**(READ ME) MATLAB code for pulmonary vascular network generation and V/Q multi-scale model**

All code in this repository are made for scientific research purposes and come with no guarantee of working. If you have and questions/comments/suggestions, please contact Andrew (Drew) Marquis at [admarqui@umich.edu](mailto:admarqui@umich.edu) . These codes encapsulate a multi-scale model of ventilation-perfusion matching in rat lungs. The model consists of realistic vascular networks, hemodynamics within the blood vessel segments, oxygen transport, and regulation by the physiological mechanism of hypoxic pulmonary vasoconstriction (HPV).

Many of the .mat files loaded and generated by this code are huge and exceed the 100 MB limits of github. Please go to [ <https://drive.google.com/drive/folders/11EaidR1VLS8ov2u1Ca5Yz65ln3uiEMQ-?usp=sharing> ] to download them and place them into the root directory (VQmodel)

***Directory Structure***

**VQmodel** –parent directory for all code. Code in this directory contains the multi-scale V/Q matching model. The model is formulated as a system of ODEs. Relevant scripts solve the model and generate plots and other visualizations from model simulations

**Cheng\_etal\_raw\_data** – This script in this directory processes data from Cheng et al. into a .mat file that can be accessed by VQmodel. Data from this paper were used to validate the model and act as boundary conditions (model inputs)

Cheng, H. Y., Croft, Q. P., Frise, M. C., Talbot, N. P., Petousi, N., Robbins, P. A., & Dorrington, K. L. (2017). Human hypoxic pulmonary vasoconstriction is unaltered by 8 h of preceding isocapnic hyperoxia. Physiological reports, 5(17), e13396.

**VascNetwork** – Code in this directory agnostically generates vascular network topologies, and ascribes geometric information to each vessel (length/radius/diameter-defined Strahler order). This directory also contains the “Polymesher” directory

**PolyMesher** – finite element 2D polygonal mesher. Code in this directory is used to generate seed points for vascular networks. Refer to the following citation for how PolyMesher works

Talischi, C., Paulino, G. H., Pereira, A., & Menezes, I. F. (2012). PolyMesher: a general- purpose mesh generator for polygonal elements written in Matlab. Structural and Multidisciplinary Optimization, 45(3), 309-328.

The general workflow is to first generate a spatial mesh with PolyMesher. The ‘’addpath” command is used in VascNetwork to have access to the .mat files made by PolyMesher. Scripts in VascNetwork are used to generate a network topology/geometry. “addpath” is again used in the runfiles of VQmodel to have access .mat files made in VascNetwork.

***VascNetwork***

**network\_run.m** – runfile to generate vascular networks. This script needs access to the “PolyMesher” directory. There are a few options you can adjust at the top of the file – “FLAG” lets you select which mesh you would like to use, “display” lets you configure how often you want the network generation algorithm to print it’s progress to the console, and “savB” is a boolean used to turn on/off automatically saving the results of network generation. This script organizes the vascular network into data structures saves to .mat files that can be loaded by runfiles in VQmodel to run simulations.

**NET\_GEN.m** – Function called in network\_run.m. Generates the topology of the vascular network based on user defined settings. Outputs data structures of the network connectivity and spatial coordinates.

**ANATOMY\_STATS.m** – Function called in network\_run.m. Takes the output from NET\_GEN.m as inputs and computes anatomical information for the vascular network. Namely, it calculates the radius of each vessel segment. Outputs data structures for vessel segment connectivity, length, and radius.

**RAD\_MIN.m** – function used to numerically estimate vessel radius – called in ANATOMY\_STATS.m via MATLAB’s fzero

**VISCOSITY.m –** function that defines Pries’s “in vivo” viscosity law. Called in RAD\_MIN.m

**DdStrahler\_order.m** – Function called in network\_run.m. Takes outputs from NET\_GEN.m and ANATOMY\_STATS.m as inputs and determines the diameter-defined Strahler orders of the vascular network.

**PLOT\_PERFUSION\_ZONES** – Function that plots quantities associated with the perfusion zones corresponding to each network terminal. Very useful for visualization.

**NETWORK\_PLOT** – Function that plots quantities associated with each vessel segment. Made originally to confirm that we were assigning radii and Strahler orders correctly.

**Scalebar**, and **gplot2** are functions NOT written by me. They were found on MathWorks file exchange. Scalebar lets you plot scalebars, and gplot2 is a slightly better version of MATLAB’s gplot (gplot2 lets you change the linewidth and other settings of the plotted graphs. gplot is still in the stone age on this front)

**COMPARE\_NETWORK\_PLOT**  and **COMPARE\_PERFUSION\_ZONES** are virtually the same as NETWORK\_PLOT.m and PLOT\_PEFUSION\_ZONES.m, but allow you to compare two networks. Note that these functions are not intended to compare different networks, they are intended to compare the same network but with different properties (ex: with and without regulation from HPV).

***VQmodel***

**model\_runfile.m** – the operative script that solves ODEs and plots model solutions. There are a handful of settings at the top of the file you can adjust – “FLAG” lets you select which network you want to simulate, “fFodP” let you use either fixed flow rate or a fixed pressure drop across the entire network (be warned that fixed flow gives you unrealistic pressure drops), “iIC” lets you initialize the model with initial conditions loaded from a .mat file, and “HPV” lets your turn on/off using the empirical HPV model or use uniform vasoconstriction. There are also boolean options that can be used to save results from simulations – in general, I would advise saving one thing at a time.

**load\_network\_par.m** – This function organizes all of the nominal parameters, initial conditions, and network connectivity/geometry info for a given network and outputs them in neatly packed structs.

**VQ\_RHS.m** – This function defines the right-hand-side of the ordinary differential equations that govern the model. We use MATLAB’s ode15s solver as the system is pretty stiff (ode45 gets stuck or takes an outrageous amount of time to finish running)

**GET\_OUT.m** – This function computes the outlet pressures for each vessel segment. Used when processing of model solutions that come from VQ\_RHS

**TT\_deconvolution.m** – this function computes the red blood cell transit time distribution. At its most basic level, it is an implementation of numerical deconvolution for indicator-dilution experiments

**NETWORK\_PREDICIONS\_FIG.m** – this script loads the results from model\_runfile.m (simulations run without regulation from HPV, run with regulation from HPV, and with uniform vasoconstriction), computes descriptive statistics for these results, organizes results for visualization, and makes pretty plots.

**COMPARE\_3PERFUSION\_ZONES.m** – effectively another version of PLOT\_PERFUSION\_ZONES but allows you to compare results from 3 versions of the same network. This function calls **calcticks.m** which is a function that I did not write – I found it on MathWorks file exchange. calcticks calculates tick labels for custom axes – here it is used to make coherent tick labels for the network visualization colorbars.