**Time-dependent Green’s function method - C version (instructions and download)**

(currently using CUDA 8.0)

In the Green’s function method for simulating solute transport from a network of vessels to a finite volume of tissue, vessels and tissue are treated as distributions of sources of sinks. The time-dependent version of this method is described in: Secomb, T.W. A Green's function method for simulation of time-dependent solute transport and reaction in realistic microvascular geometries. Mathematical Medicine and Biology 33:475-494 (2016).

All necessary files, including C source code and sample data and output files are included in a zip file:  
[GreensTD Version 1](http://physiology.arizona.edu/sites/default/files/greens16td.zip_.txt) (remove \_.txt from the file name and then unzip).

A version for GPU (graphical processing unit) computers using CUDA is also available in a zip file:  
[GreensTDGPU Version 1](http://physiology.arizona.edu/sites/default/files/greens16tdgpu.zip_.txt) (remove \_.txt from the file name and then unzip).

**Notes**

1. This method uses an “infinite-domain solution,” in which the network of vessels and the associated tissue domain are effectively embedded in an infinite space with zero net exchange of solute across the tissue domain boundaries. Such an approach has two advantages: it is applicable to tissue domains of arbitrary shape, and it avoids artifacts that can occur when specific boundary conditions, such as the no-flux condition, are imposed.

2. This method can be used for multiple reacting solutes. It allows for diffusible and non-diffusible solutes. Diffusible solutes may be specified as permeating vessel walls or non-permeating. Oxygen is handled differently from other solutes to account for binding by hemoglobin. Intravascular resistance to solute transport can be represented as a function of vessel diameter, as is the case for oxygen. For other solutes, the permeability of the vessel wall to the solute can be specified.

3. The network of vessels is approximated by a set of straight uniform segments located inside a reference cuboidal (box) shape. This cuboid must include all tissue regions associated with the given network. Each segment is then divided into sub-segments with equal length. The midpoint of each sub-segment represents a source. The tissue domain is divided into small subregions, each centered on a tissue node point. The tissue node points form a three dimension matrix. Inside each subregion, the solute reaction rates are assumed to be uniform, and depend on the solute levels at the tissue node.

4. The tissue node points within the tissue domain are determined by the function outboun.cpp, which is called from analyzenet.cpp with argument 1 or 2 to specify the method. Method 1 finds the smallest convex region inside the cuboid which excludes tissue node points that have a distance to the nearest vessel greater than a value specified by the user (Lb in network.dat). Method 2 finds all tissue points within a distance lb of the vessels. The resulting region is the simulated tissue domain, which is embedded in an infinite domain. The whole cuboidal region can be included in the tissue domain by making Lb large enough.

5. Sample input files are placed in three folders:  
**CremasterWashout**. This example gives the results shown in Figure 3 of Secomb (2016) for washout from tissue of an inert tracer.  
**Cremaster\_eq\_cylinder**. This example runs a sequence of cases to generate the data labelled “Greens equivalent cylinder” in Figure 5 of Secomb (2016). This is a Krogh-cylinder type model with equivalent vascular density and perfusion to that of the cremaster network. Results are also for washout of an inert tracer. Note that a change to input.cpp is needed to run this case: seeREADME.txt.  
**Tumor1998\_2solute**. This example includes two reacting solutes.

6. Input files must be placed in the same working directory where the C code resides. They can be copied from the folders containing sample problems. The following input files are used:  
**DefineRuns.dat**. The program provides the capability to specify multiple runs with different parameter values, by setting up multiple input files. The following integer values are specified:  
n\_diff = number of "SoluteParamsx.dat" files  
n\_perm = number of "IntravascResx.dat" files (starting with x = 0, x <= 10)  
n\_step = number of "TimeDepx.dat" files (starting with x = 0, x <= 10)  
In each case, x starts with x = 0, x <= 10. All combinations of these files are then run, i.e. the number of runs is nrun = n\_diff\*n\_perm\*n\_step.  
**IntravascRes0.dat, IntravascRes1.dat, etc**. These files contain data on the resistance of the vessel wall or intravascular resistance to transport from blood to tissue. In the case of oxygen, this refers to intravascular resistance, which is expressed in terms of a mass transfer coefficient gamma1. This is strongly diameter dependent, and is obtained by interpolating values given at a set of specific diameter values. For other solutes that permeate the wall, the resistance is expressed as the inverse of the wall permeability P in micron/s.  
**Network.dat**. This file specifies the network structure, the flow rate and hematocrit of all segments and the reference solute concentrations of boundary nodes. (Note that this value is used only for inflow nodes.) To simulate time-dependent plasma concentrations, these values can be modulated by data given in TimeDep0.dat.  
**SoluteParams0.dat, SoluteParams1.dat, etc**. These files give solute transport and reaction parameters. Oxygen is handled differently than other solutes to account for binding by hemoglobin. This file contains oxygen transport parameters, which are needed only if the “oxygen” parameter is set to 1.  
**TimeDep0.dat, TimeDep1.dat, etc**. These files control the number and size of the time steps used in the calculation. Within each run, the time step can be varied. For example, it may give better results to start the calculation with small time steps to show initial transient behaviors, and then switch to longer time steps for the approach to equilibrium. The “solute information” parameter has three possible values: 0, the inflowing concentration is zero; 1, the inflowing concentration is as specified in the last section of network.dat. 2, the inflow concentration is modulated relative to the value specified in network.dat by a factor that is specified for each time step in the following section of the TimeDep.dat file.  
**tissrate.cpp.dat**. This is a segment of C code supplied by the user that gives the rates of production mtiss[isp] of solute number isp in the tissue as a function of the solute concentrations c[1], c[2], …, c[nsp]. Also, the derivatives mptiss[isp] of mtiss[isp] with respect to c[isp] must be specified in this code. Caution: If this file is modified, the project must be rebuilt otherwise the old version will remain in the compiled code.  
**ContourParams.dat**. At each time point, the program generates a postscript file showing the vessel network projected onto a single plane and the solute contours on that plane. If there are multiple solutes, the plots are generated for each plot as separate pages within the postscript file. The position and orientation of the plane are specified by ContourParams.dat, giving the coordinates of three corners of the plane. Also, the contour levels are specified in this file.

7. The program generates several output files giving results in numerical form. It generates a folder called "Current" in the working folder (if it does not already exist), and generates postscript files showing the vessel network and the contours of each solute, in the "Current" folder. They can be viewed using a postscript viewer, such as GSview [http://pages.cs.wisc.edu/~ghost/gsview/get50.htm (link is external)](http://pages.cs.wisc.edu/~ghost/gsview/get50.htm). Also, a series of files "GreensWashoutnn.txt" is produced giving the time-dependent values of the inflowing and outflowing concentrations of each solute, together with its "ambient" concentration. These washout data are fitted to an exponential function to obtain the characteritic decay rates (lambda), which are summarized in "WashoutRate.txt." Further output piles are saved only for the final time point: VesselSources.out, VesselLevels.out, TissueSources.out, TissueLevels.out, greens.exelem, greens.exenode. The last two files can be used to obtain 3D visualizations of the network using [CMGUI (link is external)](http://sourceforge.net/projects/cmiss/files/cmgui/cmgui-wx-2.8.0/).

8. We have tested this package using Microsoft Visual C++ 2010 under Windows 7. For instructions on setting up a new project with this compiler, please see [NewProject.txt](http://physiology.arizona.edu/sites/default/files/newproject_1.txt). For error reporting and suggestions please contact Dr. Timothy W. Secomb, (520) 626-4513, email [secomb@u.arizona.edu](mailto:secomb@u.arizona.edu). We welcome your comments and suggestions.

9. This program is freely available for non-commercial use, provided appropriate acknowledgement is given. Commercial users please contact us before using this program. No assurance is given that it is free of errors and any use is at the user’s risk.

Updated 20 September 2017