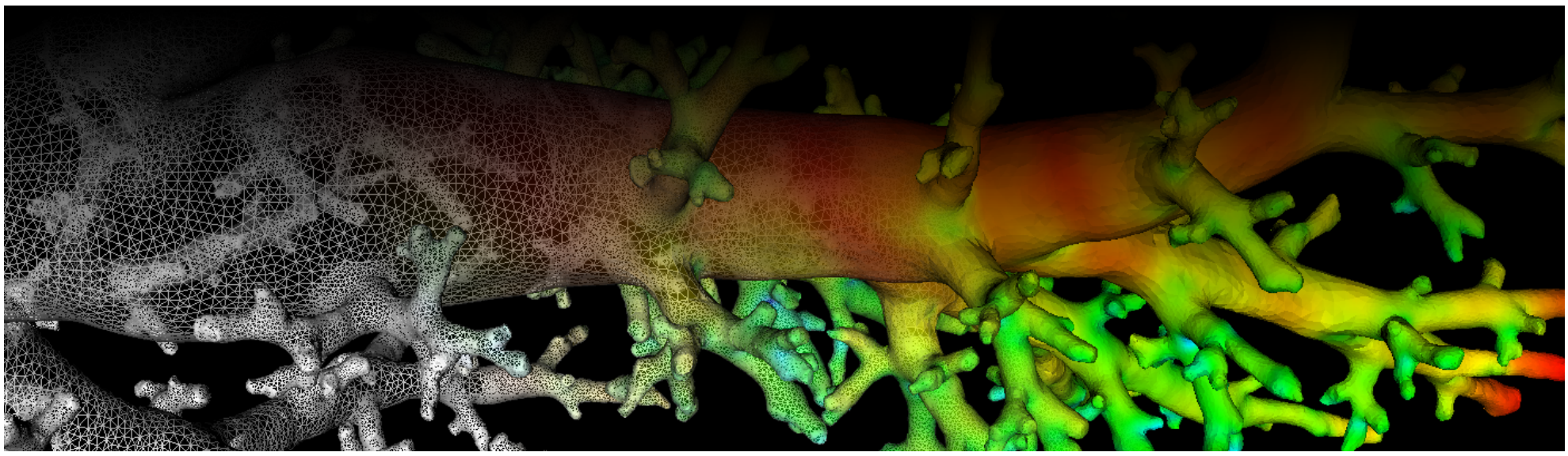
Lung Anatomy + Particle Deposition (lapd) Mouse Archive

for Modeling and Computational Toxicology



Code Examples

C++ and Python code examples demonstrating how to work with lapdMouse data:

- Working with *.csv tables and *.md info files
- Reading and writing basic data structures with ITK
- Working with airway tree structures AirwayTree.meta
- Linking information from multiple data files and deriving new information

Getting started with Examples:

- lapdMousePythonExamples repository on github.
- lapdMouseCppExamples repository on github.

Working with *.csv tables and *.md info files

Name	Description	Language
CompartmentDeposition Measurements.ipynb	Read and interpret **deposition measurement tables**: *_NearAciniDeposition.csv, *_SublobesDeposition.csv, *_LobesDeposition.csv and *_AirwaySegmentsDeposition.csv. Explains the organization of the stored information, shows how to create simple visualizations, identify and plot information.	Python
AirwayTreeTable.ipynb	Read and interpret data from *_AirwayTreeTable.csv. Explains the organization of the stored information, shows how to create simple visualization, identify and plot information.	Python

Name	Description	Language
DatabaseInfoTable.ipynb	Extract and summarize information from *_Info.md files.	Python
VentilationParameters.ipynb	Read and interpret ventilation measurement tables *_Ventilation*.csv. Explains the organization of the stored information, shows how to create simple visualizations, estimate tidal volume and I:E ratio.	Python

Reading and writing basic data structures in ITK

Name	Description	Language
readWriteImage.cpp	Read and write **intensity images** *_Aerosol*.mha, *_AerosolDeconv*.mha, *_AerosolNormalized*.mha, and *_Autofluorescent*.mha used in the lapdMouse project with proper data type.	C++
readWriteLabelmap.cpp	Read and write **labelmaps** *_AirwaySegments.nrrd, *_Lobes.nrrd, *_Sublobes.nrrd and *_NearAcini.nrrd used in the lapdMouse project with proper data type.	C++
readWriteMesh.cpp	Read and write **meshes** used in the lapdMouse project with proper data type.	C++
readWriteTree.cpp	Read and write **tree structures** *_AirwayTree.meta used in the lapdMouse project with proper data type.	C++

Working with airway tree structures AirwayTree.meta

Name	Description	Language
accessTreeData.cpp	Access information from the **tree structures** *_AirwayTree.meta used in the lapdMouse project, including accessing segments, finding child segments, identifying segment names, iterating over a segment's centerline with their coordinates and radius information.	C++
metaTree2Json Converter.cpp	Command line tool to convert *_AirwayTree.meta files used into a Java Script Object Notation (JSON) format for reading with other languages (e.g. Python)	C++
simplifyTree.cpp	Command line tool used in the lapdMouse project to convert *_AirwayTree.meta files to a simplified structure *_AirwayTreeTable.csv.	C++

Linking information from multiple data files and deriving new information

Name	Description	Language
mapOutlet2Airway Segment.cpp	Link outlets from *_AirwaySegments.vtk to airway segments stored in *_AirwayTree.meta. The example reads an AirwaySegments.vtk mesh and obtains for each labeled outlet region its center of gravity. Then, these are assigned to the airway segments in *_AirwayTree.meta by finding the closest airway segment. The resulting mapping of `outletId` to `segmentId` is printed to the command line in a Comma Separated Value (CSV) format.	C++
labelTreePathAnd Children.cpp	Shows (a) how to identify and label airway segments in a tree structure and (b) how to link information between the *_AirwaySegment.vtk mesh and the *_AirwayTree.meta. This example reads an *_AirwaySegments.vtk mesh file and an *_AirwayTree.meta. tree structure. Based on a user specified `segmentId`, the program identified (a) the set of all airway segments from the root the specified segment and (b) all of its child segments. Then, it assigns appropriate label values to all associated mesh vertex point in the input mesh. The resulting labeled mesh is written to `highlightedSegmentsMesh.vtk`. The result can get visualized using e.g. 3D Slicer.	C++

Name	Description	Language
partitionLobesIntoTerminal Compartments.cpp	Partitions the lung's *_Lobes.nrrd into disjoint compartments based the distance to *_AirwayTree.meta terminal segments. The example initially reads *_AirwayTree.meta and identifies all end points of terminal segments. Then it expands from these seed points disjoint terminal compartment regions utilizing a priority queue. During region expansion lobar boundaries are not crosses, i.e. every terminal compartment is part of one lobe only. The resulting compartments are stored in `TerminalCompartments.nrrd` The result can get visualized using e.g. 3D Slicer.	C++
imageLabelStatistics.cpp	Command line tool to calculate statistical measurements for labeled regions. The program reads a **labelmap** and an **intensity image**. It then calculates for each labeled region statistical measurements including volume, average gray-value, etc. These values will printed to the command line in a Comma Separated Value (CSV) format.	C++
GenerationDependentSummaryPlots.ipynb	Combines information from *_Info.md, *_AirwayTreeTable.csv, and *_AirwaySegmentsDeposition.csv files for all datasets in the database and creates summary plots showing average measurements as a function of airway segment generation number (or order number) grouped by mouse strain (or particle size).	Python

Updated: 11/22/19
2018 r2b