Using Carimas software for image analysis

Chunlei Han MD, Ph.D

Turku PET Centre and Turku University Hospital

Turku PET Centre Brain Imaging Course 2021

13:45-14:30, Aug. 18, 2021 Turku, Finland

Contents

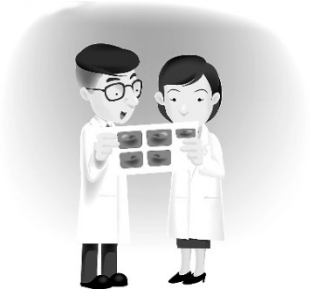
- ***General info: project purpose, website and license***
- Features
- Program structure
- Scientific applications, publications and examples
- Demonstration

Carimas

Carimas is a general medical imaging processing platform developed in Turku PET Centre in Finland.

Originally, Carimas was designed for visualization, segmentation and modelling of PET data only. However, the latest versions support processing of imaging data from most medical imaging modalities, such as CT and MRI.

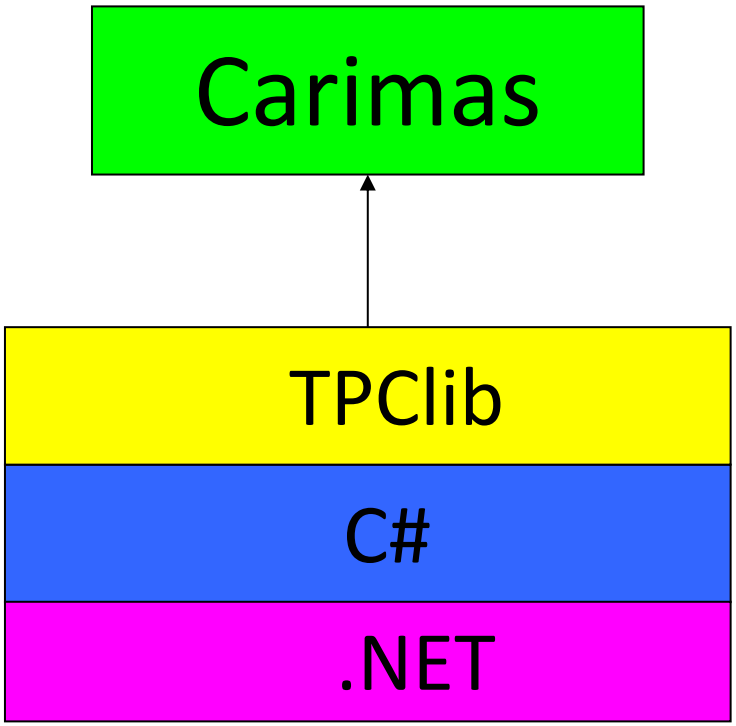
1



For researchers: to provide a complete package for analyzing their PET-related imaging data. Specially, using this package, users can perform the following tasks easily:
1) visualization, 2) segmentation, 3) modelling and 4) reporting

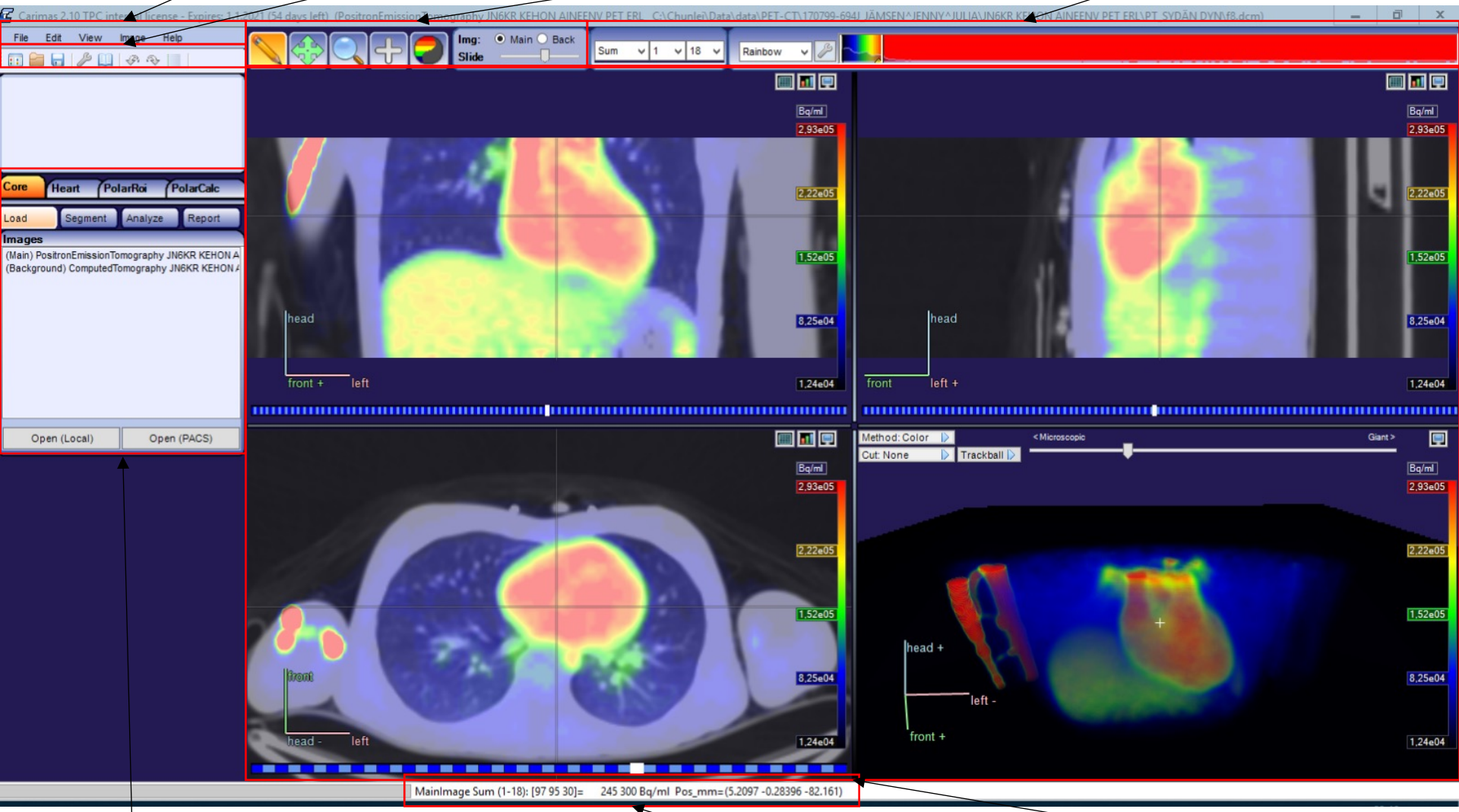
2

For software developers: to provide a development platform. Using a plug-in capability, software developers can easily develop their own applications for medical imaging visualization and analysis tools and implemented them into Carimas.



Mono will be used to generate Linux and Unix version
TPClib is developed in TPC, C# code, dataIO and modelling etc.

Main GUI of Carimas

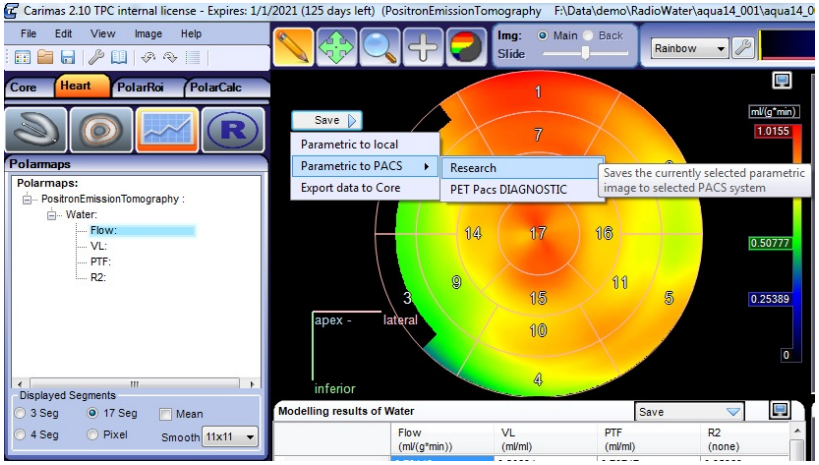


Setting of ADW workstation and PC with Carimas

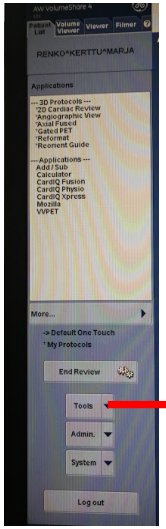
Carimas supports multi-PACs Dicom data communication

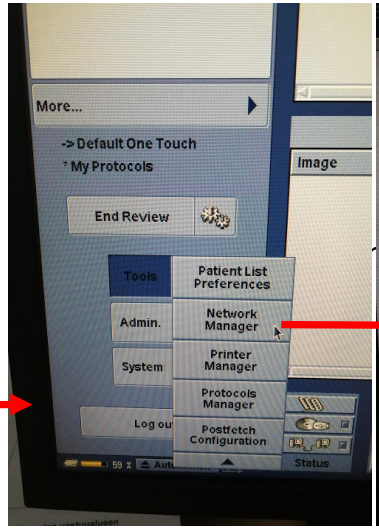
Key points:
1. IP address must be matched each other.
2. Port number must be same.
3. PC listening port must be free, not be occupied.

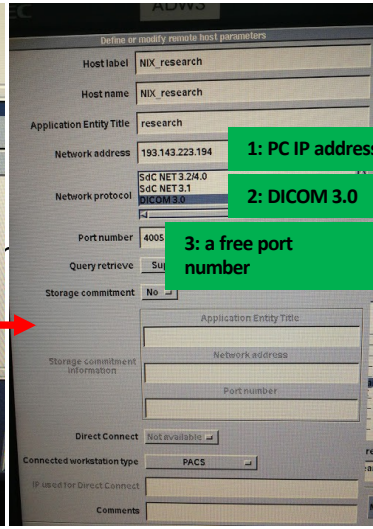
Parametric images can be sent to selected PACs system after modelling in Polarmap window



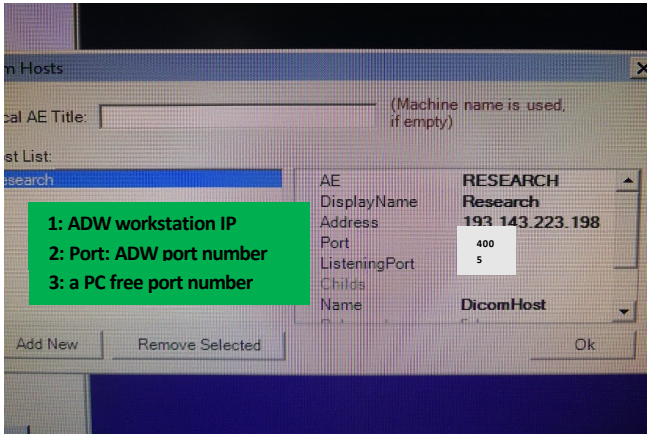
ADW







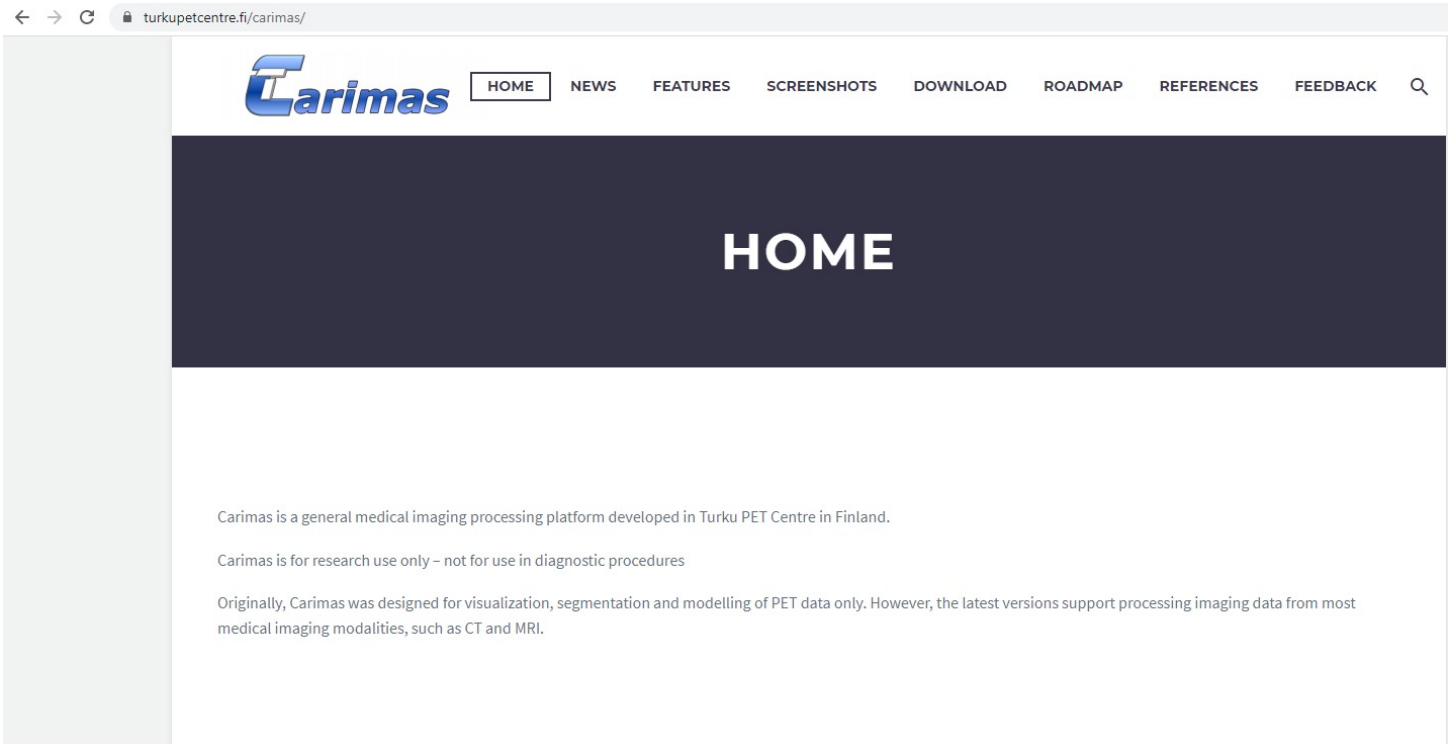
PC



Carimas website

https://turkupetcentre.fi/carimas/features/

Carimas is a general medical imaging processing platform developed in Turku PET Centre in Finland. Originally, Carimas was designed for visualization, segmentation and modelling of PET data only. However, the latest versions support processing of imaging data from most medical imaging modalities, such as CT and MRI.
Using Carimas, you can easily visualize your imaging data in many different ways, such as in 2D or 3D, or fuse images from different modalities (PET/CT, PET/MRI). Furthermore, Carimas provides a lot of advanced functions for researchers. For example, using ROI/VOI tools, the user can draw a region/volume of interest in manual, semi-automatic or automatic manner; using the modelling tools, the user can perform advanced analysis for his/her research data; using Heart tools, cardiac researchers can easily analyze their PET studies.



Installation and license

- **System requirements:**
  .NET 4.0+ for (2.10) or 2.0+ for older. Or alternative Mono framework
- Currently we offer a Lite version of Carimas without any registration. The Lite version has only two models: Linear regression and monoexponential. Using Carimas as lite version requires a lite license that can be downloaded here:
  Lite license

  License files can be loaded into Carimas from the help menu. Getting the full version requires registration. The registration can be done from the help menu. License file will be then sent to you that unlocks full functionality of the software.
- **Windows installers (msi installer):**
  Carimas 2.10
- **All operating systems (zip archive):**
  Carimas 2.10

Contents

- General info: project purpose, website and license
- ***Features***
- Program structure
- Scientific applications, publications and examples
- Demonstration

Features 1/3

- **Input**
  - Multiple image data format support***: DICOM***, ECAT, Analyse, Interfile, Nifti, Interfile, MicroPET and general bitmap formats (JPG, TIFF, PNG and BMP)
  - PACS support: available to connect to hospital PACS system
- **Visualization**
  - View images from transaxial, coronal, sagittal or any free direction
  - 3D view with color rendering or MIP
  - Move and rotate images freely in 3D space
  - Visualize image histograms or cut profiler lines
- **Static image tools**
  - Calculate VOI statistics: mean, maximum, minimum, standard deviation, volume, etc…
  - SUV and percentage units
- **Dynamic images tools**
  - Easy and fast visualization of dynamic data
  - Calculate sum or difference images, or view individual frames
  - Analyze time activity curves from VOIs or individual voxels

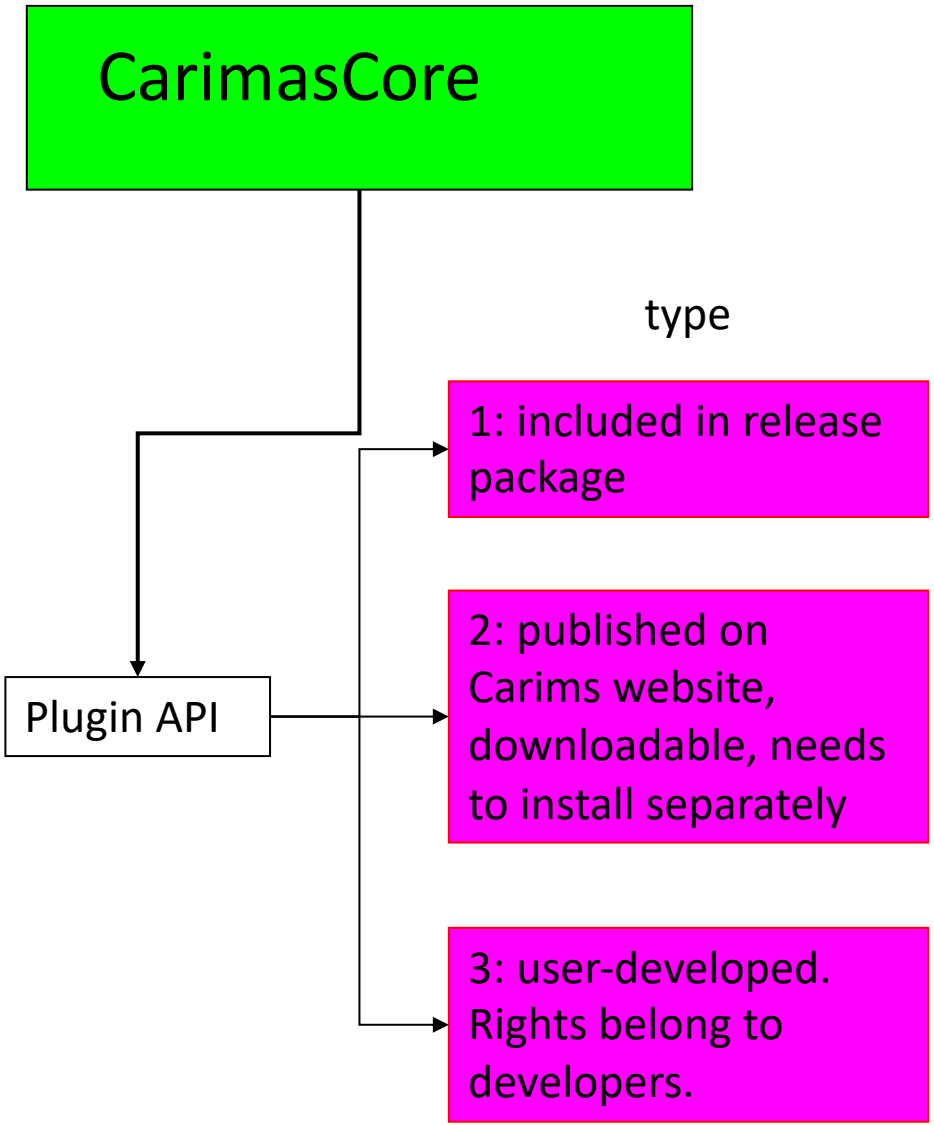
Features 2/3

- **Segmentation tools**
  - 2D ROI sets or 3D VOIs
  - Histogram tool for selecting voxels at value range.
  - Start region definition from some predefined 3D shape or draw your own
  - Create VOIs using masking and countour tools
  - Some 3D region growing segmentation tools exist also as separate plugin, like “syringe” and threshod tools
  - Scale, move rotate, smooth, combine the VOIs with easy visual tools
  - Save and reuse the VOIs in multiple studies, regardless of image type or resolution
- **Image fusion**
  - File format independent image positioning
  - Coregistrate PET/MR/CT automatically
- **Modeling**
  - Use data from images or data files
  - Specify model parameters values and limits
  - Rescale data, define time ranges and exclude time points
  - Calculate parametric images from any model and parameter (separate plugin)

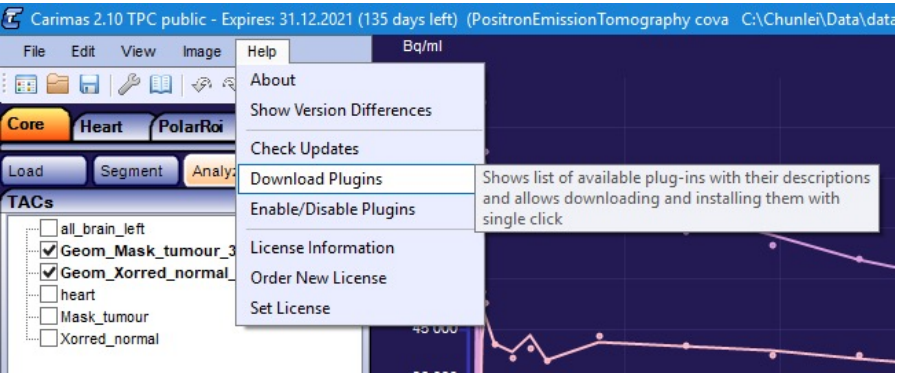
Features 3/3

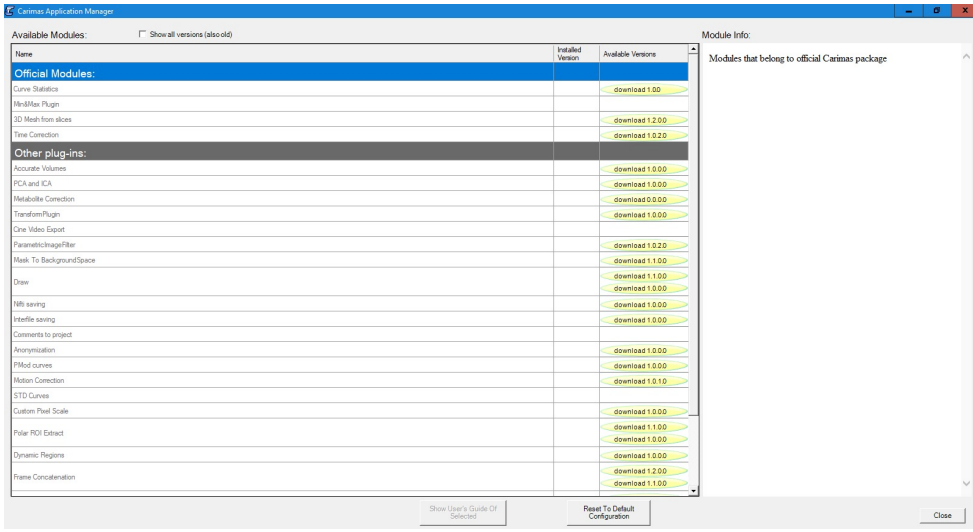
- **Existing models (trial has only first two)**
  - Linear regression
  - Exponential fitting
  - Patlak
  - Logan
  - FUR index
  - Tracer specific models for water (with special license), ammonia, rubidium, acetate and flurpiridaz
  - Generic compartment models
- **Tools for heart analysis**
  - Semi-automatic heart segmentation with manual modification tools
  - Analyze polarmaps in 3, 4 or 17 segment modes, pixel by pixel or draw your own ROI to polarmap
  - Use any models from the Carimas modelling library to create parametric polarmaps
  - Compare results side by side or save to data files
- **Lots of other features as plug-ins**
  - Add to program easily from Carimas menu by just checking them from list
  - List of currently existing plug-ins can be found here: **Go to external plugins archive**

Plugin in Carimas



Plugin is an extentional functions, which can be developed by other developers, not only from Carimas developers





The additional Carimas plug-ins

Name	Description
Accurate Volumes	Calculates more accurate volume of 3d region by placing cube with dimensions [DIM x DIM x DIM] around bounding box of VOI region and running masking. There is two plug-ins, one uses 1200 voxels for dimension, other asks it from user
PCA and ICA	Calculates principal and independent components for the dynamic image curves. User can define the amount of components extracted. The result will be parametric image for every component + dft file containing every component as curve. The plug-in will also create simulated image and error image for seeing, how much information the used components are actually carrying
Find Regions	Generate VOIs from image pixels. There is several region growing tools: contour, threshold and syringe
Metabolite Correction	Calculate TACs of authentic (unchanged) tracer and radioactive metabolite(s) from measured plasma TAC and fractions of authentic tracer
TransformPlugin	Transforms Images and VOIs from one transformation matrix space to another
Cine Video Export	Creates a video file (avi) where the 3D screen rotates whole 360 degrees around currently selected point. AForge library is used for video export. When installed, plugin can be run from Menu->Edit->Other Commands->Create Cine Video
ParametricImageFilter	With this plug-in you can calculate parametric images voxel by voxel from any existing model of Carimas. When installed, this plugin appears in the image filters menu in the Load tab. USAGE: 1) Load image 2) Select some model and fill it's possible input functions and other options in the Analyze tab 3) (optional) Select some VOI from segment tab if you want to calculate parametric map from only some specific region 4) Run the parametric image filter under the source image in the Load -tab.The parametric images will appear to image list (every parameter will have one). There can be also parametric maps of goodness of fit values among the results
Mask To BackgroundSpace	Converts mask to background image space. If the target mask has even little portion of masked area in source, the result is considered as masked, therefore the volume of result will be larger than source
Draw	Basic drawing tool for masks. Provides paint brush, trace and free hand tools for including and excluding voxels. Has also fill gaps and mask smooth features
Nifti saving	Allows saving images in Nifti -format
Interfile saving	Allows saving images in interfile -format
Comments to project	Allows writing text comments to projects. This plug-in appears to Edit->other commands -menu
Anonymization	Anonymizes Dicom images and saves them to disk
PMod curves	Allows and saving of PMod curve files
Reslicer	Reslices images to another image space. This plug-in gives two options, reslice to standard space that has component vectors of +x, +y, +z or to original image space
Motion Correction	Coregisters rigidly all image frames separately and saves the result frames as separate images
STD Curves	Calculates standard deviation curves
Custom Pixel Scale	The pixel range of image can be scaled to given range or multiplied with given factor
Polar ROI Extract	Adds the currently drawn heart polar ROI as mask VOI to Core
Image Exporting Tools	Contains several ways to export the image. All the frames or the currently visualized frame range (sum,diff,etc..) can be exported. All hidden voxels are left out from result image and automatic same value cropping can be used to save smaller image (good to be used together with voxel hiding). There is also option for reducing number of colors of the saved image
Dynamic Regions	Allows VOI region editing frame by frame so that the shape changes over time. If shape is edited in distant time points, all frames between are interpolated. The plug-in contains also tool for calculating curve from changing volume of the shape over time. Usage: Create normal 3D shape around the dynamic region. Convert it to dynamic from the tools list. Use single frame view and edit the voi with different frames. Volume curve can be created from the analyze TAB's curve tool list
Frame Concatenation	Combines many images (with same dimensions) and creates one dynamic image from them. When installed, this plug-in can be run from Edit menu->Other commands
Curve Tools	Tools for modifying Carimas curves: add, subtract, clone, interpolate, extrapolate and integrate curves or calculate frame weight curves
Nifti mask image	Saves VOIs, ROIs and masks as NIFTI images
PMod RoiSet	Loads ROI sets from PMOD .voi files.
Gausian Smooth & Spillover correction	Contains two plug-ins: Gausian convolution smoothing filter and spillover correction Spillover effect is removed by simulating the given region with given activity. After that, the area is removed with spillover included. The spilling region is assumed uniform. For small regions there exists also way to give multiplication factor for the area

Generating paramatric images in Carimas using plugin of “Parametric image filter”

Key points:

1. All models implemented in Carimas can be used to generate parametric image (pixel-based parametric image).
2. Only pixels inside of selected ROIs/VOIs are calculated.
3. Output is dicom file, each of model parametres locates in its own folder.
4. It may be a time-consuming process, depending number of selected pixels, model and hardware system.

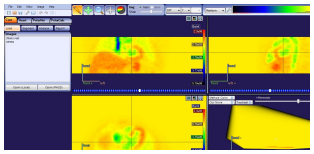
Download and installation of plugin of “Parametric image filter”.

1. This plugin is not default in Carimas.
2. It is a free-downloadable plugin from Carimas website.
3. Carimas->Help->Download plugins.
4. Select “Parametric image filter”.
5. Download and save it in a folder.
6. Carimas->Edit->Extra plugin folder: select plugin-saved folder
7. Restart Carimas.

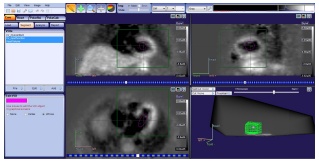
Step of “Parametric image filter”.

1. Load image.
2. Core->Segment->Define ROIs or VOIs. These include regions or volumes not only as parametric outputs, but also as input function.
3. Core->Analysis->Select a model, and define input function ROI or VOI.
4. Core->Segment->select ROIs or VOIs, in which parametric will be generated.
5. Core->Load: in image list, on select an image to click left button. On drop-down list, select “Parametric image filter”
6. On file dialog: select a folder for saving output parametric image.
7. Results: parametric images will be outputted in two ways:
   - Saved in selected folder as dicom files. Each subfolder holds a parametric. Subfolder name is parametric name.
   - Parametric images are added to image list.

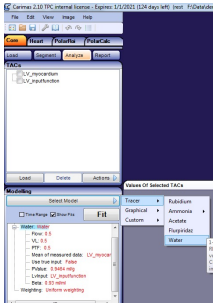
Step 1 Load image



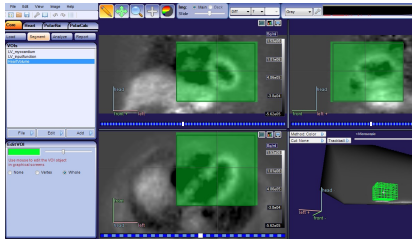
Step 2 Define ROIs/VOIs



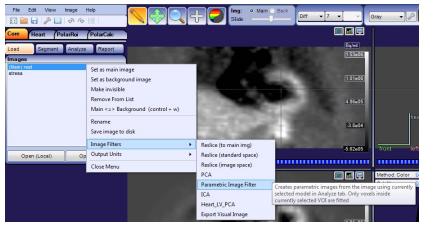
Step 3 Select a model



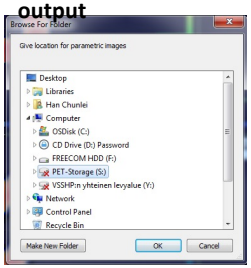
Step 4 Select ROIs/VOIs



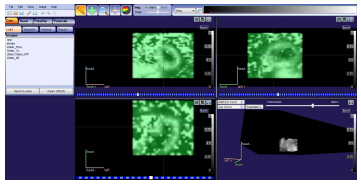
Step 5 To run parametric image filter

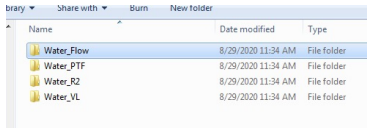


Step 6 Select folder for output



Step 7 Results



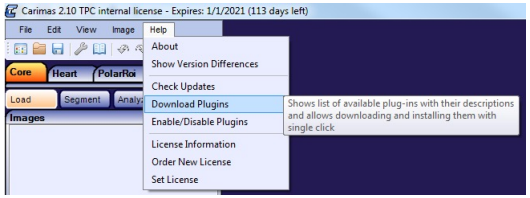


Dicom anonymization plugin of Carimas

Very effectively and easily anonymize Dicom data

Download and installation of plugin

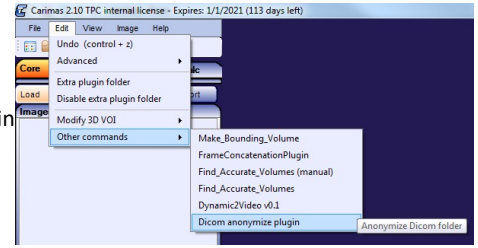
1. Help->Download Plugin
2. On Carimas plugin website, select: Dicom anonymization plugin
3. Restart Carimas.



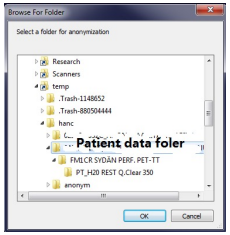
Anonymize Dicom data

Step 1:

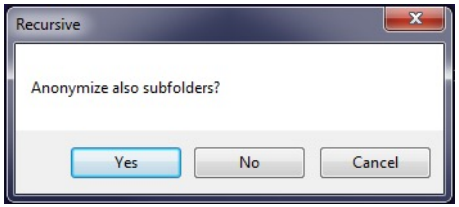
- Start Carimas
- Edit->Other commands->Dicom anonymize plugin



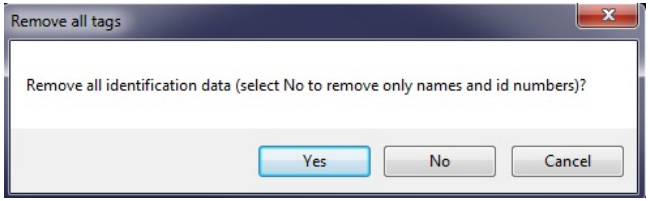
Step 2: Select data folder



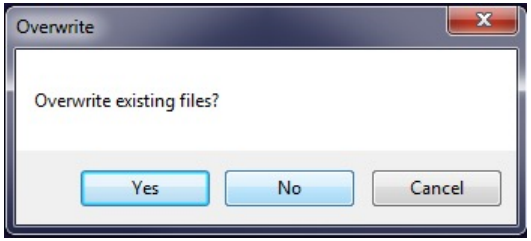
Step 3: Anonymize also subfolder?. If yes: anonymize all data in subfolder. If no: only data in selected folder will be anonymized



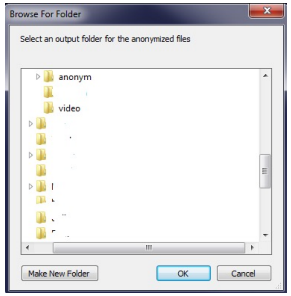
Step 4: Remove all or only name and ID



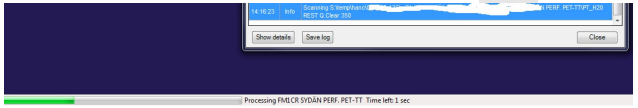
Step 5: option: overwrite or not?



Step 6: Select a folder for saving anonymize data

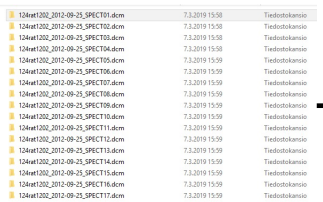


Step 7: Processing data and progress bar

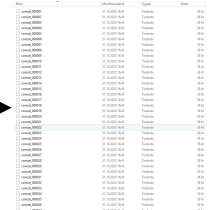


Frame concatenation plugin

One folder contains a frame image data



One folder holds all frame image data

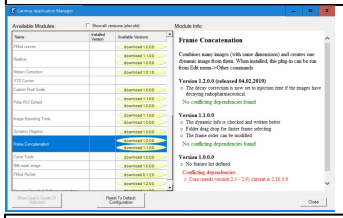


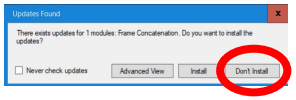
Key points:

1. This plugin is to concatenate single frame images to a complete dynamic data set.
2. From most scanners, dynamic images from all frames locate in one folder, from which Carimas can load all data at once. However, in some cases, frame images can be saved in separated folders. Therefore it needs to add all data into one folder.
3. .sif file is needed to hold frame timing information.

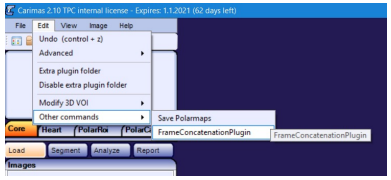
Important

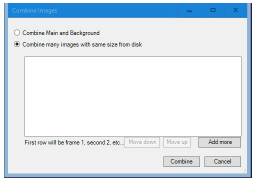
- Version of 1.2.00 has bug, please update to 1.1.00
- If you have installed 1.2.00, you have to reset: Help->Download Plugins, then Reset to Default Configuration.
- Restart Carimas, don't install.





Start: Edit->Other commands->FrameConcatenationPlugin

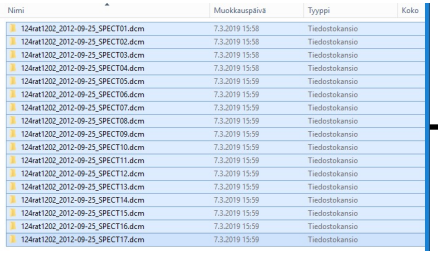




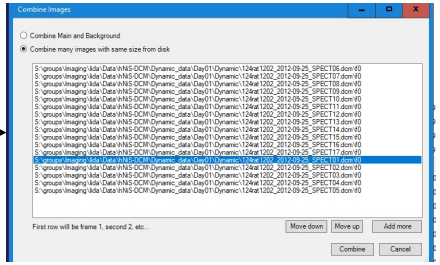
FrameConcatenationPlugin GUI:
Two options to combine

1. Main and background images.
2. Many images with same size from disk.

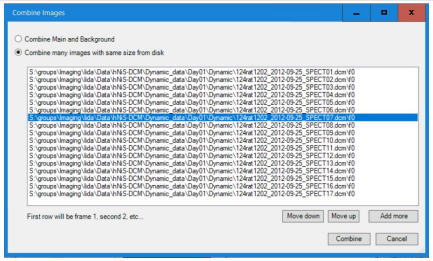
You can add frame one by one by button of "Add more", but more efficient way is to drag and drop by mouse

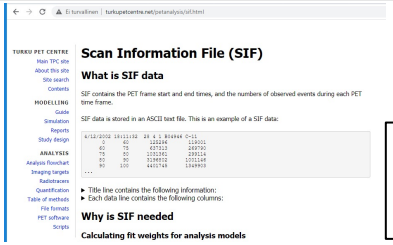


Multiple selection/Drag and drop by mouse



1. Frame order is created based on file order, which is often not correct, you can modify them by "Move down" and "Move up" buttons. Then, click button of "Combine", a dialog for saving folder will come.
2. Combined data will be saved in dicom format in selected folder.
3. You have to create a .sif file containing frame information and put it in the same folder, then Carimas will load dynamic data correctly.





.sif file is an ASCII format, more details to see from http://www.turkupetcentre.net/petanalysis/sif.html

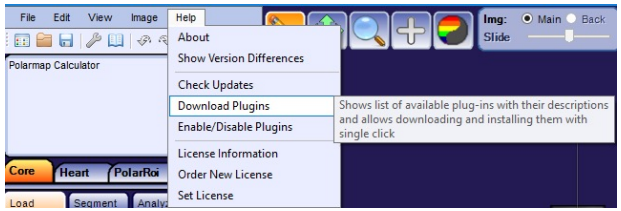
Carimas plugin: Image Export Tools/Export Visual Image

Function: to export currently visualized images in format of Dicom.

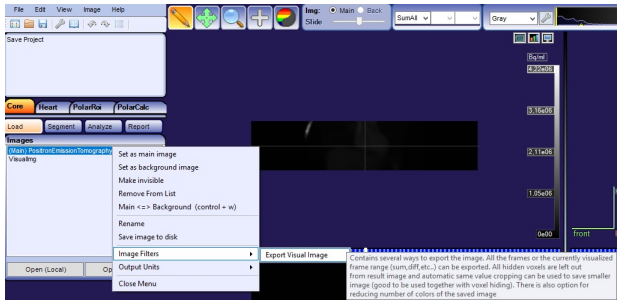
Features:

1. Currently visualized images can be exported in Dicom, such as sum image, diff image or single frame from dynamic data set.
2. Additionally, cropped images, images with user-defined value in masked regions and images with reduced bit can also be exported accordingly.

Installation of this plugin: Help->Download Plugins
In plugin list, select “Image Export Tools”



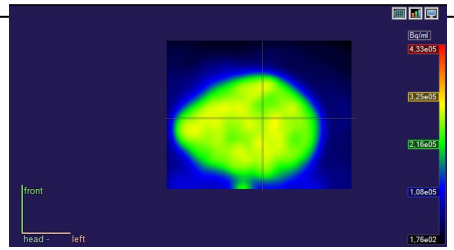
Start of plugin: Click left button of mouse on main images. On drop-down menu: Image Filter->Export Visual Image



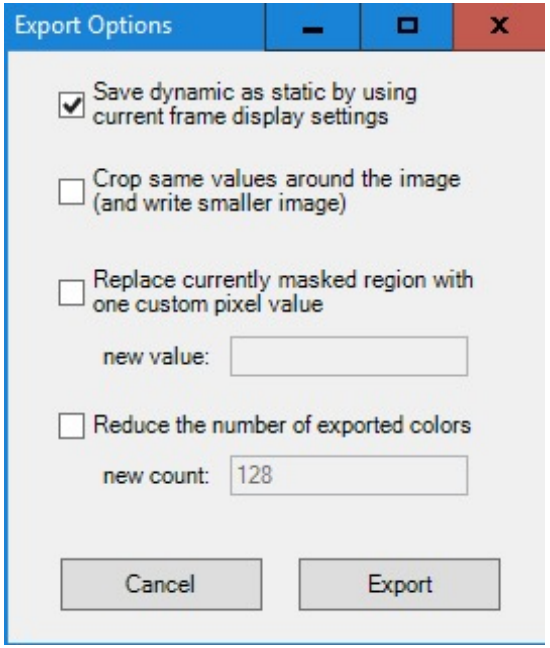
Usage

1. Save dynamic as static by using current frame display settings: depending on user’s sitting, sum images, diff images, or single frame images can be saved and exported.
2. Crop same values around the image (and write smaller image):
   - VOIs are needed firstly defined and selected
   - Image->Show/Hide voxels->Show voxels outside…
3. Replace currently masked region with one custom pixel value: voxel in cropped volume will be filled with new value.
4. Reduce the number of exported colors: exported image bit will be replaced with new value.

Example: heart volume is generated as a new data set



Main GUI of Image Export Tools



Generating paramatric images in Carimas using plugin of “Parametric image filter”

Key points:

1. All models implemented in Carimas can be used to generate parametric image (pixel-based parametric image).
2. Only pixels inside of selected ROIs/VOIs are calculated.
3. Output is dicom file, each of model parametres locates in its own folder.
4. It may be a time-consuming process, depending number of selected pixels, model and hardware system.

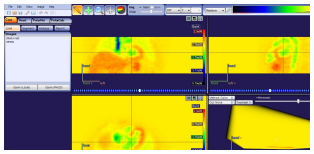
Download and installation of plugin of “Parametric image filter”.

1. This plugin is not default in Carimas.
2. It is a free-downloadable plugin from Carimas website.
3. Carimas->Help->Download plugins.
4. Select “Parametric image filter”.
5. Download and save it in a folder.
6. Carimas->Edit->Extra plugin folder: select plugin-saved folder
7. Restart Carimas.

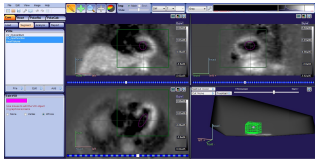
Step of “Parametric image filter”.

1. Load image.
2. Core->Segment->Define ROIs or VOIs. These include regions or volumes not only as parametric outputs, but also as input function.
3. Core->Analysis->Select a model, and define input function ROI or VOI.
4. Core->Segment->select ROIs or VOIs, in which parametric will be generated.
5. Core->Load: in image list, on select an image to click left button. On drop-down list, select “Parametric image filter”
6. On file dialog: select a folder for saving output parametric image.
7. Results: parametric images will be outputted in two ways:
   - Saved in selected folder as dicom files. Each subfolder holds a parametric. Subfolder name is parametric name.
   - Parametric images are added to image list.

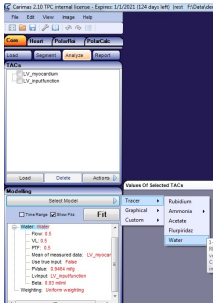
Step 1 Load image



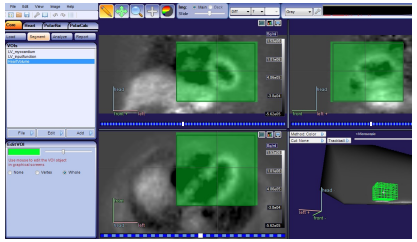
Step 2 Define ROIs/VOIs



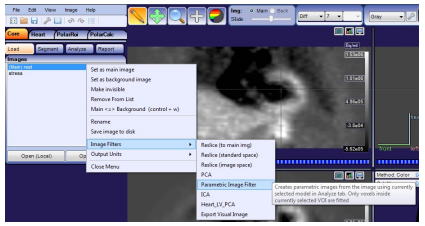
Step 3 Select a model



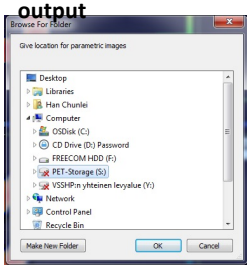
Step 4 Select ROIs/VOIs



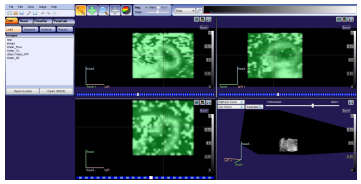
Step 5 To run parametric image filter

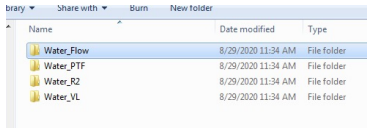


Step 6 Select folder for output



Step 7 Results

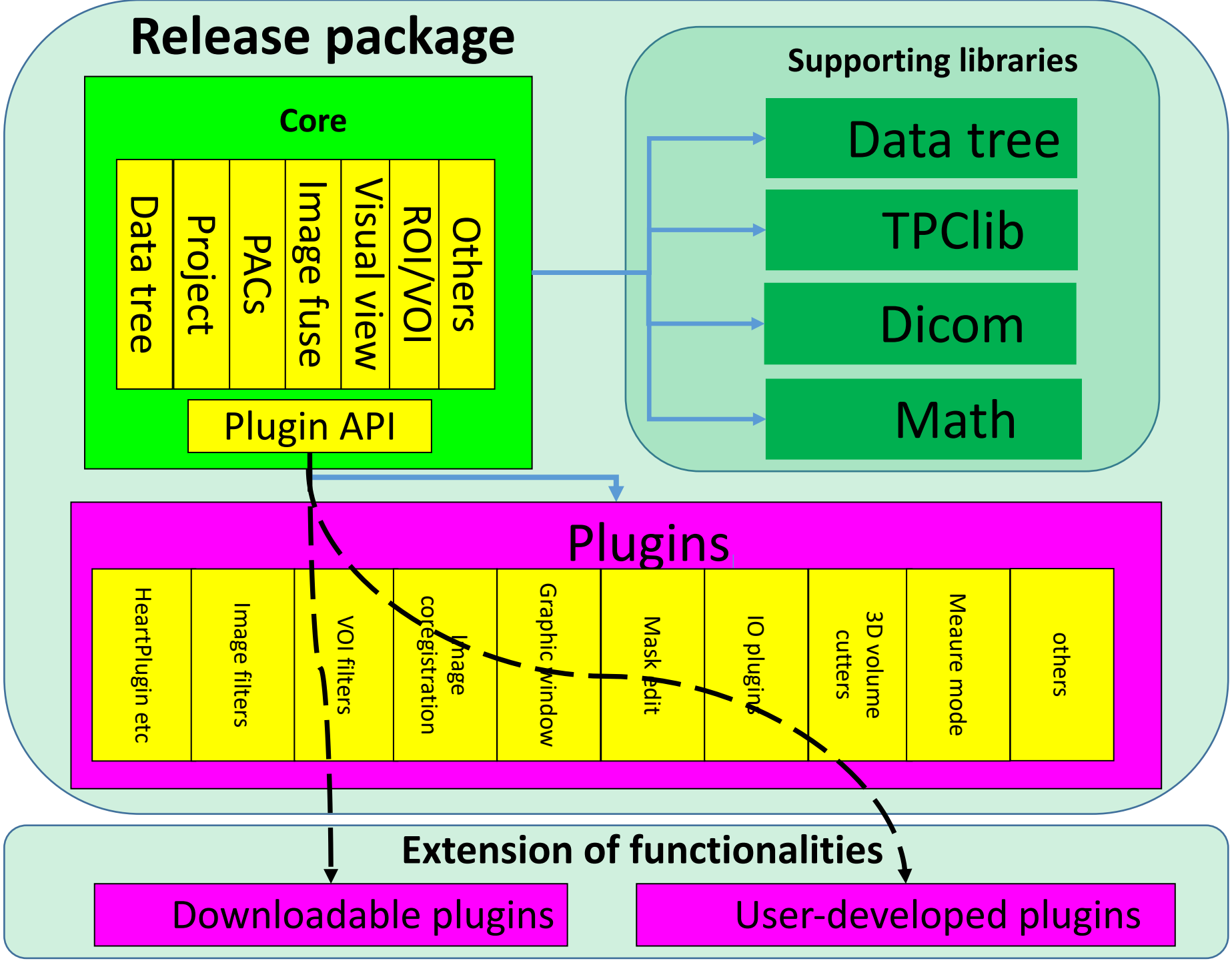


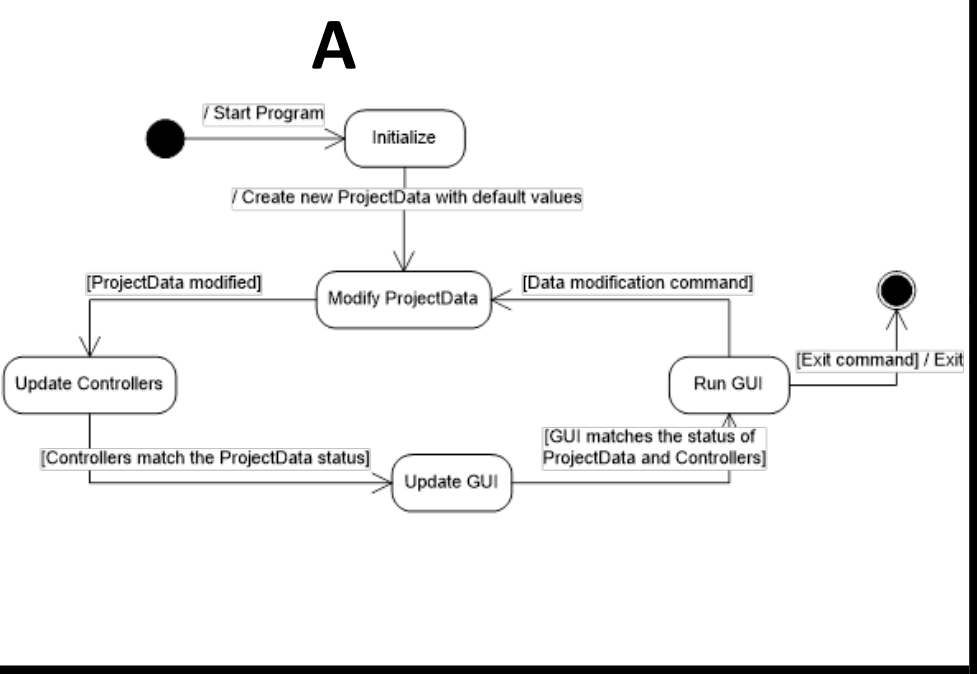


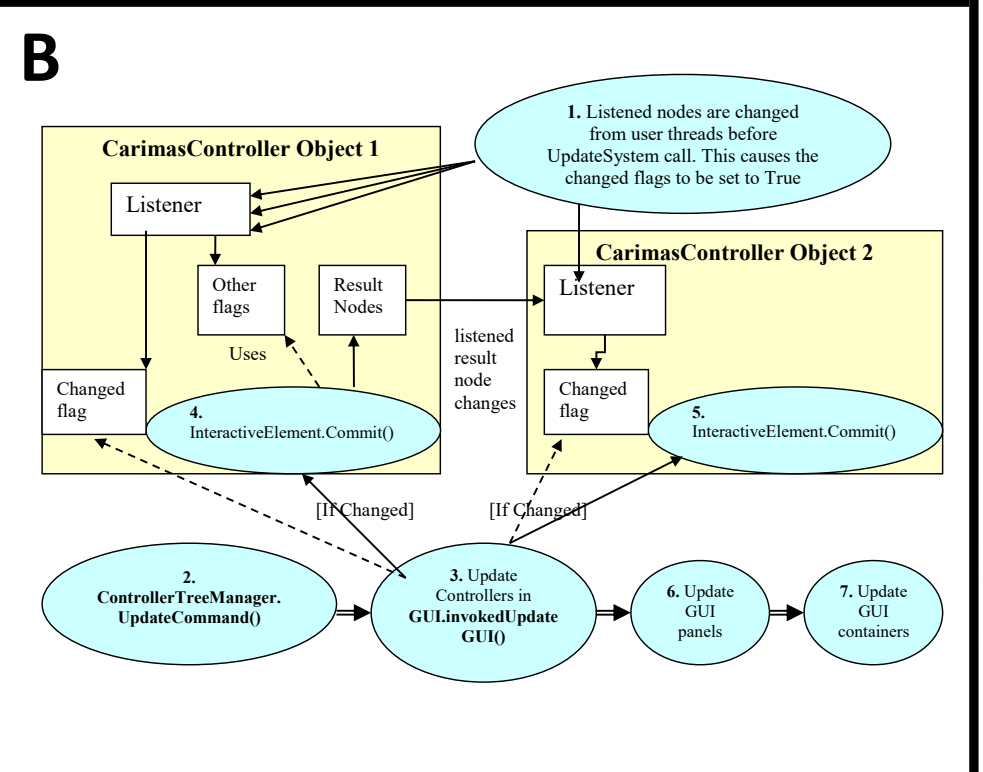
By Chunlei Han, Aug. 30,2020, Turku PET Centre, Turku, Finland

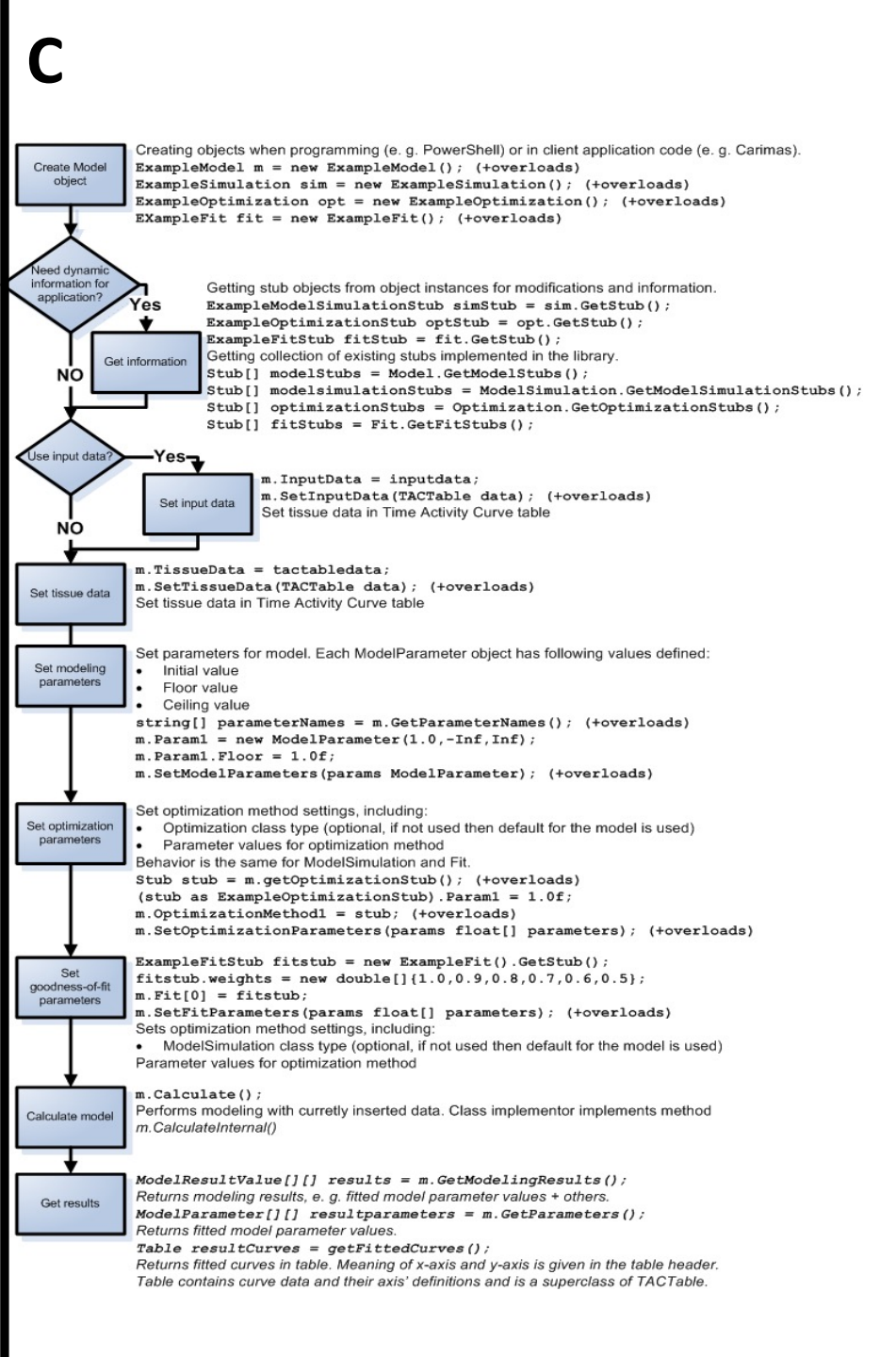
Contents

- General info: project purpose, website and license
- Features
- ***Program structure***
- Scientific applications, publications and examples
- Demonstration

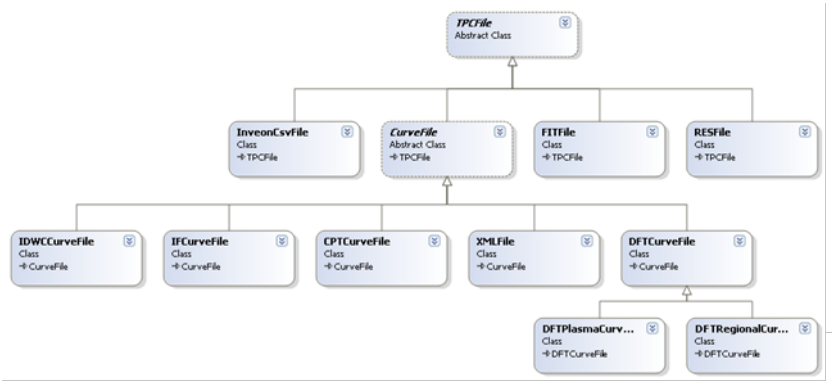

Release package
Core
Data tree
Project
PACs
Image fuse
Visual view
ROI/VOI
Others
Plugin API
Supporting libraries
Data tree
TPClib
Dicom
Math
Plugins
HeartPlugin etc
Image filters
VOI filters
Image coregistration
Graphic Window
Mask Edit
IO plugins
3D volume cutters
Meaure mode
others
Extension of functionalities
Downloadable plugins
User-developed plugins

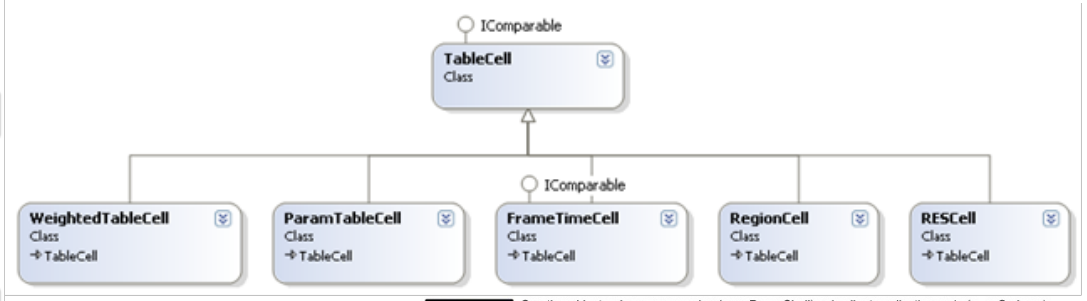

A
/ Start Program
Initialize
/ Create new ProjectData with default values
Modify ProjectData
[ProjectData modified]
[Data modification command]
Update Controllers
Run GUI
[Exit command] / Exit
[Controllers match the ProjectData status]
Update GUI
[GUI matches the status of ProjectData and Controllers]

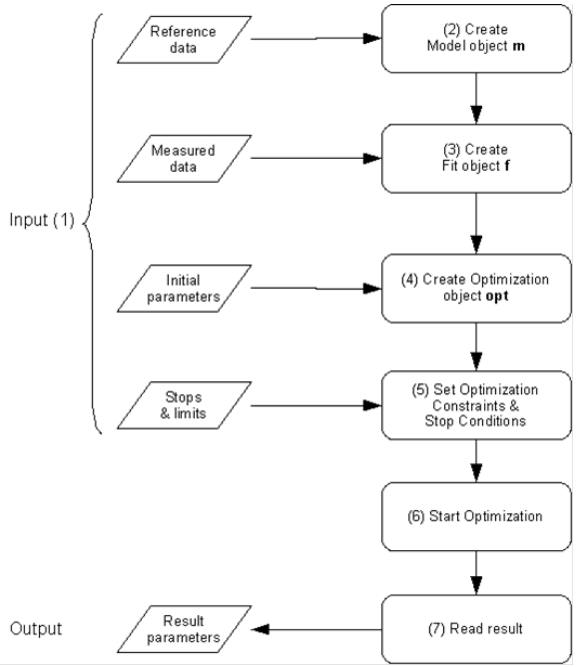

B
CarimasController Object 1
Listener
Other flags
Result Nodes
Uses
Changed flag
4. InteractiveElement.Commit()
1. Listened nodes are changed from user threads before UpdateSystem call. This causes the changed flags to be set to True
listened result node changes
CarimasController Object 2
Listener
Changed flag
5. InteractiveElement.Commit()
[If Changed]
[If Changed]
2. ControllerTreeManager. UpdateCommand()
3. Update Controllers in GUI.invokedUpdate GUI()
6. Update GUI panels
7. Update GUI containers

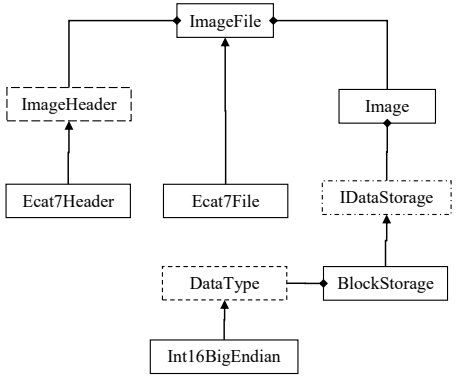

C
Create Model object
Creating objects when programming (e. g. PowerShell) or in client application code (e. g. Carimas).
ExampleModel m = new ExampleModel(); (+overloads)
ExampleSimulation sim = new ExampleSimulation(); (+overloads)
ExampleOptimization opt = new ExampleOptimization(); (+overloads)
EXampleFit fit = new ExampleFit(); (+overloads)
Need dynamic information for application?
Yes
NO
Get information
Getting stub objects from object instances for modifications and information.
ExampleModelSimulationStub simStub = sim.GetStub();
ExampleOptimizationStub optStub = opt.GetStub();
ExampleFitStub fitStub = fit.GetStub();
Getting collection of existing stubs implemented in the library.
Stub[] modelStubs = Model.GetModelStubs();
Stub[] modelsimulationStubs = ModelSimulation.GetModelSimulationStubs();
Stub[] optimizationStubs = Optimization.GetOptimizationStubs();
Stub[] fitStubs = Fit.GetFitStubs();
Use input data?
Yes
NO
Set input data
m.InputData = inputdata;
m.SetInputData(TACTable data); (+overloads)
Set tissue data in Time Activity Curve table
Set tissue data
m.TissueData = tactabledata;
m.SetTissueData(TACTable data); (+overloads)
Set tissue data in Time Activity Curve table
Set modeling parameters
Set parameters for model. Each ModelParameter object has following values defined:
Initial value
Floor value
Ceiling value
string[] parameterNames = m.GetParameterNames(); (+overloads)
m.Param1 = new ModelParameter(1.0,-Inf,Inf);
m.Param1.Floor = 1.0f;
m.SetModelParameters(params ModelParameter); (+overloads)
Set optimization parameters
Set optimization method settings, including:
Optimization class type (optional, if not used then default for the model is used)
Parameter values for optimization method
Behavior is the same for ModelSimulation and Fit.
Stub stub = m.getOptimizationStub(); (+overloads)
(stub as ExampleOptimizationStub).Param1 = 1.0f;
m.OptimizationMethod1 = stub; (+overloads)
m.SetOptimizationParameters(params float[] parameters); (+overloads)
Set goodness-of-fit parameters
ExampleFitStub fitstub = new ExampleFit().GetStub();
fitstub.weights = new double[]{1.0,0.9,0.8,0.7,0.6,0.5};
m.Fit[0] = fitstub;
m.SetFitParameters(params float[] parameters); (+overloads)
Sets optimization method settings, including:
ModelSimulation class type (optional, if not used then default for the model is used)
Parameter values for optimization method
Calculate model
m.Calculate();
Performs modeling with curretly inserted data. Class implementor implements method
m.CalculateInternal()
Get results
ModelResultValue[][] results = m.GetModelingResults();
Returns modeling results, e. g. fitted model parameter values + others.
ModelParameter[][] resultparameters = m.GetParameters();
Returns fitted model parameter values.
Table resultCurves = getFittedCurves();
Returns fitted curves in table. Meaning of x-axis and y-axis is given in the table header.
Table contains curve data and their axis' definitions and is a superclass of TACTable.

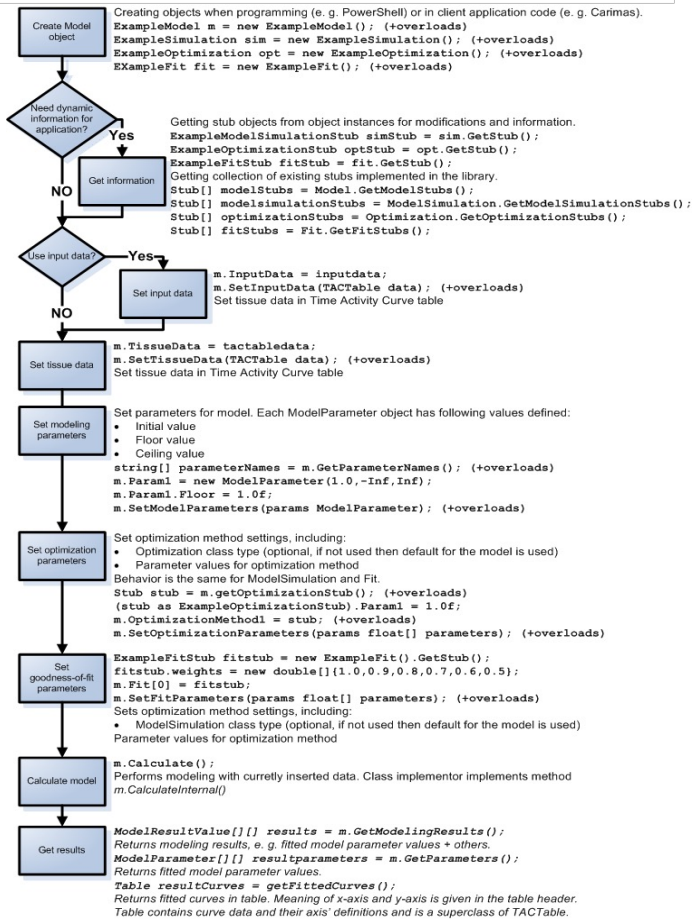
TPClib


TPCFile
Abstract Class
InveonCsvFile
Class
TPCFile
CurveFile
Abstract Class
TPCFile
FITFile
Class
TPCFile
RESFile
Class
TPCFile
IDWCCurveFile
Class
CurveFile
IFCurveFile
Class
CurveFile
CPTCurveFile
Class
CurveFile
XMLFile
Class
CurveFile
DFTCurveFile
Class
CurveFile
DFTPlasmaCurv...
Class
DFTCurveFile
DFTRegionalCur...
Class
DFTCurveFile


IComparable
TableCell
Class
IComparable
WeightedTableCell
Class
TableCell
ParamTableCell
Class
TableCell
FrameTimeCell
Class
TableCell
RegionCell
Class
TableCell
RESCell
Class
TableCell


Input (1)
Reference data
Measured data
Initial parameters
Stops & limits
(2) Create Model object m
(3) Create Fit object f
(4) Create Optimization object opt
(5) Set Optimization Constraints & Stop Conditions
(6) Start Optimization
(7) Read result
Output
Result parameters


ImageFile
ImageHeader
Image
Ecat7Header
Ecat7File
IDataStorage
DataType
BlockStorage
Int16BigEndian

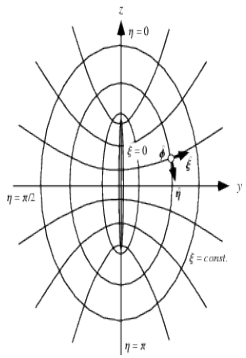

Create Model object
Creating objects when programming (e. g. PowerShell) or in client application code (e. g. Carimas).
ExampleModel m = new ExampleModel(); (+overloads)
ExampleSimulation sim = new ExampleSimulation(); (+overloads)
ExampleOptimization opt = new ExampleOptimization(); (+overloads)
EXampleFit fit = new ExampleFit(); (+overloads)
Need dynamic information for application?
Yes
Get information
Getting stub objects from object instances for modifications and information.
ExampleModelSimulationStub simStub = sim.GetStub();
ExampleOptimizationStub optStub = opt.GetStub();
ExampleFitStub fitStub = fit.GetStub();
Getting collection of existing stubs implemented in the library.
Stub[] modelStubs = Model.GetModelStubs();
Stub[] modelsimulationStubs = ModelSimulation.GetModelSimulationStubs();
Stub[] optimizationStubs = Optimization.GetOptimizationStubs();
Stub[] fitStubs = Fit.GetFitStubs();
NO
Use input data?
Yes
Set input data
m.InputData = inputdata;
m.SetInputData(TACTable data); (+overloads)
Set tissue data in Time Activity Curve table
NO
Set tissue data
m.TissueData = tactabledata;
m.SetTissueData(TACTable data); (+overloads)
Set tissue data in Time Activity Curve table
Set modeling parameters
Set parameters for model. Each ModelParameter object has following values defined:
Initial value
Floor value
Ceiling value
string[] parameterNames = m.GetParameterNames(); (+overloads)
m.Param1 = new ModelParameter(1.0,-Inf,Inf);
m.Param1.Floor = 1.0f;
m.SetModelParameters(params ModelParameter); (+overloads)
Set optimization parameters
Set optimization method settings, including:
Optimization class type (optional, if not used then default for the model is used)
Parameter values for optimization method
Behavior is the same for ModelSimulation and Fit.
Stub stub = m.getOptimizationStub(); (+overloads)
(stub as ExampleOptimizationStub).Param1 = 1.0f;
m.OptimizationMethod1 = stub; (+overloads)
m.SetOptimizationParameters(params float[] parameters); (+overloads)
Set goodness-of-fit parameters
ExampleFitStub fitstub = new ExampleFit().GetStub();
fitstub.weights = new double[]{1.0,0.9,0.8,0.7,0.6,0.5};
m.Fit[0] = fitstub;
m.SetFitParameters(params float[] parameters); (+overloads)
Sets optimization method settings, including:
ModelSimulation class type (optional, if not used then default for the model is used)
Parameter values for optimization method
Calculate model
m.Calculate();
Performs modeling with curretly inserted data. Class implementor implements method
m.CalculateInternal()
Get results
ModelResultValue[][] results = m.GetModelingResults();
Returns modeling results, e. g. fitted model parameter values + others.
ModelParameter[][] resultparameters = m.GetParameters();
Returns fitted model parameter values.
Table resultCurves = getFittedCurves();
Returns fitted curves in table. Meaning of x-axis and y-axis is given in the table header.
Table contains curve data and their axis' definitions and is a superclass of TACTable.

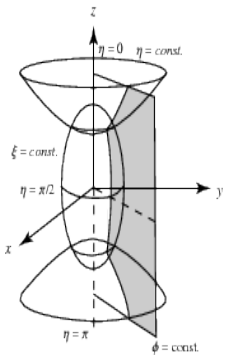
Heart

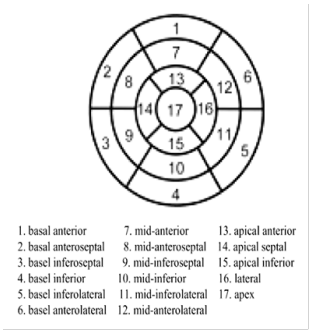
Heart plug-in

TAB Plugin interface

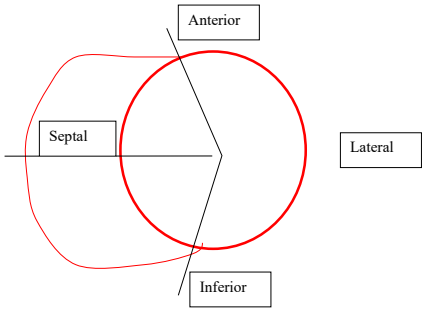
Carimas Core (Carimas framework)







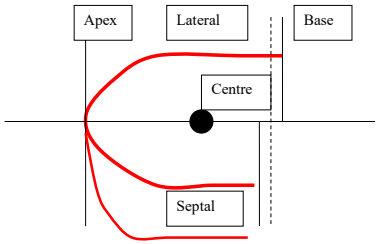
$$x = a \sinh \xi \sin \eta \cos \phi$$
$$y = a \sinh \xi \sin \eta \sin \phi$$
$$z = a \cosh \xi \cos \eta$$



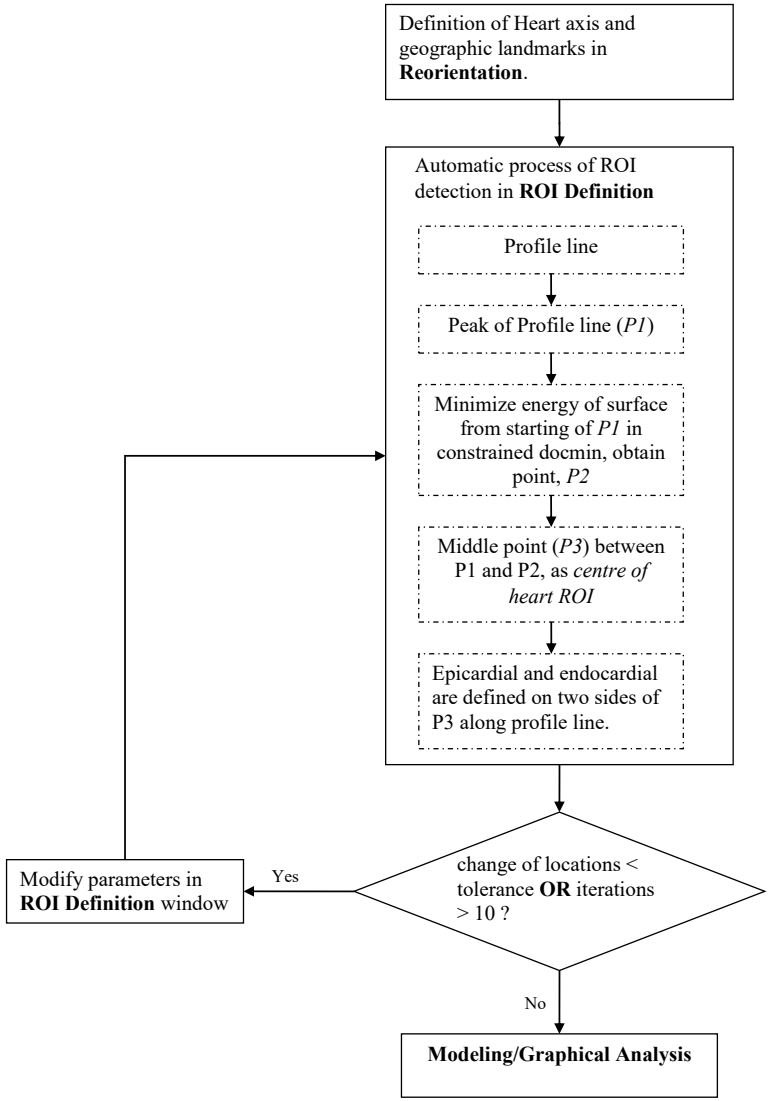
17-segment polarmap regions of 2D projection

6	1	2	3	4	5
12	7	8	9	10	11
16	13	14		15	16
17					

Long axis section of heart



Heart ROI searching algorithm in HeartShapeSegment.search_LVROIs()



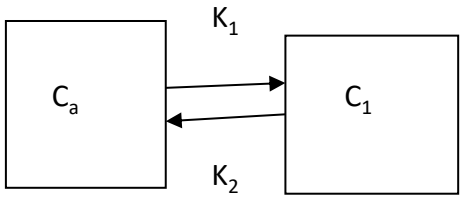
Models implemented in Carimas

1. Compartment model
    1. Tracer-specific model for heart
        1. 15O labeled water perfusion model (1TCM)
        2. 11C labeled acetate perfusion model (1TCM)
        3. 82Rb rubidium perfusion model (1TCM)
        4. 13N labeled ammonia perfusion model
            1. UCLA model (simplified 1TCM)
            2. Duke model (simplified 2TCM-K3)
            3. Michigan model (full 2TCM)
        5. 18F labeled flurpiridaz perfusion model (2TCM,K3)
    2. General model
        1. 1-tissue compartment model
        2. 2-tissue compartment model(k3)
        3. 2-tissue compartment model (k4)
        4. 3-tissue compartment model in parallel
        5. 3-tissue compartment model in series
2. Graphical analysis method
    1. Patlak
    2. Logan
    3. Kmono

TCM=tissue compartment model

15O-labeled radio water cardiac perfusion model

1-tissue compartment model



The perfusion calculations are based on three equations:

$$C_i(T) = f\int_0^T a(t)dt - \frac{f}{p}\int_0^T C_i(t)dt \quad (1)$$

$$ROI(T) = \alpha C_i(T) + V_a a(T) \quad (2)$$

$$LV(T) = \beta a(T) + (1-\beta)C_i(T) \quad (3)$$

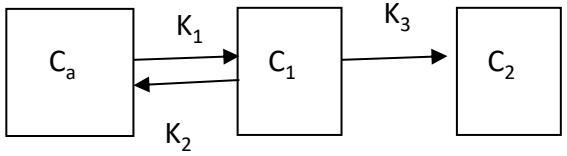
Table 1. Definition of symbols:

$C_i(t)$	True myocardial tissue radioactivity concentration at time *t*; radioactivity of perfusable myocardium [kBq ml^{-1}]
$a(t)$	True input function; radioactivity concentration in (coronary) arterial blood [kBq ml^{-1}]
$ROI(t)$	Time-activity curve of radioactivity of region-of-interest (ROI) which is drawn on the left ventricular (LV) myocardial region [kBq ml^{-1}]
$LV(t)$	Time-activity curve of radioactivity of ROI which is drawn on the LV cavity [kBq ml^{-1}]
f	Regional MBF; the blood flow of perfusable tissue [ml min^{-1} ml^{-1}]
p	Myocardium-to-blood partition coefficient of water [ml ml^{-1}]
α	Tissue fraction; volume of perfusable tissue in ROI [ml ml^{-1}]
V_a	Arterial blood volume; volume of arterial vascular space (including the spill-over from the chamber) in ROI [ml ml^{-1}]
β	Recovery coefficient of left-ventricular ROI ($0<\beta\leq 1$)
λ	Physical decay constant of ^{15}O [s^{-1}]

Modelling

UCLA Model

simplified 2-tissue compartment model (k3)



Fit parametres: , k_1,V_a
Fit range: ***2 min***

$$\frac{dC_1}{dt} = k_1 C_a - (k_2 + k_3) C_1$$

$$\frac{dC_2}{dt} = k_2 C_1$$

$$\frac{k_1}{k_2} = 0.8$$

$$C_{PET} = V_a C_m + (1 - V_a)(C_1 + C_2)$$

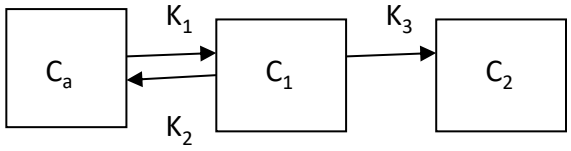
$$k_1 = f$$

Extraction fraction

$$K_3 = f\left(1.65 e^{\frac{1.25}{f}} - 1\right)$$

Michigan Model

full 2-tissue compartment model (k3)



k_1, k_2, k_3, V_a
10 min

$$\frac{dC_1}{dt} = k_1 C_a - (k_1 + k_2) C_1$$

$$\frac{dC_2}{dt} = k_3 C_1$$

$$C_{PET} = V_a C_m + (1 - V_a)(C_1 + C_2)$$

C_a is metabolite corrected C_m

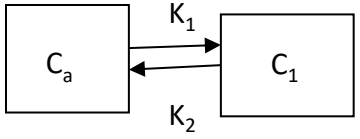
C_m is image - based LV input function

Extraction fraction option

1. $K_1 = f\left(1 - e^{-(108 + 2.34 f)/f}\right)$
2. Flow $= K_1$

Duke Model

1-tissue compartment model



k_1, k_2,V_a
4 min

$$\frac{dC_1}{dt} = k_1 C_a - k_2 C_1$$

$$C_{PET} = V_a C_m + (1 - V_a) C_1$$

Input function metabolite correction options:

1. van den Hoff

$C_a(t) = C_m(t)$ if $t <= t_{1/2}$

$C_a(t) = e^{-\ln 2\left(\frac{t - t_0}{t_{1/2}}\right)} C_m(t)$ if $t > t_{1/2}$

$t_0 = 0.48$ min

$t_{1/2} = 6.69$ min

2. DeGrado

$C_a(t) = C_m(1 - 0.4t)$ for dog

$C_a(t) = C_m(1 - 0.07t)$ for human

time unit is minute

Models implemented in Carimas

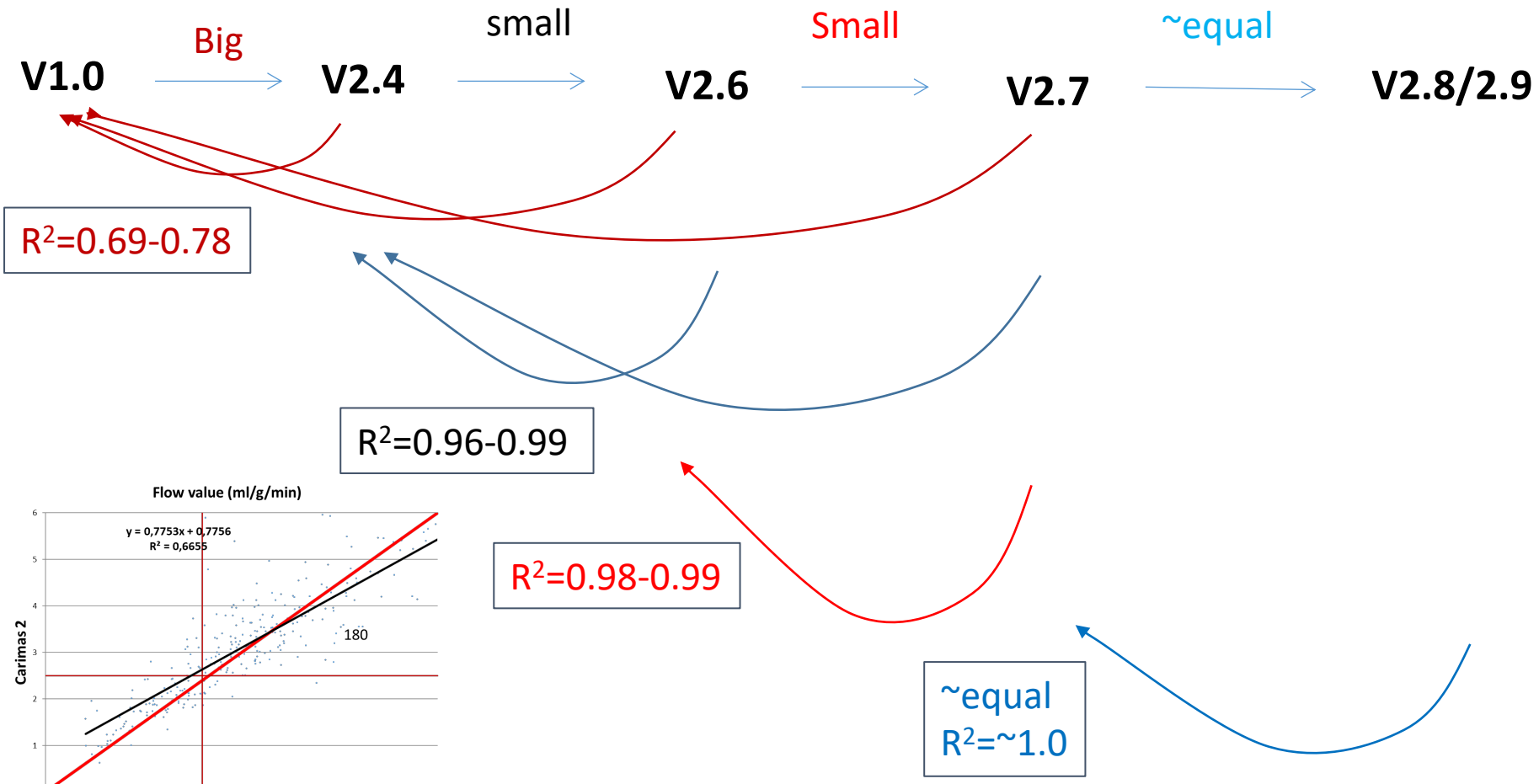
1. Compartment model
   1. Tracer-specific model for heart
      1. 15O labeled water perfusion model (1TCM)
      2. 11C labeled acetate perfusion model (1TCM)
      3. 82Rb rubidium perfusion model (1TCM)
      4. 13N labeled ammonia perfusion model
         1. UCLA model (simplified 1TCM)
         2. Duke model (simplified 2TCM-K3)
         3. Michigan model (full 2TCM)
      5. 18F labeled flurpiridaz perfusion model (2TCM,K3)
   2. General model
      1. 1-tissue compartment model
      2. 2-tissue compartment model(k3)
      3. 2-tissue compartment model (k4)
      4. 3-tissue compartment model in parallel
      5. 3-tissue compartment model in series
2. Graphical analysis method
   1. Patlak
   2. Logan
   3. Kmono

TCM=tissue compartment model

Summary of 3 common-used N13-ammonia models

Quality control, test and validation

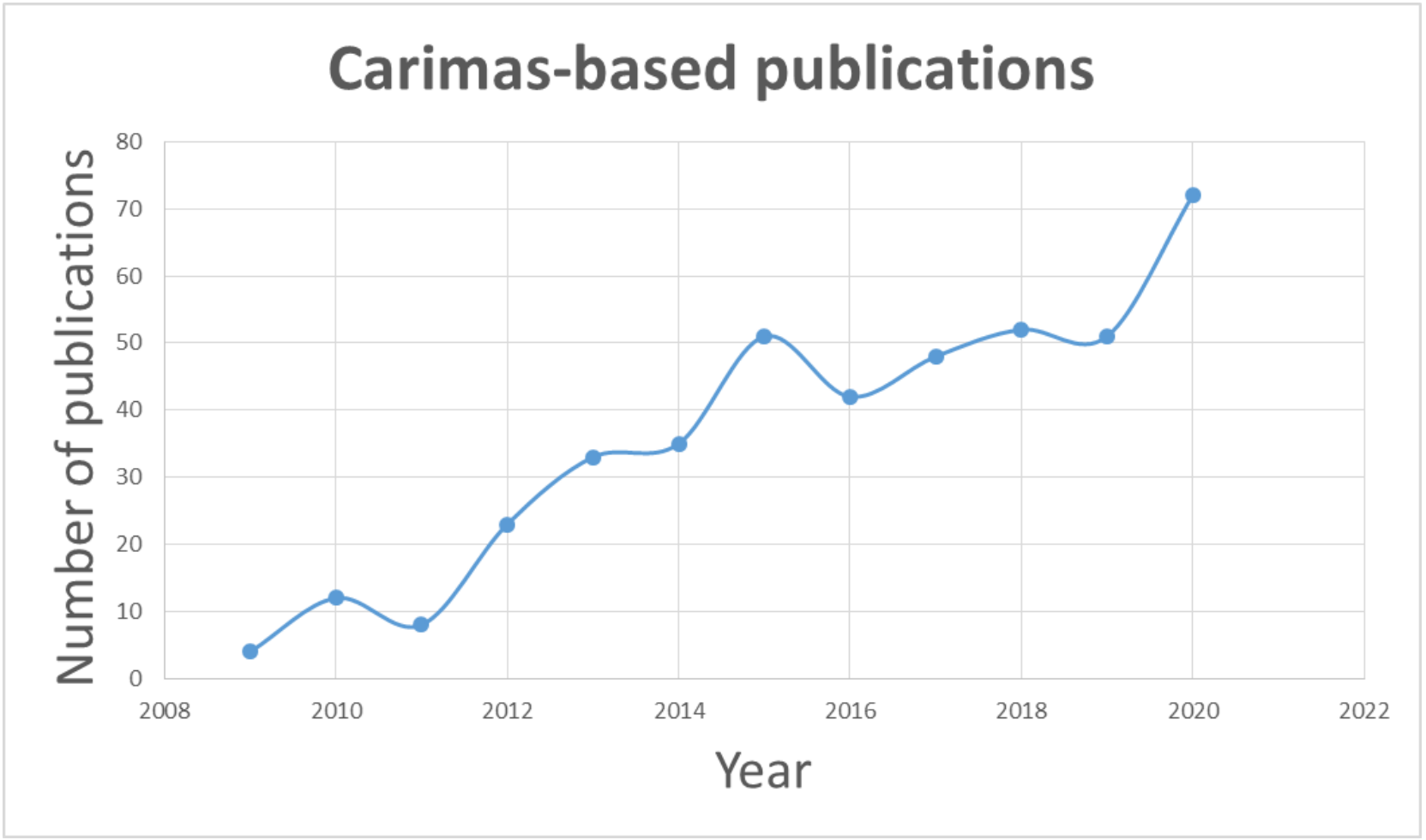
Correlation of flow values from water study in differnt Carimas versions



Contents

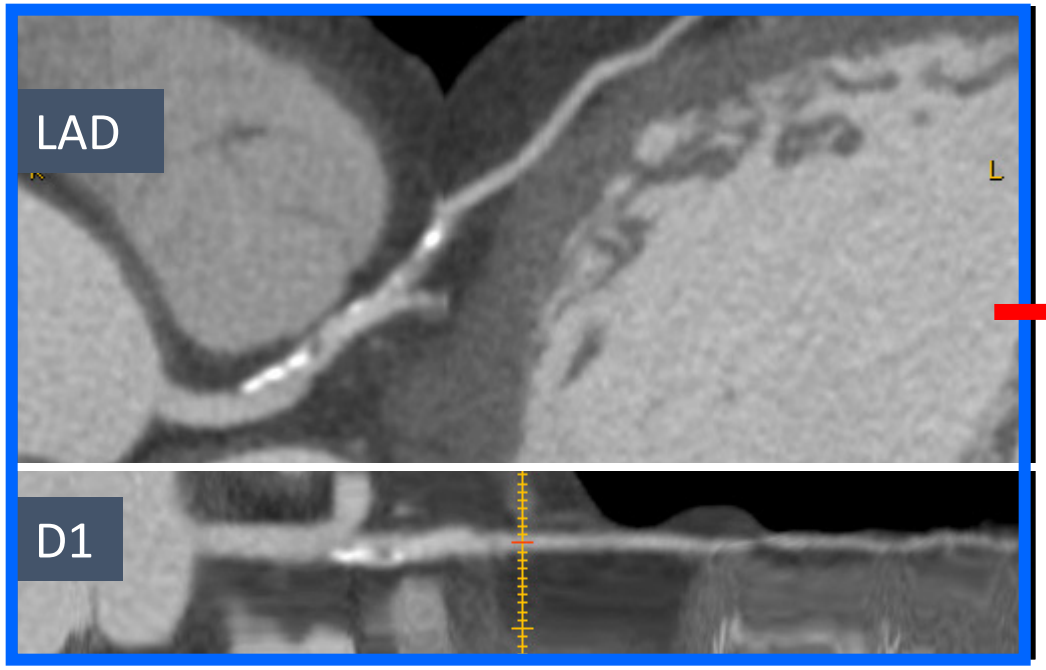
- General info: project purpose, website and license
- Features
- Program structure
- ***Scientific applications, publications and examples***
- Demonstration

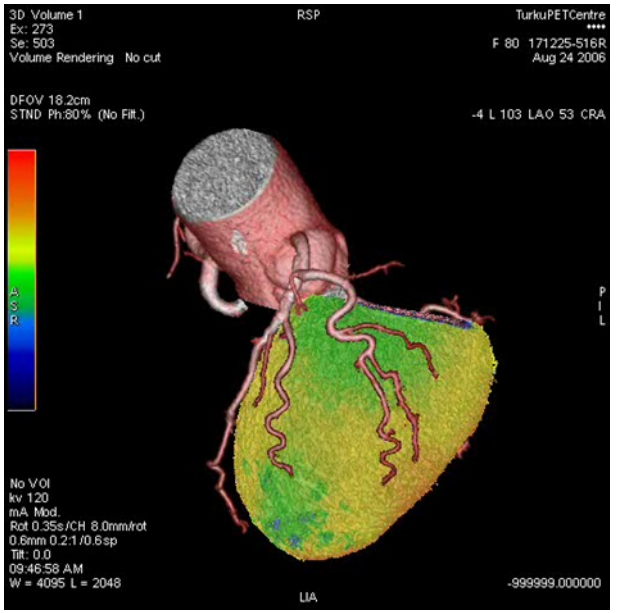
Carimas-based publication

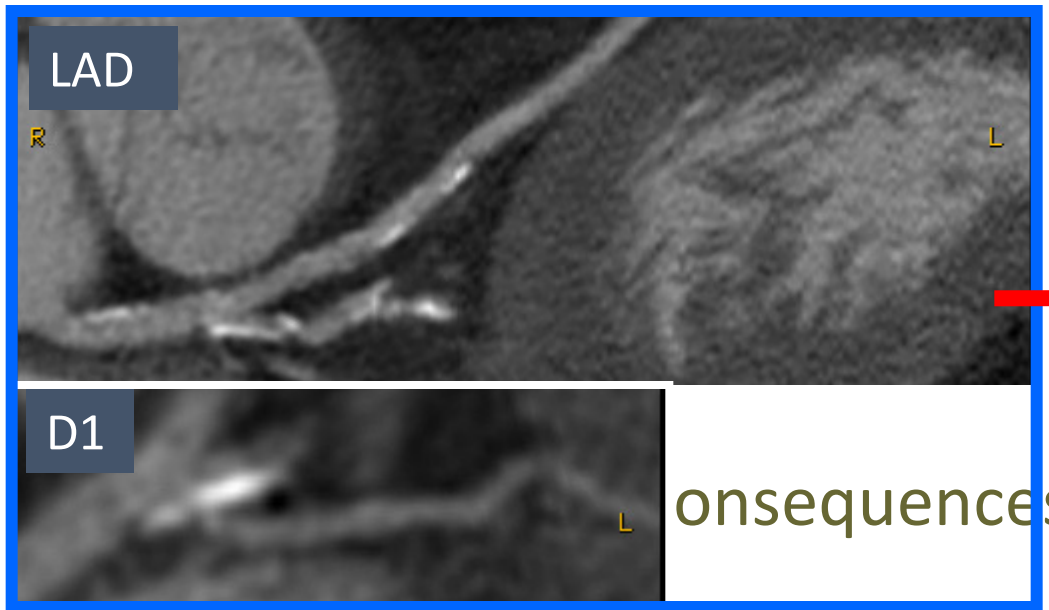


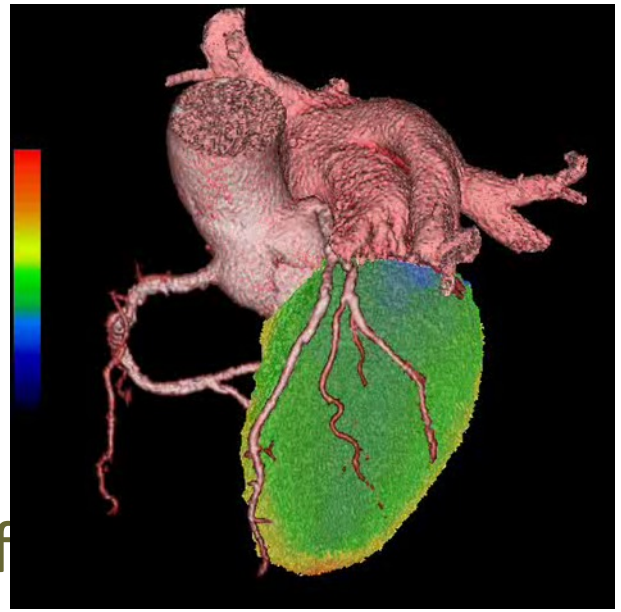
Total:	431
Included:	article, abstract, thesis
Keyword:	Carimas
Field:	all (tittle, abstract, body, reference etc)
Search engine:	Google scholar

heart









onsequences of

Turku PET Centre, Finland

ESC European Society of Cardiology

European Heart Journal (2017) **38**, 2547–2555
doi:10.1093/eurheartj/ehx352

CLINICAL RESEARCH
Coronary artery disease

Impact factor: 22.6(2019)

Adenoviral intramyocardial VEGF-D$^{\Delta N \Delta C}$ gene transfer increases myocardial perfusion reserve in refractory angina patients: a phase I/IIa study with 1-year follow-up

Juha Hartikainen[1,2], Iiro Hassinen[1], Antti Hedman[1], Antti Kivelä[1], Antti Saraste[3], Juhani Knuuti[3], Minna Husso[4], Hanna Mussalo[4], Marja Hedman[1,4], Tuomas T. Rissanen[1,5], Pyry Toivanen[6], Tommi Heikura[6], Joseph L. V Sotirios Tsimikas[7], and Seppo Ylä-Herttuala[1,6,8]*

[1]Heart Center, Kuopio University Hospital, Kuopio 70029, Finland; [2]Institute of Clinical Medicine, University of Eastern Finland, Kuopio 70211, Finl University Hospital, Turku 20521, Finland; [4]Center of Diagnostic Imaging, Kuopio University Hospital, Kuopio 70029, Finland; [5]Heart Center, Centr Joensuu 80210, Finland; [6]A.I. Virtanen Institute, University of Eastern Finland, Kuopio 70211, Finland; [7]University of California San Diego, La Jolla, Therapy Unit, Kuopio University Hospital, Kuopio 70029, Finland

Received 22 December 2016; revised 30 March 2017; editorial decision 12 May 2017; accepted 2 June 2017; online publish-ahead-of-print 31 July 2017

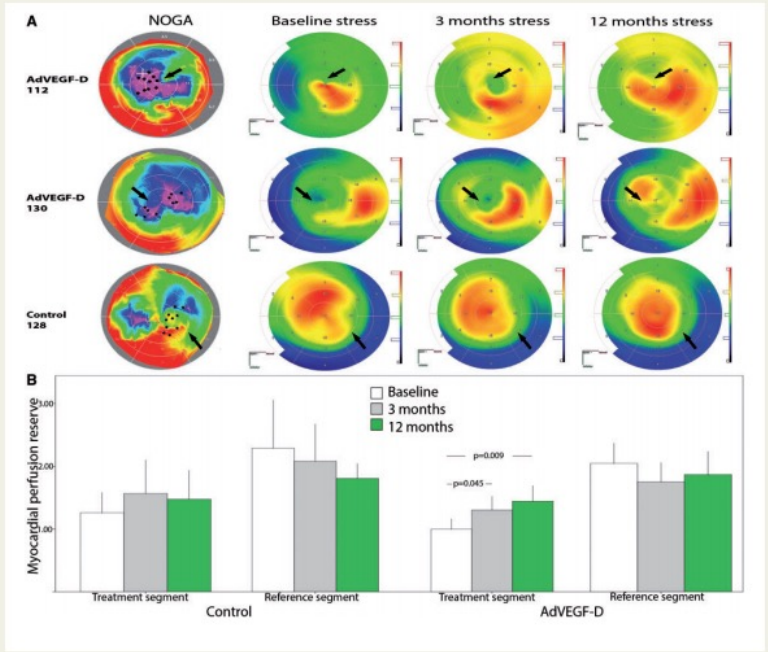
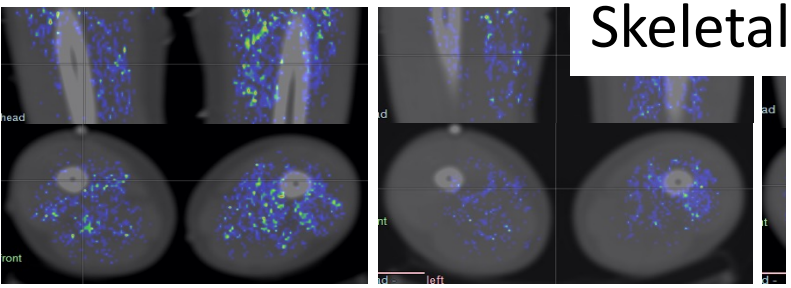
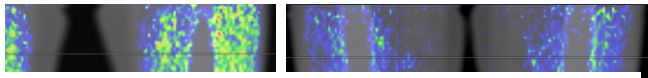


Figure 3 (*A*) Representative images of combined NOGA and stress PET radiowater images of two AdVEGF-D treated patients and one control patient. Black dots and arrows indicate sites for gene injections in viable but poorly perfused myocardium. Myocardial blood flow improved in the AdVEGF-D patients visualized as increases in red colour during the follow-up. Perfusion did not increase in the control patient. (*B*) Myocardial perfusion reserve in the treated and reference segments of the control and AdVEGF-D$^{\Delta N \Delta C}$-treated patients. Colour scales in NOGA and PET maps as in *Figure 1*. Values are mean ± standard deviation.

Skeletal muscle perfusion



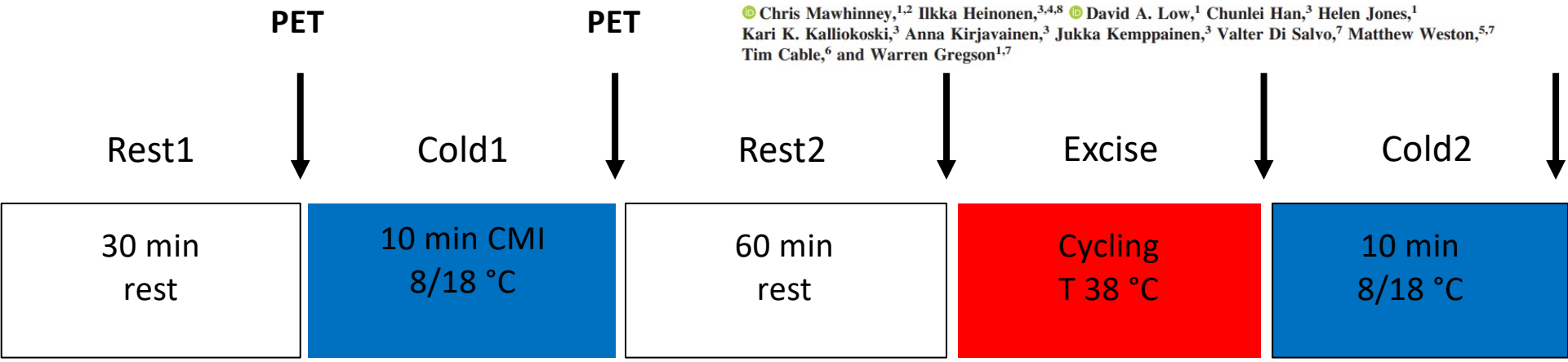


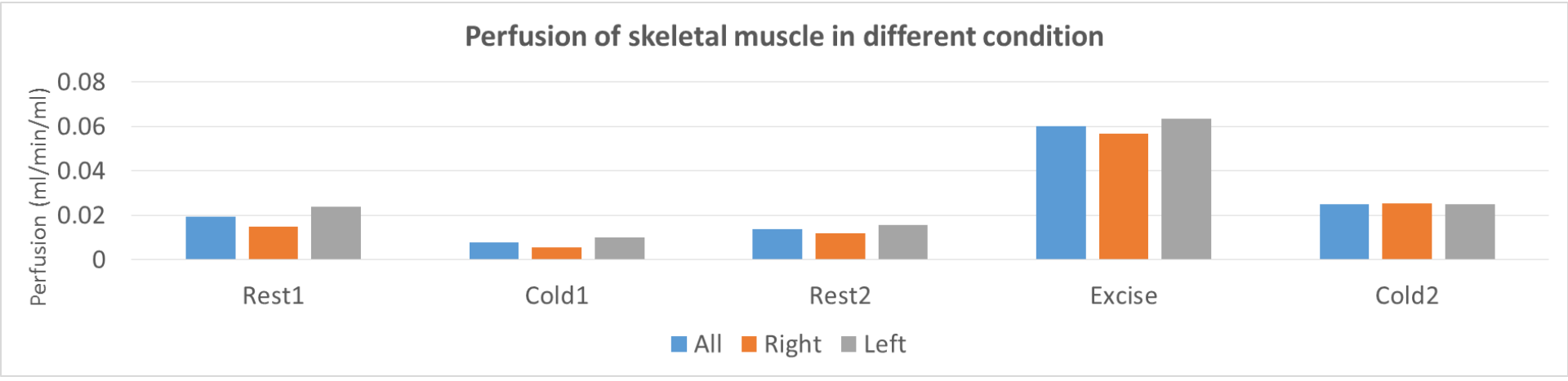
J Appl Physiol 128: 1392–1401, 2020.
First published April 30, 2020; doi:10.1152/japplphysiol.00833.2019.

RESEARCH ARTICLE

Changes in quadriceps femoris muscle perfusion following different degrees of cold-water immersion

Chris Mawhinney,[1,2] Ilkka Heinonen,[3,4,8] David A. Low,[1] Chunlei Han,[3] Helen Jones,[1] Kari K. Kalliokoski,[3] Anna Kirjavainen,[3] Jukka Kemppainen,[3] Valter Di Salvo,[7] Matthew Weston,[5,7] Tim Cable,[6] and Warren Gregson[1,7]





J Appl Physiol 128: 1392–1401, 2020.
First published April 30, 2020; doi:10.1152/japplphysiol.00833.2019.

RESEARCH ARTICLE

Changes in quadriceps femoris muscle perfusion following different degrees of cold-water immersion

Chris Mawhinney,[1,2] Ilkka Heinonen,[3,4,8] David A. Low,[1] Chunlei Han,[3] Helen Jones,[1] Kari K. Kalliokoski,[3] Anna Kirjavainen,[3] Jukka Kemppainen,[3] Valter Di Salvo,[7] Matthew Weston,[5,7] Tim Cable,[6] and Warren Gregson[1,7]

[1]Research Institute for Sport and Exercise Sciences, Liverpool John Moores University, Liverpool, United Kingdom; [2]College of Sports Science and Technology, Mahidol University, Salaya, Thailand; [3]Turku PET Centre, University of Turku and Turku University Hospital, Turku, Finland; [4]Department of Clinical Physiology and Nuclear Medicine, University of Turku, Turku, Finland; [5]School of Health and Social Care, Teesside University, Middlesbrough, United Kingdom; [6]School of Sport, Exercise and Rehabilitation Sciences, University of Birmingham, Birmingham, United Kingdom; [7]Football Performance and Science Department, Aspire Academy, Doha, Qatar; and [8]Rydberg Laboratory of Applied Sciences, University of Halmstad, Halmstad, Sweden

Submitted 2 December 2019; accepted in final form 23 April 2020

Brain function

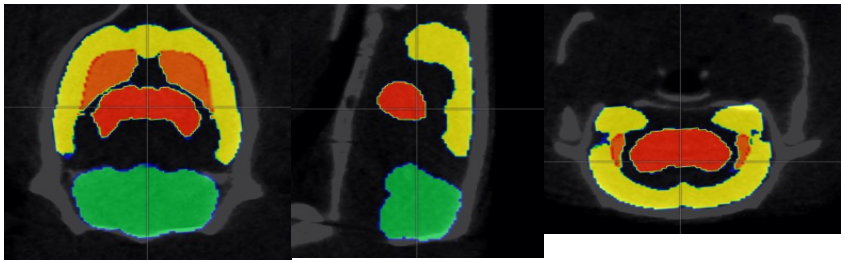
Seasonal Variation in the Brain μ-Opioid Receptor Availability

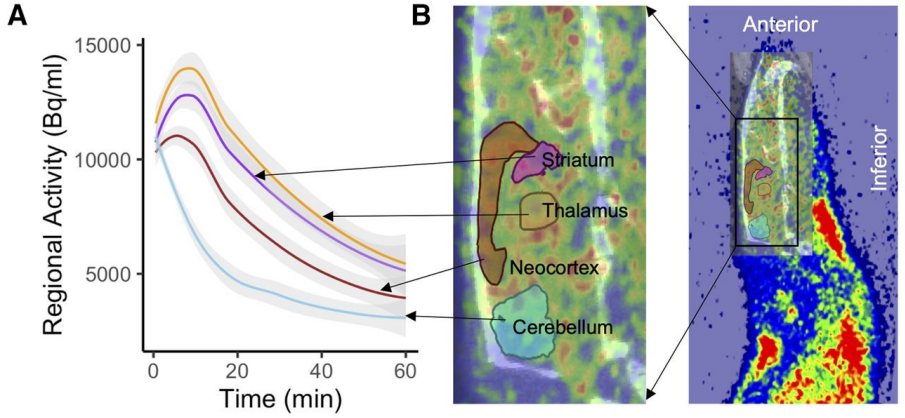
Lihua Sun, Jing Tang, Heidi Liljenbäck, Aake Honkaniemi, Jenni Virta, Janne Isojärvi, Tomi Karjalainen, Tatu Kantonen, Pirjo Nuutila, Jarmo Hietala, Valtteri Kaasinen, Kari Kalliokoski, Jussi Hirvonen, Harry Scheinin, Semi Helin, Kim Eerola, Eriika Savontaus, Emrah Yatkin, Juha O. Rinne, Anne Roivainen, and Lauri Nummenmaa

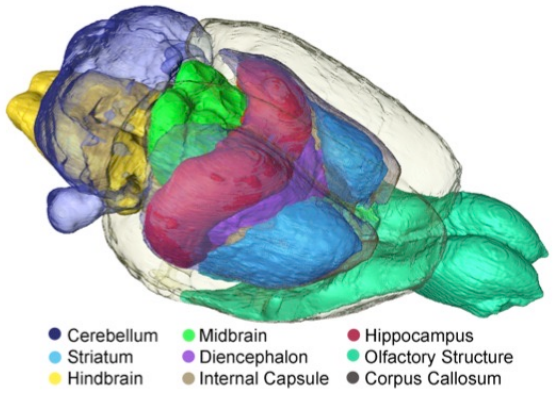
Journal of Neuroscience 10 February 2021, 41 (6) 1265-1273; DOI: https://doi.org/10.1523/JNEUROSCI.2380-20.2020

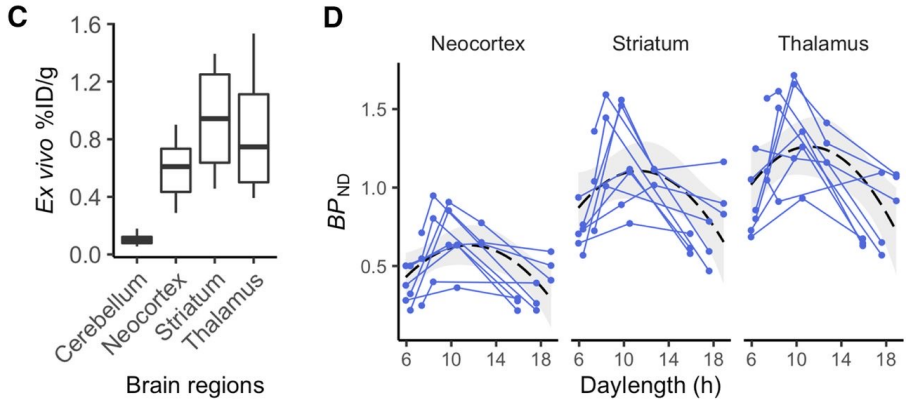
Dynamic PET images were analyzed using Carimas software (version 2.10.3.0) developed at the Turku PET Center. The PET datasets were reconstructed in 20 time frames using the OSEM3D

Journal of Neuroscience: IF=6.16



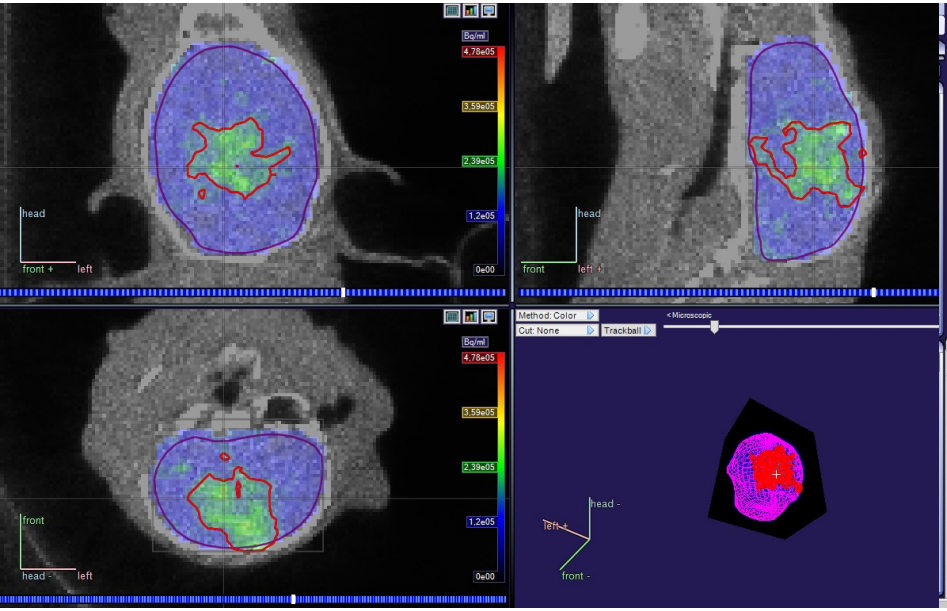


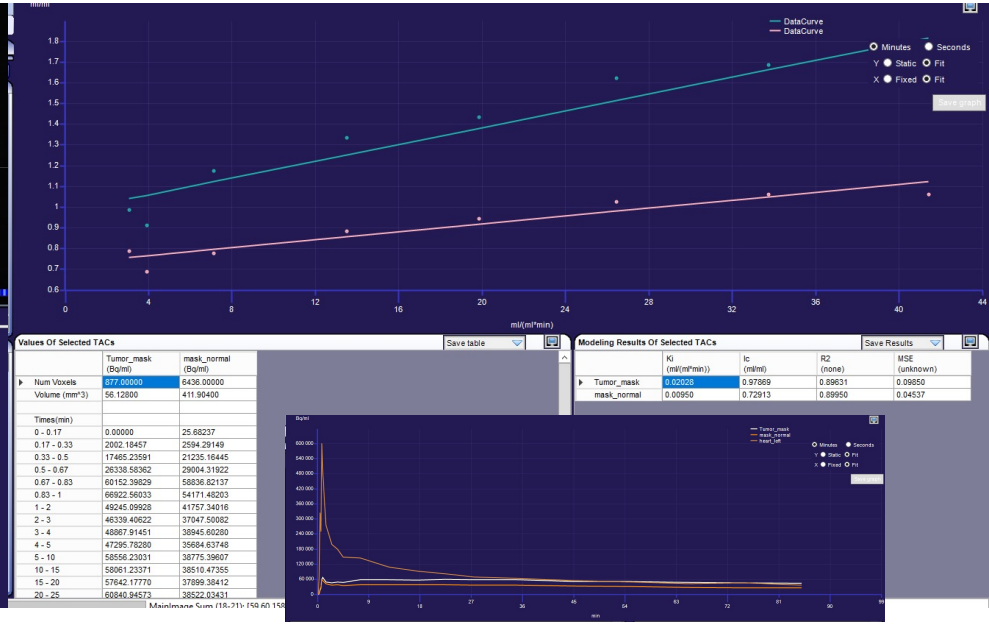




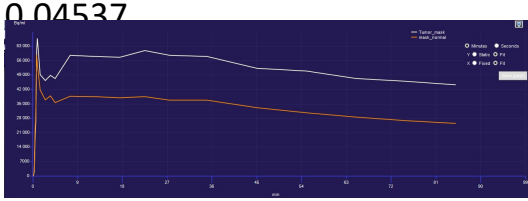
Brain tumour

Patlak





	Ki (ml/(ml*min))	Ic (ml/ml)	R2 (none)	MSE (unknown)
Tumor	0.02028	0.97869	0.89631	0.09850
Normal	0.00950	0.72913	0.89950	0.04537



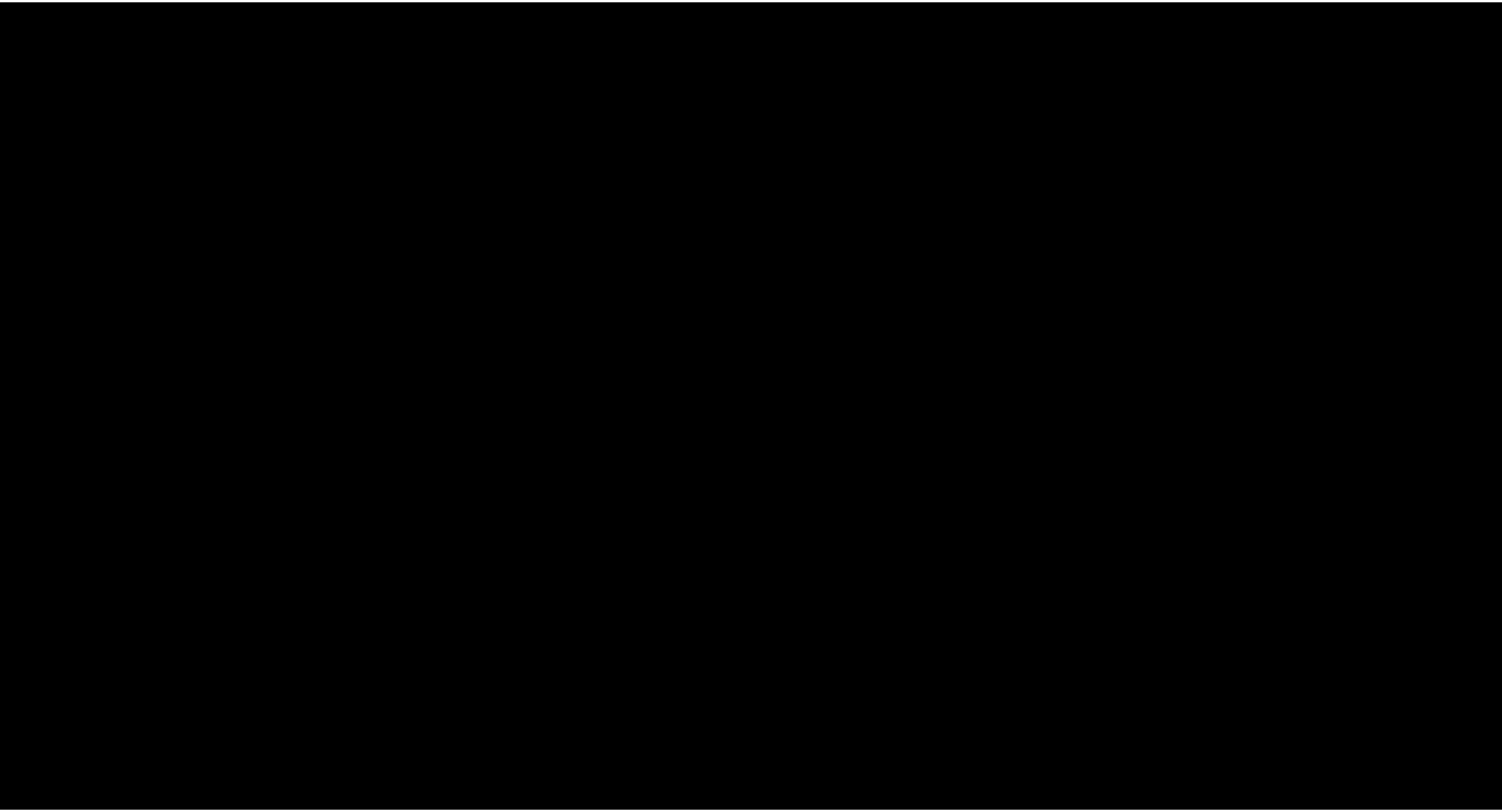
High Ki in tumour means that uptake in tumor tissue is than normal tissue

Contents

- General info: project purpose, website and license
- Features
- Program structure
- Scientific applications, publications and examples
- ***Demonstration***

Demo

Scanner: Molecube, PET/CT
Animal: two mice
Tumour:



Conclusions

- Carimas is useful tools for medical imaging-related researchers.
- Carimas is free for academic users.
- Additional functions can be very easily extended by plugin.


Kiitos
Thank for your attention