBLAS for vector and matrix operations and CVODE for solving systems of ordinary differential equations.

Thanks to the aforementioned developments, and integration with Chaste, the models of vascular tumour growth included in this release, and presented as sample applications, represent the state-of-the-art in the field. Presented applications range from well known snail-trail models [], allowing comparison with results of previous studies, through to three dimensional off-lattice angiogenesis models on complex geometries with mechanical interaction between discrete cells.

The structure of the report is now over-viewed. In Section 2 code design and implementation details are introduced. The general design of Chaste is briefly summarized, however the reader should refer to [] for further details. In Section 3 sample applications, including typical angiogenesis and vascular tumour growth problems are demonstrated, solved with a variety of solution techniques. While the focus is on cell centred based vascular tumour growth, it is noted that the framework is widely applicable to vascular tissues, and can be used with other functionality in Chaste, including vertex and Cellular Potts type simulations. In Section 4 code availability, future directions and other potential applications are discussed.

2 Design and Implementation

Chaste is a C++ library with code that can be used to build a range of models in the area of computational biology. The code is over-viewed in detail in several publications [][][]. In brief, the core components are:

- global - basic mathematics, check-pointing and time stepping
- io - reading and writing of various file formats, including HDF5
- mesh - handling linear and quadratic meshes, including reading and writing
- linalg - linear algebra, including PETSc and uBLAS wrappers
- ode - ODE system solvers, including CVODE interface
- pde - PDE solvers using the finite element method
- continuum mechanics - non-linear elasticity problems

Beyond the core components further code is specialized for discrete cell problems and cardiac electrophysiology problems. The code for modelling discrete cell problems is further divided into []:

- cell - cell description, cell cycle models and mutation state definitions
- population - collections of cells described using cellular automaton, mesh based, node based, vertex based and cellular potts frameworks
- simulation - tissue growth using lattice and off lattice approaches
- writers - solution output to ascii and VTK files

Chaste can be extended by add-on user projects, which can share the same dependencies and build and test frameworks []. The vascular tumour growth functionality is implemented as such an add-on project, but with additional functionality implemented through Python bindings. As such the code is divided into a C++ framework, which follows a similar design to cell-based Chaste framework [] and a Python package, which encompasses most of the C++ functionality, but with easier integration with other solvers and libraries.

The primary C++ components are:

- vessel - vessel network reading, writing, analysis and management
- geometry - simple geometric descriptions and operations
- mesh - finite element meshing and regular grid tools
- pde - extensions to core PDE solvers with finite difference and semi-analytical methods

- simulation - solvers for flow, angiogenesis and structural adaptation. Modifiers for integration with cell based Chaste.

The code heavily uses Chaste core functionality, but is only loosely integrated with cell-based functionality through a 'simulation modifier' pattern. This pattern allows loose coupling during simulations between discrete representations of vessels and cells, through the solution of PDEs in which either agent can be a discrete sinks or sources or through simple models of mechanical contact. The new framework is developed with Python binding in mind. This leads to extensive use of shared pointers and factory constructor methods and reduction in the use of templating relative to the remainder of the Chaste code. Unrelated to Python binding, there is also limited parallelism and checkpointing functionality at present.

The Python framework is available as a standard Python package []. Due to fundamental differences between C++ and Python in regard to templating and memory management it does not encompass all of the functionality of the C++ framework. In particular all simulations assume a spatial dimension of 3 (i.e. 3D) and some extra wrappers are needed to generate meshes, cell populations and simulation instances. Extra functionality is implemented in Python, giving the following additional components:

- geometry - CAD operations and centreline extraction
- image - image processing and segmentation
- interface - interfaces with external finite element solvers
- plot - two dimensional and three dimensional post-processing and visualization
- gui - graphical user interface for model pre-processing steps

Both C++ and Python components are subject to unit-testing are documented and are accompanied with a collection of tutorials. Functionality with application in the solution of vascular tumour growth problems is demonstrated in the next section.

3 Results and exemplar simulations

3.1 Avascular tumour spheroid growth

3.1.1 On-lattice

3.1.2 Off-lattice

3.2 Vascular tumour growth

3.2.1 On-lattice

3.2.2 Off-lattice

3.3 An off-lattice model of corneal angiogenesis on a complex domain

4 Discussion and future work