PMOD Artificial Intelligence Framework (PAI)

USER MANUAL
Version 4.4

PMOD is a software
FOR RESEARCH USE ONLY (RUO)
and must not be used for diagnosis or treatment of patients.

1 Introduction

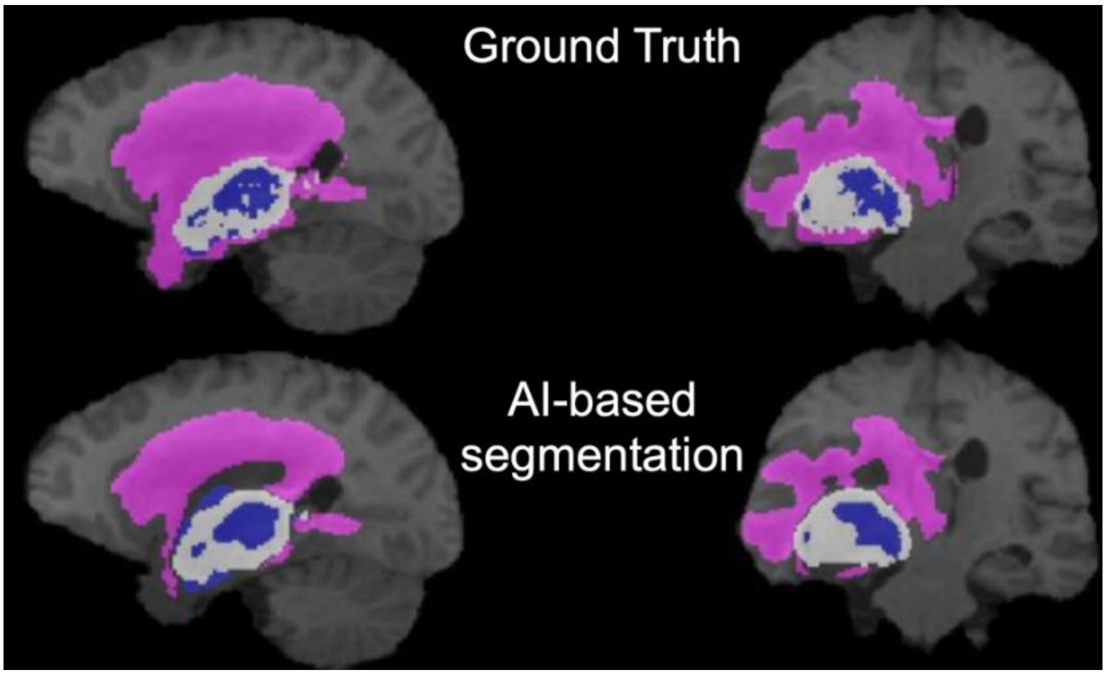
Computer algorithms based on Artificial Intelligence (AI) have become a success story and are now part of daily life. Machine Learning (ML) is a subset of AI in which humans provide the input data and expected results, and the computer determines "rules" with which it can process the data to approach the expected results. These "rules" may be considered as representations of the data. Deep Learning (DL) is considered a further subset of ML, in which there are successive layers of representations. ML methods have resulted in solutions ranging from facial recognition to web-based language translation. They have also been applied in many domains of biomedical research. However, the setup and use of ML toolkits is a task requiring a lot of methodological insight as well as specialized IT expertise.

The aim of PAI is to drastically lower the entry barrier to the ML methodology for researchers analyzing biomedical images. PAI is designed as a framework, allowing users to develop their own tailored ML-based image segmentation solution while working entirely within the familiar PMOD environment.

1.1 PAI Purpose

Segmentation of pathology or of organs/regions that do not conform to common templates can be a tedious, time-consuming and subjective procedure. The use of machine learning to automatically perform such a segmentation has the potential to save large amounts of time and improve reproducibility.

In the example shown below, regions of necrotic, gadolinium-enhancing and penumbra of a brain tumor are shown in color on a gray-scale anatomical T1-weighted MR image. Contrast from four MR series (T1-weighted, T2-weighted, T2-FLAIR, gadolinium-enhanced T1-weighted) were used to define these regions. For the top row, created by an expert reviewer using manual segmentation, this process takes many minutes or even hours. In the bottom row, the broadly similar segmentation result was generated by a trained convolutional neural network and took seconds.



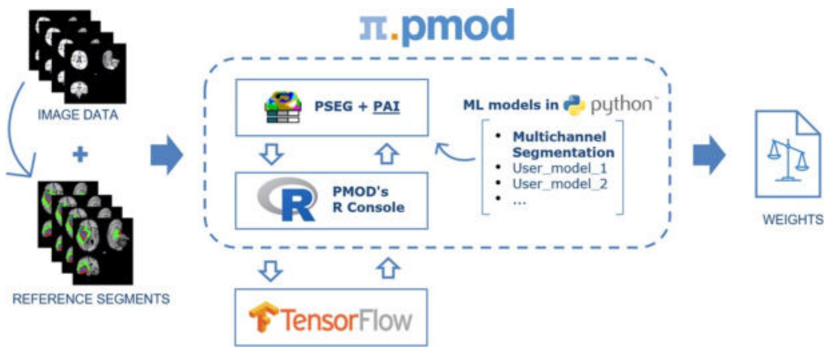
However, training a convolutional neural network to perform such a segmentation task requires a substantial amount of data, time and effort.

PMOD's PAI framework aims to make training and deploying ML-based segmentation more accessible to non-expert users. PMOD's well-tested tools for image processing and traditional segmentation provide an excellent base to prepare the training data needed for supervised machine learning.

In addition to AI-based segmentation, PAI also supports classification tasks. An example of classification in imaging is the assignment of a label "amyloid positive" or "amyloid negative" to amyloid PET images for tracers such as ^{11}C-PiB. PMOD's database functionality provides the base to organise data into classes for training of a neural network. The trained neural network then returns a probability of a new image belonging to a given class.

1.2 PAI Overview

The structure of the PAI framework is illustrated below.



The actual ML platform used by PAI is the well-known TensorFlow solution. The neural network structure and the training method are correspondingly developed as Python scripts suitable for TensorFlow and constitute the ML model. The data for supervised learning are prepared in PAI, communication with TensorFlow is implemented via the R console in PMOD.

Learning Set

The **Learning Set** in PAI consists of references to the training data (i.e. Links to the input series in a PMOD database) and a specification of the preprocessing steps required to bring the data into a format suitable for machine learning. The training data itself consists of data samples. Each data sample consists of an input (one or several images) and its expected segmentation or classification result (one or several segments, in the format of label maps that can be associated with the input images, or label for a given class).

Training

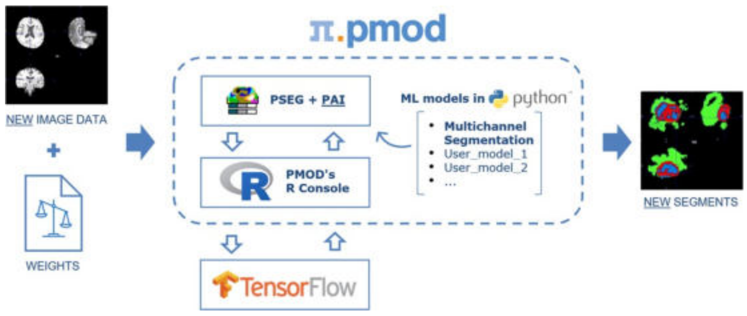
Training is performed in TensorFlow using a Learning Set and an ML model. There are different mechanisms available, which will be explained below. Basically, training can be performed locally on your local machine, or delegated to a more powerful infrastructure such as Cloud-Computing.

Training Result

The result of training is a "trained model" - a set of **Weights** for the layers in the neural network, and a **Manifest** file containing information about the training process. These results are added to the **Learning Set**, making it ready for use in **Prediction** (i.e. automated segmentation or classification). As new training data become available, it can be added to the Learning Set and incremental training performed to improve prediction. An export functionality allows transfer of the result to other PMOD installations for prediction (sometimes known as Deployment).

Prediction

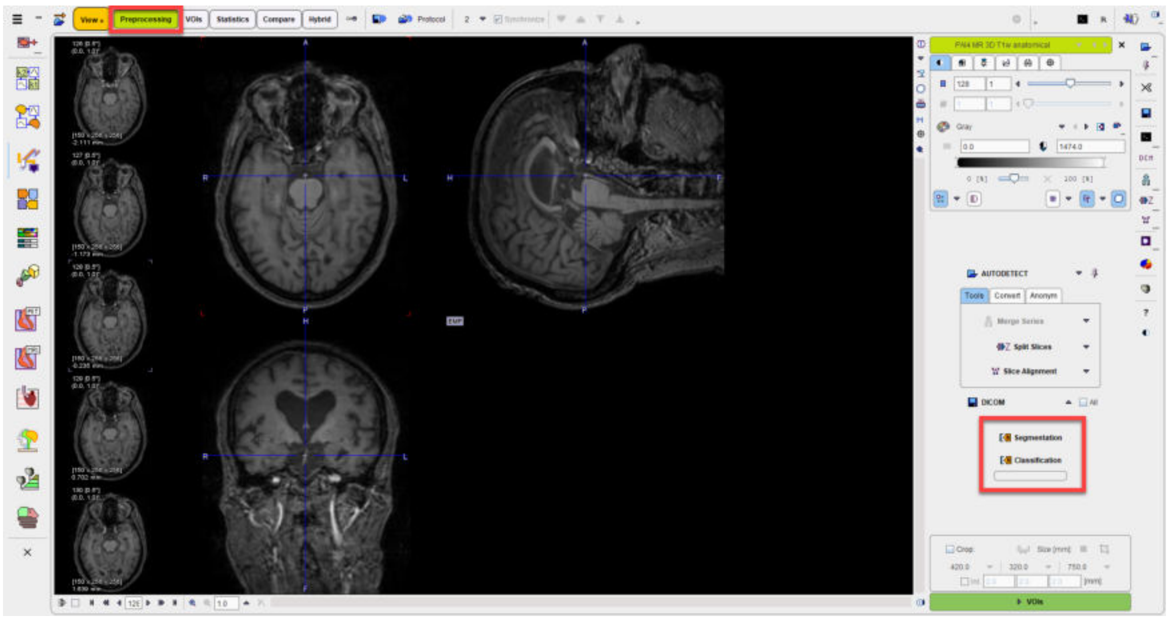
Prediction applies the trained model to a new set of input image data, resulting in a segmentation or classification result. Segmentation results may then be converted to VOIs and used for quantification.

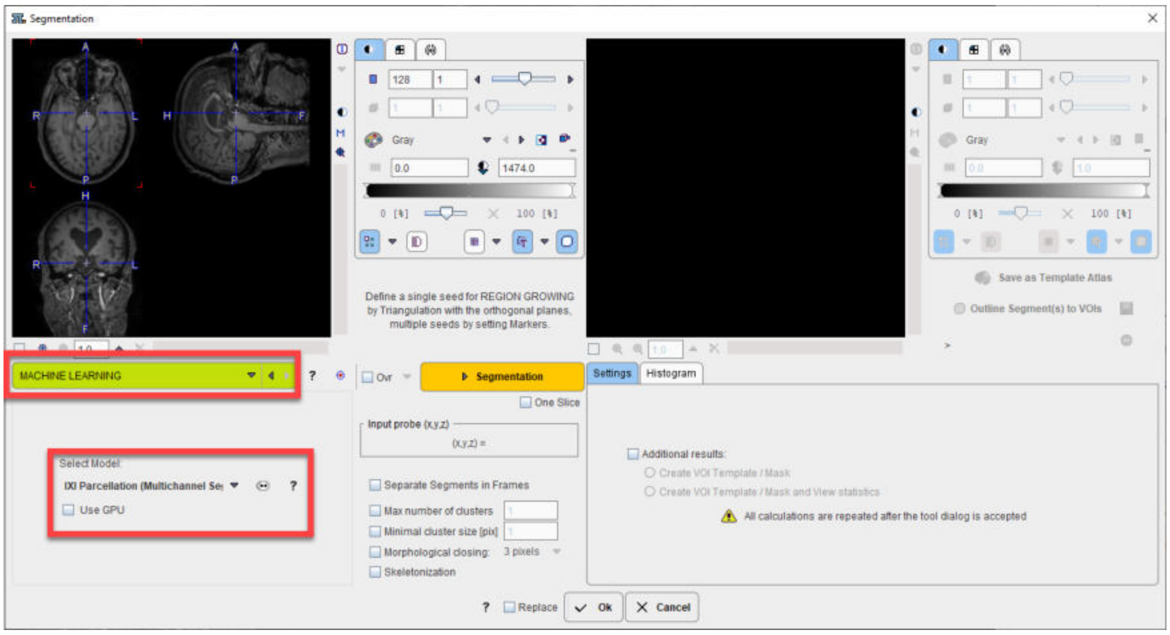


Implementation in PMOD tools

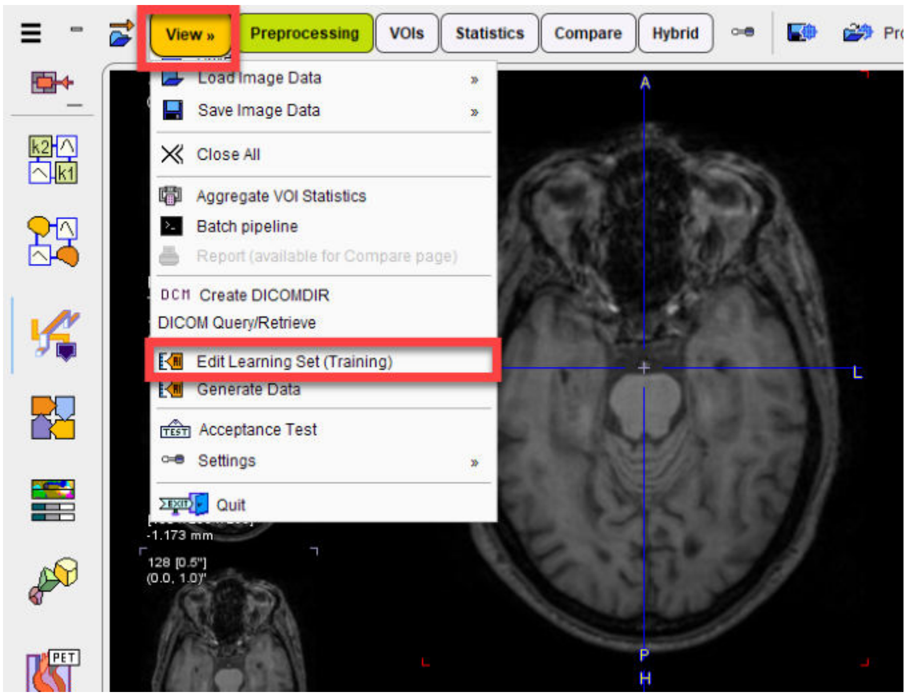
PAI functionality is available in PBAS, PSEG, PCARDM and PNEURO.

In View, shortcuts to AI Segmentation and Classification become available on the Prprocess tab, and Machine Learning is available as an option in the Segmentation interface:

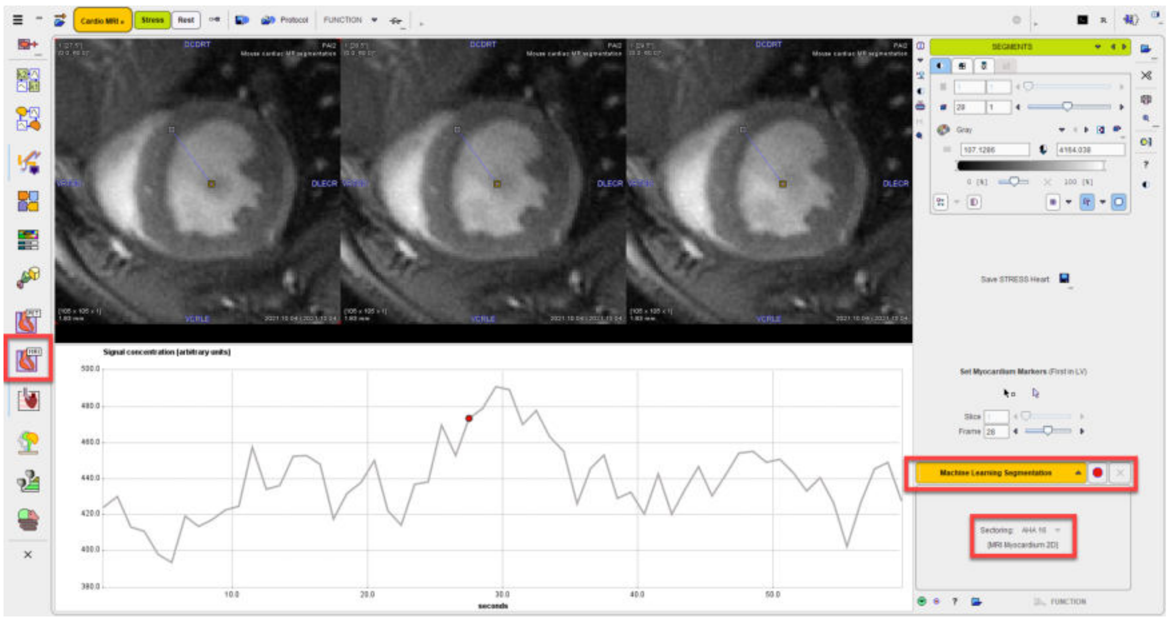



Segmentation
MACHINE LEARNING
Select Model:
IXI Parcellation (Multichannel Se
Use GPU
Segmentation
Define a single seed for REGION GROWING by Triangulation with the orthogonal planes, multiple seeds by setting Markers.
Settings
Histogram
Additional results:
All calculations are repeated after the tool dialog is accepted
Replace
Ok
Cancel

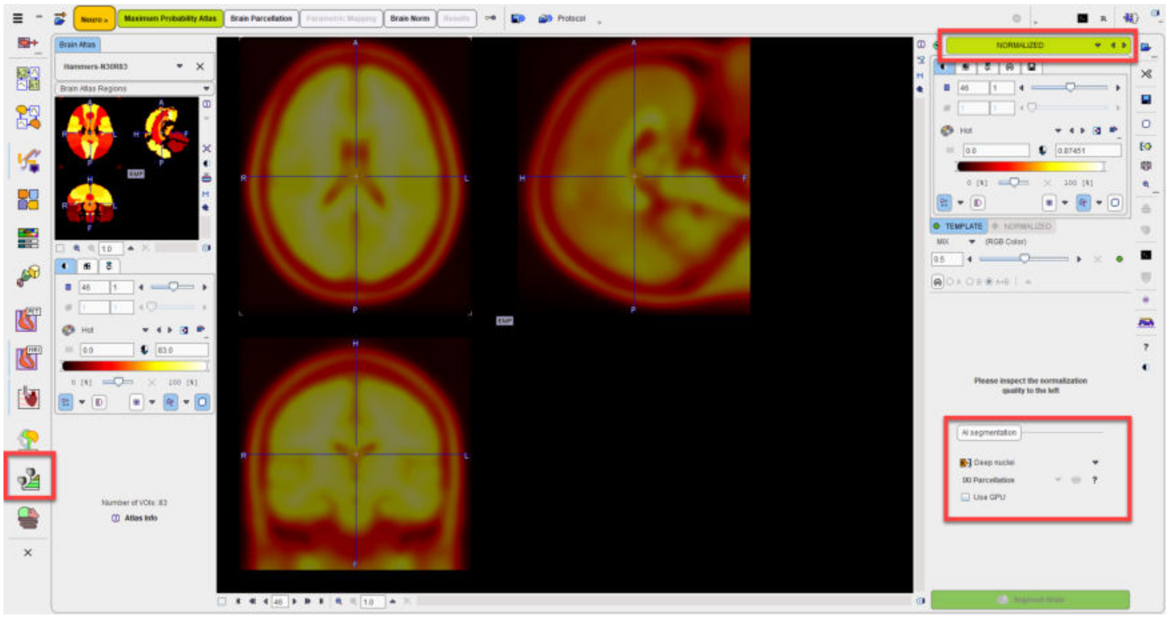
Learning Sets for AI projects are prepared via the Edit Learning Set option in the main View tool menu:



In PCARDM the Machine Learning Segmentation option becomes available for either MRI_Myocardium_2D (mouse) or MRI_Human_Myocardium model:



In PNEURO, the IXI Parcellation model is available for replacement of the deep nuclei segments on the Normalized page in the Maximum Probability Atlas workflow:



1.3 Architectures included in Distribution

Multiple neural network architectures are provided with PAI. Some have already been trained with appropriate data as part of our Case Studies. The architectures are available in the Pmod4.4/resources/pai folder, labeled by type.

1.3.1 uNET

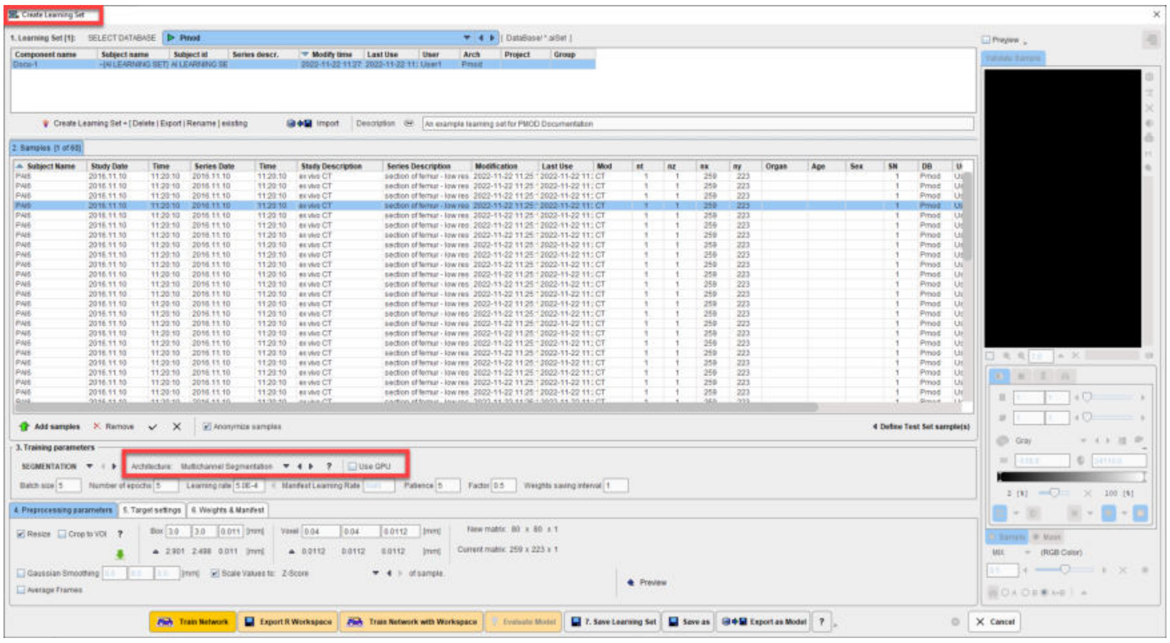
uNET is a convolutional neural network developed for image segmentation (Ronneberger et al., 2015): https://en.wikipedia.org/wiki/U-Net

A modified uNET was used by Isensee et al. (2018) in the MICCAI Brain Tumor Segmentation Challenge (BraTS). The BraTS data and this modified uNET were used for our first case study and the Multichannel Segmentation architecture for segmentation in PAI.

Additional case studies were made using this architecture:

- Rat brain dopaminergic PET segmentation
- Human brain MRI deep nuclei segmentation (IXI)
- Human cardiac MRI segmentation for function analysis

This architecture is available as **unet_002** in the Create Learning Set dialog:



Create Learning Set

1. Learning Set [1]: SELECT DATABASE Pmod [DataBase! * aiSet]

Component name	Subject name	Subject id	Series descr.	Modify time	Last Use	User	Arch	Project	Group
Doco-1	-[AI LEARNING SET] AI LEARNING SE			2022-11-22 11:27	2022-11-22 11:	User1	Pmod		

Create Learning Set + [Delete | Export | Rename | existing Import Description: An example learning set for PMOD Documentation

2. Samples [1 of 60]

Subject Name	Study Date	Time	Series Date	Time	Study Description	Series Description	Modification	Last Use	Mod
PA6	2016.11.10	11:20:10	2016.11.10	11:20:10	ex vivo CT	section of femur - low res	2022-11-22 11:25	2022-11-22 11:	CT

Add samples Remove Anonymize samples

3. Training parameters

SEGMENTATION Architecture: Multichannel Segmentation Use GPU

Batch size 5 Number of epochs 5 Learning rate 5.0E-4 Manifest Learning Rate Patience 5 Factor 0.5 Weights saving interval 1

4. Preprocessing parameters | 5. Target settings | 6. Weights & Manifest

Resize ✓ Crop to VOI Box 3.0 3.0 0.011 [mm] Voxel 0.04 0.04 0.0112 [mm] New matrix: 80 x 80 x 1

2.901 2.498 0.011 [mm] 0.0112 0.0112 0.0112 [mm] Current matrix: 259 x 223 x 1

Gaussian Smoothing [mm] Scale Values to: ✓ Z-Score of sample.

Average Frames Preview

A classical uNET architecture is also available for segmentation projects. It is available as **unet_001** in the Create Learning Set dialog.

This architecture is used for mouse cardiac MRI segmentation in the PCARDM tool, and it has additionally been used for the following case study:

- mouse femur/tibia trabecular segmentation

The loss function used for both architectures is Dice Coefficient:

https://en.wikipedia.org/wiki/S%C3%B8rensen%E2%80%93Dice_coefficient

References:

Isensee et al. (2018), Brain Tumor Segmentation and Radiomics Survival Prediction: Contribution to the BRATS 2017 Challenge, https://arxiv.org/abs/1802.10508

Ronneberger et al. (2015), U-Net: Convolutional Networks for Biomedical Image Segmentation, https://arxiv.org/abs/1505.04597

1.3.2 Support Vector Machine (SVM)

The SVM architecture for classification in PAI is implemented in accordance with the Python library scikit-learn https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

Support vector machines are supervised learning models used for classification and regression:

https://en.wikipedia.org/wiki/Support-vector_machine

SVM is particularly suited to image reduction strategies such as using the average voxel value in a set of VOIs (e.g. brain atlas VOIs for brain PET data - see Amyloid PET classification case study). Linear, RBF, Sigmoid and Poly kernels are available according to the scikit-learn library (see Learning Set Preparation). Our testing of the architecture in PAI has used up to 10 classes.

The output of a trained SVM classification model is a probability that an input to prediction belongs to one of the classes included in training.

An example of the SVM architecture in practice is provided in our Amyloid PET classification case study. Data to test the Amyloid PET SVM classification model is available in our Demo database.

1.3.3 Generative Adversarial Network (GAN)

Generative adversarial networks, generally known as GAN, are used in image generation, often to generate an "improved" image from a given input. Examples include image denoising and creation of "super-resolution" images. GAN generally consist of a generator and a classifier (Discriminator). Both are trained separately. The role of the discriminator is to detect that the image provided from the generator is fake and the original image is real, but the role of the generator is to fool the discriminator by generating such good images that the discriminator classifies them as real. That is why they are called adversarial.

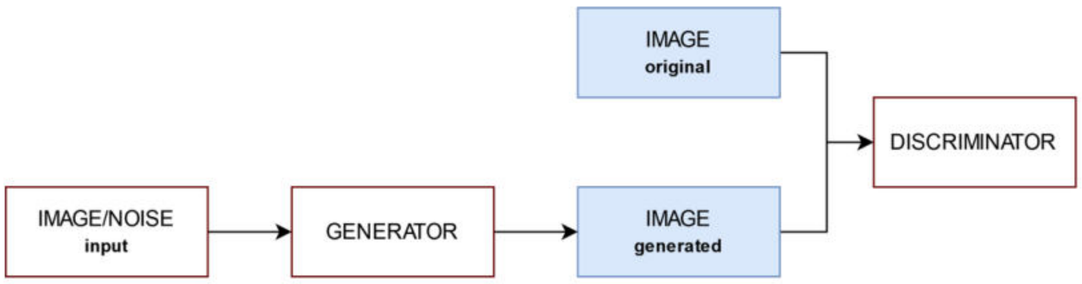
The first GAN available in PAI, StyleGAN2 (from noise), generates images from noise, so the input to the generator is a matrix of randomly generated noise of a fixed size (normal distribution, mean 0, variance 1). The second GAN available in PAI, Generate from uNet, is also known as Pix2Pix because a new image is generated from an existing one, for example ‘super-resolution’ images.

An important aspect is that the generator never sees the image it should generate, only the discriminator sees the original images.

Two GAN architectures are under development:

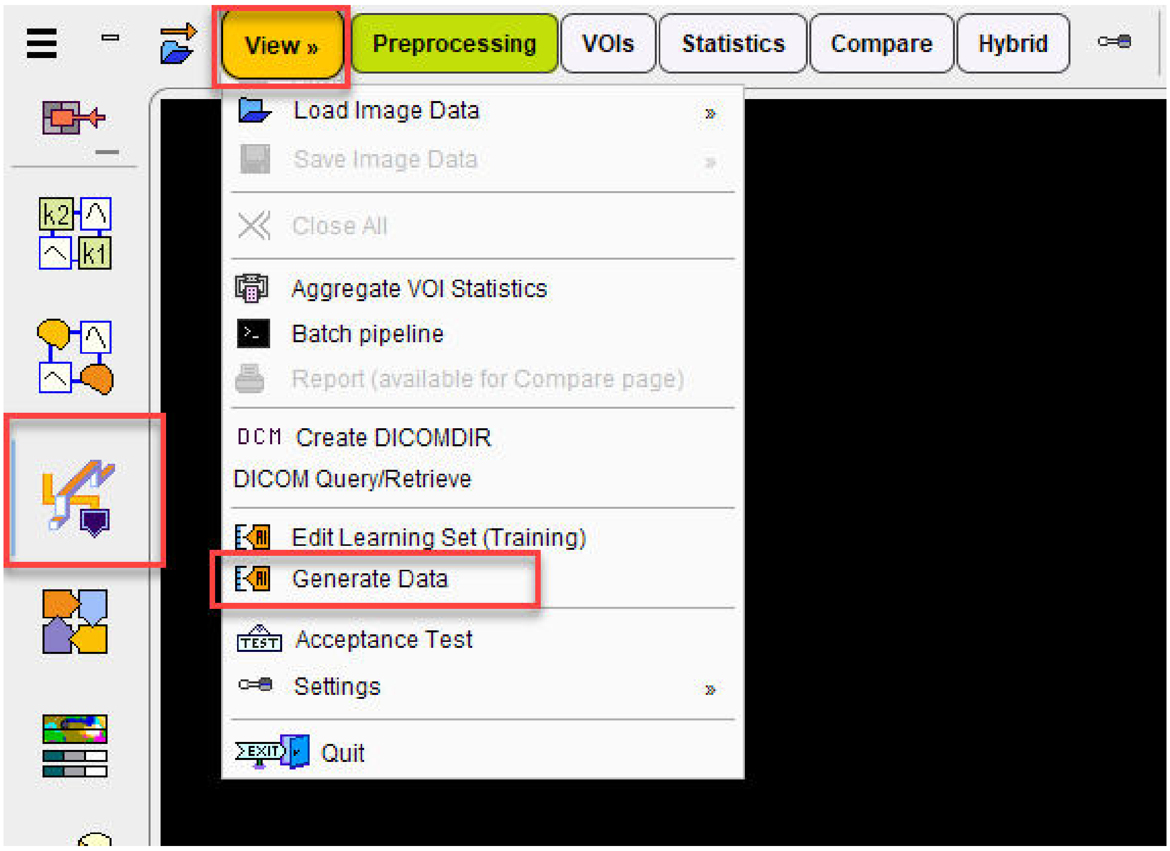
- Generate from uNet (the generator architecture is based on the classical uNet architecture)

- StyleGAN2 from noise (see Karras et al. [1])

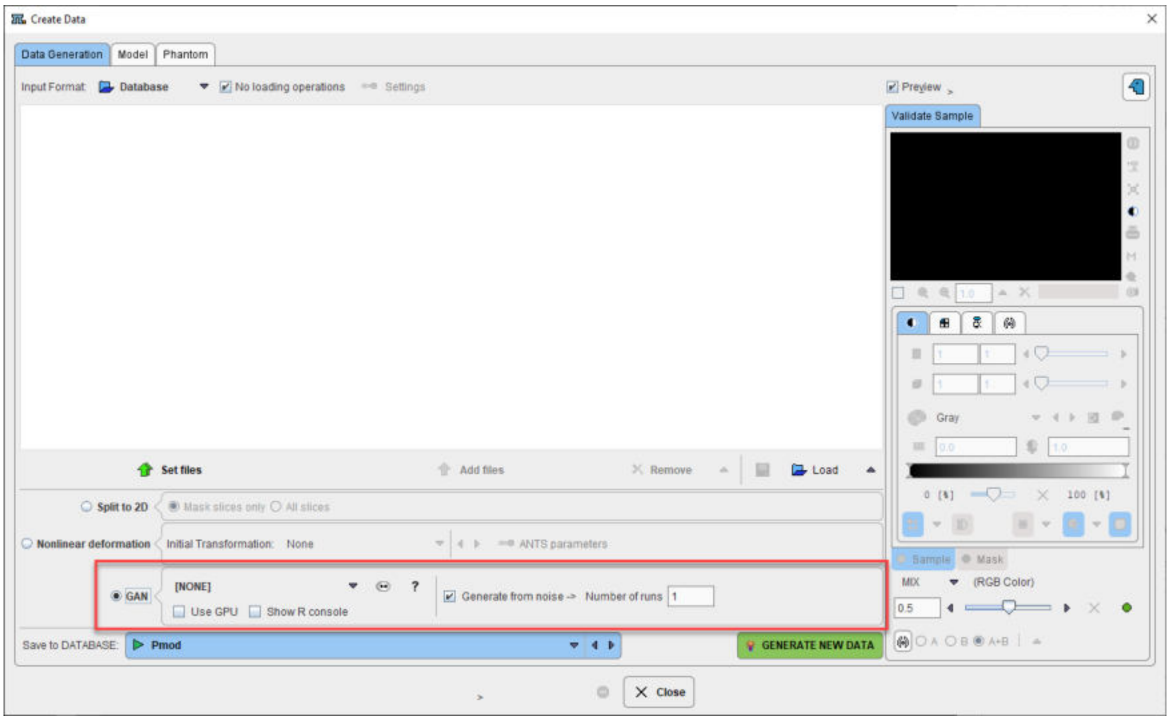


These architectures are only designed to accept 2D data. Contact us for more information.

Once trained, GAN are available in the Generate Data tool available in the View menu:



GAN accepts input images specified in the upper panel, or images can be generated from noise, depending on how the model was trained:



Input Format: Database
[✓] No loading operations
[✓] Preview
Split to 2D: ● Mask slices only ○ All slices
○ Nonlinear deformation: Initial Transformation: None
● GAN: [NONE]
☐ Use GPU ☐ Show R console
[✓] Generate from noise -> Number of runs 1
Save to DATABASE: Pmod
Gray 0.0 1.0
0 [%] 100 [%]
MIX (RGB Color) 0.5
○ A ○ B ● A+B

The generated images are saved to the Database specified.

References

[1] Karras et al., (2020) Analyzing and Improving the Image Quality of StyleGAN. https://arxiv.org/abs/1912.04958

1.3.4 ResNet50

ResNet-50 is written in TensorFlow, the architecture contains residual blocks and 50 layers.

The network is dedicated for classification. In PAI, the available ResNet-50 architecture is adapted for 2D and 3D data. This architecture is available as ResNet 50 in the Create Learning Set dialog. The loss function used for this architecture is categorical cross entropy.

References

[1] Keras repo, ResNet50 implementation

https://github.com/keras-team/keras-applications/blob/master/keras_applications/resnet50.py

[2] Priya Dwivedi, towards data science

https://towardsdatascience.com/understanding-and-coding-a-resnet-in-keras-446d7ff84d33

https://github.com/priya-dwivedi/Deep-Learning/blob/master/resnet_keras/Residual_Networks_yourself.ipynb

1.3.5 Image Classification

VGG-16 is written for 2D data in TensorFlow, and contains 16 layers. It is dedicated to classification.

The number of dense units was reduced from 4096 to 1024. This architecture is available as Image Classification in the Create Learning Set dialog. The loss function used for this architecture is categorical cross entropy.

References:

https://neurohive.io/en/popular-networks/vgg16/

2 Installation of PAI Infrastructure

PAI requires the following elements:

- PMOD with PAI licensed (please use the latest build available in our download area)
- Local installation of R version 4.2.2 with the required packages. Please refer to the *PMOD Base Functionality User Guide* for details about the integration of R with PMOD.
- Configuration of the PMOD R Console to use a local R installation
- Local installation of Python (version 3.8 is required)
- Installation of TensorFlow, Scikit-Learn and PyTorch via Python

Please follow the installation instructions applicable for your operating system.

2.1 Windows

Windows 10 or Windows Server 2019 are required as the operating system.

2.1.1 Python and TensorFlow Installation

The additional packages required for PAI on Windows should be installed from their respective websites.

Please follow these steps:

1. Install Microsoft Visual Studio 2015, 2017, 2019 Runtime (i.e VC_redist.x86.exe). https://support.microsoft.com/en-us/help/2977003/the-latest-supported-visual-c-downloads
2. Check whether you have a compatible GPU. TensorFlow only supports NVIDIA GPUs in combination with NVIDIA's CUDA Toolkit. Tables of compatible GPUs are available on https://developer.nvidia.com/cuda-gpus
3. If your GPU is compatible, install NVIDIA drivers, Toolkit and models for TensorFlow and PyTorch:
   a. Drivers: https://www.nvidia.com/download/index.aspx?lang=en-us
   b. CUDA Toolkit 11.2 (TensorFlow) 11.6 (PyTorch): https://developer.nvidia.com/cuda-toolkit-archive
   c. cuDNN 8.1 for CUDA 11.2 (TensorFlow) and cuDNN 8.6 for CUDA 11.6: https://developer.nvidia.com/rdp/cudnn-download
4. Install Python 3.8 64-bit (select "Add Python to PATH", enable pip option and long paths). https://www.python.org/downloads/windows/
5. Upgrade pip by entering in a command terminal:
   ```
   pip3 install --upgrade pip
   ```
6. Install TensorFlow by entering in a command terminal:
   ```
   pip3 install -U tensorflow==2.10.0
   ```
7. Check that `tensorflow` appears in the list of installed packages:
   ```
   pip3 list
   ```
8. Test TensorFlow by entering in a command terminal:
   ```
   python -c "import tensorflow as tf;print(\"Num GPUs Available: \",
   len(tf.config.experimental.list_physical_devices('GPU')))"
   ```

This test returns the number of compatible GPUs available for PAI. Zero is an acceptable result if you do not have a CUDA-compatible NVIDIA GPU.

9. Install scikit-learn by entering in a command terminal:
```
pip3 install -U scikit-learn
```

10. Test scikit-learn by entering in a command terminal:
```
python -c "import sklearn; print(sklearn.__version__)"
```

11. Install PyTorch.
   For CPU:
   ```
   pip3 install -U torch
   ```
   For GPU:
   ```
   pip3 install -U torch --extra-index-url
https://download.pytorch.org/whl/cu116
   ```

Note: GPU support for Windows (point 3. above) was tested for TensorFlow 2.104 and PyTorch 1.13

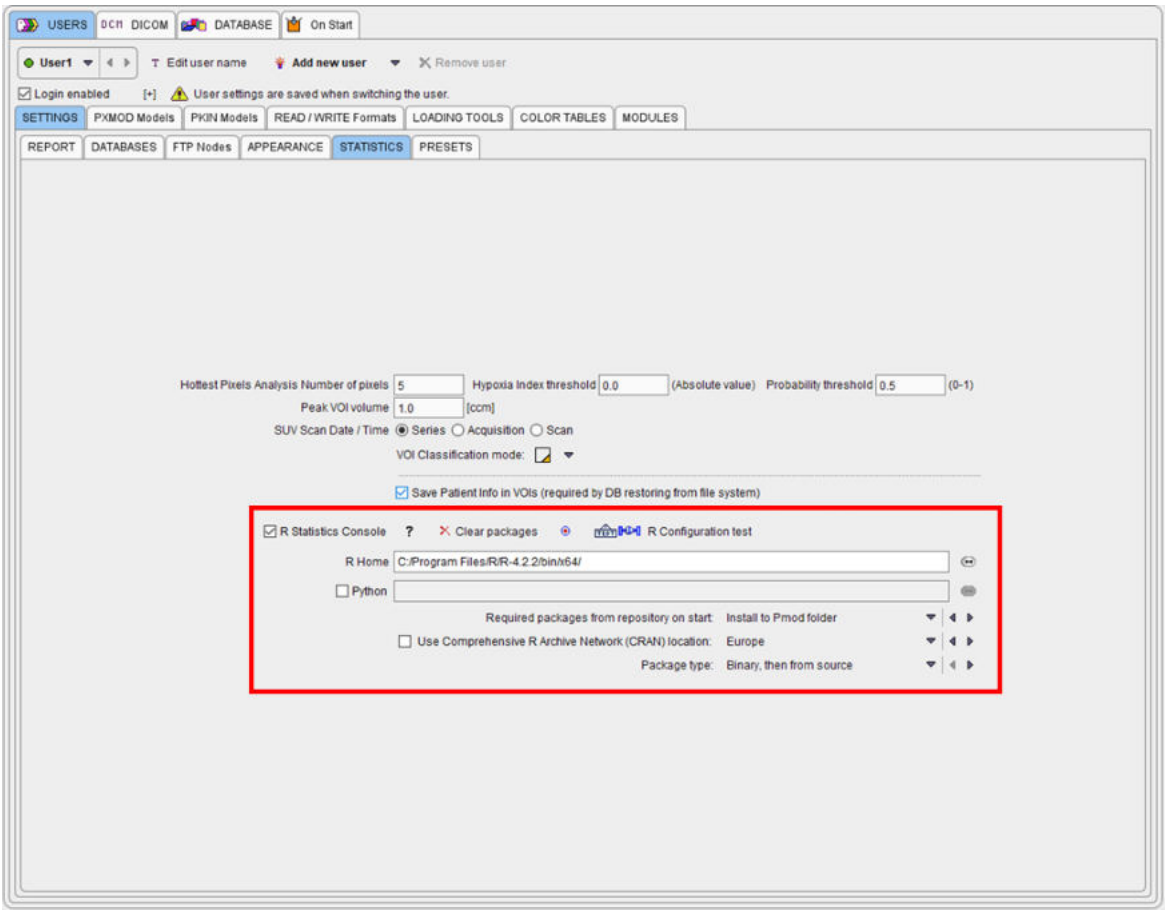
2.1.2 R Installation

Please download and install R version 4.2.2 for Windows from https://cran.r-project.org/

There is no need to manually install additional R packages. PMOD will automatically download and install the necessary packages when the PMOD R Console is started for the first time. If no R functionality besides PAI is used in a particular PMOD installation, the installation can be restricted to a minimal package set as described in Minimal R Configuration 18 below.

2.1.2.1 Default R Configuration

Following R installation, start PMOD and open the **Configuration** facility from the main ToolBox.



Hottest Pixels Analysis Number of pixels: 5 — Hypoxia Index threshold: 0.0 (Absolute value) — Probability threshold: 0.5 (0-1)

Peak VOI volume: 1.0 [ccm]

SUV Scan Date / Time: [✓ Series] Acquisition Scan

VOI Classification mode:

[✓] Save Patient Info in VOIs (required by DB restoring from file system)

[✓] R Statistics Console ? Clear packages R Configuration test

R Home: C:/Program Files/R/R-4.2.2/bin/x64/

[ ] Python

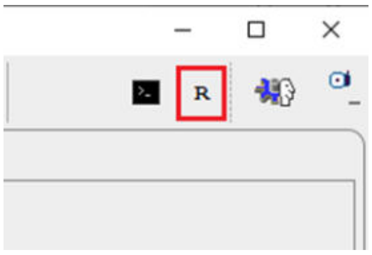
Required packages from repository on start: Install to Pmod folder

[ ] Use Comprehensive R Archive Network (CRAN) location: Europe

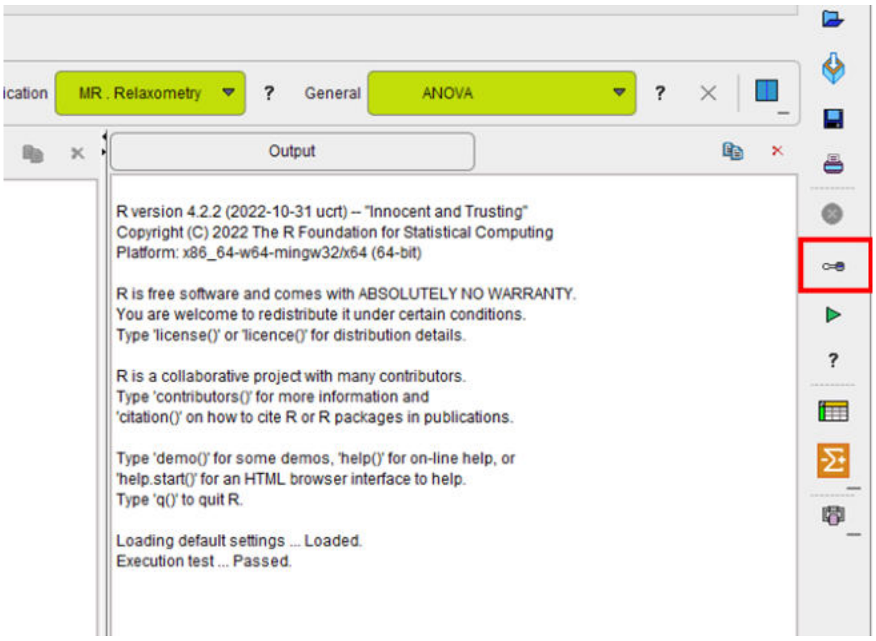
Package type: Binary, then from source

On the Users/Settings **STATISTICS** tab ensure that the checkbox R Statistics Console is checked and verify that the path to the local R installation is correct. Select **Install to Pmod folder** to avoid permission problems when installing the R packages.

Restart PMOD and wait for the **R** icon on the main ToolBox to become active.

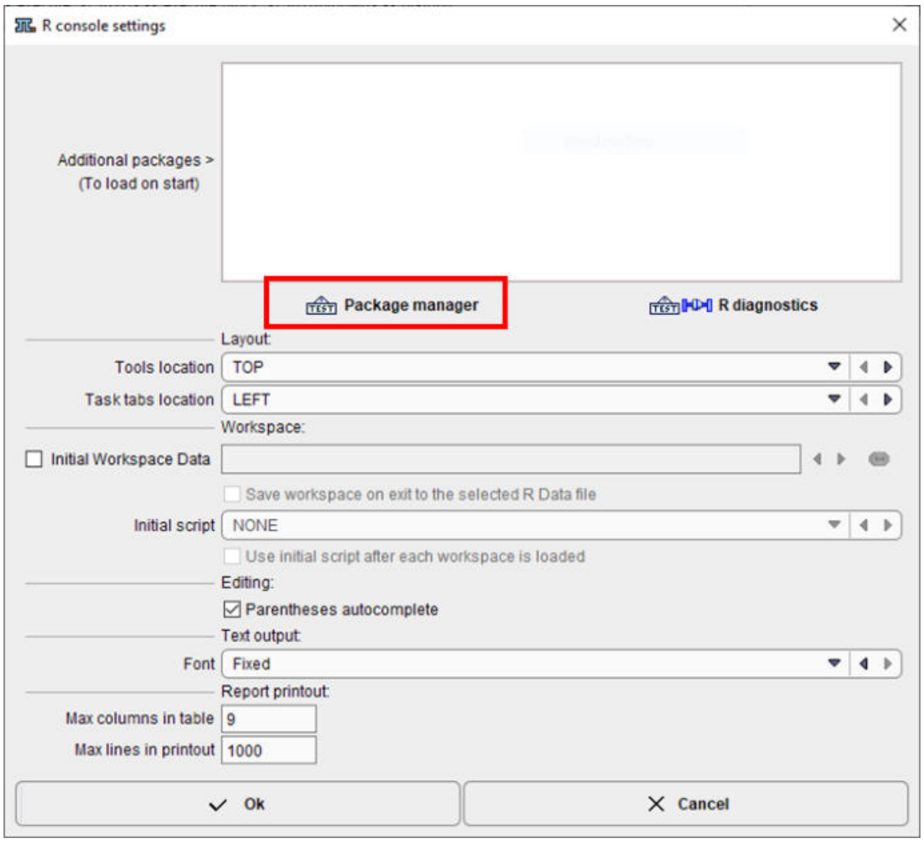


Then click on the **R** icon to open the **PMOD Console**. The required packages are downloaded and installed, followed by an execution test and printing of the R version information:



```
R version 4.2.2 (2022-10-31 ucrt) -- "Innocent and Trusting"
Copyright (C) 2022 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

Loading default settings ... Loaded.
Execution test ... Passed.
```

The settings button indicated above opens the dialog window below.



R console settings

Additional packages >
(To load on start)

Package manager   R diagnostics

Layout:
Tools location: TOP
Task tabs location: LEFT

Workspace:
☐ Initial Workspace Data
☐ Save workspace on exit to the selected R Data file
Initial script: NONE
☐ Use initial script after each workspace is loaded

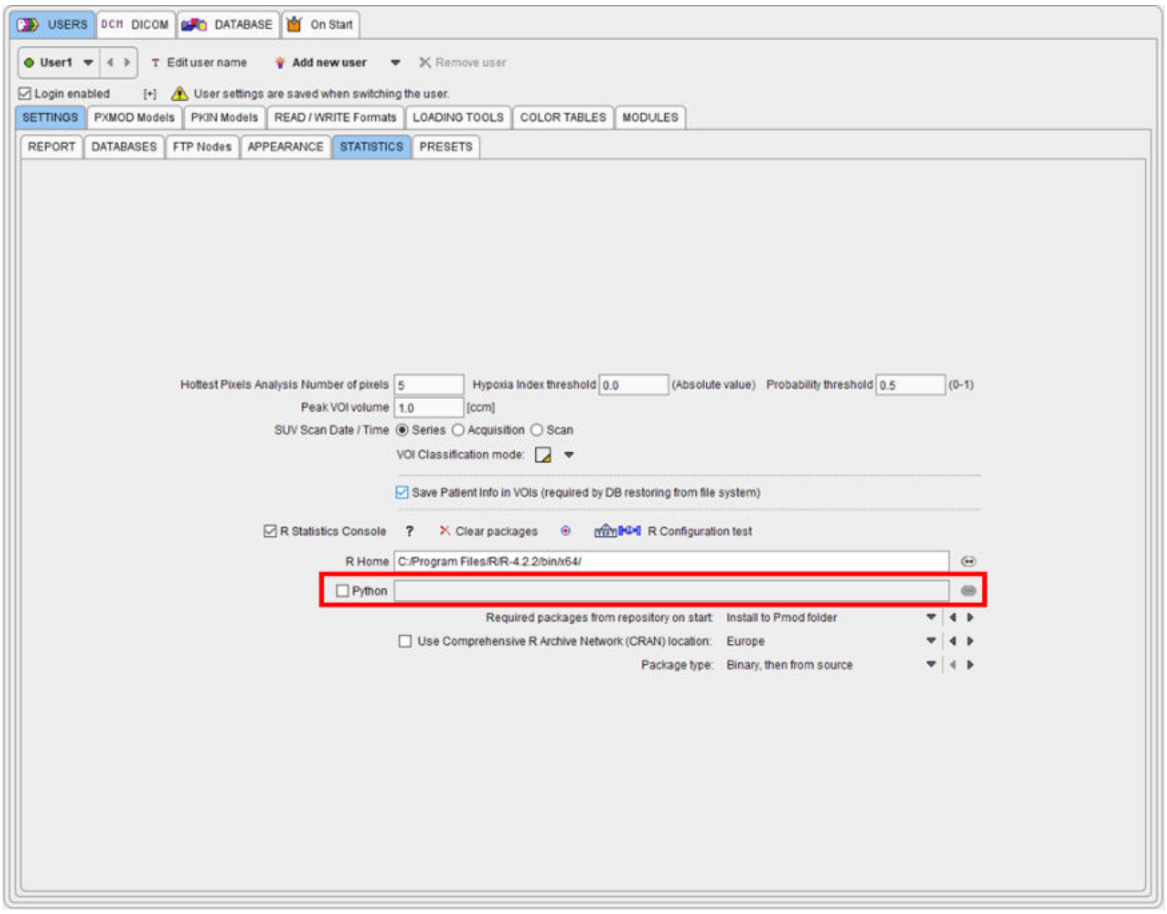
Editing:
☑ Parentheses autocomplete

Text output:
Font: Fixed

Report printout:
Max columns in table: 9
Max lines in printout: 1000

Ok   Cancel

Note that in case of problems connecting to Python when using PAI, the Path to Python can be configured in the **Configuration** facility from the main ToolBox:



Login enabled ✓

User settings are saved when switching the user.

Hottest Pixels Analysis Number of pixels: 5
Hypoxia Index threshold: 0.0 (Absolute value)
Probability threshold: 0.5 (0-1)
Peak VOI volume: 1.0 [ccm]
SUV Scan Date / Time: [✓ Series] Acquisition Scan
VOI Classification mode:

Save Patient Info in VOIs (required by DB restoring from file system) ✓

R Statistics Console ✓

R Home: C:/Program Files/R/R-4.2.2/bin/x64/

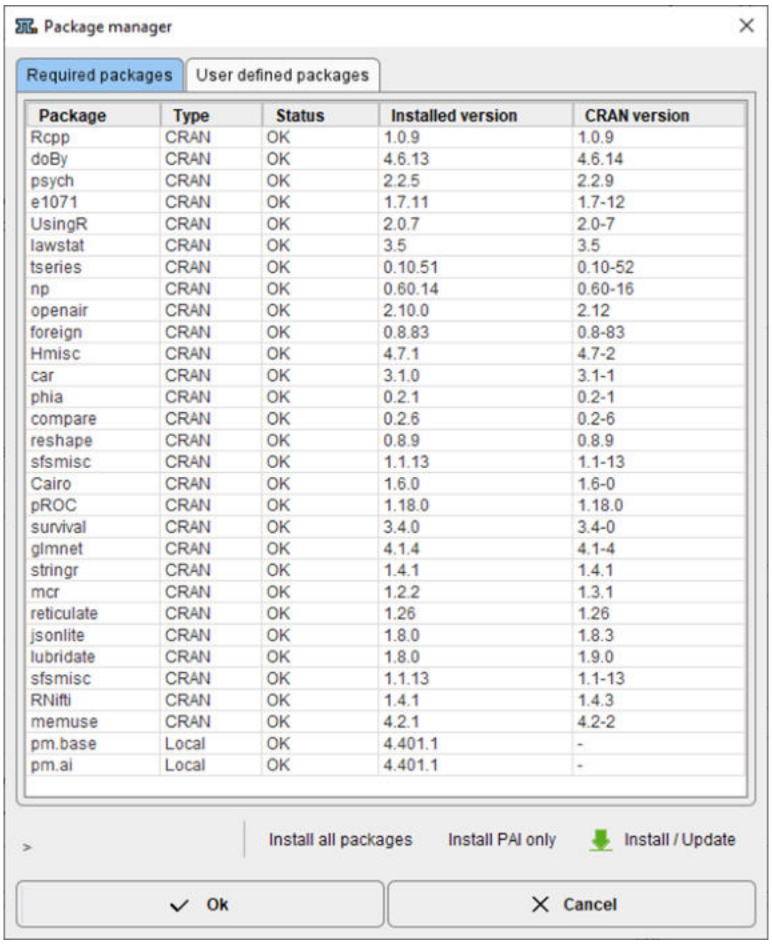
Python ☐

Required packages from repository on start: Install to Pmod folder

Use Comprehensive R Archive Network (CRAN) location: ☐ Europe

Package type: Binary, then from source

Next open the **Package Manager**. All packages should have status **OK**.



Package	Type	Status	Installed version	CRAN version
Rcpp	CRAN	OK	1.0.9	1.0.9
doBy	CRAN	OK	4.6.13	4.6.14
psych	CRAN	OK	2.2.5	2.2.9
e1071	CRAN	OK	1.7.11	1.7-12
UsingR	CRAN	OK	2.0.7	2.0-7
lawstat	CRAN	OK	3.5	3.5
tseries	CRAN	OK	0.10.51	0.10-52
np	CRAN	OK	0.60.14	0.60-16
openair	CRAN	OK	2.10.0	2.12
foreign	CRAN	OK	0.8.83	0.8-83
Hmisc	CRAN	OK	4.7.1	4.7-2
car	CRAN	OK	3.1.0	3.1-1
phia	CRAN	OK	0.2.1	0.2-1
compare	CRAN	OK	0.2.6	0.2-6
reshape	CRAN	OK	0.8.9	0.8.9
sfsmisc	CRAN	OK	1.1.13	1.1-13
Cairo	CRAN	OK	1.6.0	1.6-0
pROC	CRAN	OK	1.18.0	1.18.0
survival	CRAN	OK	3.4.0	3.4-0
glmnet	CRAN	OK	4.1.4	4.1-4
stringr	CRAN	OK	1.4.1	1.4.1
mcr	CRAN	OK	1.2.2	1.3.1
reticulate	CRAN	OK	1.26	1.26
jsonlite	CRAN	OK	1.8.0	1.8.3
lubridate	CRAN	OK	1.8.0	1.9.0
sfsmisc	CRAN	OK	1.1.13	1.1-13
RNifti	CRAN	OK	1.4.1	1.4.3
memuse	CRAN	OK	4.2.1	4.2-2
pm.base	Local	OK	4.401.1	-
pm.ai	Local	OK	4.401.1	-

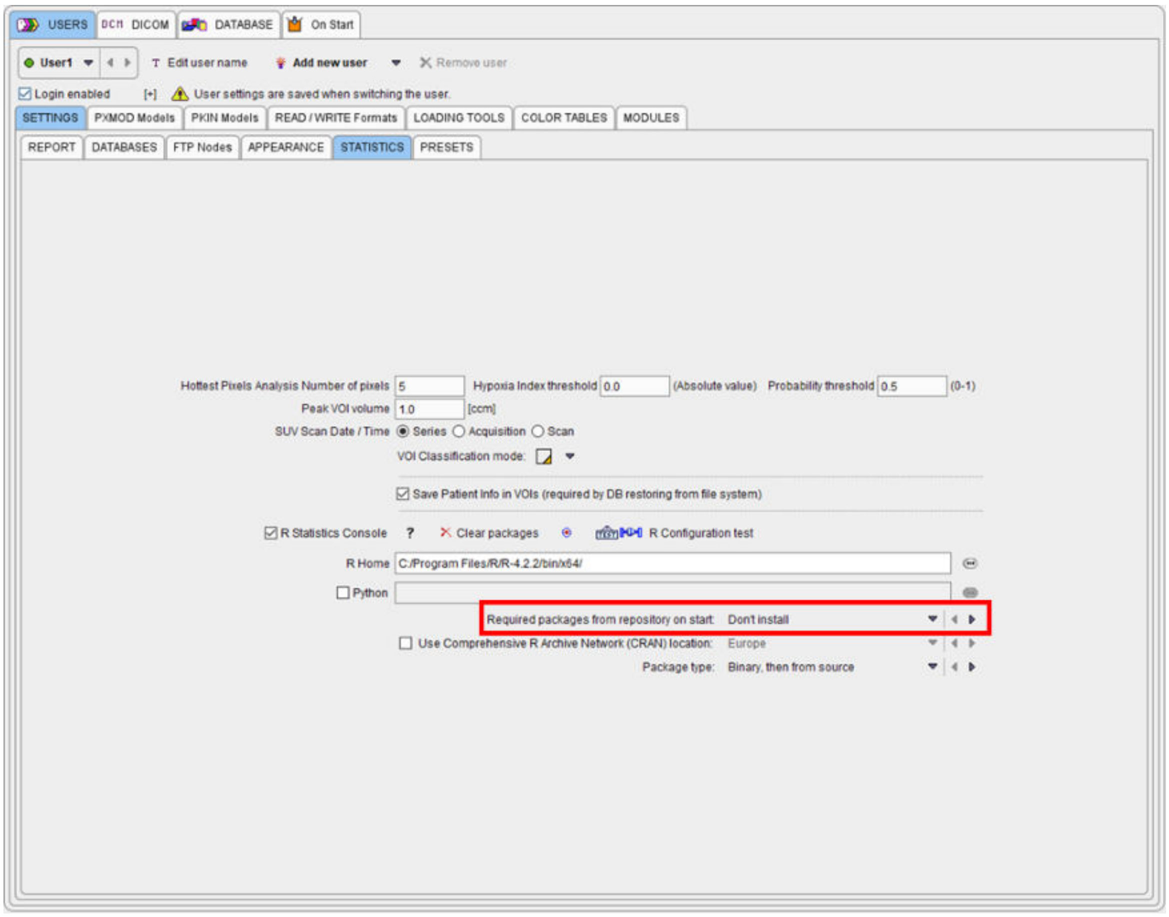
Note: If package installation fails, please check your firewall settings or contact your IT service.

2.1.2.2 Minimal R Configuration

For instances of PMOD that will only be used for PAI it is possible to install a reduced set of R packages. To achieve this, perform the following configuration and installation procedure.

Following R installation, start PMOD and open the **Configuration** facility from the main ToolBox.

On the **STATISTICS** tab ensure that the checkbox R Statistics Console is checked and and verify that the path to the local R installation is correct. Select **Don't install.**



☑ Login enabled

User settings are saved when switching the user.

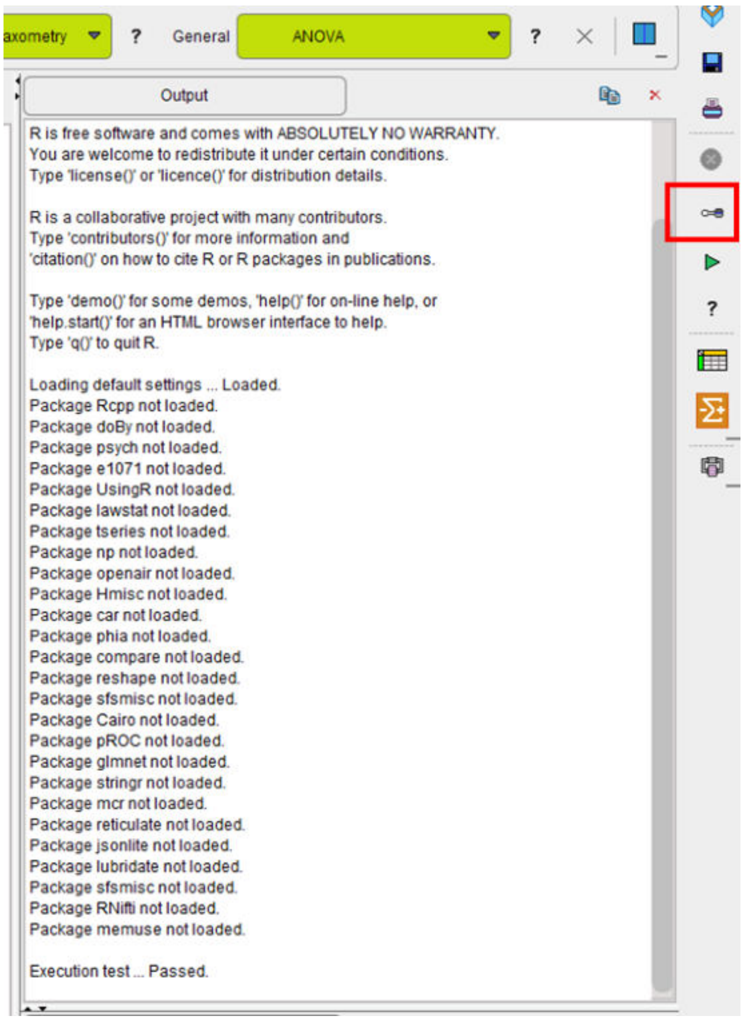
Hottest Pixels Analysis Number of pixels: 5
Hypoxia Index threshold: 0.0 (Absolute value)
Probability threshold: 0.5 (0-1)
Peak VOI volume: 1.0 [ccm]
SUV Scan Date / Time: [✓ Series] Acquisition Scan
VOI Classification mode:

☑ Save Patient Info in VOIs (required by DB restoring from file system)

☑ R Statistics Console
R Home: C:/Program Files/R/R-4.2.2/bin/x64/
☐ Python
Required packages from repository on start: Don't install
☐ Use Comprehensive R Archive Network (CRAN) location: Europe
Package type: Binary, then from source

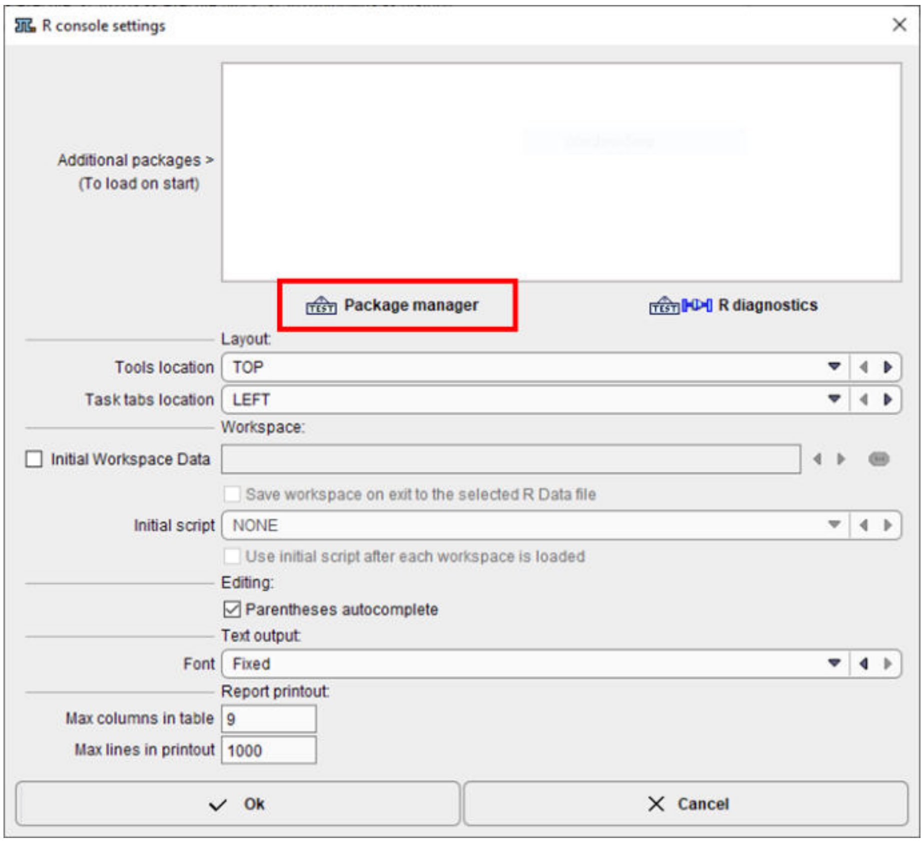
Restart PMOD and wait for the **R** icon on the main ToolBox to become active.

Then click on the **R** icon to open the **PMOD Console**. The internal packages **pm.base** and **pm.ai** were installed automatically whereas the remaining packages are skipped and **not loaded** messages listed:



```
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

Loading default settings ... Loaded.
Package Rcpp not loaded.
Package doBy not loaded.
Package psych not loaded.
Package e1071 not loaded.
Package UsingR not loaded.
Package lawstat not loaded.
Package tseries not loaded.
Package np not loaded.
Package openair not loaded.
Package Hmisc not loaded.
Package car not loaded.
Package phia not loaded.
Package compare not loaded.
Package reshape not loaded.
Package sfsmisc not loaded.
Package Cairo not loaded.
Package pROC not loaded.
Package glmnet not loaded.
Package stringr not loaded.
Package mcr not loaded.
Package reticulate not loaded.
Package jsonlite not loaded.
Package lubridate not loaded.
Package sfsmisc not loaded.
Package RNifti not loaded.
Package memuse not loaded.

Execution test ... Passed.
```

The **Settings** button indicated above opens the dialog window below.



R console settings

Additional packages > (To load on start)

Layout:

Tools location	TOP
Task tabs location	LEFT

Workspace:

☐ Initial Workspace Data

☐ Save workspace on exit to the selected R Data file

Initial script	NONE

☐ Use initial script after each workspace is loaded

Editing:

☑ Parentheses autocomplete

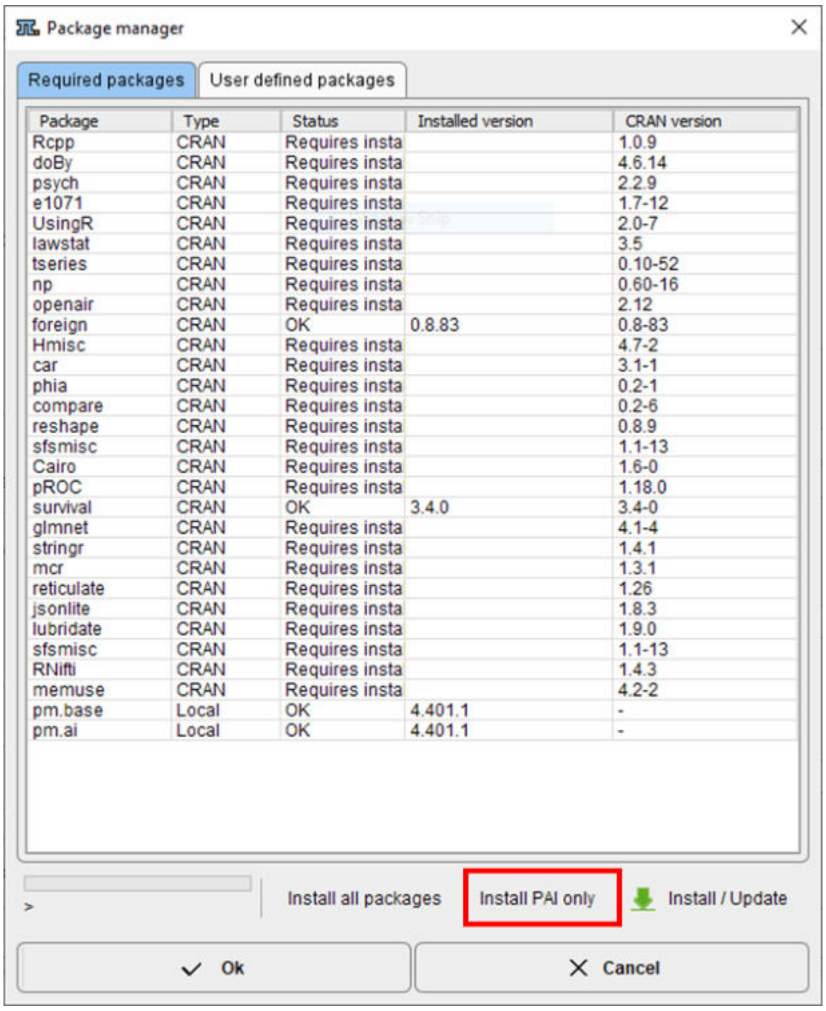
Text output:

Font	Fixed

Report printout:

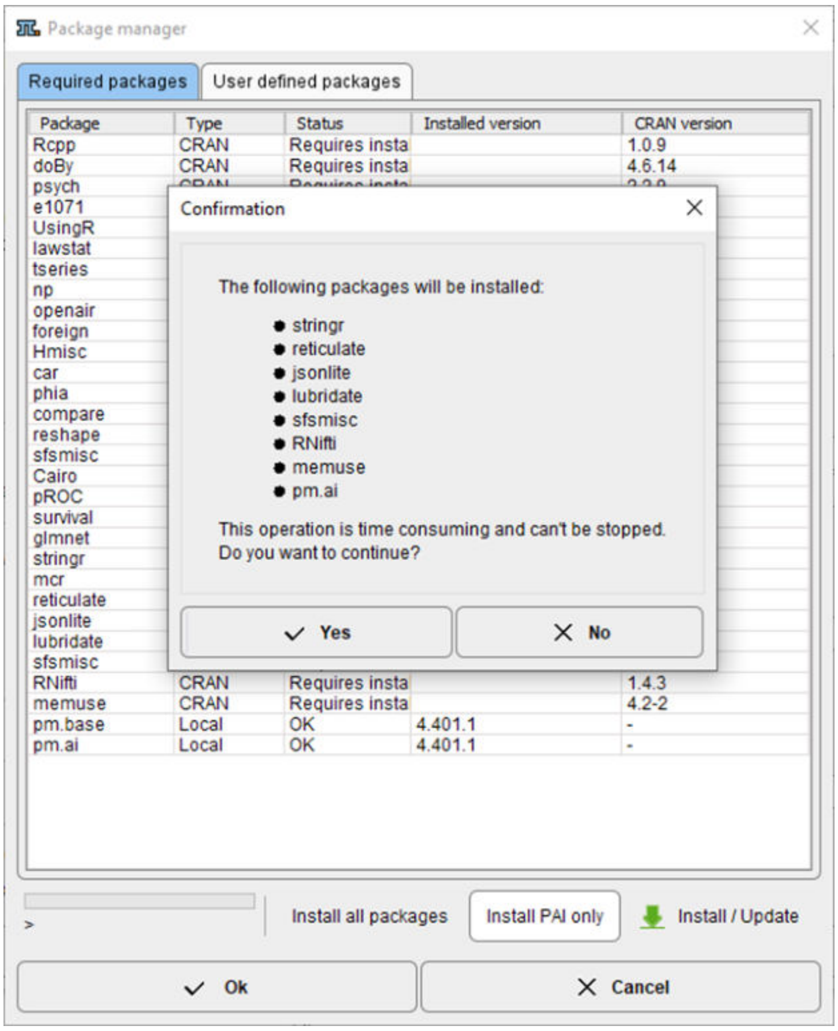
Max columns in table	9
Max lines in printout	1000

Open the **Package Manager**. Most packages will have **Requires installation** in the **Status** column



Package	Type	Status	Installed version	CRAN version
Rcpp	CRAN	Requires insta		1.0.9
doBy	CRAN	Requires insta		4.6.14
psych	CRAN	Requires insta		2.2.9
e1071	CRAN	Requires insta		1.7-12
UsingR	CRAN	Requires insta		2.0-7
lawstat	CRAN	Requires insta		3.5
tseries	CRAN	Requires insta		0.10-52
np	CRAN	Requires insta		0.60-16
openair	CRAN	Requires insta		2.12
foreign	CRAN	OK	0.8.83	0.8-83
Hmisc	CRAN	Requires insta		4.7-2
car	CRAN	Requires insta		3.1-1
phia	CRAN	Requires insta		0.2-1
compare	CRAN	Requires insta		0.2-6
reshape	CRAN	Requires insta		0.8.9
sfsmisc	CRAN	Requires insta		1.1-13
Cairo	CRAN	Requires insta		1.6-0
pROC	CRAN	Requires insta		1.18.0
survival	CRAN	OK	3.4.0	3.4-0
glmnet	CRAN	Requires insta		4.1-4
stringr	CRAN	Requires insta		1.4.1
mcr	CRAN	Requires insta		1.3.1
reticulate	CRAN	Requires insta		1.26
jsonlite	CRAN	Requires insta		1.8.3
lubridate	CRAN	Requires insta		1.9.0
sfsmisc	CRAN	Requires insta		1.1-13
RNifti	CRAN	Requires insta		1.4.3
memuse	CRAN	Requires insta		4.2-2
pm.base	Local	OK	4.401.1	-
pm.ai	Local	OK	4.401.1	-

To install only the packages necessary for PAI, please select **Install PAI only** and then **Yes**



Package	Type	Status	Installed version	CRAN version
Rcpp	CRAN	Requires insta		1.0.9
doBy	CRAN	Requires insta		4.6.14
psych	CRAN	Requires insta		2.2.9
e1071				
UsingR				
lawstat				
tseries				
np				
openair				
foreign				
Hmisc				
car				
phia				
compare				
reshape				
sfsmisc				
Cairo				
pROC				
survival				
glmnet				
stringr				
mcr				
reticulate				
jsonlite				
lubridate				
sfsmisc				
RNifti	CRAN	Requires insta		1.4.3
memuse	CRAN	Requires insta		4.2-2
pm.base	Local	OK	4.401.1	-
pm.ai	Local	OK	4.401.1	-

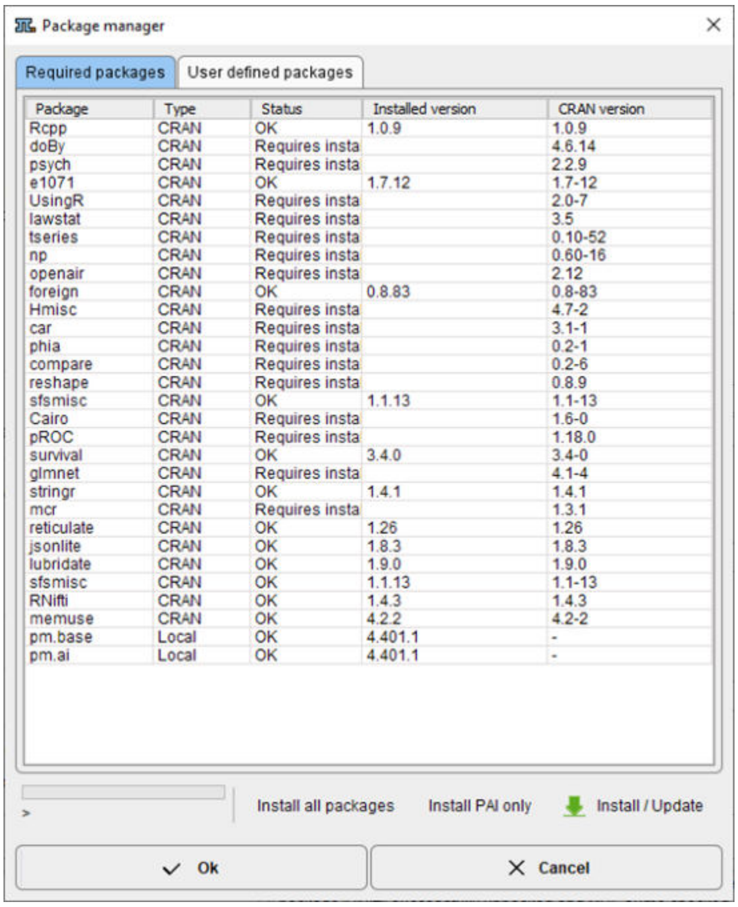
Confirmation

The following packages will be installed:

- stringr
- reticulate
- jsonlite
- lubridate
- sfsmisc
- RNifti
- memuse
- pm.ai

This operation is time consuming and can't be stopped.
Do you want to continue?

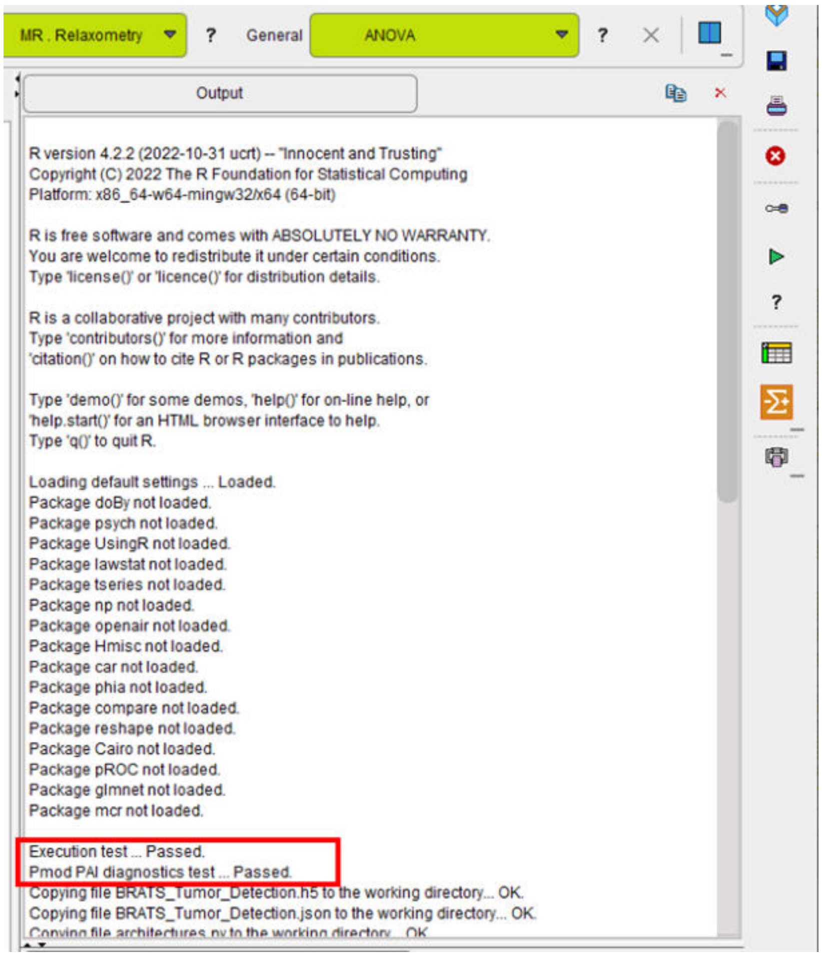
Following installation the **Status** entries will be updated:



Package	Type	Status	Installed version	CRAN version
Rcpp	CRAN	OK	1.0.9	1.0.9
doBy	CRAN	Requires insta		4.6.14
psych	CRAN	Requires insta		2.2.9
e1071	CRAN	OK	1.7.12	1.7-12
UsingR	CRAN	Requires insta		2.0-7
lawstat	CRAN	Requires insta		3.5
tseries	CRAN	Requires insta		0.10-52
np	CRAN	Requires insta		0.60-16
openair	CRAN	Requires insta		2.12
foreign	CRAN	OK	0.8.83	0.8-83
Hmisc	CRAN	Requires insta		4.7-2
car	CRAN	Requires insta		3.1-1
phia	CRAN	Requires insta		0.2-1
compare	CRAN	Requires insta		0.2-6
reshape	CRAN	Requires insta		0.8.9
sfsmisc	CRAN	OK	1.1.13	1.1-13
Cairo	CRAN	Requires insta		1.6-0
pROC	CRAN	Requires insta		1.18.0
survival	CRAN	OK	3.4.0	3.4-0
glmnet	CRAN	Requires insta		4.1-4
stringr	CRAN	OK	1.4.1	1.4.1
mcr	CRAN	Requires insta		1.3.1
reticulate	CRAN	OK	1.26	1.26
jsonlite	CRAN	OK	1.8.3	1.8.3
lubridate	CRAN	OK	1.9.0	1.9.0
sfsmisc	CRAN	OK	1.1.13	1.1-13
RNifti	CRAN	OK	1.4.3	1.4.3
memuse	CRAN	OK	4.2.2	4.2-2
pm.base	Local	OK	4.401.1	-
pm.ai	Local	OK	4.401.1	-

Closing the window with **Ok** completes the installation.

Note: When prediction is launched at a later time point, the R Console will report the packages that were not installed, but the **Execution test** will still pass as illustrated below.

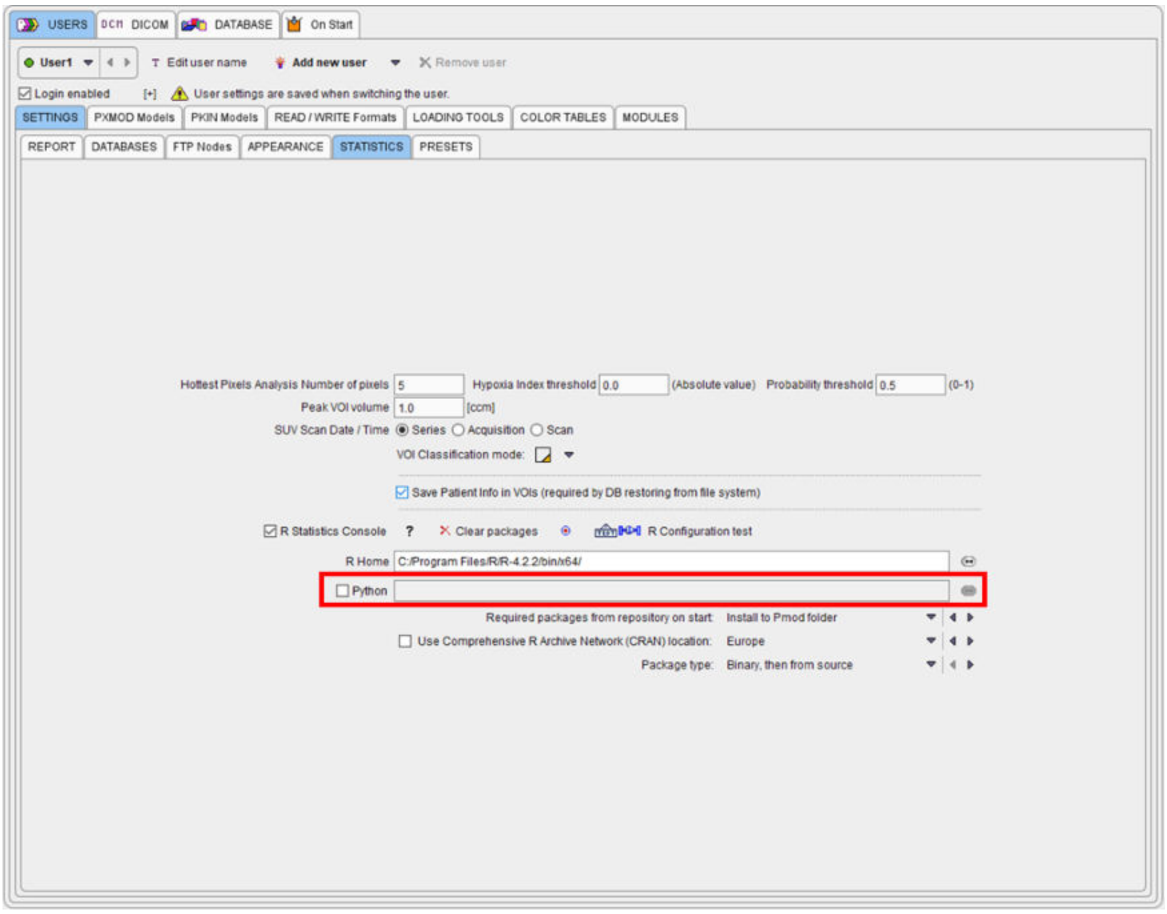


```
R version 4.2.2 (2022-10-31 ucrt) -- "Innocent and Trusting"
Copyright (C) 2022 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

Loading default settings ... Loaded.
Package doBy not loaded.
Package psych not loaded.
Package UsingR not loaded.
Package lawstat not loaded.
Package tseries not loaded.
Package np not loaded.
Package openair not loaded.
Package Hmisc not loaded.
Package car not loaded.
Package phia not loaded.
Package compare not loaded.
Package reshape not loaded.
Package Cairo not loaded.
Package pROC not loaded.
Package glmnet not loaded.
Package mcr not loaded.

Execution test ... Passed.
Pmod PAI diagnostics test ... Passed.
Copying file BRATS_Tumor_Detection.h5 to the working directory... OK.
Copying file BRATS_Tumor_Detection.json to the working directory... OK.
Copying file architectures.py to the working directory... OK
```

2.1.3 Selection of Python installation

Multiple installations and versions of Python may cause the PAI infrastructure test to fail.

In this situation you should define the path to Python 3.8 in the **Configuration** facility from the main ToolBox:



[✓] Login enabled

Hottest Pixels Analysis Number of pixels: 5 — Hypoxia Index threshold: 0.0 (Absolute value) — Probability threshold: 0.5 (0-1)
Peak VOI volume: 1.0 [ccm]
SUV Scan Date / Time: [✓ Series] Acquisition Scan
VOI Classification mode:
[✓] Save Patient Info in VOIs (required by DB restoring from file system)
[✓] R Statistics Console
R Home: C:/Program Files/R/R-4.2.2/bin/x64/
[ ] Python
Required packages from repository on start: Install to Pmod folder
[ ] Use Comprehensive R Archive Network (CRAN) location: Europe
Package type: Binary, then from source

Restart PMOD after setting the path to Python 3.8.

2.2 MacOS

MacOS from Big Sur onwards is recommended. Older MacOS may not support the PAI infrastructure.

Note: the XQuartz package is required for plotting in the R console (to support X11 libraries). The XQuartz project is officially supported by Apple: https://www.xquartz.org/

2.2.1 Python and TensorFlow Installation

Although MacOS provides Python libraries by default we recommend installing the newest available Python version 3.8 from the website https://www.python.org/downloads/mac-osx/

1. Install TensorFlow using the Terminal command:

```
python3 -m pip3 install --upgrade tensorflow==2.10.0
```

2. Check that tensorflow appears in the list of installed packages:

```
python3 -m pip list
```

3. Test TensorFlow using the command:

```
python3 -c "import tensorflow as tf;print(\"Num GPUs Available: \",
len(tf.config.experimental.list_physical_devices('GPU')))"
```

This test returns the number of compatible GPUs available for PAI (zero is an acceptable result if you do not have a compatible NVIDIA GPU).

4. Install scikit-learn by entering in Terminal:

```
python3 -m pip3 install --upgrade scikit-learn
```

5. Test scikit-learn by entering in Terminal:

```
python3 -c "import sklearn; print(sklearn.__version__)"
```

6. Install PyTorch by entering in Terminal:

```
python3 -m pip3 install --upgrade torch
```

2.2.2 R Installation

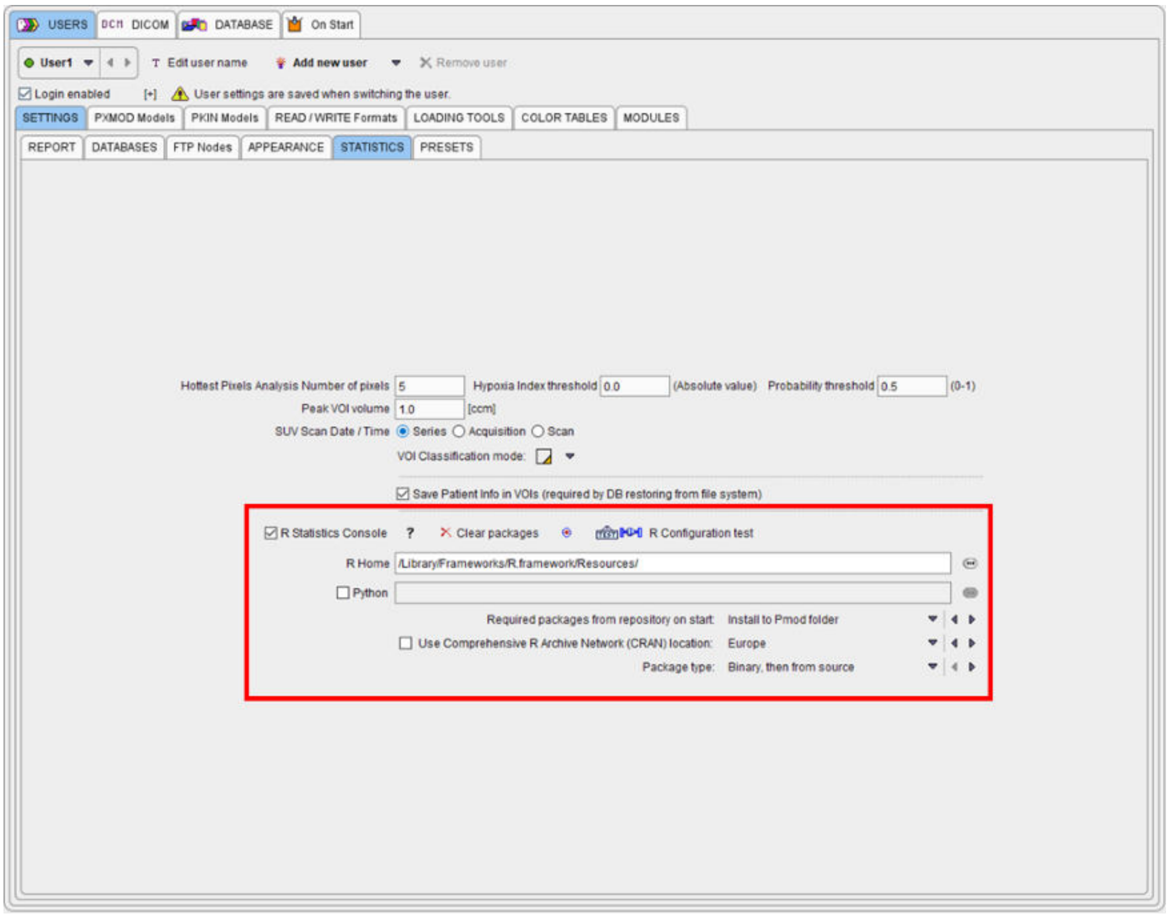
Please download and install R version 4.2.2 for MacOS from https://cran.r-project.org/

There is no need to manually install additional R packages. PMOD will automatically download and install the necessary packages when the PMOD R Console is started for the first time. If no R functionality besides PAI is used in a particular PMOD installation, the installation can be restricted to a minimal package set as described in Minimal R Configuration 18 below.

2.2.2.1 Default R Configuration

Following R installation, start PMOD and open the **Configuration** facility from the main ToolBox.

On the **STATISTICS** tab ensure that the checkbox R Statistics Console is checked.Select **Install to Pmod folder** to avoid permission problems when installing the R packages. We recommend selecting **Use Comprehensive R Archive Network (CRAN) location**. In rare cases the connection to CRAN may be unsuccessful. In this case, unselect **Use Comprehensive R Archive Network (CRAN) location**, then select the CRAN location from the native R dialog that appears when R Console is started. This situation has been encountered on MacOS Catalina.



Hottest Pixels Analysis Number of pixels: 5 | Hypoxia Index threshold: 0.0 (Absolute value) | Probability threshold: 0.5 (0-1)

Peak VOI volume: 1.0 [ccm]

SUV Scan Date / Time: [✓ Series] Acquisition Scan

VOI Classification mode:

[✓] Save Patient Info in VOIs (required by DB restoring from file system)

[✓] R Statistics Console ? Clear packages R Configuration test

R Home: /Library/Frameworks/R.framework/Resources/

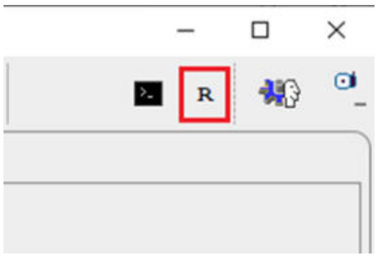
[ ] Python

Required packages from repository on start: Install to Pmod folder

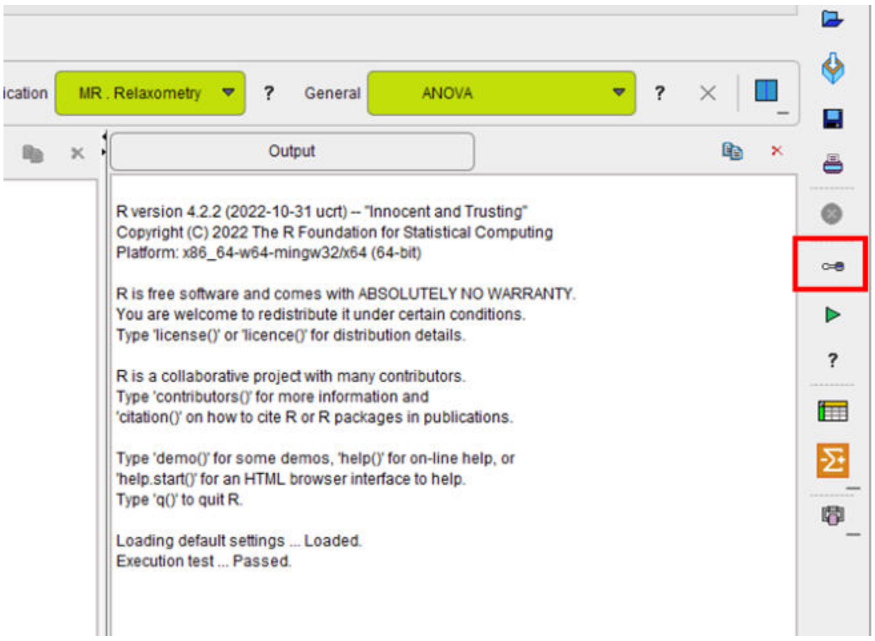
[ ] Use Comprehensive R Archive Network (CRAN) location: Europe

Package type: Binary, then from source

Restart PMOD and wait for the **R** icon on the main ToolBox to become active.

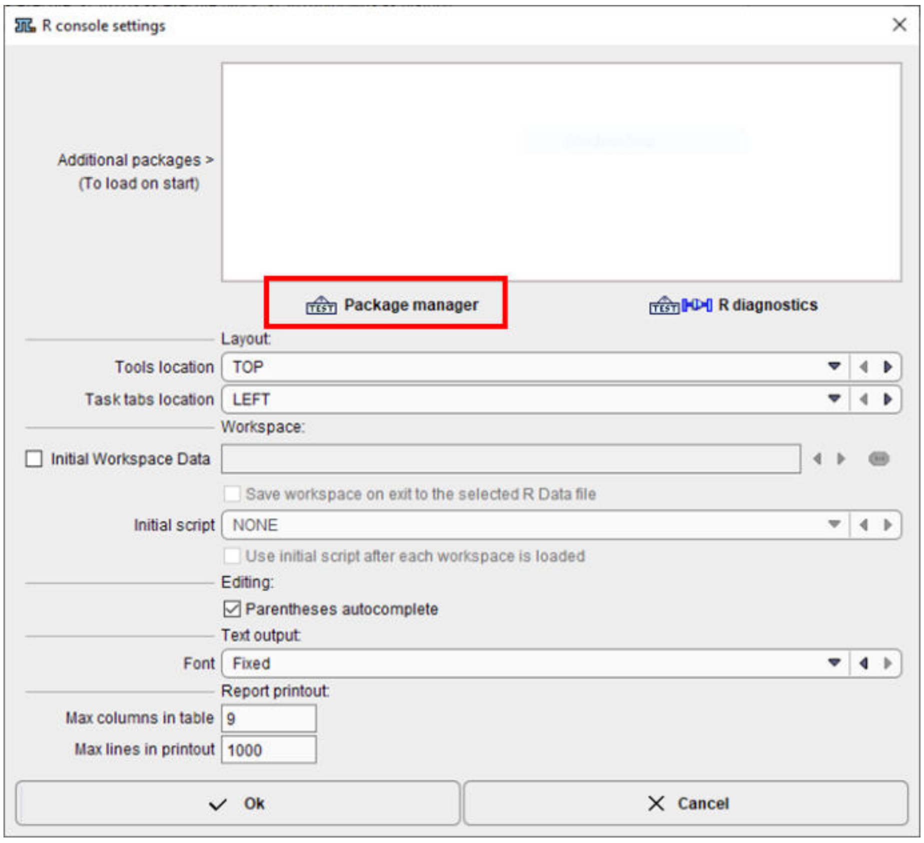


Then click on the **R** icon to open the **PMOD Console**. The required packages are downloaded and installed, followed by an execution test and printing of the R version information:



```
R version 4.2.2 (2022-10-31 ucrt) -- "Innocent and Trusting"
Copyright (C) 2022 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

Loading default settings ... Loaded.
Execution test ... Passed.
```

The **Settings** button indicated above opens the dialog window below.



Additional packages > (To load on start)

Layout:
Tools location: TOP
Task tabs location: LEFT

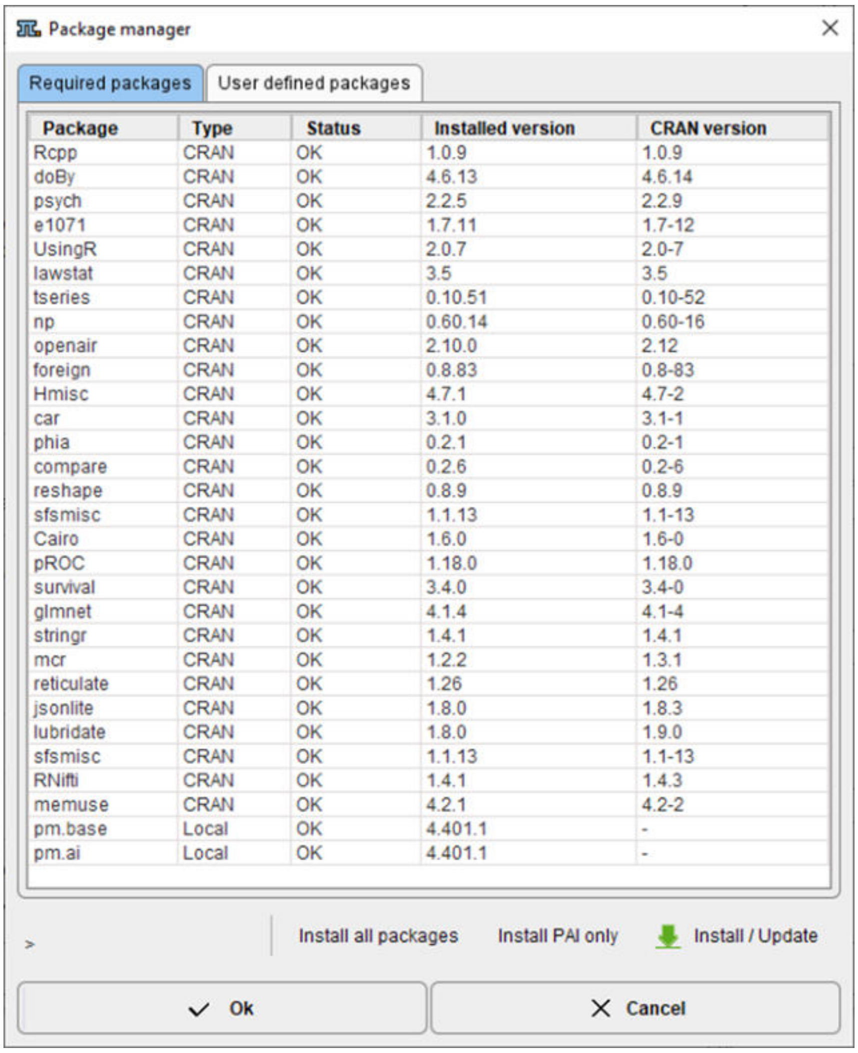
Workspace:
☐ Initial Workspace Data
☐ Save workspace on exit to the selected R Data file
Initial script: NONE
☐ Use initial script after each workspace is loaded

Editing:
☑ Parentheses autocomplete

Text output:
Font: Fixed

Report printout:
Max columns in table: 9
Max lines in printout: 1000

Open the **Package Manager**. All packages should have status **OK**.



Package	Type	Status	Installed version	CRAN version
Rcpp	CRAN	OK	1.0.9	1.0.9
doBy	CRAN	OK	4.6.13	4.6.14
psych	CRAN	OK	2.2.5	2.2.9
e1071	CRAN	OK	1.7.11	1.7-12
UsingR	CRAN	OK	2.0.7	2.0-7
lawstat	CRAN	OK	3.5	3.5
tseries	CRAN	OK	0.10.51	0.10-52
np	CRAN	OK	0.60.14	0.60-16
openair	CRAN	OK	2.10.0	2.12
foreign	CRAN	OK	0.8.83	0.8-83
Hmisc	CRAN	OK	4.7.1	4.7-2
car	CRAN	OK	3.1.0	3.1-1
phia	CRAN	OK	0.2.1	0.2-1
compare	CRAN	OK	0.2.6	0.2-6
reshape	CRAN	OK	0.8.9	0.8.9
sfsmisc	CRAN	OK	1.1.13	1.1-13
Cairo	CRAN	OK	1.6.0	1.6-0
pROC	CRAN	OK	1.18.0	1.18.0
survival	CRAN	OK	3.4.0	3.4-0
glmnet	CRAN	OK	4.1.4	4.1-4
stringr	CRAN	OK	1.4.1	1.4.1
mcr	CRAN	OK	1.2.2	1.3.1
reticulate	CRAN	OK	1.26	1.26
jsonlite	CRAN	OK	1.8.0	1.8.3
lubridate	CRAN	OK	1.8.0	1.9.0
sfsmisc	CRAN	OK	1.1.13	1.1-13
RNifti	CRAN	OK	1.4.1	1.4.3
memuse	CRAN	OK	4.2.1	4.2-2
pm.base	Local	OK	4.401.1	-
pm.ai	Local	OK	4.401.1	-

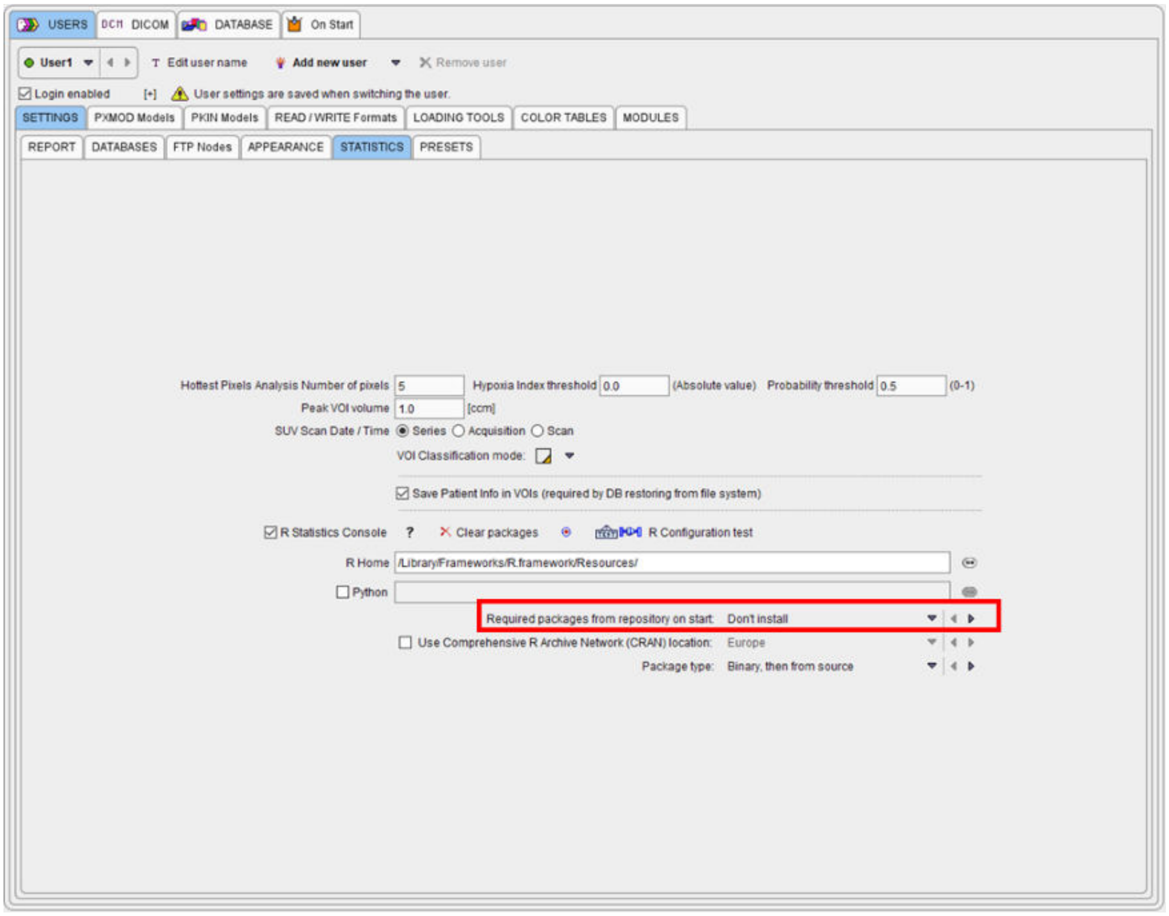
Note: If package installation fails, please check your firewall settings or contact your IT service.

2.2.2.2 Minimal R Configuration

For instances of PMOD that will only be used for PAI it is possible to install a reduced set of R packages. To achieve this, perform the following configuration and installation procedure.

Following R installation, start PMOD and open the **Configuration** facility from the main ToolBox.

On the **STATISTICS** tab ensure that the checkbox R Statistics Console is checked and verify that the path to the local R installation is correct. Select **Don't install.**We recommend selecting **Use Comprehensive R Archive Network (CRAN) location**. In rare cases the connection to CRAN may be unsuccessful. In this case, unselect **Use Comprehensive R Archive Network (CRAN) location**, then select the CRAN location from the native R dialog that appears when R Console is started.



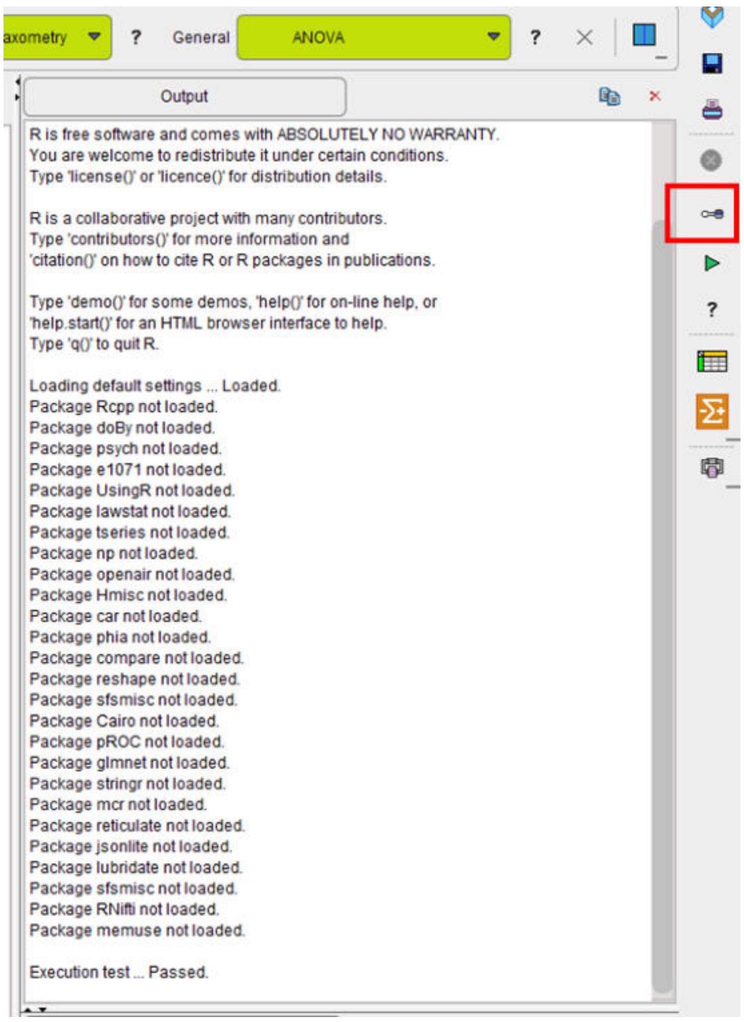
Login enabled [✓]

User settings are saved when switching the user.

Hottest Pixels Analysis Number of pixels: 5
Hypoxia Index threshold: 0.0 (Absolute value)
Probability threshold: 0.5 (0-1)
Peak VOI volume: 1.0 [ccm]
SUV Scan Date / Time: [✓ Series] Acquisition Scan
VOI Classification mode:
Save Patient Info in VOIs (required by DB restoring from file system) [✓]
R Statistics Console [✓]
R Home: /Library/Frameworks/R.framework/Resources/
Python [ ]
Required packages from repository on start: Don't install
Use Comprehensive R Archive Network (CRAN) location [ ]: Europe
Package type: Binary, then from source

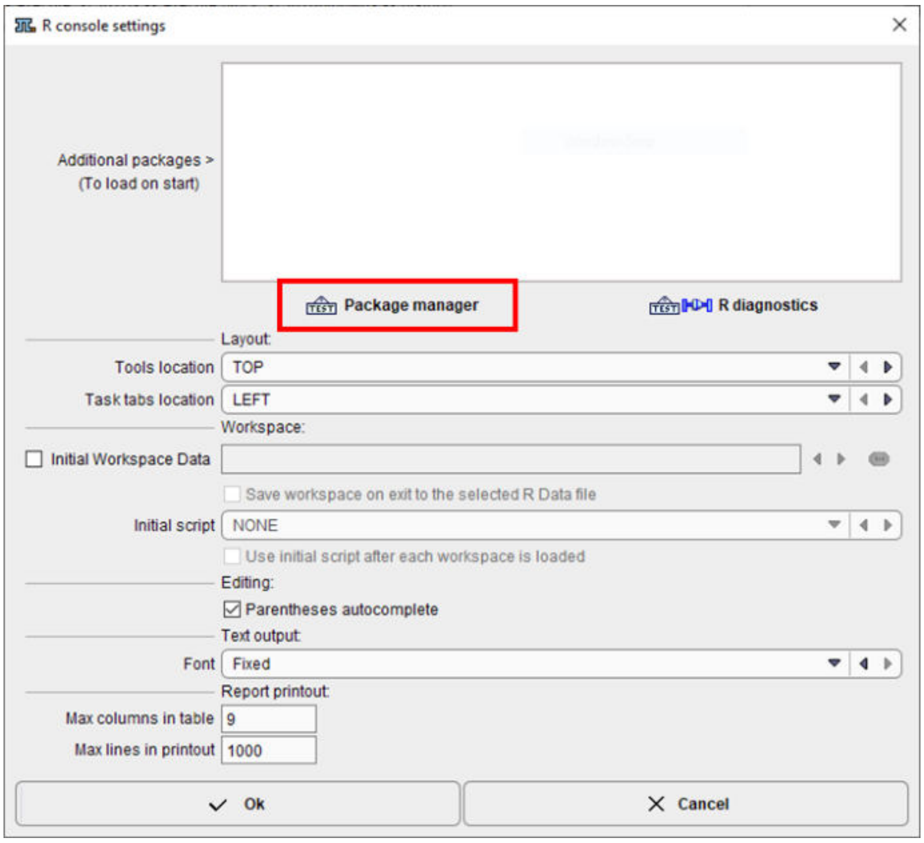
Restart PMOD and wait for the **R** icon on the main ToolBox to become active.

Then click on the **R** icon to open the **PMOD Console**. The internal packages **pm.base** and **pm.ai** were installed automatically whereas the remaining packages are skipped and **not loaded** messages listed:



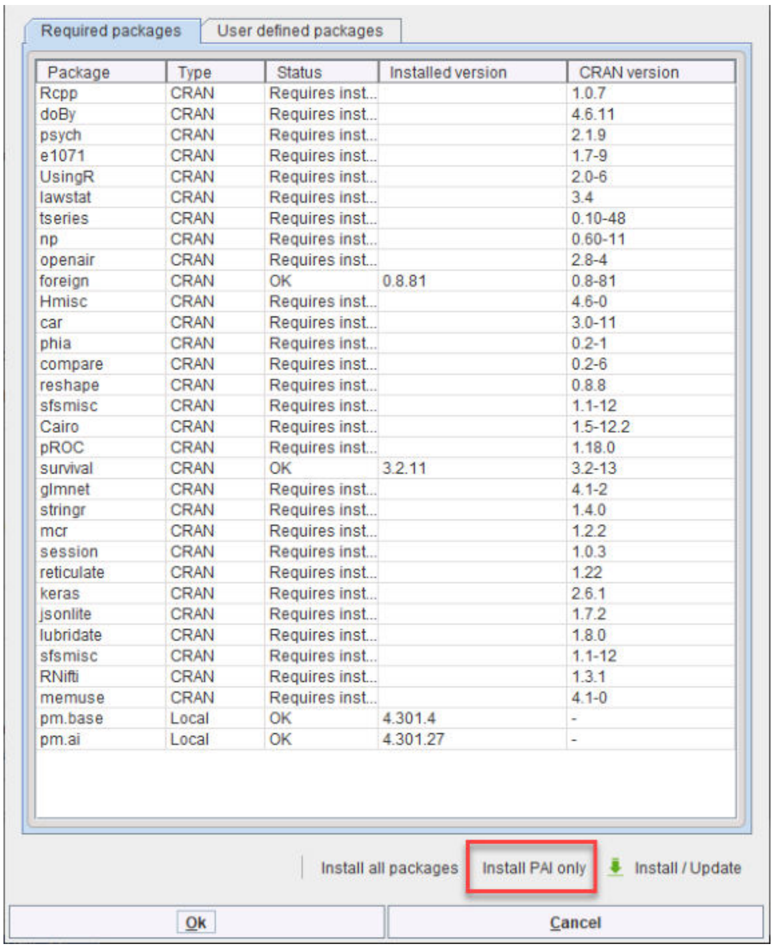
```
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

Loading default settings ... Loaded.
Package Rcpp not loaded.
Package doBy not loaded.
Package psych not loaded.
Package e1071 not loaded.
Package UsingR not loaded.
Package lawstat not loaded.
Package tseries not loaded.
Package np not loaded.
Package openair not loaded.
Package Hmisc not loaded.
Package car not loaded.
Package phia not loaded.
Package compare not loaded.
Package reshape not loaded.
Package sfsmisc not loaded.
Package Cairo not loaded.
Package pROC not loaded.
Package glmnet not loaded.
Package stringr not loaded.
Package mcr not loaded.
Package reticulate not loaded.
Package jsonlite not loaded.
Package lubridate not loaded.
Package sfsmisc not loaded.
Package RNifti not loaded.
Package memuse not loaded.

Execution test ... Passed.
```

The **Settings** button indicated above opens the dialog window below.



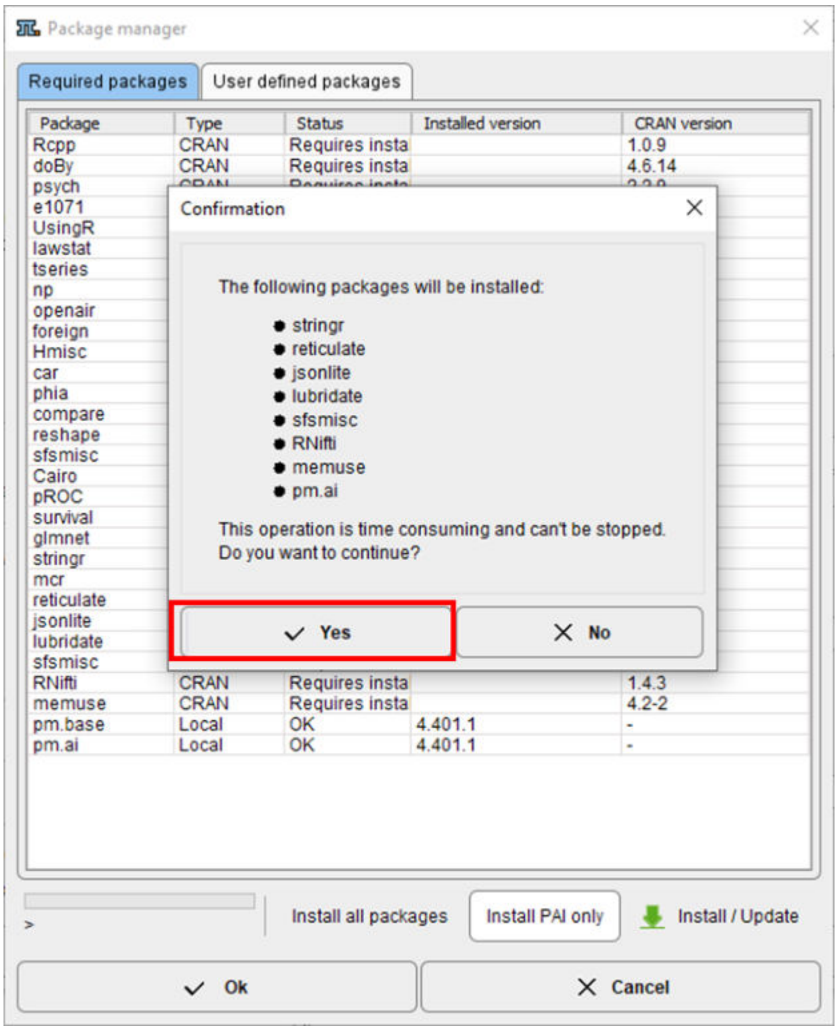
Additional packages > (To load on start)	
Layout:	
Tools location	TOP
Task tabs location	LEFT
Workspace:	
☐ Initial Workspace Data	
	☐ Save workspace on exit to the selected R Data file
Initial script	NONE
	☐ Use initial script after each workspace is loaded
Editing:	
	☑ Parentheses autocomplete
Text output:	
Font	Fixed
Report printout:	
Max columns in table	9
Max lines in printout	1000

Open the **Package Manager**. Most packages will have **Requires installation** in the **Status** column



Package	Type	Status	Installed version	CRAN version
Rcpp	CRAN	Requires inst...		1.0.7
doBy	CRAN	Requires inst...		4.6.11
psych	CRAN	Requires inst...		2.1.9
e1071	CRAN	Requires inst...		1.7-9
UsingR	CRAN	Requires inst...		2.0-6
lawstat	CRAN	Requires inst...		3.4
tseries	CRAN	Requires inst...		0.10-48
np	CRAN	Requires inst...		0.60-11
openair	CRAN	Requires inst...		2.8-4
foreign	CRAN	OK	0.8.81	0.8-81
Hmisc	CRAN	Requires inst...		4.6-0
car	CRAN	Requires inst...		3.0-11
phia	CRAN	Requires inst...		0.2-1
compare	CRAN	Requires inst...		0.2-6
reshape	CRAN	Requires inst...		0.8.8
sfsmisc	CRAN	Requires inst...		1.1-12
Cairo	CRAN	Requires inst...		1.5-12.2
pROC	CRAN	Requires inst...		1.18.0
survival	CRAN	OK	3.2.11	3.2-13
glmnet	CRAN	Requires inst...		4.1-2
stringr	CRAN	Requires inst...		1.4.0
mcr	CRAN	Requires inst...		1.2.2
session	CRAN	Requires inst...		1.0.3
reticulate	CRAN	Requires inst...		1.22
keras	CRAN	Requires inst...		2.6.1
jsonlite	CRAN	Requires inst...		1.7.2
lubridate	CRAN	Requires inst...		1.8.0
sfsmisc	CRAN	Requires inst...		1.1-12
RNifti	CRAN	Requires inst...		1.3.1
memuse	CRAN	Requires inst...		4.1-0
pm.base	Local	OK	4.301.4	-
pm.ai	Local	OK	4.301.27	-

To install only the packages necessary for PAI, please select **Install PAI only** and then **Install/Update**



Package	Type	Status	Installed version	CRAN version
Rcpp	CRAN	Requires insta		1.0.9
doBy	CRAN	Requires insta		4.6.14
psych				
e1071				
UsingR				
lawstat				
tseries				
np				
openair				
foreign				
Hmisc				
car				
phia				
compare				
reshape				
sfsmisc				
Cairo				
pROC				
survival				
glmnet				
stringr				
mcr				
reticulate				
jsonlite				
lubridate				
sfsmisc				
RNifti	CRAN	Requires insta		1.4.3
memuse	CRAN	Requires insta		4.2-2
pm.base	Local	OK	4.401.1	-
pm.ai	Local	OK	4.401.1	-

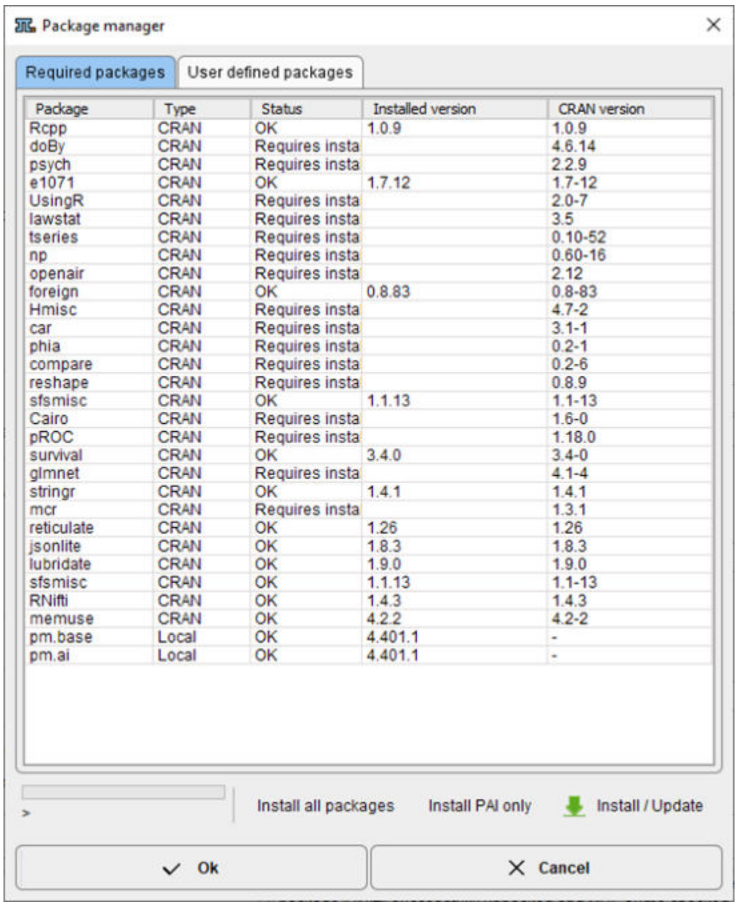
Confirmation

The following packages will be installed:

- stringr
- reticulate
- jsonlite
- lubridate
- sfsmisc
- RNifti
- memuse
- pm.ai

This operation is time consuming and can't be stopped.
Do you want to continue?

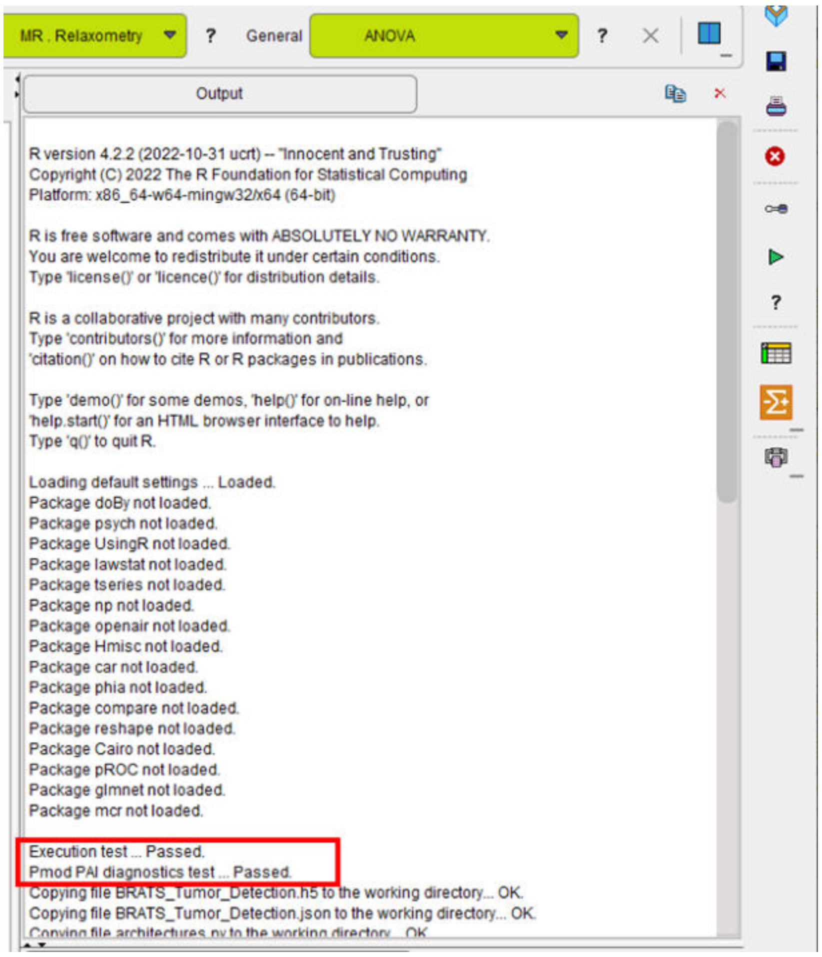
Following installation the **Status** entries will be updated:



Package	Type	Status	Installed version	CRAN version
Rcpp	CRAN	OK	1.0.9	1.0.9
doBy	CRAN	Requires insta		4.6.14
psych	CRAN	Requires insta		2.2.9
e1071	CRAN	OK	1.7.12	1.7-12
UsingR	CRAN	Requires insta		2.0-7
lawstat	CRAN	Requires insta		3.5
tseries	CRAN	Requires insta		0.10-52
np	CRAN	Requires insta		0.60-16
openair	CRAN	Requires insta		2.12
foreign	CRAN	OK	0.8.83	0.8-83
Hmisc	CRAN	Requires insta		4.7-2
car	CRAN	Requires insta		3.1-1
phia	CRAN	Requires insta		0.2-1
compare	CRAN	Requires insta		0.2-6
reshape	CRAN	Requires insta		0.8.9
sfsmisc	CRAN	OK	1.1.13	1.1-13
Cairo	CRAN	Requires insta		1.6-0
pROC	CRAN	Requires insta		1.18.0
survival	CRAN	OK	3.4.0	3.4-0
glmnet	CRAN	Requires insta		4.1-4
stringr	CRAN	OK	1.4.1	1.4.1
mcr	CRAN	Requires insta		1.3.1
reticulate	CRAN	OK	1.26	1.26
jsonlite	CRAN	OK	1.8.3	1.8.3
lubridate	CRAN	OK	1.9.0	1.9.0
sfsmisc	CRAN	OK	1.1.13	1.1-13
RNifti	CRAN	OK	1.4.3	1.4.3
memuse	CRAN	OK	4.2.2	4.2-2
pm.base	Local	OK	4.401.1	-
pm.ai	Local	OK	4.401.1	-

Closing the window with **Ok** completes the installation.

Note: When prediction is launched at a later time point, the R Console will report the packages that were not installed, but the **Execution test** will still pass as illustrated below.

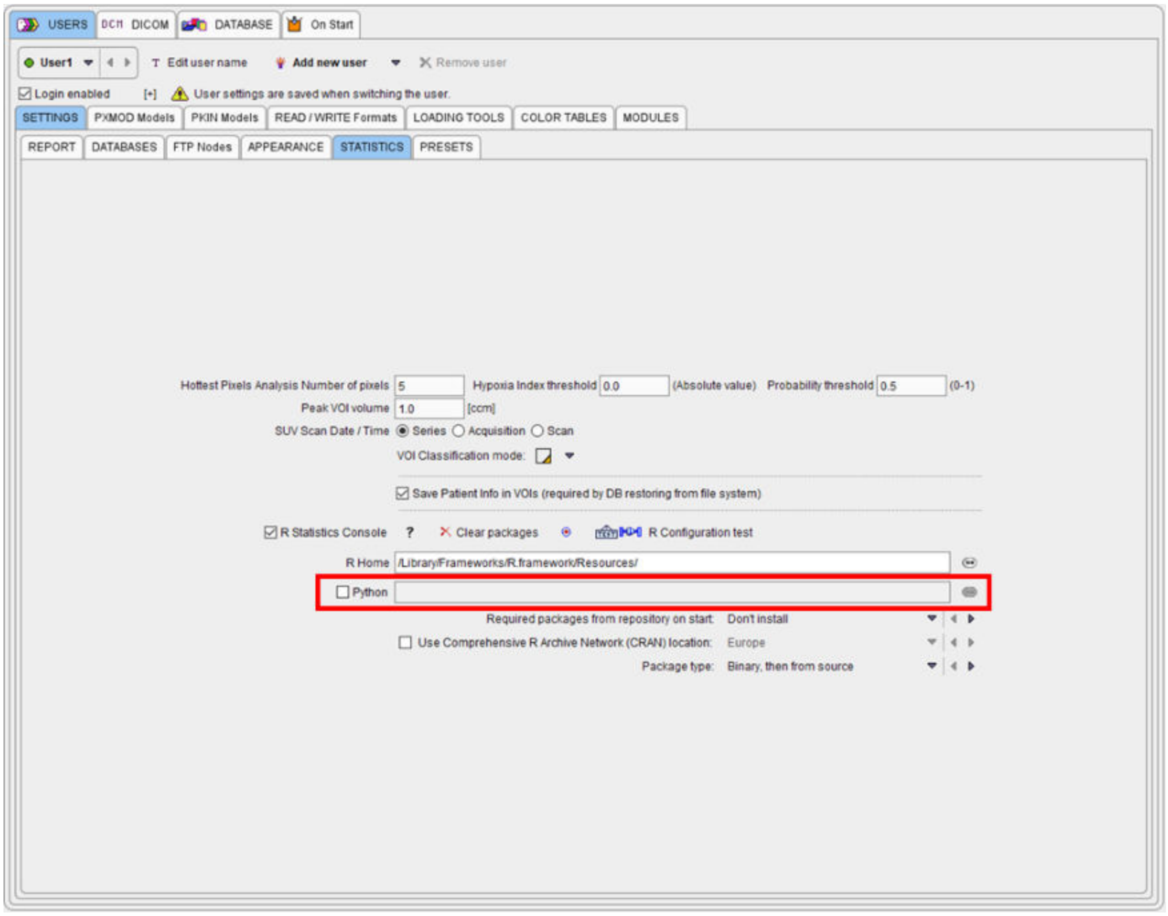


```
R version 4.2.2 (2022-10-31 ucrt) -- "Innocent and Trusting"
Copyright (C) 2022 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

Loading default settings ... Loaded.
Package doBy not loaded.
Package psych not loaded.
Package UsingR not loaded.
Package lawstat not loaded.
Package tseries not loaded.
Package np not loaded.
Package openair not loaded.
Package Hmisc not loaded.
Package car not loaded.
Package phia not loaded.
Package compare not loaded.
Package reshape not loaded.
Package Cairo not loaded.
Package pROC not loaded.
Package glmnet not loaded.
Package mcr not loaded.

Execution test ... Passed.
Pmod PAI diagnostics test ... Passed.
Copying file BRATS_Tumor_Detection.h5 to the working directory... OK.
Copying file BRATS_Tumor_Detection.json to the working directory... OK.
Copying file architectures.py to the working directory... OK
```

2.2.3 Selection of Python installation

Multiple installations and versions of Python may cause the PAI infrastructure test to fail.

In this situation you should define the path to Python 3.8 in the **Configuration** facility from the main ToolBox:



☑ Login enabled [+] User settings are saved when switching the user.

Hottest Pixels Analysis Number of pixels: 5 — Hypoxia Index threshold: 0.0 (Absolute value) — Probability threshold: 0.5 (0-1)

Peak VOI volume: 1.0 [ccm]

SUV Scan Date / Time: [✓ Series] Acquisition Scan

VOI Classification mode:

☑ Save Patient Info in VOIs (required by DB restoring from file system)

☑ R Statistics Console

R Home: /Library/Frameworks/R.framework/Resources/

☐ Python

Required packages from repository on start: Don't install

☐ Use Comprehensive R Archive Network (CRAN) location: Europe

Package type: Binary, then from source

Restart PMOD after setting the path to Python 3.8.

2.3 Linux Platforms

2.3.1 R Installation

PAI requires R version 4.2.2. Please perform the following installation steps (Ubuntu 18.04):

Install R:

```
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys
E298A3A825C0D65DFD57CBB651716619E084DAB9

sudo add-apt-repository 'deb https://cloud.r-
project.org/bin/linux/ubuntu bionic-cran40/' sudo apt update

sudo apt install r-recommended
```

Install the required libraries:

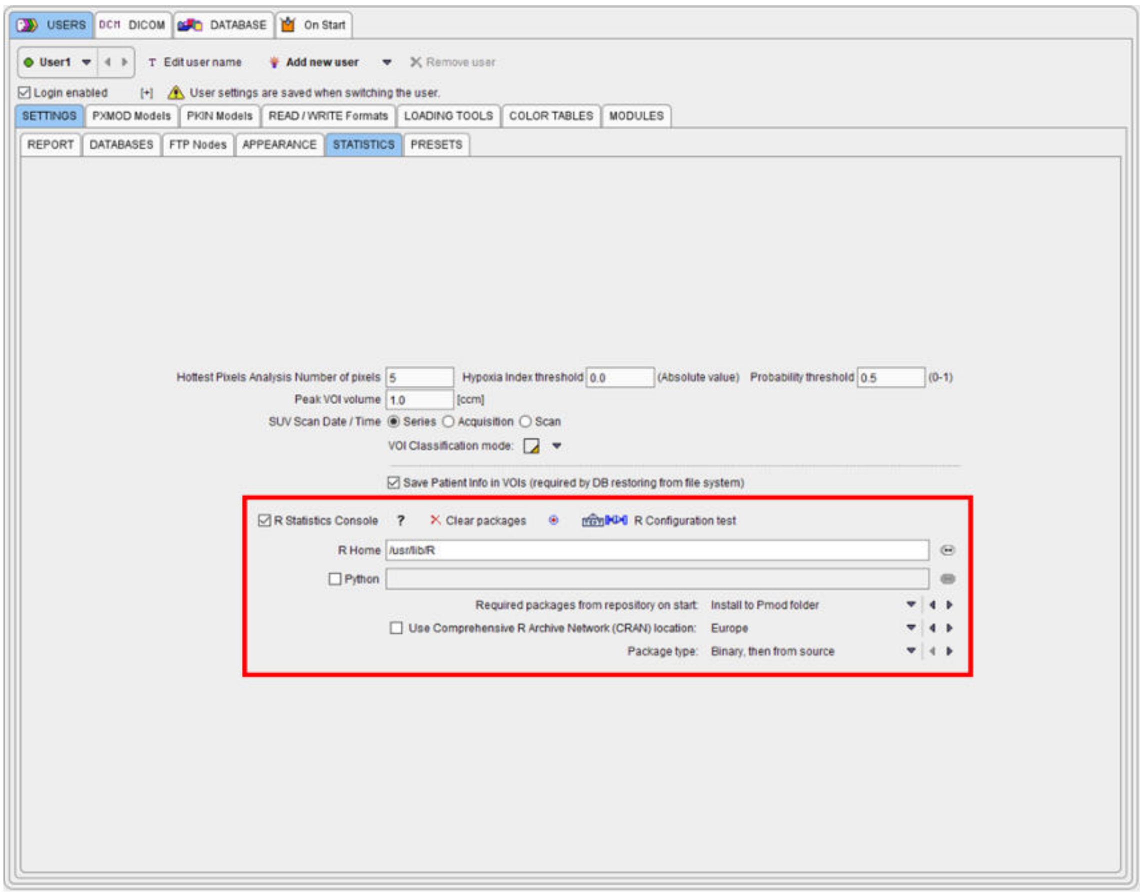
```
sudo apt install libcurl4-openssl-dev

sudo apt install libcairo2-dev

sudo apt install xorg-dev

sudo apt install libssl-dev
sudo add-apt-repository ppa:c2d4u.team/c2d4u4.0+

sudo apt-get update

sudo apt-get install r-cran-lme4

sudo apt-get install r-cran-snow

sudo apt-get install r-cran-vgam
```

Start R at the command line as administrator (“sudo R”) and install the required packages:

```
install.packages("doBy")

install.packages("psych")

install.packages("e1071")

install.packages("UsingR")

install.packages("lawstat")

install.packages("tseries")

install.packages("np")

install.packages("openair")

install.packages("foreign")

install.packages("Hmisc")

install.packages("car")

install.packages("phia")

install.packages("compare")

install.packages("reshape")

install.packages("sfsmisc")

install.packages("Cairo")

install.packages("pROC")

install.packages("survival")

install.packages("glmnet")

install.packages("mcr")

install.packages("stringr")

install.packages("reticulate")

install.packages("jsonlite")
install.packages("lubridate")

install.packages("sfsmisc")

install.packages("RNifti")

install.packages("memuse")
```

Main Configuration in PMOD User Interface

Following R and package installation, launch PMOD and check that R Statistics Console is activated in the Main Configuration, on the USERS, STATISTICS tab:



☑ Login enabled

Hottest Pixels Analysis Number of pixels: 5
Hypoxia Index threshold: 0.0 (Absolute value)
Probability threshold: 0.5 (0-1)
Peak VOI volume: 1.0 [ccm]
SUV Scan Date / Time: [✓ Series] Acquisition Scan
VOI Classification mode:
☑ Save Patient Info in VOIs (required by DB restoring from file system)

☑ R Statistics Console
R Home: /usr/lib/R
☐ Python
Required packages from repository on start: Install to Pmod folder
☐ Use Comprehensive R Archive Network (CRAN) location: Europe
Package type: Binary, then from source

2.3.2 Python and TensorFlow Installation

Python version 3.8 is required. Install Python, TensorFlow and Scikit-Learn as follows:

```
sudo apt update

sudo apt upgrade

sudo apt install build-essential

sudo apt install python3
sudo apt install python3-pip

pip3 install -U pip

pip3 install -U tensorflow==2.10.0

pip3 install -U scikit-learn
```

Install PyTorch either for CPU only:

```
pip3 install -U torch
```

or install PyTorch for GPU:

```
pip3 install -U torch --extra-index-url
https://download.pytorch.org/whl/cu116
```

Note: TensorFlow can alternatively be installed in Python’s virtual environment. scikit-learn can be tested using the command: `python3 -c “import sklearn; print(sklearn.__version__)”`

3 Preparation of Training Data and Neural Network

3.1 Data Preparation

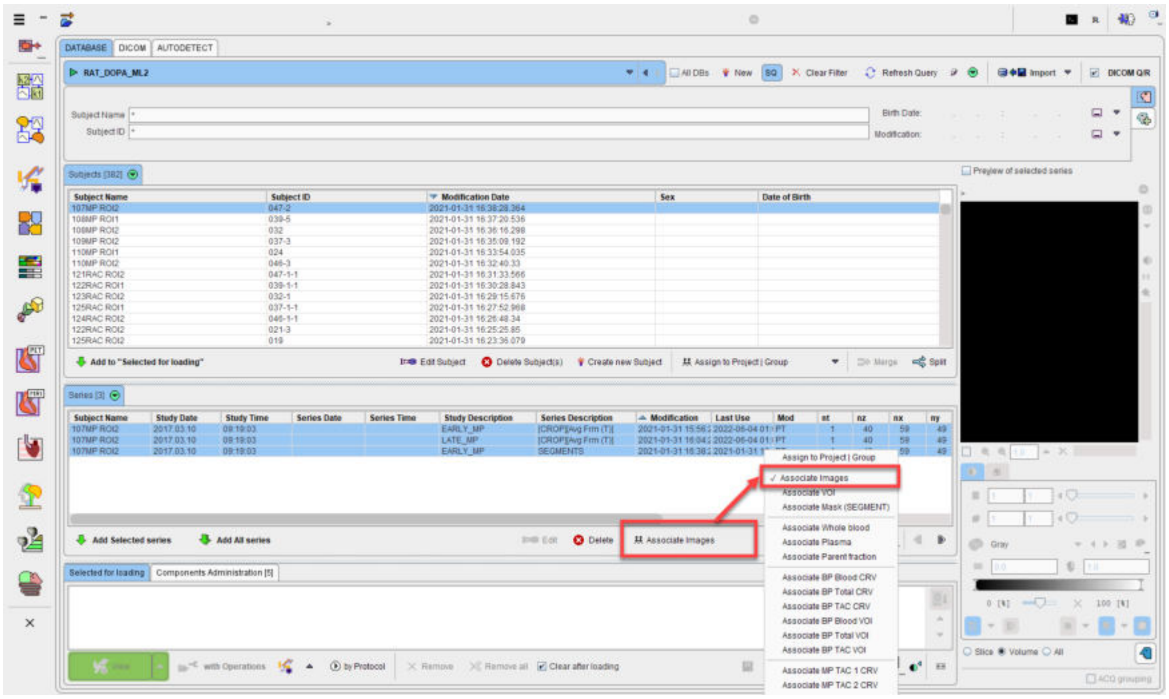
Image Import

The use of a database is a prerequisite for developing an ML model in PAI. Please refer to the *PMOD Basic Functionality User Guide* for instructions how to create and use databases. In the example below a database called **BraTS** was created and the data from the MICCAI BraTS Challenge imported.

Image Association

A training sample consists of one or several image series, and either the segmentation reference result or class for classification from which the neural network should learn. All of these images need to be associated in the database so that when a single image is referenced all related images are identified.

To associate the images, select a subject in the **Subjects** list and then all series to be associated in the **Series** list. From the option menu indicated below select **Associate Images**, which brings up a dialog window confirming association of the selected series.



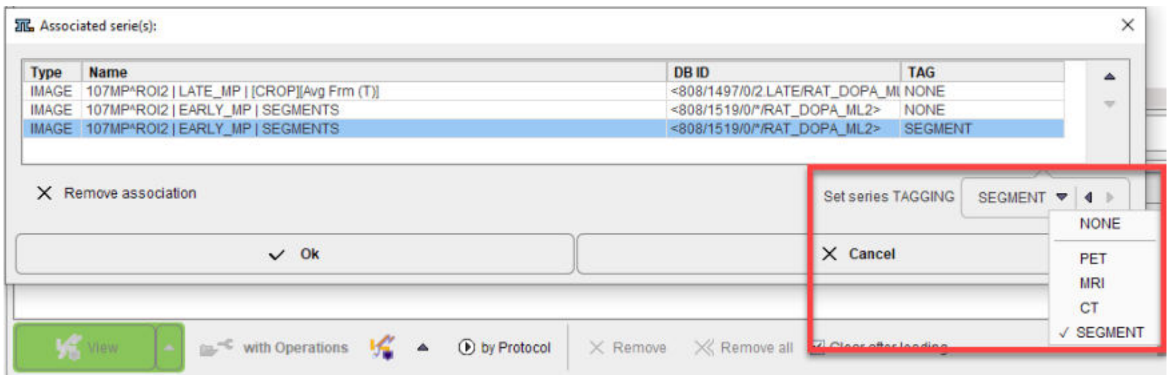
Subject Name: *
Subject ID: *

Subjects [382]

Subject Name	Subject ID	Modification Date	Sex	Date of Birth
107MP ROI2	047-2	2021-01-31 16:38:28.364		
108MP ROI1	039-5	2021-01-31 16:37:20.536		
108MP ROI2	032	2021-01-31 16:36:16.298		
109MP ROI2	037-3	2021-01-31 16:35:09.192		
110MP ROI1	024	2021-01-31 16:33:54.035		
110MP ROI2	046-3	2021-01-31 16:32:40.33		
121RAC ROI2	047-1-1	2021-01-31 16:31:33.566		
122RAC ROI1	039-1-1	2021-01-31 16:30:28.843		
123RAC ROI2	032-1	2021-01-31 16:29:15.676		
125RAC ROI1	037-1-1	2021-01-31 16:27:52.968		
124RAC ROI2	046-1-1	2021-01-31 16:26:48.34		
122RAC ROI2	021-3	2021-01-31 16:25:25.85		
125RAC ROI2	019	2021-01-31 16:23:36.079		

Series [3]

Subject Name	Study Date	Study Time	Series Date	Series Time	Study Description	Series Description	Modification	Last Use	Mod	nt	nz	nx	ny
107MP ROI2	2017.03.10	09:19:03			EARLY_MP	[CROP][Avg Frm (T)]	2021-01-31 15:56:2	2022-06-04 01:	PT	1	40	59	49
107MP ROI2	2017.03.10	09:19:03			LATE_MP	[CROP][Avg Frm (T)]	2021-01-31 16:04:2	2022-06-04 01:	PT	1	40	59	49
107MP ROI2	2017.03.10	09:19:03			EARLY_MP	SEGMENTS	2021-01-31 16:38:2	2021-01-31 1				59	49

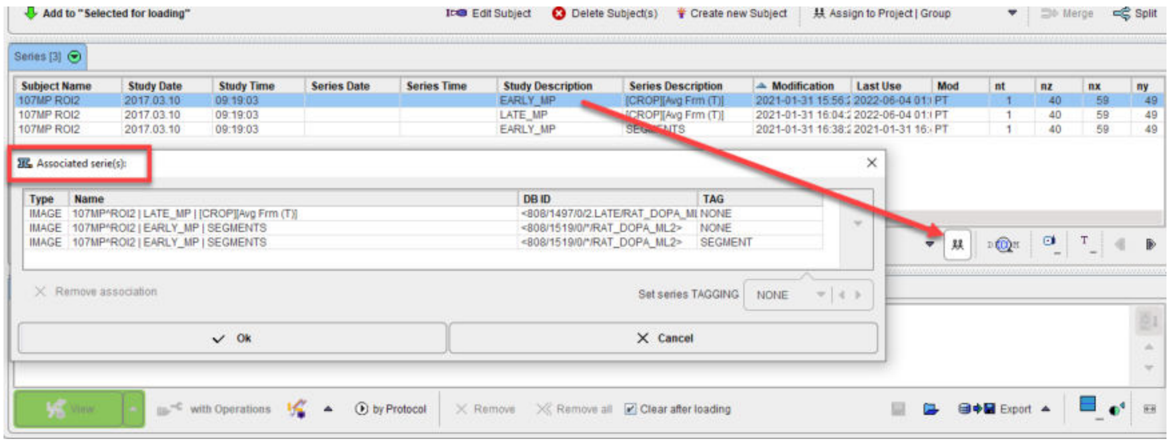


Type	Name	DB ID	TAG
IMAGE	107MP^ROI2 | LATE_MP | [CROP][Avg Frm (T)]	<808/1497/0/2.LATE/RAT_DOPA_ML	NONE
IMAGE	107MP^ROI2 | EARLY_MP | SEGMENTS	<808/1519/0/^/RAT_DOPA_ML2>	NONE
IMAGE	107MP^ROI2 | EARLY_MP | SEGMENTS	<808/1519/0/^/RAT_DOPA_ML2>	SEGMENT

Set series TAGGING: SEGMENT

To identify which image series is the reference segment map, select it in the list, then the **TAG** column, and in the menu that appears select the **SEGMENT** entry. If more than one input image is required for the segmentation, it is important that they always appear in the same order in the association list. Please use the arrow buttons to the right of the list for shifting the position of a selected element.

Existing associations can be checked by selecting one of the image series and activating the button indicated below:



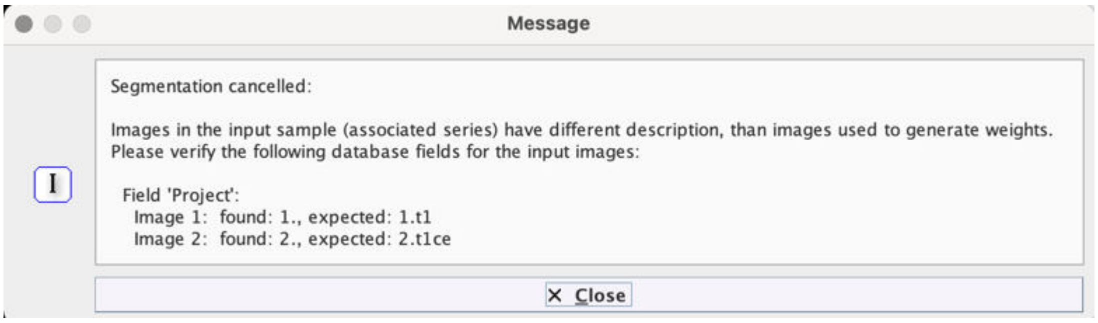
Subject Name	Study Date	Study Time	Series Date	Series Time	Study Description	Series Description	Modification	Last Use	Mod	nt	nz	nx	ny
107MP ROI2	2017.03.10	09:19:03			EARLY_MP	[CROP][Avg Frm (T)]	2021-01-31 15:56:2	2022-06-04 01:	PT	1	40	59	49
107MP ROI2	2017.03.10	09:19:03			LATE_MP	[CROP][Avg Frm (T)]	2021-01-31 16:04:2	2022-06-04 01:	PT	1	40	59	49
107MP ROI2	2017.03.10	09:19:03			EARLY_MP	SEGMENTS	2021-01-31 16:38:2	2021-01-31 16:	PT	1	40	59	49

Type	Name	DB ID	TAG
IMAGE	107MP^ROI2 | LATE_MP | [CROP][Avg Frm (T)]	<808/1497/0/2.LATE/RAT_DOPA_MI	NONE
IMAGE	107MP^ROI2 | EARLY_MP | SEGMENTS	<808/1519/0/*/RAT_DOPA_ML2>	NONE
IMAGE	107MP^ROI2 | EARLY_MP | SEGMENTS	<808/1519/0/*/RAT_DOPA_ML2>	SEGMENT

Adding a Descriptive Variable or Class for Training (Project Description)

For segmentation projects we strongly recommend adding a descriptive label to the series used for training by defining the Project using **Assign to Project | Group**. This description will be used to check that new data used for Prediction has the same content as that used for Training.

If a difference in the Project description (or number of studies) in Training/Prediction is detected, warning messages based on the following structure will be returned:

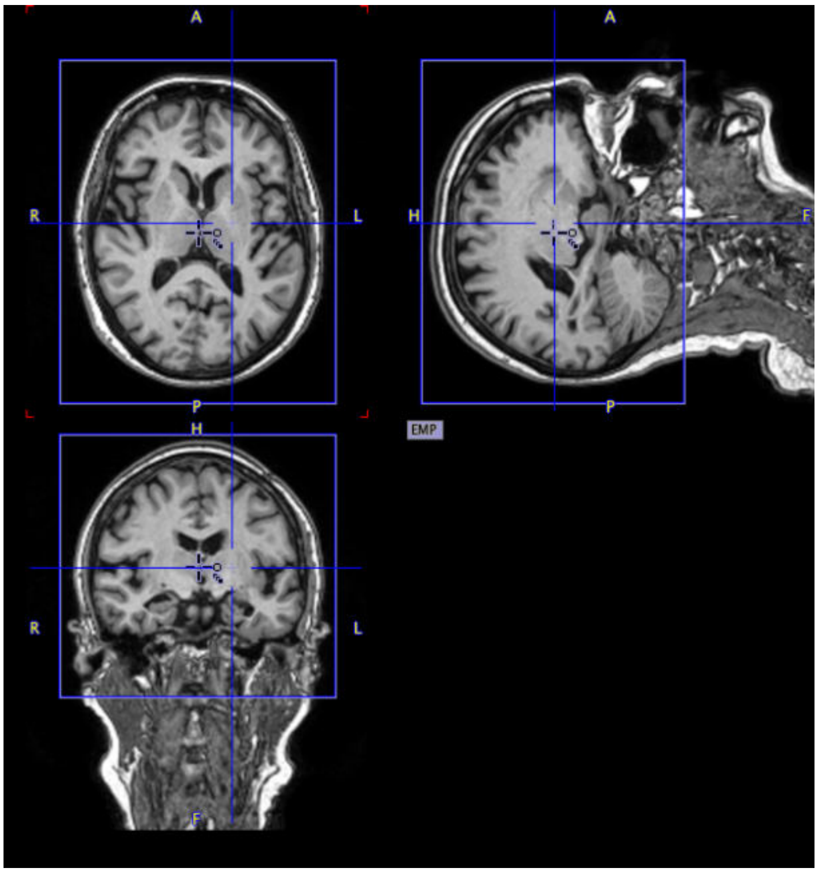


```
Segmentation cancelled:

Images in the input sample (associated series) have different description, than images used to generate weights.
Please verify the following database fields for the input images:

 Field 'Project':
   Image 1:  found: 1., expected: 1.t1
   Image 2:  found: 2., expected: 2.t1ce
```

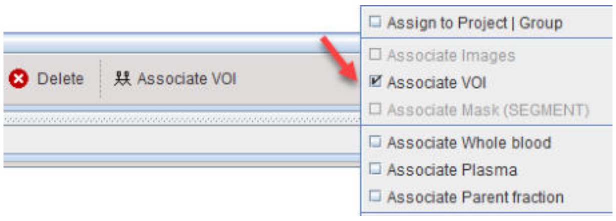
For classification projects the Project label is required as assignment of the Class that the sample belongs to. For example, see the Amyloid PET classification Case Study: samples are in either "AD" or "YC" class (Alzheimer's disease or Young Control) and the model is trained to Predict (with probability) which of these classes a new sample belongs.

Data Cropping

Another part of the data preparation consists of reducing the data volume to the relevant portion. In the brain segmentation example the image should be restricted to the brain. This process can be included in the training set definition by creating a VOI that will serve as the cropping box and associating it with the input data using the same tools as image association.



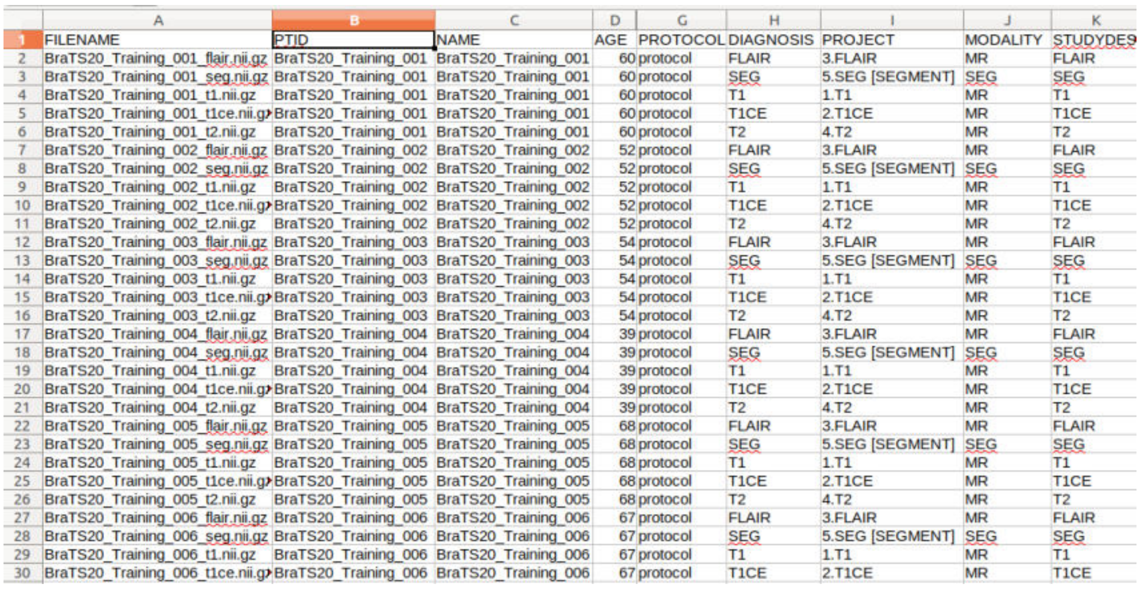
To achieve this, open the input image, create a suitable VOI such as a box, position it properly and save it to the database. Then select the input image in the **Series** list on the DB Load page, followed by **Associate VOI** from the same menu where the images were associated.



In the dialog window which opens select the saved VOI and activate **Set Selected**.

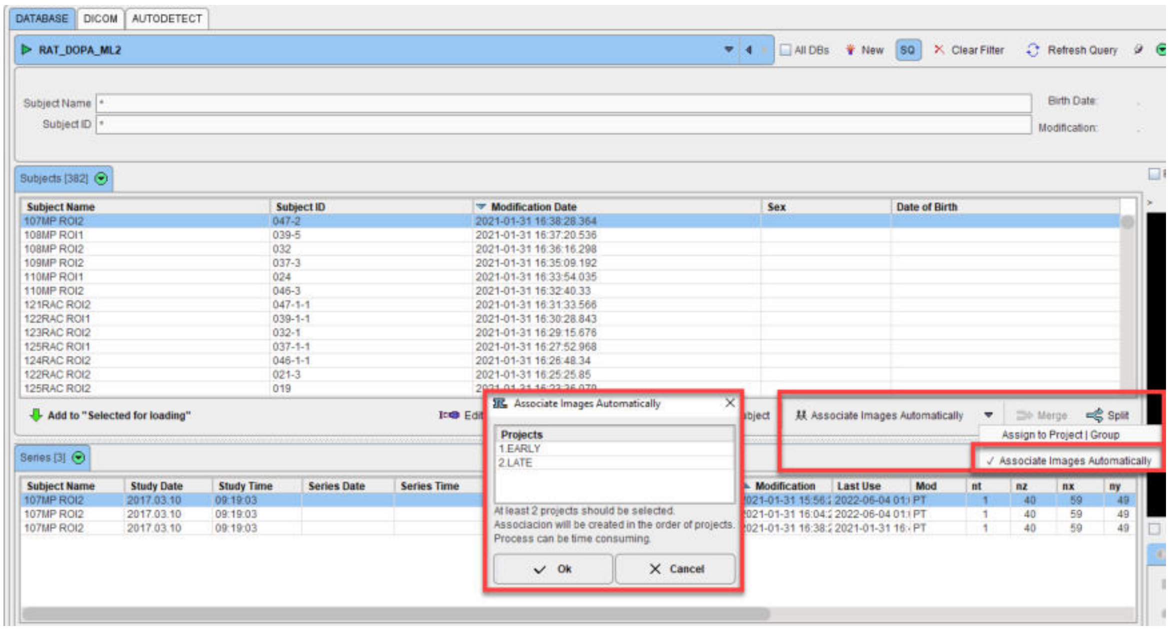
Automatic Association Creation

The neural network training process requires the preparation of a large number of samples. To make this process easier a mechanism for the automatic association of the images is available. It uses either the **Incoming Folder** method or batch assignment of Project using database queries. A folder that is regularly checked for data to be imported into the database is defined in the **DICOM Server** configuration. It takes into account information prepared in a csv file that must also be located in the incoming folder. The structure of such a csv file is illustrated below:



	A	B	C	D	G	H	I	J	K
1	FILENAME	PTID	NAME	AGE	PROTOCOL	DIAGNOSIS	PROJECT	MODALITY	STUDYDES
2	BraTS20_Training_001_flair.nii.gz	BraTS20_Training_001	BraTS20_Training_001	60	protocol	FLAIR	3.FLAIR	MR	FLAIR
3	BraTS20_Training_001_seg.nii.gz	BraTS20_Training_001	BraTS20_Training_001	60	protocol	SEG	5.SEG [SEGMENT]	SEG	SEG
4	BraTS20_Training_001_t1.nii.gz	BraTS20_Training_001	BraTS20_Training_001	60	protocol	T1	1.T1	MR	T1
5	BraTS20_Training_001_t1ce.nii.g	BraTS20_Training_001	BraTS20_Training_001	60	protocol	T1CE	2.T1CE	MR	T1CE
6	BraTS20_Training_001_t2.nii.gz	BraTS20_Training_001	BraTS20_Training_001	60	protocol	T2	4.T2	MR	T2
7	BraTS20_Training_002_flair.nii.gz	BraTS20_Training_002	BraTS20_Training_002	52	protocol	FLAIR	3.FLAIR	MR	FLAIR
8	BraTS20_Training_002_seg.nii.gz	BraTS20_Training_002	BraTS20_Training_002	52	protocol	SEG	5.SEG [SEGMENT]	SEG	SEG
9	BraTS20_Training_002_t1.nii.gz	BraTS20_Training_002	BraTS20_Training_002	52	protocol	T1	1.T1	MR	T1
10	BraTS20_Training_002_t1ce.nii.g	BraTS20_Training_002	BraTS20_Training_002	52	protocol	T1CE	2.T1CE	MR	T1CE
11	BraTS20_Training_002_t2.nii.gz	BraTS20_Training_002	BraTS20_Training_002	52	protocol	T2	4.T2	MR	T2
12	BraTS20_Training_003_flair.nii.gz	BraTS20_Training_003	BraTS20_Training_003	54	protocol	FLAIR	3.FLAIR	MR	FLAIR
13	BraTS20_Training_003_seg.nii.gz	BraTS20_Training_003	BraTS20_Training_003	54	protocol	SEG	5.SEG [SEGMENT]	SEG	SEG
14	BraTS20_Training_003_t1.nii.gz	BraTS20_Training_003	BraTS20_Training_003	54	protocol	T1	1.T1	MR	T1
15	BraTS20_Training_003_t1ce.nii.g	BraTS20_Training_003	BraTS20_Training_003	54	protocol	T1CE	2.T1CE	MR	T1CE
16	BraTS20_Training_003_t2.nii.gz	BraTS20_Training_003	BraTS20_Training_003	54	protocol	T2	4.T2	MR	T2
17	BraTS20_Training_004_flair.nii.gz	BraTS20_Training_004	BraTS20_Training_004	39	protocol	FLAIR	3.FLAIR	MR	FLAIR
18	BraTS20_Training_004_seg.nii.gz	BraTS20_Training_004	BraTS20_Training_004	39	protocol	SEG	5.SEG [SEGMENT]	SEG	SEG
19	BraTS20_Training_004_t1.nii.gz	BraTS20_Training_004	BraTS20_Training_004	39	protocol	T1	1.T1	MR	T1
20	BraTS20_Training_004_t1ce.nii.g	BraTS20_Training_004	BraTS20_Training_004	39	protocol	T1CE	2.T1CE	MR	T1CE
21	BraTS20_Training_004_t2.nii.gz	BraTS20_Training_004	BraTS20_Training_004	39	protocol	T2	4.T2	MR	T2
22	BraTS20_Training_005_flair.nii.gz	BraTS20_Training_005	BraTS20_Training_005	68	protocol	FLAIR	3.FLAIR	MR	FLAIR
23	BraTS20_Training_005_seg.nii.gz	BraTS20_Training_005	BraTS20_Training_005	68	protocol	SEG	5.SEG [SEGMENT]	SEG	SEG
24	BraTS20_Training_005_t1.nii.gz	BraTS20_Training_005	BraTS20_Training_005	68	protocol	T1	1.T1	MR	T1
25	BraTS20_Training_005_t1ce.nii.g	BraTS20_Training_005	BraTS20_Training_005	68	protocol	T1CE	2.T1CE	MR	T1CE
26	BraTS20_Training_005_t2.nii.gz	BraTS20_Training_005	BraTS20_Training_005	68	protocol	T2	4.T2	MR	T2
27	BraTS20_Training_006_flair.nii.gz	BraTS20_Training_006	BraTS20_Training_006	67	protocol	FLAIR	3.FLAIR	MR	FLAIR
28	BraTS20_Training_006_seg.nii.gz	BraTS20_Training_006	BraTS20_Training_006	67	protocol	SEG	5.SEG [SEGMENT]	SEG	SEG
29	BraTS20_Training_006_t1.nii.gz	BraTS20_Training_006	BraTS20_Training_006	67	protocol	T1	1.T1	MR	T1
30	BraTS20_Training_006_t1ce.nii.g	BraTS20_Training_006	BraTS20_Training_006	67	protocol	T1CE	2.T1CE	MR	T1CE

The label defined in the **Project** column is assigned to the imported image series. Once imported, **Associate Images Automatically** can be used to generate the associations. Note that in the example, four images in each sample are used as input for the segmentation according to the requirements for the MICCAI BraTS Challenge. To establish a consistent order, numbers are used in the labels.



Subject Name	Subject ID	Modification Date	Sex	Date of Birth
107MP ROI2	047-2	2021-01-31 16:38:28.364		
108MP ROI1	039-5	2021-01-31 16:37:20.536		
108MP ROI2	032	2021-01-31 16:36:16.298		
109MP ROI2	037-3	2021-01-31 16:35:09.192		
110MP ROI1	024	2021-01-31 16:33:54.035		
110MP ROI2	046-3	2021-01-31 16:32:40.33		
121RAC ROI2	047-1-1	2021-01-31 16:31:33.566		
122RAC ROI1	039-1-1	2021-01-31 16:30:28.843		
123RAC ROI2	032-1	2021-01-31 16:29:15.676		
125RAC ROI1	037-1-1	2021-01-31 16:27:52.968		
124RAC ROI2	046-1-1	2021-01-31 16:26:48.34		
122RAC ROI2	021-3	2021-01-31 16:25:25.85		
125RAC ROI2	019	[illegible]		

Subject Name	Study Date	Study Time	Series Date	Series Time	Modification	Last Use	Mod	nt	nz	nx	ny
107MP ROI2	2017.03.10	09:19:03			021-01-31 15:56:	2022-06-04 01:	PT	1	40	59	49
107MP ROI2	2017.03.10	09:19:03			021-01-31 16:04:	2022-06-04 01:	PT	1	40	59	49
107MP ROI2	2017.03.10	09:19:03			021-01-31 16:38:	2021-01-31 16:	PT	1	40	59	49

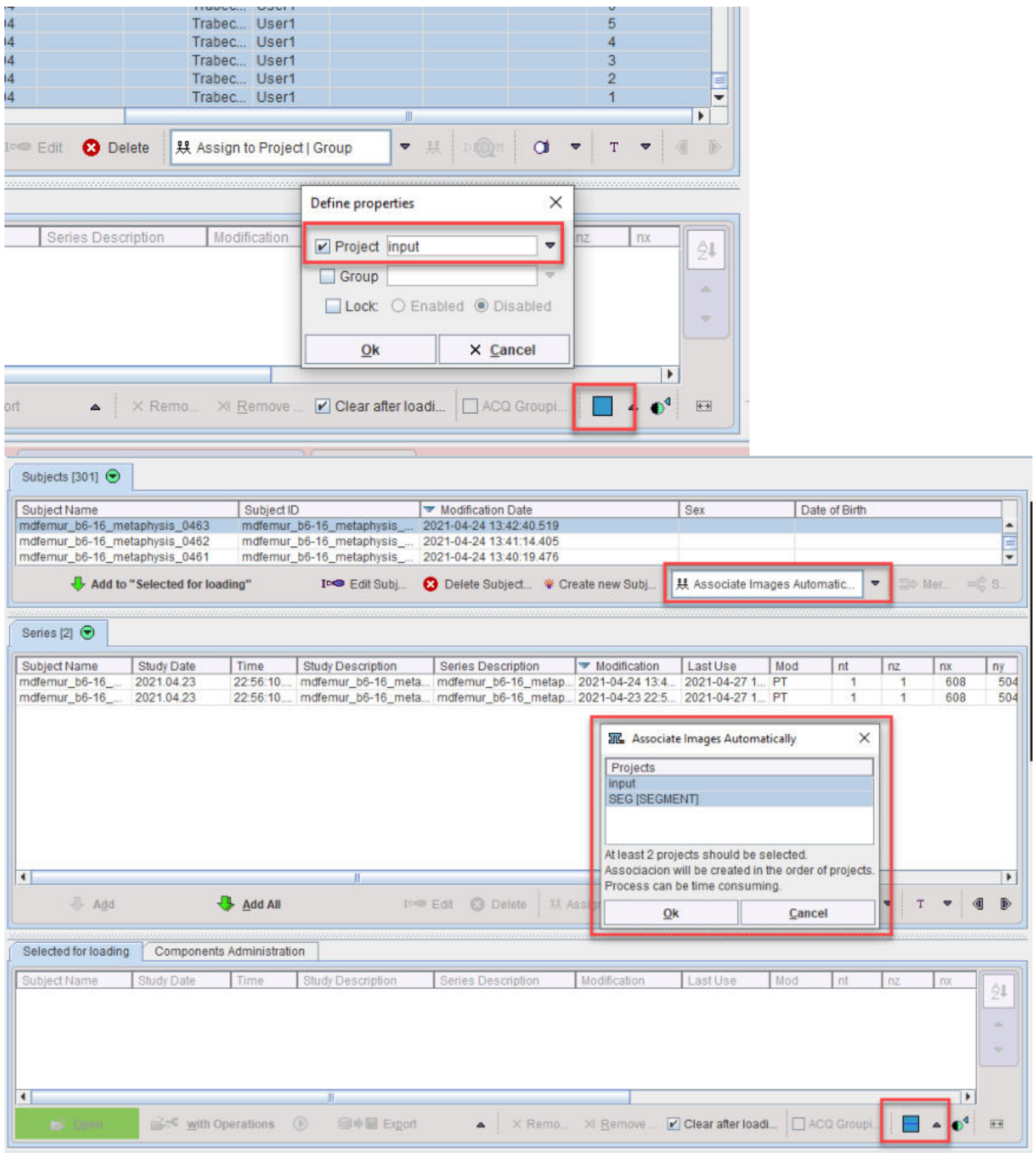
Associate Images Automatically

Projects
1.EARLY
2.LATE

At least 2 projects should be selected.
Associacion will be created in the order of projects.
Process can be time consuming.

Ok | Cancel

Automatic association may also be used for data already in a database, when advanced database query can be used to easily batch label series. For example, when each subject in the database has only a single image series and single segment series, all image series can be given the Project “input” and all segments the Project “[SEGMENT]”. These two Projects are then selected in the Associate Images Automatically dialog.



Define properties

- [✓] Project: input
- [ ] Group
- [ ] Lock: ○ Enabled ◉ Disabled

Ok | Cancel

Subjects [301]

Subject Name	Subject ID	Modification Date	Sex	Date of Birth
mdfemur_b6-16_metaphysis_0463	mdfemur_b6-16_metaphysis_...	2021-04-24 13:42:40.519		
mdfemur_b6-16_metaphysis_0462	mdfemur_b6-16_metaphysis_...	2021-04-24 13:41:14.405		
mdfemur_b6-16_metaphysis_0461	mdfemur_b6-16_metaphysis_...	2021-04-24 13:40:19.476		

Series [2]

Subject Name	Study Date	Time	Study Description	Series Description	Modification	Last Use	Mod	nt	nz	nx	ny
mdfemur_b6-16_...	2021.04.23	22:56:10....	mdfemur_b6-16_meta...	mdfemur_b6-16_metap...	2021-04-24 13:4...	2021-04-27 1...	PT	1	1	608	504
mdfemur_b6-16_...	2021.04.23	22:56:10....	mdfemur_b6-16_meta...	mdfemur_b6-16_metap...	2021-04-23 22:5...	2021-04-27 1...	PT	1	1	608	504

Associate Images Automatically

Projects
input
SEG [SEGMENT]

At least 2 projects should be selected.
Associacion will be created in the order of projects.
Process can be time consuming.

Ok | Cancel

3.2 Control Mechanism

Data Consistency

A prerequisite of training a neural network is that all samples are consistent. For example, the MICCAI BraTS example described in the Multichannel MRI Segmentation (brain tumor) Case Study expects four input MR images, each from a particular type of MR sequence, and reference segments identifying three tissue types for segmentation. Adding a sample with only two input MR images, or reference segments with five labels provokes a failure. Likewise, prediction using the trained neural network will fail with data of a different structure.

In a classification project, all samples must have a Class assigned or a warning that unassigned samples are present will be returned and the samples excluded from training. During prediction the use of input data with different dimensionality to the data used for training will return a warning, as will missing VOIs used as part of data reduction in the SVM architecture (see Amyloid PET classification Case Study).

Manifest File

To ensure consistency, PAI uses a manifest file (JSON format) to store information about the training process and history. The consistency check includes the number and type of datasets included, their descriptions, as well as the pre-processing procedures. The manifest file is based on the first valid sample ("leading sample") when the first training occurs. The manifest file is additionally associated with the weights file resulting from training of a model. If a mismatch between manifest and weights is detected when importing/retrieving these files in the Edit Learning Set dialog, a warning will be returned when additional training, or prediction, is attempted.

Data Checks

The following checks are included;

- Proper loading of the associated images – checking whether all associated images are properly loaded.
- Number of dimensions – the number of dimensions of the input images must be consistent with the number of dimensions of the images used for the original training. It must also be consistent with the number of dimensions supported by the chosen model.
- Number of associated images – the number of associated input images must be the same as the number of images associated in the samples used for the original training.
- Segmentation: "Project" field, serving as image description – the description of the associated input images stored in the "Project" database field must be the same as in the images used for the original training
- Classification: "Group" field, class assignment - the classes detected are reported before training is started and warnings are returned for samples missing a class, or if less than two classes are detected, or if the selected samples do not contain all classes detected in the learning set (manifest)
- Modalities – the modalities of the associated input images must be the same as for the images used for the original training.

If one of the above requirements is not met, PAI will do the following according to the workflow in use:

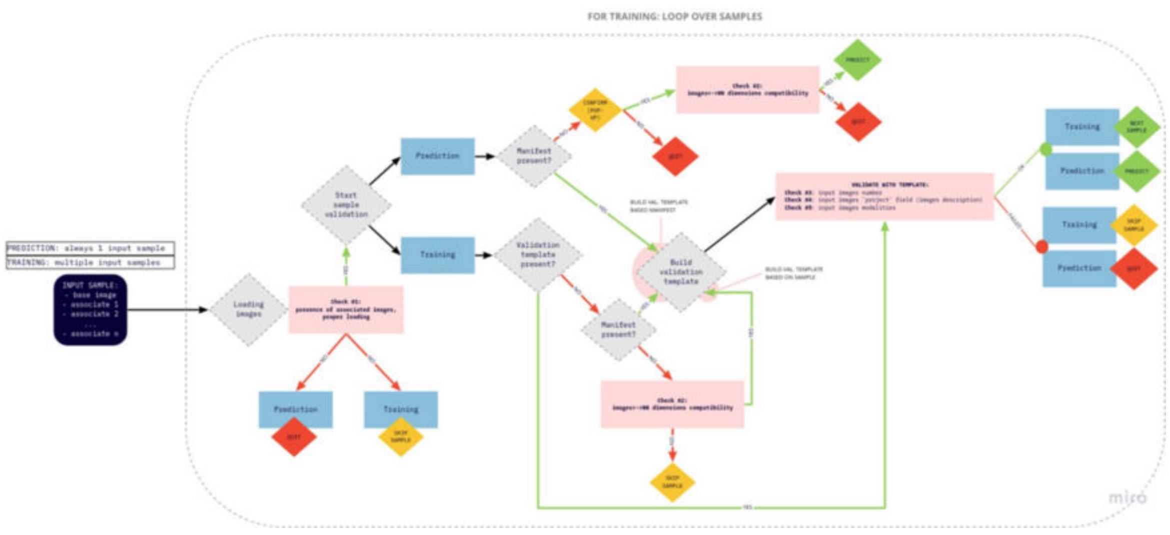
- Prediction: Cancel processing and inform the user
- Re-training: Skip the sample and print the information in the log

Model Configuration Check

The final control checks the settings in the user interface. Every time the user starts a training or saves the learning set, the content of the selected manifest file will be compared to the current settings in the user interface. This avoids accidental use of training parameters that differ from previous training sessions. The user is prompted to copy either all or only critical settings between manifest and learning set.

Control Overview

The data and model consistency checks are performed in the background during the workflows. Issues will be automatically detected and the user guided to correct them.


FOR TRAINING: LOOP OVER SAMPLES
PREDICTION: always 1 input sample
TRAINING: multiple input samples
INPUT SAMPLE:
- base image
- associate 1
- associate 2
...
- associate n
Loading images
Check #1: presence of associated images, proper loading
Prediction
QUIT
Training
SKIP SAMPLE
Start sample validation
Prediction
Training
Manifest present?
Validation template present?
CONFIRM (POP-UP)
Check #2: images<->NN dimensions compatibility
PROJECT
QUIT
BUILD VAL. TEMPLATE BASED MANIFEST
Build validation template
BUILD VAL. TEMPLATE BASED ON SAMPLE
Manifest present?
Check #2: images<->NN dimensions compatibility
SKIP SAMPLE
VALIDATE WITH TEMPLATE:
Check #3: input images number
Check #4: input images 'project' field (images description)
Check #5: input images modalities
Training
NEXT SAMPLE
Prediction
PROJECT
Training
SKIP SAMPLE
Prediction
QUIT
YES
NO
miro

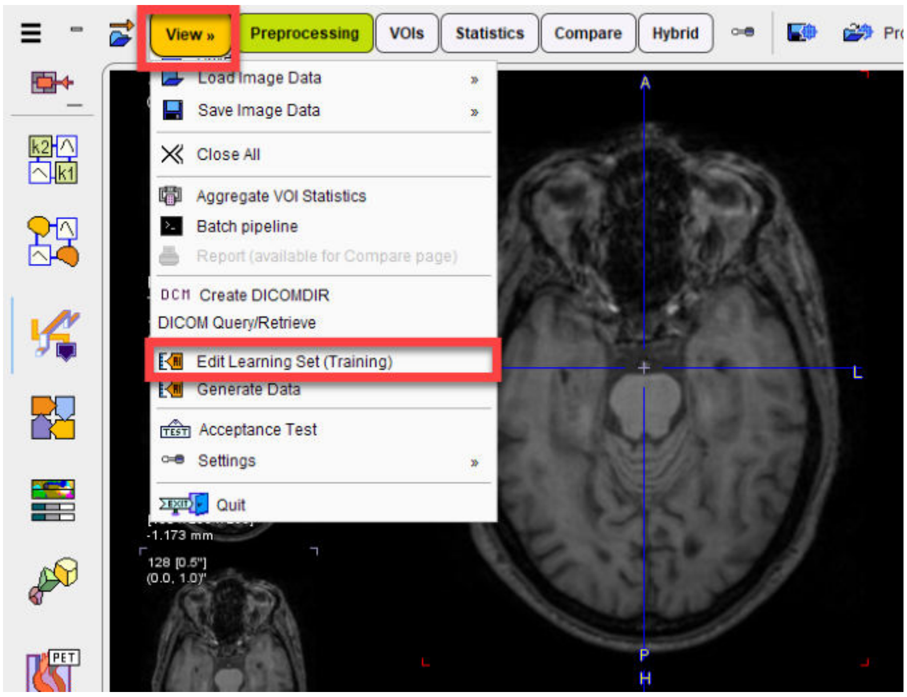
3.3 Learning Sets

In PAI, a **Learning Set** organizes all of the necessary data and parameters for training, including the data samples, the data preparation parameters and the neural network settings.

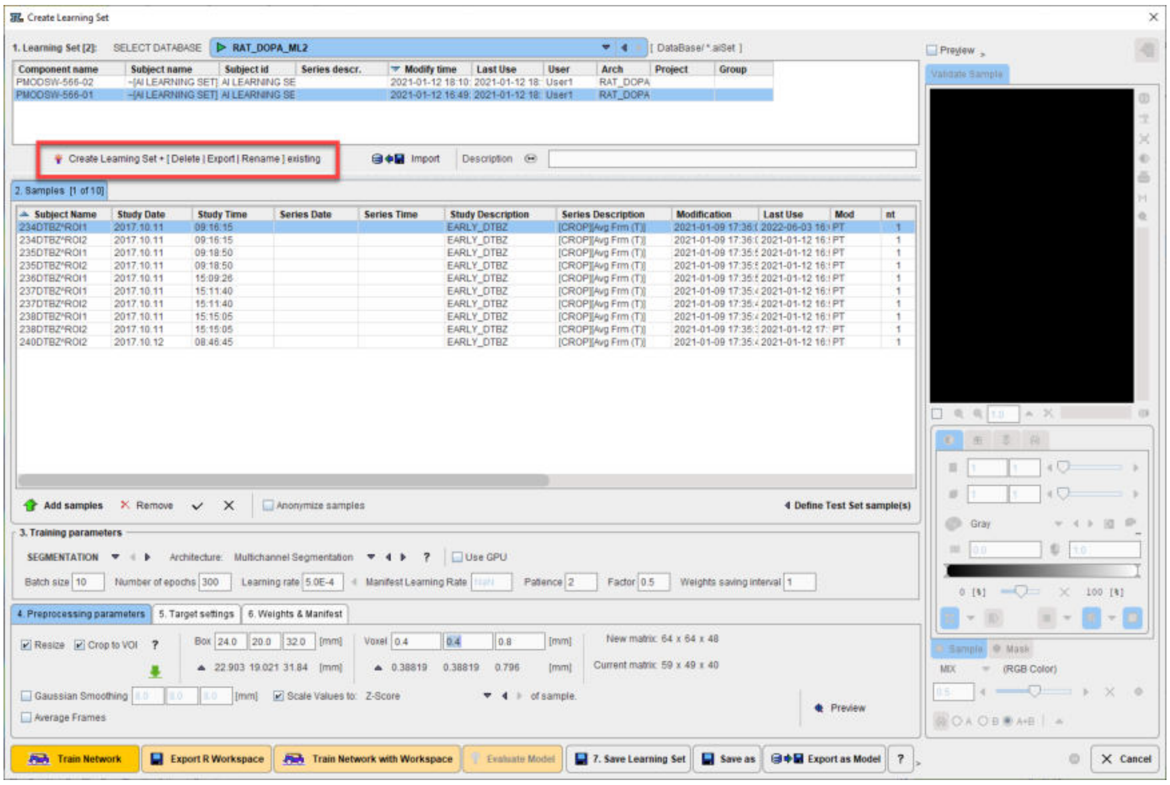
Note that any necessary data preparation 41 should be done before creating or extending a **Leaning Set**.

Learning Set Creation

To create a **Learning Set** or start a training run, select **Edit Learning Set (Training)** from the **View + AI** menu:



A dialog window appears that lists the existing learning sets in the upper section. It allows new learning sets to be created and existing leaning sets to be extended in the lower section.



Create Learning Set

1. Learning Set [2]: SELECT DATABASE RAT_DOPA_ML2 [DataBase/ *.aiSet]

Component name	Subject name	Subject id	Series descr.	Modify time	Last Use	User	Arch	Project	Group
PMODSW-566-02	~[AI LEARNING SET]	AI LEARNING SE		2021-01-12 18:10:	2021-01-12 18:	User1	RAT_DOPA		
PMODSW-566-01	~[AI LEARNING SET]	AI LEARNING SE		2021-01-12 16:49:	2021-01-12 18:	User1	RAT_DOPA		

Create Learning Set + [Delete | Export | Rename] existing | Import | Description

2. Samples [1 of 10]

Subject Name	Study Date	Study Time	Series Date	Series Time	Study Description	Series Description	Modification	Last Use	Mod	nt
234DTBZ^ROI1	2017.10.11	09:16:15			EARLY_DTBZ	[CROP][Avg Frm (T)]	2021-01-09 17:36:(	2022-06-03 16:!	PT	1
234DTBZ^ROI2	2017.10.11	09:16:15			EARLY_DTBZ	[CROP][Avg Frm (T)]	2021-01-09 17:36:(	2021-01-12 16:!	PT	1
235DTBZ^ROI1	2017.10.11	09:18:50			EARLY_DTBZ	[CROP][Avg Frm (T)]	2021-01-09 17:35:!	2021-01-12 16:!	PT	1
235DTBZ^ROI2	2017.10.11	09:18:50			EARLY_DTBZ	[CROP][Avg Frm (T)]	2021-01-09 17:35:!	2021-01-12 16:!	PT	1
236DTBZ^ROI1	2017.10.11	15:09:26			EARLY_DTBZ	[CROP][Avg Frm (T)]	2021-01-09 17:35:!	2021-01-12 16:!	PT	1
237DTBZ^ROI1	2017.10.11	15:11:40			EARLY_DTBZ	[CROP][Avg Frm (T)]	2021-01-09 17:35:4	2021-01-12 16:!	PT	1
237DTBZ^ROI2	2017.10.11	15:11:40			EARLY_DTBZ	[CROP][Avg Frm (T)]	2021-01-09 17:35:4	2021-01-12 16:!	PT	1
238DTBZ^ROI1	2017.10.11	15:15:05			EARLY_DTBZ	[CROP][Avg Frm (T)]	2021-01-09 17:35:4	2021-01-12 16:!	PT	1
238DTBZ^ROI2	2017.10.11	15:15:05			EARLY_DTBZ	[CROP][Avg Frm (T)]	2021-01-09 17:35:3	2021-01-12 17:	PT	1
240DTBZ^ROI2	2017.10.12	08:46:45			EARLY_DTBZ	[CROP][Avg Frm (T)]	2021-01-09 17:35:4	2021-01-12 16:!	PT	1

Add samples | Remove | Anonymize samples | Define Test Set sample(s)

3. Training parameters

SEGMENTATION | Architecture: Multichannel Segmentation | Use GPU

Batch size 10 | Number of epochs 300 | Learning rate 5.0E-4 | Manifest Learning Rate | Patience 2 | Factor 0.5 | Weights saving interval 1

4. Preprocessing parameters | 5. Target settings | 6. Weights & Manifest

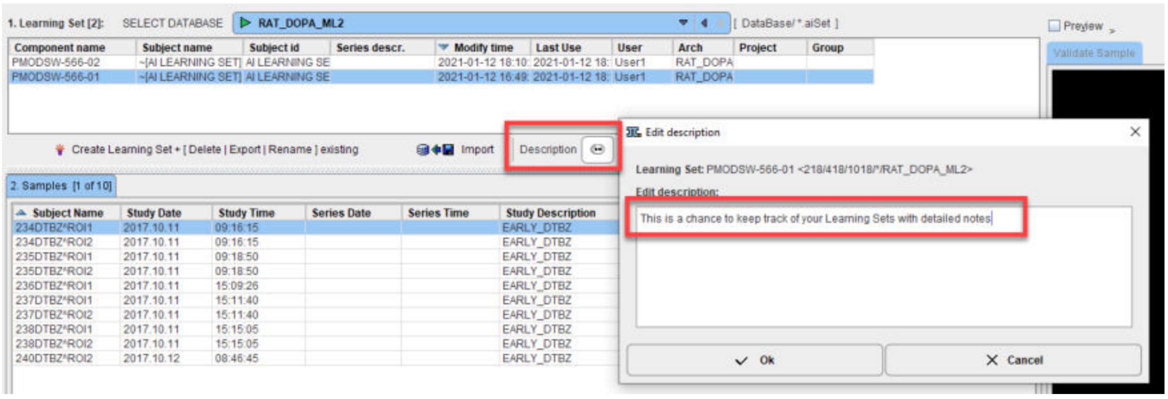
[✓] Resize [✓] Crop to VOI | Box 24.0 20.0 32.0 [mm] | Voxel 0.4 0.4 0.8 [mm] | New matrix: 64 x 64 x 48

22.903 19.021 31.84 [mm] | 0.38819 0.38819 0.796 [mm] | Current matrix: 59 x 49 x 40

[ ] Gaussian Smoothing 8.0 8.0 8.0 [mm] | [✓] Scale Values to: Z-Score of sample. | Preview

[ ] Average Frames

A text description can be appended to the **Learning Set** using the **Description** navigation button:



1. Learning Set [2]: SELECT DATABASE RAT_DOPA_ML2 [DataBase/ *.aiSet]

Component name	Subject name	Subject id	Series descr.	Modify time	Last Use	User	Arch	Project	Group
PMODSW-566-02	~[AI LEARNING SET]	AI LEARNING SE		2021-01-12 18:10:	2021-01-12 18:	User1	RAT_DOPA		
PMODSW-566-01	~[AI LEARNING SET]	AI LEARNING SE		2021-01-12 16:49:	2021-01-12 18:	User1	RAT_DOPA		

Create Learning Set + [Delete | Export | Rename] existing | Import | Description

2. Samples [1 of 10]

Subject Name	Study Date	Study Time	Series Date	Series Time	Study Description
234DTBZ^ROI1	2017.10.11	09:16:15			EARLY_DTBZ
234DTBZ^ROI2	2017.10.11	09:16:15			EARLY_DTBZ
235DTBZ^ROI1	2017.10.11	09:18:50			EARLY_DTBZ
235DTBZ^ROI2	2017.10.11	09:18:50			EARLY_DTBZ
236DTBZ^ROI1	2017.10.11	15:09:26			EARLY_DTBZ
237DTBZ^ROI1	2017.10.11	15:11:40			EARLY_DTBZ
237DTBZ^ROI2	2017.10.11	15:11:40			EARLY_DTBZ
238DTBZ^ROI1	2017.10.11	15:15:05			EARLY_DTBZ
238DTBZ^ROI2	2017.10.11	15:15:05			EARLY_DTBZ
240DTBZ^ROI2	2017.10.12	08:46:45			EARLY_DTBZ

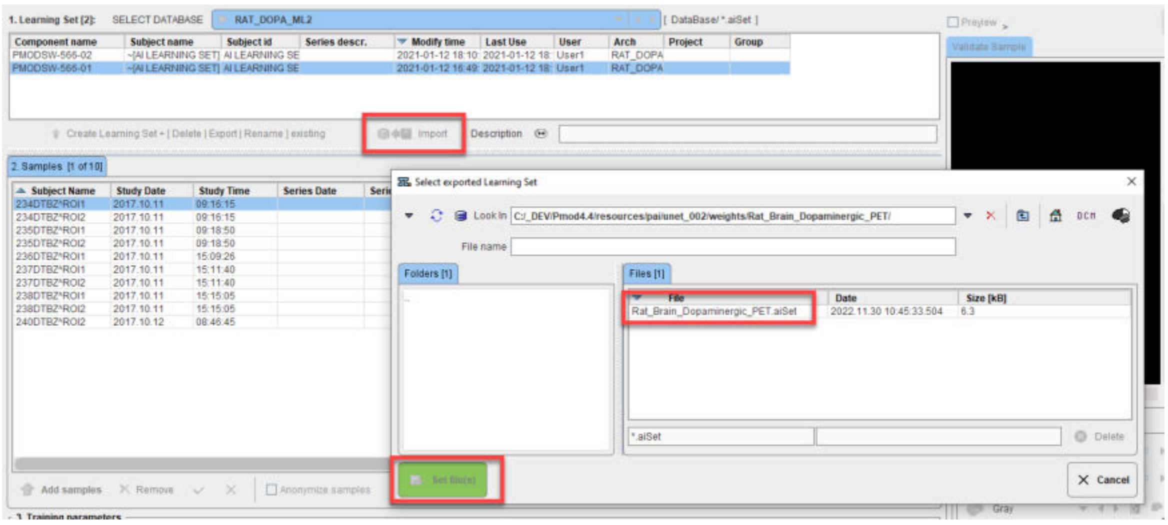
Edit description

Learning Set: PMODSW-566-01 <218/418/1018/*/RAT_DOPA_ML2>

Edit description:

This is a chance to keep track of your Learning Sets with detailed notes

Existing Models can also be imported using the Import Learning Set button. A database containing the same samples as used for training, with same database and sample naming, is required. The corresponding weights and manifest files will also be imported and the Model available via the database for Prediction. This mechanism can be used to import models that were trained on different machines (or cloud computing).



1. Learning Set [2]: SELECT DATABASE RAT_DOPA_ML2 [DataBase/ *.aiSet]

Component name	Subject name	Subject id	Series descr.	Modify time	Last Use	User	Arch	Project	Group
PMODSW-566-02	~[AI LEARNING SET]	AI LEARNING SE		2021-01-12 18:10	2021-01-12 18:	User1	RAT_DOPA		
PMODSW-566-01	~[AI LEARNING SET]	AI LEARNING SE		2021-01-12 16:49	2021-01-12 18:	User1	RAT_DOPA		

2. Samples [1 of 10]

Subject Name	Study Date	Study Time	Series Date
234DTBZ^ROI1	2017.10.11	09:16:15	
234DTBZ^ROI2	2017.10.11	09:16:15	
235DTBZ^ROI1	2017.10.11	09:18:50	
235DTBZ^ROI2	2017.10.11	09:18:50	
236DTBZ^ROI1	2017.10.11	15:09:26	
237DTBZ^ROI1	2017.10.11	15:11:40	
237DTBZ^ROI2	2017.10.11	15:11:40	
238DTBZ^ROI1	2017.10.11	15:15:05	
238DTBZ^ROI2	2017.10.11	15:15:05	
240DTBZ^ROI2	2017.10.12	08:46:45	

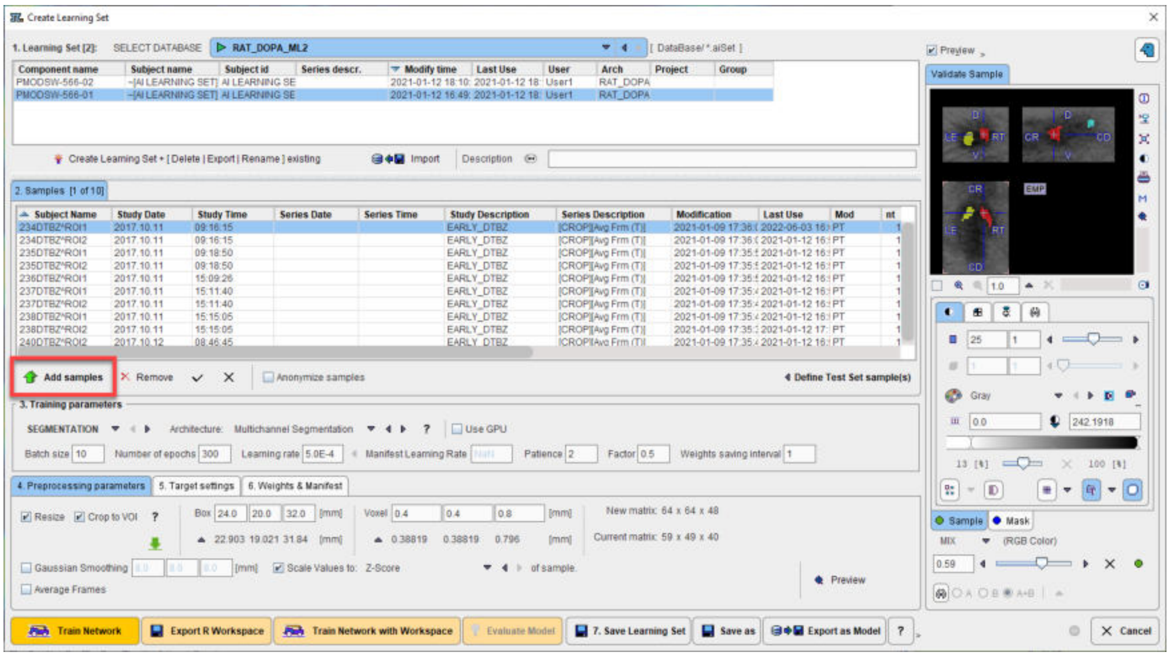
Select exported Learning Set

Look In: C:/_DEV/Pmod4.4/resources/pai/unet_002/weights/Rat_Brain_Dopaminergic_PET/

File	Date	Size [kB]
Rat_Brain_Dopaminergic_PET.aiSet	2022.11.30 10:45:33.504	6.3

Learning Set Content

The next step is to add training samples. First select the learning set in the **1. Learning Set** list, then select **Add Samples** at the bottom of **2. Samples**.



Create Learning Set

1. Learning Set [2]: SELECT DATABASE RAT_DOPA_ML2 [DataBase/ *.aiSet]

Component name	Subject name	Subject id	Series descr.	Modify time	Last Use	User	Arch	Project	Group
PMODSW-566-02	~[AI LEARNING SET]	AI LEARNING SE		2021-01-12 18:10	2021-01-12 18:	User1	RAT_DOPA		
PMODSW-566-01	~[AI LEARNING SET]	AI LEARNING SE		2021-01-12 16:49	2021-01-12 18:	User1	RAT_DOPA		

2. Samples [1 of 10]

Subject Name	Study Date	Study Time	Series Date	Series Time	Study Description	Series Description	Modification	Last Use	Mod	nt
234DTBZ^ROI1	2017.10.11	09:16:15			EARLY_DTBZ	[CROP][Avg Frm (T)]	2021-01-09 17:36:	2022-06-03 16:	PT	1
234DTBZ^ROI2	2017.10.11	09:16:15			EARLY_DTBZ	[CROP][Avg Frm (T)]	2021-01-09 17:36:	2021-01-12 16:	PT	1
235DTBZ^ROI1	2017.10.11	09:18:50			EARLY_DTBZ	[CROP][Avg Frm (T)]	2021-01-09 17:35:	2021-01-12 16:	PT	1
235DTBZ^ROI2	2017.10.11	09:18:50			EARLY_DTBZ	[CROP][Avg Frm (T)]	2021-01-09 17:35:	2021-01-12 16:	PT	1
236DTBZ^ROI1	2017.10.11	15:09:26			EARLY_DTBZ	[CROP][Avg Frm (T)]	2021-01-09 17:35:	2021-01-12 16:	PT	1
237DTBZ^ROI1	2017.10.11	15:11:40			EARLY_DTBZ	[CROP][Avg Frm (T)]	2021-01-09 17:35:	2021-01-12 16:	PT	1
237DTBZ^ROI2	2017.10.11	15:11:40			EARLY_DTBZ	[CROP][Avg Frm (T)]	2021-01-09 17:35:	2021-01-12 16:	PT	1
238DTBZ^ROI1	2017.10.11	15:15:05			EARLY_DTBZ	[CROP][Avg Frm (T)]	2021-01-09 17:35:	2021-01-12 16:	PT	1
238DTBZ^ROI2	2017.10.11	15:15:05			EARLY_DTBZ	[CROP][Avg Frm (T)]	2021-01-09 17:35:	2021-01-12 17:	PT	1
240DTBZ^ROI2	2017.10.12	08:46:45			EARLY_DTBZ	[CROP][Avg Frm (T)]	2021-01-09 17:35:	2021-01-12 16:	PT	1

3. Training parameters

SEGMENTATION Architecture: Multichannel Segmentation; Use GPU [ ]

Batch size 10; Number of epochs 300; Learning rate 5.0E-4; Manifest Learning Rate NaN; Patience 2; Factor 0.5; Weights saving interval 1

4. Preprocessing parameters

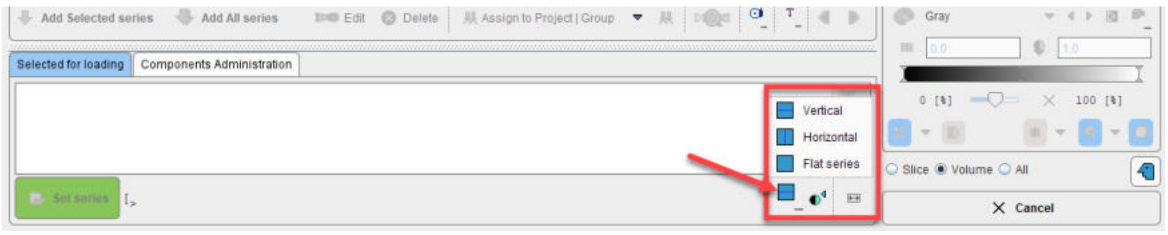
[✓] Resize [✓] Crop to VOI; Box 24.0 20.0 32.0 [mm]; Voxel 0.4 0.4 0.8 [mm]; New matrix: 64 x 64 x 48

22.903 19.021 31.84 [mm]; 0.38819 0.38819 0.796 [mm]; Current matrix: 59 x 49 x 40

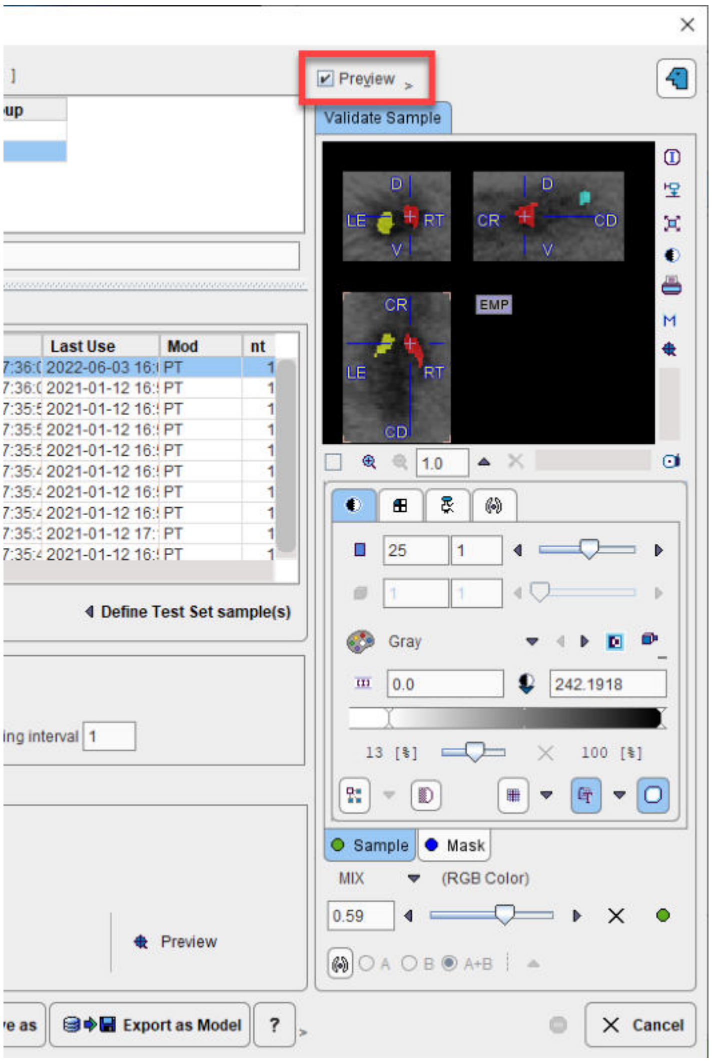
[ ] Gaussian Smoothing 8.0 8.0 8.0 [mm]; [✓] Scale Values to: Z-Score of sample.

[ ] Average Frames

A dialog window appears for selection of the input images. The flat database view

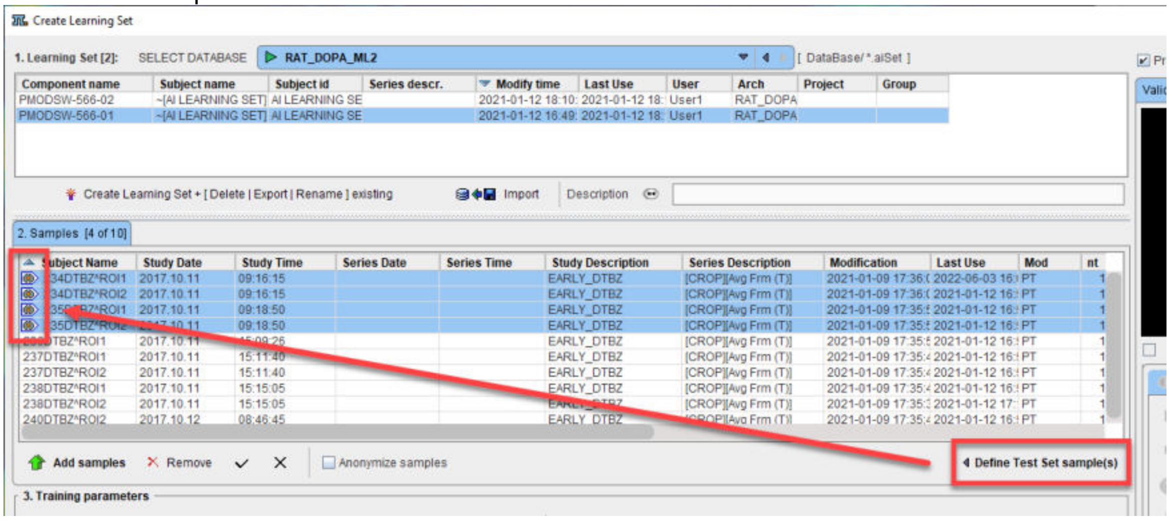


and advanced filtering options are useful to list the only the image series which are first in the **Associated** list. Select all appropriate series from the filtered list and **Set Selected**. The dialog window is closed and the samples added are listed in **2. Samples**. For a quick quality control, the fusion of the sample image and the corresponding reference segment can be shown in the **Validate Sample** area by activating the **Preview** box:



	Last Use	Mod	nt
7:36:(	2022-06-03 16:(	PT	1
7:36:(	2021-01-12 16::	PT	1
7:35:5	2021-01-12 16::	PT	1
7:35:5	2021-01-12 16::	PT	1
7:35:5	2021-01-12 16::	PT	1
7:35:4	2021-01-12 16::	PT	1
7:35:4	2021-01-12 16::	PT	1
7:35:4	2021-01-12 16::	PT	1
7:35:3	2021-01-12 17::	PT	1
7:35:4	2021-01-12 16::	PT	1

The Define Test Set Samples option is used as part of model evaluation after successful training. A number of samples may be excluded from training and "held back" for model evaluation. These samples can be selected in 2. Samples and marked as Test samples using the Define Test Set Samples button. Following training, an automated model evaluation process can be started using these samples using the Evaluate Model button. Prediction will be run in R Console and each predicted segment compared to the reference associated with the sample. An average loss value for the test set samples will be returned.



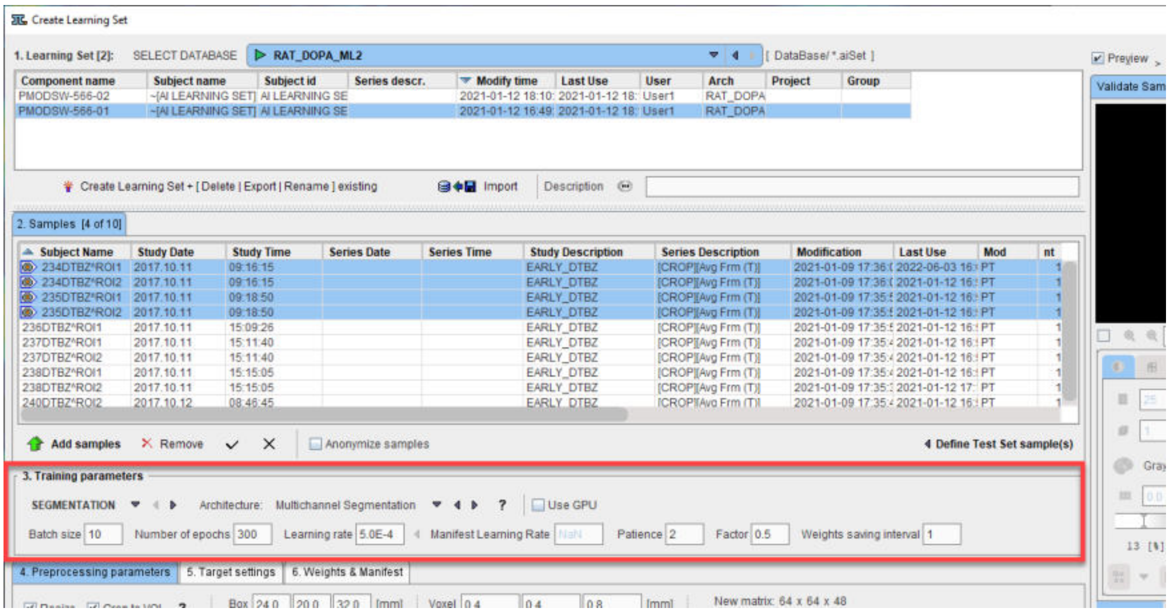
Component name	Subject name	Subject id	Series descr.	Modify time	Last Use	User	Arch	Project	Group
PMODSW-566-02	~[AI LEARNING SET]	AI LEARNING SE		2021-01-12 18:10:	2021-01-12 18:	User1	RAT_DOPA		
PMODSW-566-01	~[AI LEARNING SET]	AI LEARNING SE		2021-01-12 16:49:	2021-01-12 18:	User1	RAT_DOPA		

Subject Name	Study Date	Study Time	Series Date	Series Time	Study Description	Series Description	Modification	Last Use	Mod	nt
34DTBZ^ROI1	2017.10.11	09:16:15			EARLY_DTBZ	[CROP][Avg Frm (T)]	2021-01-09 17:36:(	2022-06-03 16::	PT	1
34DTBZ^ROI2	2017.10.11	09:16:15			EARLY_DTBZ	[CROP][Avg Frm (T)]	2021-01-09 17:36:(	2021-01-12 16::	PT	1
35DTBZ^ROI1	2017.10.11	09:18:50			EARLY_DTBZ	[CROP][Avg Frm (T)]	2021-01-09 17:35:5	2021-01-12 16::	PT	1
35DTBZ^ROI2	2017.10.11	09:18:50			EARLY_DTBZ	[CROP][Avg Frm (T)]	2021-01-09 17:35:5	2021-01-12 16::	PT	1
TBZ^ROI1	2017.10.11	15:09:26			EARLY_DTBZ	[CROP][Avg Frm (T)]	2021-01-09 17:35:5	2021-01-12 16::	PT	1
237DTBZ^ROI1	2017.10.11	15:11:40			EARLY_DTBZ	[CROP][Avg Frm (T)]	2021-01-09 17:35:4	2021-01-12 16::	PT	1
237DTBZ^ROI2	2017.10.11	15:11:40			EARLY_DTBZ	[CROP][Avg Frm (T)]	2021-01-09 17:35:4	2021-01-12 16::	PT	1
238DTBZ^ROI1	2017.10.11	15:15:05			EARLY_DTBZ	[CROP][Avg Frm (T)]	2021-01-09 17:35:4	2021-01-12 16::	PT	1
238DTBZ^ROI2	2017.10.11	15:15:05			EARLY_DTBZ	[CROP][Avg Frm (T)]	2021-01-09 17:35:3	2021-01-12 17::	PT	1
240DTBZ^ROI2	2017.10.12	08:46:45			EARLY_DTBZ	[CROP][Avg Frm (T)]	2021-01-09 17:35:4	2021-01-12 16::	PT	1

Samples may be Anonymized for training by checking the checkbox **Anonymize samples**. This is particularly relevant for training on cloud-computing infrastructure.

Architecture Selection & Settings

In **3. Training parameters**, architecture to be used and the training settings are configured.


Create Learning Set

1. Learning Set [2]: SELECT DATABASE RAT_DOPA_ML2 [DataBase/*.aiSet]

Component name	Subject name	Subject id	Series descr.	Modify time	Last Use	User	Arch	Project	Group
PMODSW-566-02	~[AI LEARNING SET]	AI LEARNING SE		2021-01-12 18:10:	2021-01-12 18:	User1	RAT_DOPA		
PMODSW-566-01	~[AI LEARNING SET]	AI LEARNING SE		2021-01-12 16:49:	2021-01-12 18:	User1	RAT_DOPA		

Create Learning Set + [Delete | Export | Rename] existing Import Description

2. Samples [4 of 10]

Subject Name	Study Date	Study Time	Series Date	Series Time	Study Description	Series Description	Modification	Last Use	Mod	nt
234DTBZ^ROI1	2017.10.11	09:16:15			EARLY_DTBZ	[CROP][Avg Frm (T)]	2021-01-09 17:36:(	2022-06-03 16:	PT	1
234DTBZ^ROI2	2017.10.11	09:16:15			EARLY_DTBZ	[CROP][Avg Frm (T)]	2021-01-09 17:36:(	2021-01-12 16:	PT	1
235DTBZ^ROI1	2017.10.11	09:18:50			EARLY_DTBZ	[CROP][Avg Frm (T)]	2021-01-09 17:35:!	2021-01-12 16:	PT	1
235DTBZ^ROI2	2017.10.11	09:18:50			EARLY_DTBZ	[CROP][Avg Frm (T)]	2021-01-09 17:35:!	2021-01-12 16:	PT	1
236DTBZ^ROI1	2017.10.11	15:09:26			EARLY_DTBZ	[CROP][Avg Frm (T)]	2021-01-09 17:35:!	2021-01-12 16:	PT	1
237DTBZ^ROI1	2017.10.11	15:11:40			EARLY_DTBZ	[CROP][Avg Frm (T)]	2021-01-09 17:35:4	2021-01-12 16:	PT	1
237DTBZ^ROI2	2017.10.11	15:11:40			EARLY_DTBZ	[CROP][Avg Frm (T)]	2021-01-09 17:35:4	2021-01-12 16:	PT	1
238DTBZ^ROI1	2017.10.11	15:15:05			EARLY_DTBZ	[CROP][Avg Frm (T)]	2021-01-09 17:35:4	2021-01-12 16:	PT	1
238DTBZ^ROI2	2017.10.11	15:15:05			EARLY_DTBZ	[CROP][Avg Frm (T)]	2021-01-09 17:35:3	2021-01-12 17:	PT	1
240DTBZ^ROI2	2017.10.12	08:46:45			EARLY_DTBZ	[CROP][Avg Frm (T)]	2021-01-09 17:35:4	2021-01-12 16:	PT	1

Add samples Remove Anonymize samples Define Test Set sample(s)

3. Training parameters

SEGMENTATION Architecture: Multichannel Segmentation ? Use GPU

Batch size 10 Number of epochs 300 Learning rate 5.0E-4 Manifest Learning Rate NaN Patience 2 Factor 0.5 Weights saving interval 1

4. Preprocessing parameters 5. Target settings 6. Weights & Manifest

Box 24.0 20.0 32.0 [mm] Voxel 0.4 0.4 0.8 [mm] New matrix: 64 x 64 x 48

The AI task (Segmentation or Classification) is selected on the left, followed by the neural network **Architecture.**

The neural network architecture is selected from the drop-down menu **Architecture**. The list corresponds to the the content of the Pmod4.4/resources/pai folder, where the neural network configurations are stored in sub-directories. See Architectures included in Distribution 9

Note that the architectures provided by PMOD are designed to be retrainable. For example, the Multichannel Segmentation architecture was initially tested for the 4 input series, 3 segment output MICCAI BraTS example case, and was successfully reapplied for the Rat Brain Dopaminergic PET example case. This retraining is described as a Case Study later in this documentation.

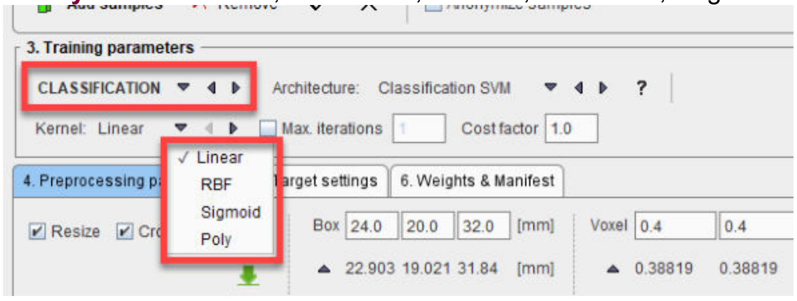
The checkbox **Use GPU** allows you to choose between training using the CPU or available/compatible GPU. Note that the Classification architecture SVM can only use CPU.

The training parameters for **SEGMENTATION** are:

- **Batch Size**: This parameter defines the number of samples that are processed before the internal model parameters are updated.
- **Number of Epochs**: Defines the number of times that the learning algorithms processes the entire training data sets. The length of the vector of loss values recorded in the **Manifest** corresponds to the number of epochs. Hence multiple epochs are required to observe an evolution of the loss value through training. During training a plot of the loss value (y-axis) by epoch (x-axis) is displayed, allowing the user to gauge the progress of training (and stop training if a plateau in loss value is observed).
- **Learning Rate**: Defines the rate of change of the **Weights**. (For a Learning Set that has been used for training the final learning rate reported in the **Manifest** is shown)
- **Manifest Learning Rate**: if the model is being retrained and there is an existing manifest, the Learning Rate recorded in that manifest is displayed here (the field is not editable)
- **Patience**: Defines how frequently (after which number of epochs) the validation loss is compared (and hence learning rate altered)
- **Factor**: Defines the factor by which the Learning Rate will be adjusted when the validation loss is found to increased from one epoch to the next (or for interval of epochs defined by Patience)
- **Weights saving interval**: Defines how frequently the weights file will be saved (e.g. Weights saving interval = 1, save after every epoch)

The training parameters for **CLASSIFICATION** vary by SVM **kernel** selected (see https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html for additional details):

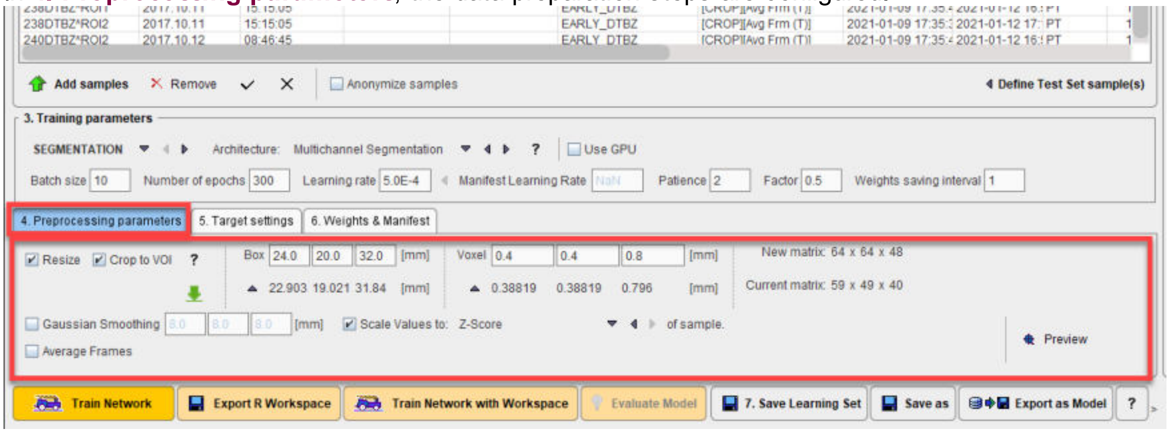
- **Linear**: Max. iterations, Cost factor
- **RBF**: Max. iterations, Cost factor, Gamma (Auto, Scale, User defined)
- **Sigmoid**: Max. iterations, Cost factor, Gamma, Coefficient
- **Poly**: Max. iterations, Cost factor, Gamma, Coefficient, Degree



Additional architectures available for Classification are ResNet50 and Image Classification. Contact us for more information.

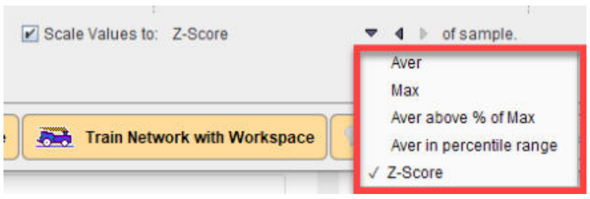
Data Preparation

In **4. Preprocessing parameters**, the data preparation steps are configured.

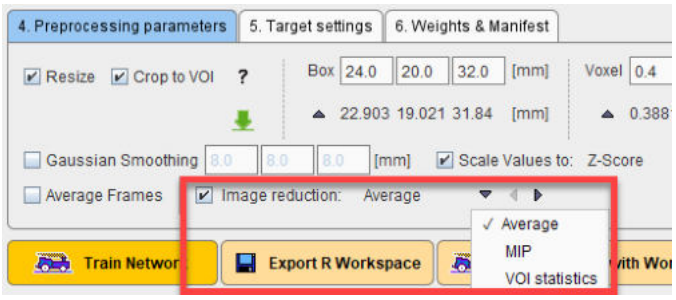


The available data preparation steps are:

- **Resize**: input samples may not all have the same image dimensions, pixel size or field-of-view. Standardization is beneficial to training the neural network. This can be achieved through a combination of cropping and interpolation. **Crop to VOI**: Enables cropping to the associated VOI as described in [Data Preparation]()41 or to a fixed **Box** and **Voxel** size, defining the **New matrix**. The icon retrieves the current box and voxel size from the sample selected in **2. Samples** to make a comparison between the **New matrix** and **Current matrix** easier.
- **Gaussian Smoothing**: Input image smoothing to reduce noise.
- **Scale Values to**: Normalization of the pixel value range by scaling according to a method selected:



- **Average Frames**: Reduce the dimensions of the input data by averaging the available frames.
- For **Classification** additional options for **Image Reduction** are available. The tool tips provide additional information about the methods available:



4. Preprocessing parameters | 5. Target settings | 6. Weights & Manifest

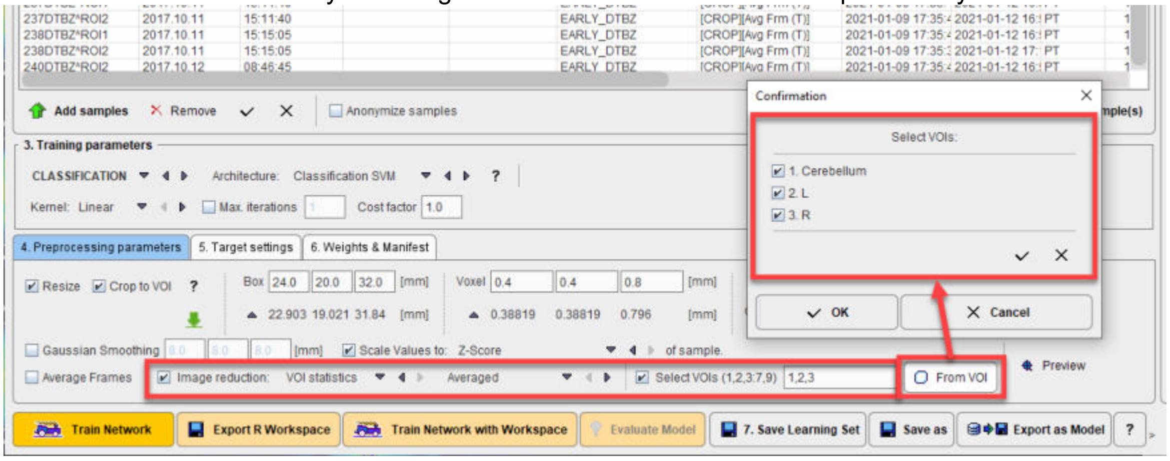
✓ Resize ✓ Crop to VOI ? Box 24.0 20.0 32.0 [mm] Voxel 0.4

22.903 19.021 31.84 [mm] 0.388

Gaussian Smoothing 8.0 8.0 8.0 [mm] ✓ Scale Values to: Z-Score

Average Frames ✓ Image reduction: Average

- **Average**: average of available slices in z direction, creating a 2D image from 3D input
- **MIP**: calculates the maximum intensity projection in z direction, creating a 2D image
- **VOI Statistics**: reduces the sample to a vector by calculating a chosen statistic in associated VOI(s). A subset of the VOIs available in an associated VOI file can be defined using the **Select VOIs** checkbox. The VOIs to be selected can either be defined by manually entering the VOI numbers to be used or by checking the desired VOIs in the interface provided by **From VOIs**.



237DTBZ^ROI2	2017.10.11	15:11:40	EARLY_DTBZ	[CROP][Avg Frm (T)]	2021-01-09 17:35	2021-01-12 16	PT
238DTBZ^ROI1	2017.10.11	15:15:05	EARLY_DTBZ	[CROP][Avg Frm (T)]	2021-01-09 17:35	2021-01-12 16	PT
238DTBZ^ROI2	2017.10.11	15:15:05	EARLY_DTBZ	[CROP][Avg Frm (T)]	2021-01-09 17:35	2021-01-12 17	PT
240DTBZ^ROI2	2017.10.12	08:46:45	EARLY_DTBZ	[CROP][Avg Frm (T)]	2021-01-09 17:35	2021-01-12 16	PT

3. Training parameters

CLASSIFICATION Architecture: Classification SVM

Kernel: Linear Max. iterations Cost factor 1.0

✓ Resize ✓ Crop to VOI Box 24.0 20.0 32.0 [mm] Voxel 0.4 0.4 0.8 [mm]

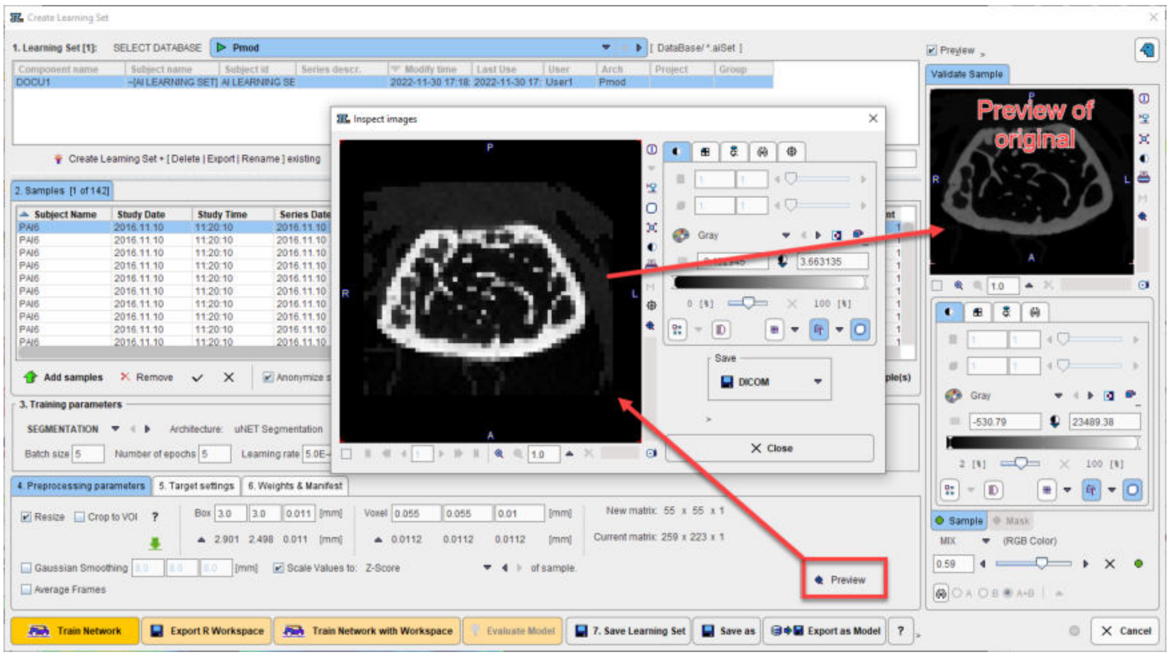
22.903 19.021 31.84 [mm] 0.38819 0.38819 0.796 [mm]

Gaussian Smoothing 8.0 8.0 8.0 [mm] ✓ Scale Values to: Z-Score of sample.

Average Frames ✓ Image reduction: VOI statistics Averaged ✓ Select VOIs (1,2,3,7,9) 1,2,3

Confirmation — Select VOIs: ✓ 1. Cerebellum ✓ 2. L ✓ 3. R

The effect of the selected Preprocessing parameters can be examined by generating a Preview of the sample after preprocessing using the Preview button:



1. Learning Set [1]: SELECT DATABASE Pmod [DataBase/*.aiSet]

Component name	Subject name	Subject id	Series descr.	Modify time	Last Use	User	Arch	Project	Group
DOCU1		-[AI LEARNING SET] AI LEARNING SE		2022-11-30 17:18	2022-11-30 17:	User1	Pmod		

2. Samples [1 of 142]

Subject Name	Study Date	Study Time	Series Date
PAI6	2016.11.10	11:20:10	2016.11.10
PAI6	2016.11.10	11:20:10	2016.11.10
PAI6	2016.11.10	11:20:10	2016.11.10
PAI6	2016.11.10	11:20:10	2016.11.10
PAI6	2016.11.10	11:20:10	2016.11.10
PAI6	2016.11.10	11:20:10	2016.11.10
PAI6	2016.11.10	11:20:10	2016.11.10
PAI6	2016.11.10	11:20:10	2016.11.10
PAI6	2016.11.10	11:20:10	2016.11.10
PAI6	2016.11.10	11:20:10	2016.11.10

3. Training parameters

SEGMENTATION Architecture: uNET Segmentation

Batch size 5 Number of epochs 5 Learning rate 5.0E-

✓ Resize Crop to VOI Box 3.0 3.0 0.011 [mm] Voxel 0.055 0.055 0.01 [mm] New matrix: 55 x 55 x 1

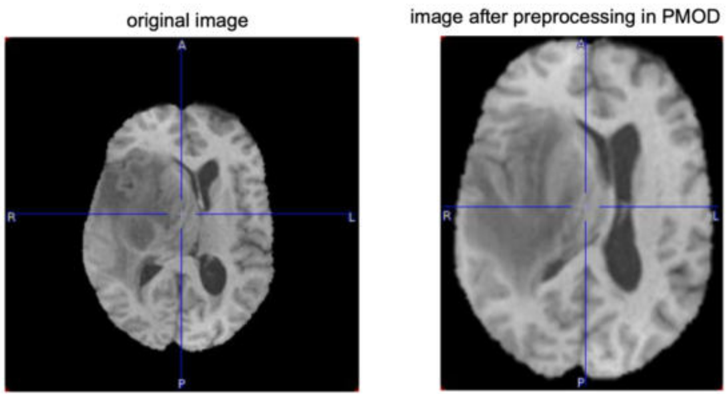
2.901 2.498 0.011 [mm] 0.0112 0.0112 0.0112 [mm] Current matrix: 259 x 223 x 1

Gaussian Smoothing 8.0 8.0 8.0 [mm] ✓ Scale Values to: Z-Score of sample.

Average Frames

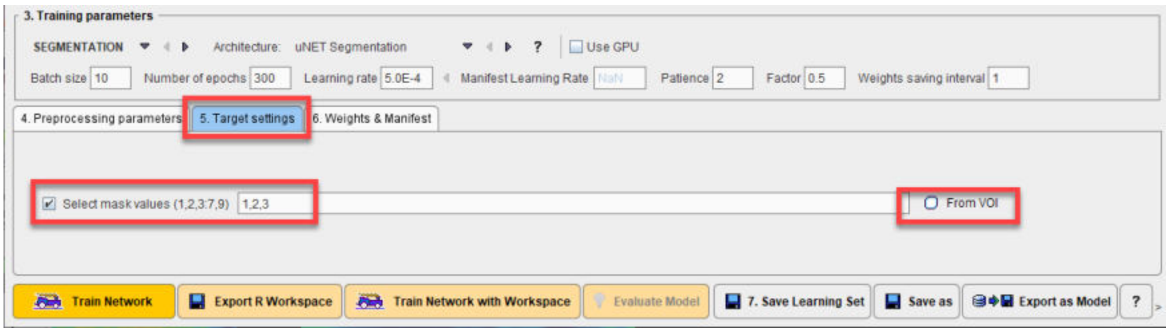
Note that as an alternative to such pre-processing steps, the input images could be (manually) pre-processed in other PMOD tools. In this case the user must ensure that identical operations are applied to the input images before prediction.

Data preparation helps to reduces the amount of unnecessary information in the sample and standardize images which were not acquired using the same protocol.



Definition of Target Settings

The target settings, defined on **5. Target Settings**, that are required depend on whether a segmentation or classification task has been defined. For segmentation tasks a list of the integer labeled segments or VOI numbers to be used must be provided if more than one segment is present in the reference. The reference segment image may contain more segments than actually required. The option **Select mask values** allows the integer value of the required segments to be specified by entering the label values of the target segments, separated by a “comma”. The **From VOI** interface may be used where VOIs are available instead of or as well as segments.


3. Training parameters

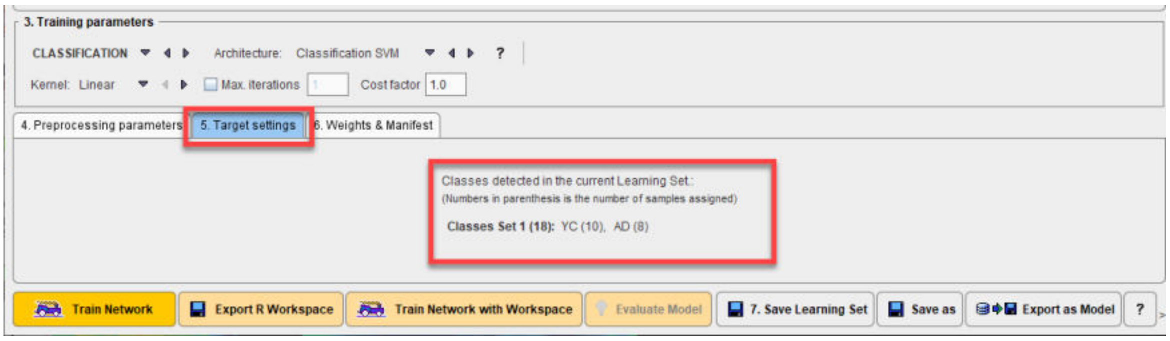
SEGMENTATION Architecture: uNET Segmentation ? Use GPU

Batch size 10 Number of epochs 300 Learning rate 5.0E-4 Manifest Learning Rate NaN Patience 2 Factor 0.5 Weights saving interval 1

4. Preprocessing parameters | 5. Target settings | 6. Weights & Manifest

Select mask values (1,2,3:7,9) 1,2,3 From VOI

For classification tasks the classes defined in the samples are reported. If the list of classes is incorrect the Project label per sample should be edited using the database interface.


3. Training parameters

CLASSIFICATION Architecture: Classification SVM ?

Kernel: Linear Max. iterations 1 Cost factor 1.0

4. Preprocessing parameters | 5. Target settings | 6. Weights & Manifest

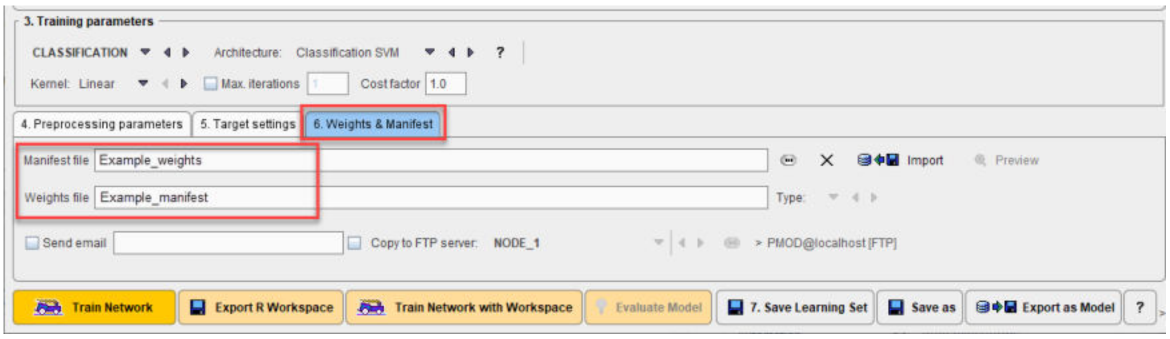
Classes detected in the current Learning Set:
(Numbers in parenthesis is the number of samples assigned)

Classes Set 1 (18): YC (10), AD (8)

Saving of Training Result

Two files result from a neural network training, **Weights** and **Manifest**. The **Weights** file contains the weighting given to each layer in the network. The **Manifest** file contains details of the **Learning Set** and the training process such as the samples used for training, samples used for validation (every fifth sample), number of epochs used, batch size, volume size, and the segments in the output. Weights and Manifest are inherently linked, and references to the Weights are included in the Manifest. Multiple Weights files may be created depending on the model architecture (e.g. Generative Adversarial Models, GAN).

The file locations are defined on the 6. **Weights & Manifest** panel, and the files will be logically attached to the current learning set. In case of Additive Training, these defined files provide the starting point for further training of the neural network with new samples.


3. Training parameters

CLASSIFICATION

Architecture: Classification SVM

Kernel: Linear

Max. iterations 1

Cost factor 1.0

Manifest file Example_weights

Weights file Example_manifest

Type:

Send email

Copy to FTP server: NODE_1 > PMOD@localhost [FTP]

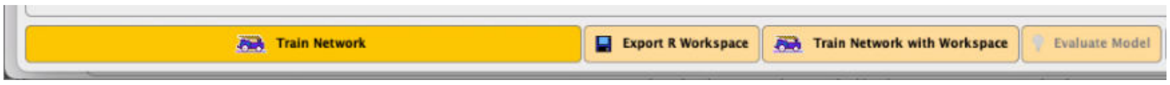
Before starting training we recommend saving the learning set in its current state using **7. Save Learning Set**. **Save As** may be used when editing an existing learning set (e.g. changing pre-processing parameters) and wishing to save it as a new copy in the database.

The **Send email** option is provided in case of training on cloud computing infrastructure. When training has been completed an email will be sent to the address provided (we suggest testing this functionality with a limited training run to ensure that your institutional firewall settings allow receipt of the email). The weights resulting from training (on cloud computing) may also be copied to an FTP server according to the settings provided.

Export as Model is used to save the required information to use a trained neural network for Prediction 64 in the PMOD installation *resources/pai* folder. The resulting model in the architecture/weights folder can be copied to other PMOD installations. **Import Model** as described above is particularly useful for this purpose. Trained models can also be used for Prediction in the same PMOD installation without **Export as Model**, and will be available as Learning Sets in the database.

4 *Training of Neural Network*

Training of the neural network can be performed in three different ways represented by the three buttons at the bottom of the dialog window:



1. The **Train Network** button directly starts the configured training locally. Depending on the checkbox **Use GPU**, either your CPU or GPU will be used. Note that only the samples explicitly selected (left-click, ctrl+click, shift+click) in the **2. Samples** list will be used for the training, according to the settings in **3. Training Parameters** and **4. Preprocessing parameters**. As a result, the weights and manifest files will be updated.
2. Using **Export R workspace** 62 **,** the configured preprocessing operations are applied to the selected data and the resulting images are exported together with the training configuration in the form of a compact R workspace. The workspace can then be transferred to a more powerful processing environment for the actual training. This can either be another PMOD installation on a more powerful machine or in the cloud.
3. The **Train Network with Workspace** button opens a dialog window in order to load a previously exported R workspace and starts the training locally.

The process is described in detail below.

Deployment

After completion of the training, the resulting **Weights** and **Manifest** files are combined with the Learning Set definition (*.aiset file) to form the trained model. The trained model can be used in the same PMOD installation used for training by selecting the Learning Set from the database where AI functionality is available. The trained model can also be transferred to other PMOD installations for prediction. This is most easily achieved by using Export as Model and copying the resulting files from the PMOD installation folder *resources/pai/architecture/weights* (e.g. *C:\Pmod4.4\resources\pai\unet_002\weights\IXI_Parcellation*\) to another PMOD installation.

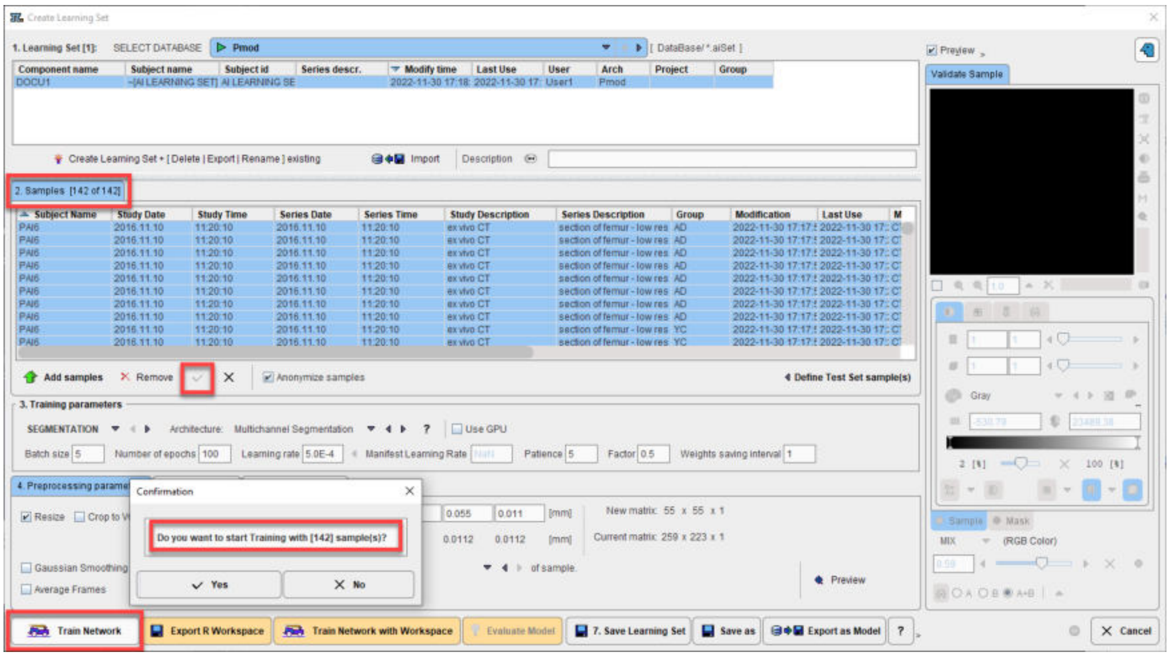
Recommendations

On typical personal computers local training is only recommended for tests with a limited amount of data. Performance may be acceptable with data that has a small matrix size (e.g. 50 x 50 x 50 for cropped PET data) and low number of input series for multichannel segmentation (e.g. 1 or 2). The total time required for training cannot be estimated. While training is running you will see a significant load on CPU/GPU and a plot of loss value by epoch in PMOD's R Console. Even for powerful workstations, training with hundreds of samples may take many hours. Training on a cloud computing infrastructure with virtual machines accessing several GPUs is likely to be more time- and cost-efficient.

It is advisable to always perform a small "infrastructure check" training before launching training with your full data set and many epochs. This can be performed using the minimum requirement for input samples (2 samples), a batch size of 1 and a low number of epochs (1 is acceptable, but 2 or 3 will reveal changes in the loss value in the Manifest). If the input data has a high matrix size (e.g. 200 x 200 x 200) and/or there are multiple input series in the sample, the data volume could be reduced by using a larger pixel size for this training test (e.g. 2 x 2 x 2 mm instead of planned 1 x 1 x 1 mm).

Training Progress and Output

For data prepared on your local system, training is started by selecting the desired samples in the **Learning Set** and clicking **Train Network**:



1. Learning Set [1]: SELECT DATABASE Pmod [DataBase/ *.aiSet]

Component name	Subject name	Subject id	Series descr.	Modify time	Last Use	User	Arch	Project	Group
DOCU1	-[AI LEARNING SET]	AI LEARNING SE		2022-11-30 17:18:	2022-11-30 17:	User1	Pmod		

Create Learning Set + [Delete | Export | Rename] existing · Import · Description

2. Samples [142 of 142]

Subject Name	Study Date	Study Time	Series Date	Series Time	Study Description	Series Description	Group	Modification	Last Use	M
PAI6	2016.11.10	11:20:10	2016.11.10	11:20:10	ex vivo CT	section of femur - low res	AD	2022-11-30 17:17:!	2022-11-30 17::	C'
PAI6	2016.11.10	11:20:10	2016.11.10	11:20:10	ex vivo CT	section of femur - low res	AD	2022-11-30 17:17:!	2022-11-30 17::	C'
PAI6	2016.11.10	11:20:10	2016.11.10	11:20:10	ex vivo CT	section of femur - low res	AD	2022-11-30 17:17:!	2022-11-30 17::	C'
PAI6	2016.11.10	11:20:10	2016.11.10	11:20:10	ex vivo CT	section of femur - low res	AD	2022-11-30 17:17:!	2022-11-30 17::	C'
PAI6	2016.11.10	11:20:10	2016.11.10	11:20:10	ex vivo CT	section of femur - low res	AD	2022-11-30 17:17:!	2022-11-30 17::	C'
PAI6	2016.11.10	11:20:10	2016.11.10	11:20:10	ex vivo CT	section of femur - low res	AD	2022-11-30 17:17:!	2022-11-30 17::	C'
PAI6	2016.11.10	11:20:10	2016.11.10	11:20:10	ex vivo CT	section of femur - low res	AD	2022-11-30 17:17:!	2022-11-30 17::	C'
PAI6	2016.11.10	11:20:10	2016.11.10	11:20:10	ex vivo CT	section of femur - low res	AD	2022-11-30 17:17:!	2022-11-30 17::	C'
PAI6	2016.11.10	11:20:10	2016.11.10	11:20:10	ex vivo CT	section of femur - low res	YC	2022-11-30 17:17:!	2022-11-30 17::	C'
PAI6	2016.11.10	11:20:10	2016.11.10	11:20:10	ex vivo CT	section of femur - low res	YC	2022-11-30 17:17:!	2022-11-30 17::	C'

Add samples · Remove · Anonymize samples · Define Test Set sample(s)

3. Training parameters

SEGMENTATION · Architecture: Multichannel Segmentation · Use GPU

Batch size 5 · Number of epochs 100 · Learning rate 5.0E-4 · Manifest Learning Rate NaN · Patience 5 · Factor 0.5 · Weights saving interval 1

4. Preprocessing parame...

Resize · Crop to V... · 0.055 0.011 [mm] · New matrix: 55 x 55 x 1 · 0.0112 0.0112 [mm] · Current matrix: 259 x 223 x 1

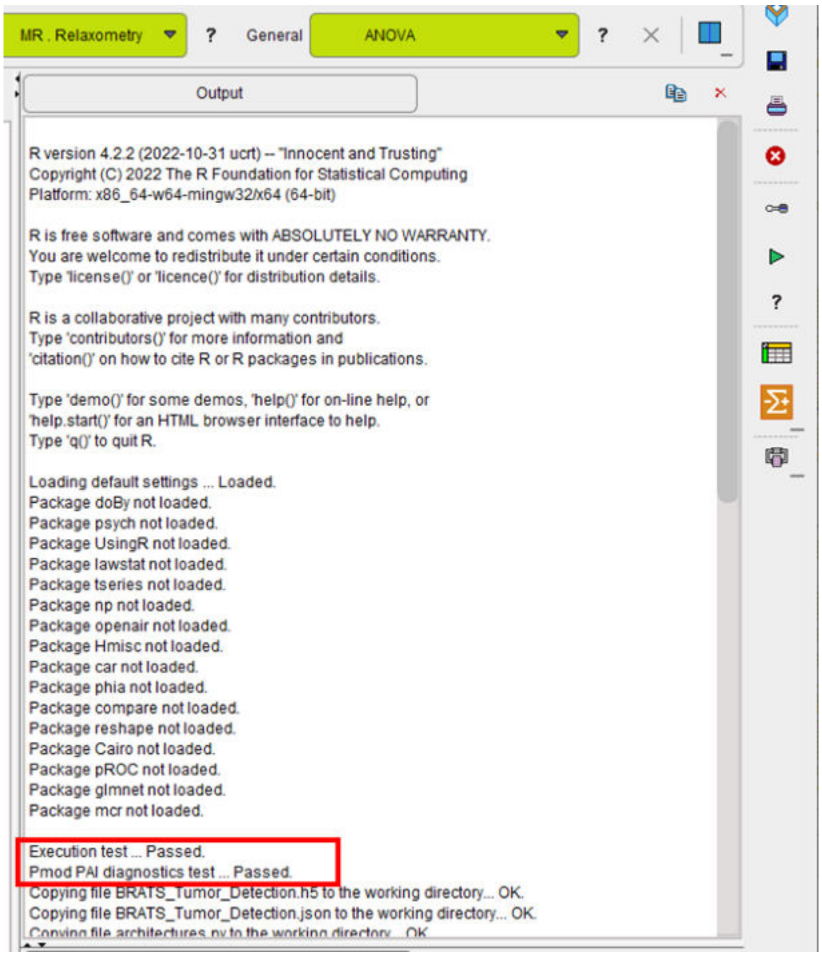
Gaussian Smoothing · Average Frames · of sample · Preview

Confirmation

Do you want to start Training with [142] sample(s)?

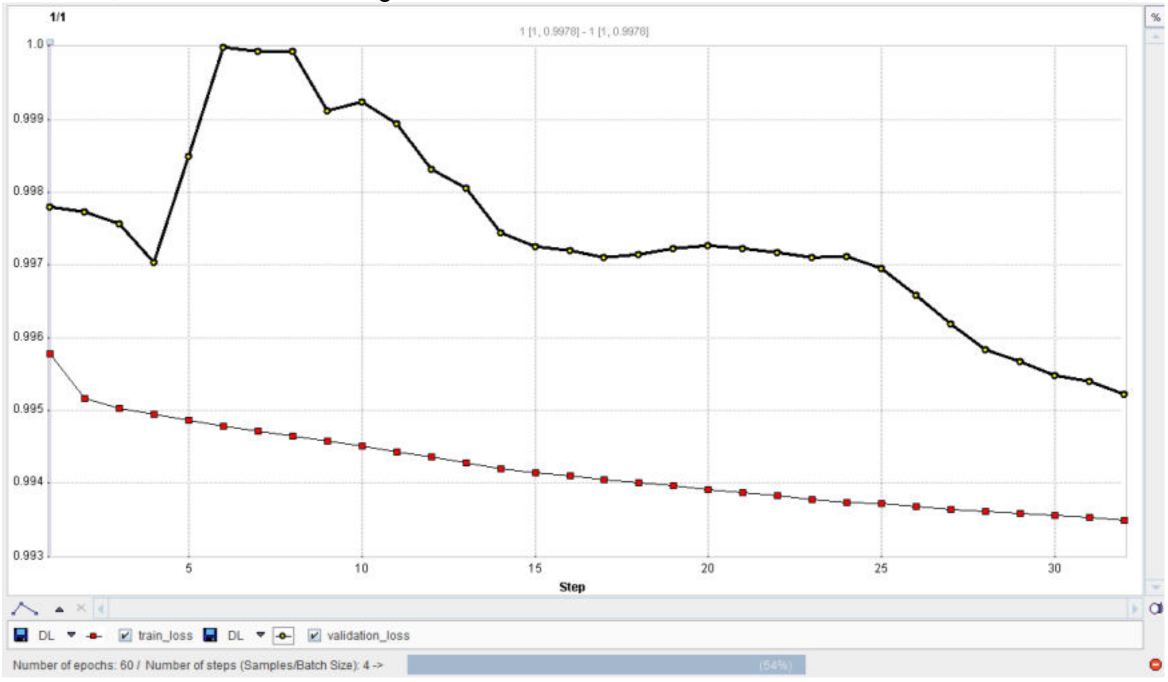
Yes · No

The **RConsole** opens and the Execution test and PAI diagnostics test are performed. If the tests are passed the selected samples (all input series and associated Segments) are loaded:



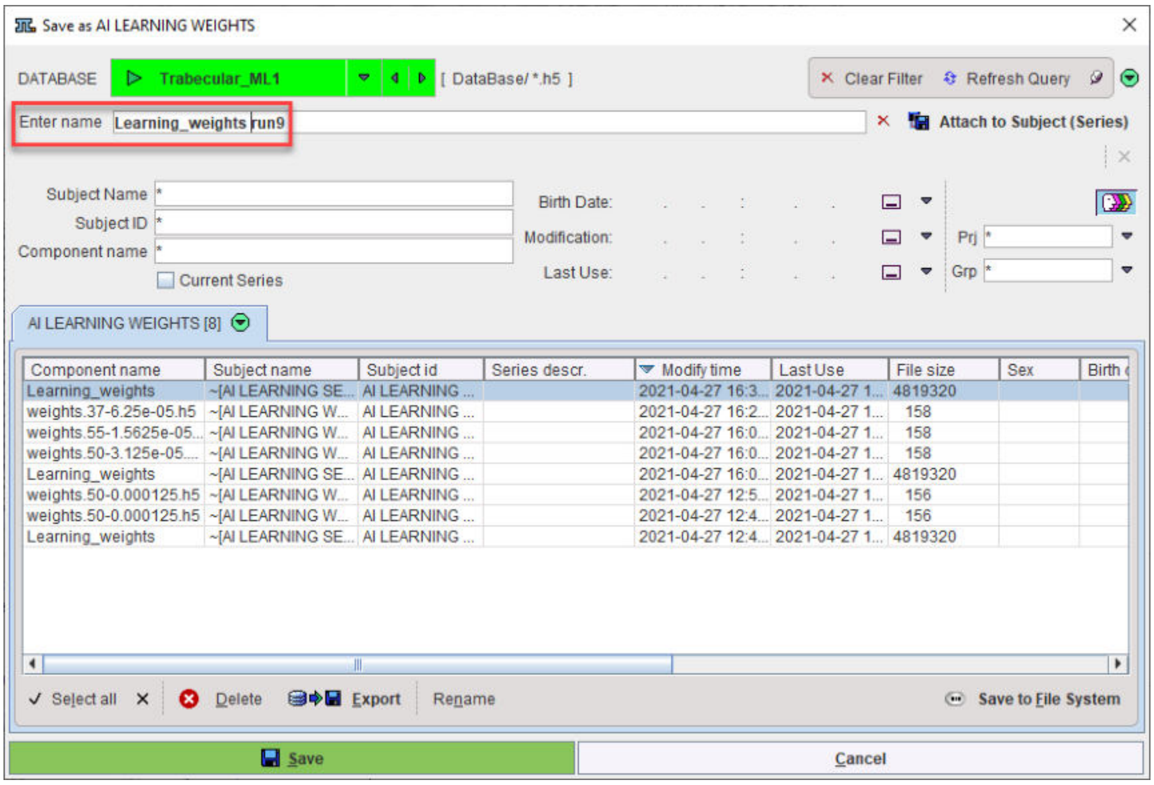
```
R version 4.2.2 (2022-10-31 ucrt) -- "Innocent and Trusting"
Copyright (C) 2022 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

Loading default settings ... Loaded.
Package doBy not loaded.
Package psych not loaded.
Package UsingR not loaded.
Package lawstat not loaded.
Package tseries not loaded.
Package np not loaded.
Package openair not loaded.
Package Hmisc not loaded.
Package car not loaded.
Package phia not loaded.
Package compare not loaded.
Package reshape not loaded.
Package Cairo not loaded.
Package pROC not loaded.
Package glmnet not loaded.
Package mcr not loaded.

Execution test ... Passed.
Pmod PAI diagnostics test ... Passed.
Copying file BRATS_Tumor_Detection.h5 to the working directory... OK.
Copying file BRATS_Tumor_Detection.json to the working directory... OK.
Copying file architectures.py to the working directory... OK.
```

During training the loss value is plotted with each epoch. Note that the loss value depends on the architecture selected and large changes on the y-axis are possible. In the example shown the loss value is 1 - Dice Coefficient for training of the unet_002 Multichannel Segmentation architecture with rat brain dopaminergic PET data as described in our Case Study. At the start of training the loss value is 1, and successful training results in a reduction of the loss value towards zero:



Number of epochs: 60 / Number of steps (Samples/Batch Size): 4 ->

Once training is complete a dialog appears to save the **Weights**. They can be saved to the database or file system (the same database as the **Learning Set** is recommended):



Save as AI LEARNING WEIGHTS

DATABASE Trabecular_ML1 [DataBase/ *.h5] Clear Filter Refresh Query

Enter name Learning_weights run9 Attach to Subject (Series)

Subject Name *
Subject ID *
Component name *
Current Series

Birth Date:
Modification:
Last Use:
Prj *
Grp *

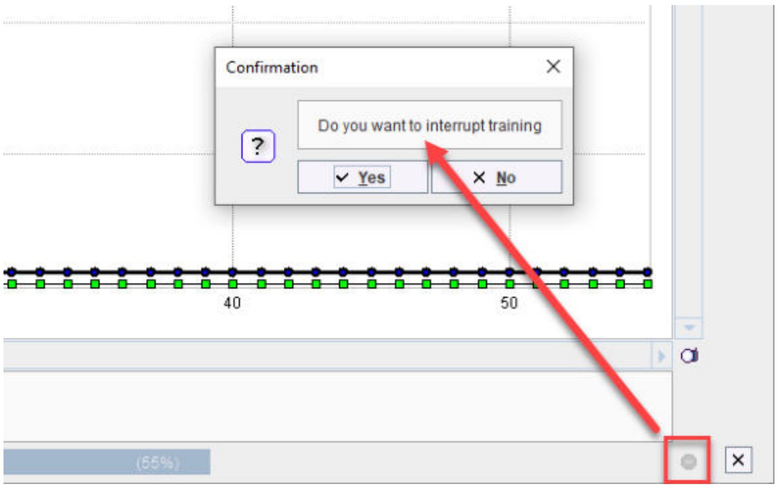
AI LEARNING WEIGHTS [8]

Component name	Subject name	Subject id	Series descr.	Modify time	Last Use	File size	Sex	Birth
Learning_weights	~[AI LEARNING SE...	AI LEARNING ...		2021-04-27 16:3...	2021-04-27 1...	4819320		
weights.37-6.25e-05.h5	~[AI LEARNING W...	AI LEARNING ...		2021-04-27 16:2...	2021-04-27 1...	158		
weights.55-1.5625e-05...	~[AI LEARNING W...	AI LEARNING ...		2021-04-27 16:0...	2021-04-27 1...	158		
weights.50-3.125e-05....	~[AI LEARNING W...	AI LEARNING ...		2021-04-27 16:0...	2021-04-27 1...	158		
Learning_weights	~[AI LEARNING SE...	AI LEARNING ...		2021-04-27 16:0...	2021-04-27 1...	4819320		
weights.50-0.000125.h5	~[AI LEARNING W...	AI LEARNING ...		2021-04-27 12:5...	2021-04-27 1...	156		
weights.50-0.000125.h5	~[AI LEARNING W...	AI LEARNING ...		2021-04-27 12:4...	2021-04-27 1...	156		
Learning_weights	~[AI LEARNING SE...	AI LEARNING ...		2021-04-27 12:4...	2021-04-27 1...	4819320		

Select all | Delete | Export | Rename | Save to File System

Save | Cancel

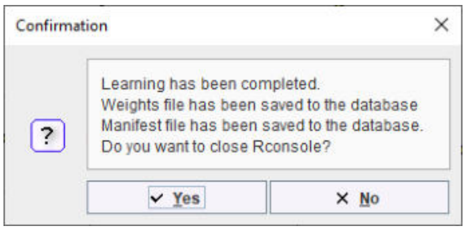
Training may also be stopped by the user. This is useful when training appears to be unsuccessful with no improvement in the loss value or when training has been particularly successful and a plateau in loss value has been achieved with many epochs remaining:



Confirmation

Do you want to interrupt training

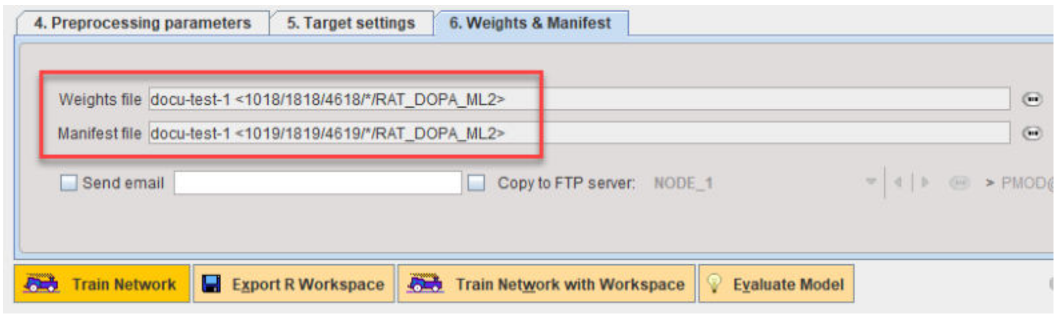
A confirmation dialog in the R Console confirms that learning was completed and the components saved. The R Console can be closed:



Confirmation

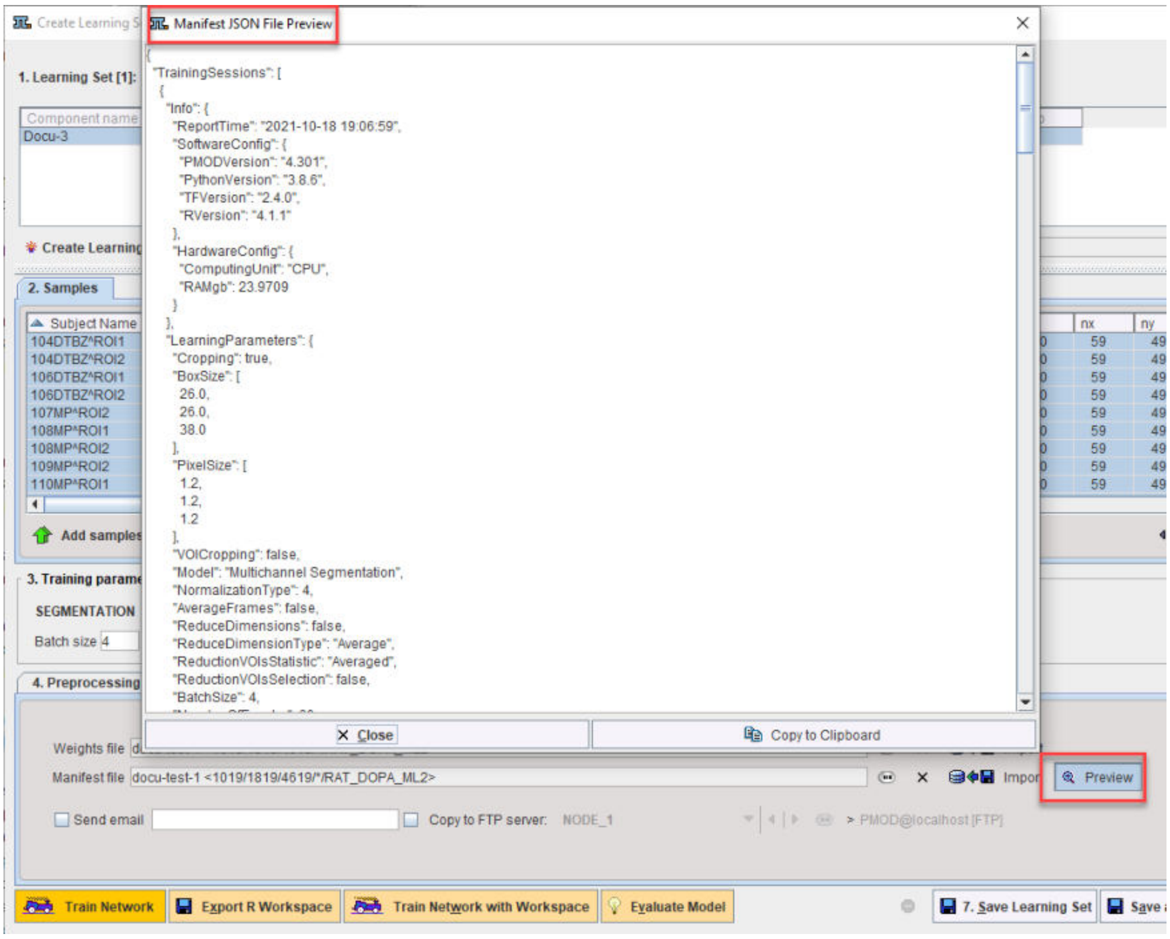
Learning has been completed.
Weights file has been saved to the database
Manifest file has been saved to the database.
Do you want to close Rconsole?

The **Weights** and **Manifest** are now attached to the **Learning Set**:



Weights file docu-test-1 <1018/1818/4618/*/RAT_DOPA_ML2>
Manifest file docu-test-1 <1019/1819/4619/*/RAT_DOPA_ML2>
Send email
Copy to FTP server: NODE_1

Details of the training are recorded in the Manifest. It can be reviewed using the **Preview** function on **6. Weights & Manifest**:



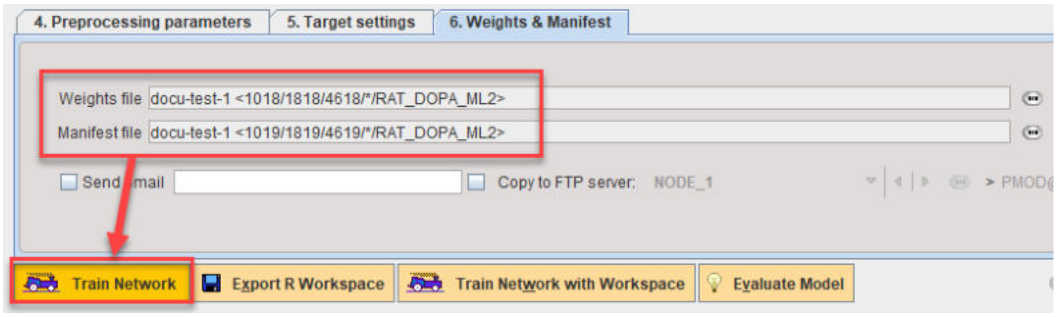
```
{
 "TrainingSessions": [
  {
   "Info": {
    "ReportTime": "2021-10-18 19:06:59",
    "SoftwareConfig": {
     "PMODVersion": "4.301",
     "PythonVersion": "3.8.6",
     "TFVersion": "2.4.0",
     "RVersion": "4.1.1"
    },
    "HardwareConfig": {
     "ComputingUnit": "CPU",
     "RAMgb": 23.9709
    }
   },
   "LearningParameters": {
    "Cropping": true,
    "BoxSize": [
     26.0,
     26.0,
     38.0
    ],
    "PixelSize": [
     1.2,
     1.2,
     1.2
    ],
    "VOICropping": false,
    "Model": "Multichannel Segmentation",
    "NormalizationType": 4,
    "AverageFrames": false,
    "ReduceDimensions": false,
    "ReduceDimensionType": "Average",
    "ReductionVOIsStatistic": "Averaged",
    "ReductionVOIsSelection": false,
    "BatchSize": 4,
```

Manifest file docu-test-1 <1019/1819/4619/*/RAT_DOPA_ML2>

Additive Training

The best results are achieved by training in a single session with the maximum amount of data available. However, in a situation where the initial number of samples is limited and new samples will become available on a regular basis it is possible to try additive training. For example, where 50 samples are available at the start of a project and 10 new samples will be preprocessed every two weeks.

Additive training is achieved by adding the new samples to your existing Learning Set, selecting a subset of the total Learning Set (a combination of existing samples and new samples is recommended, e.g. select the 10 new samples and 10 existing samples), then launching **Train Network** based on the existing **Weights** and **Manifest** (identified on **5. Weights & Manifest**).

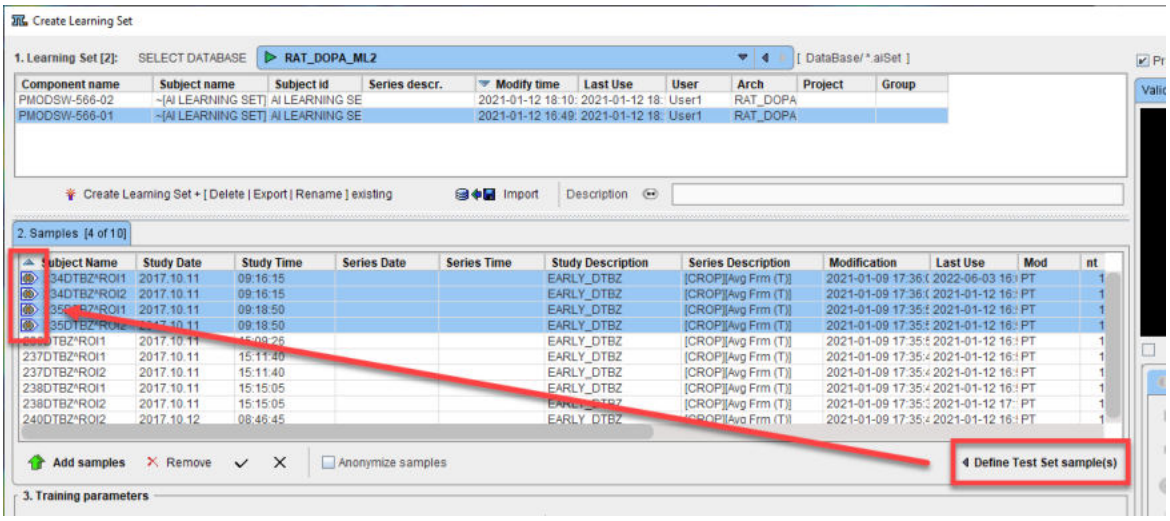


Weights file docu-test-1 <1018/1818/4618/*/RAT_DOPA_ML2>

Manifest file docu-test-1 <1019/1819/4619/*/RAT_DOPA_ML2>

Evaluation of the Model

Once the model has been trained, a subset of samples in 2. Samples can be designated as Test Set samples and used as an additional evaluation of the model. Samples can be labeled as Test Set prior to training so that they are excluded for this use in evaluation.


Create Learning Set

1. Learning Set [2]: SELECT DATABASE RAT_DOPA_ML2 [DataBase/*.aiSet]

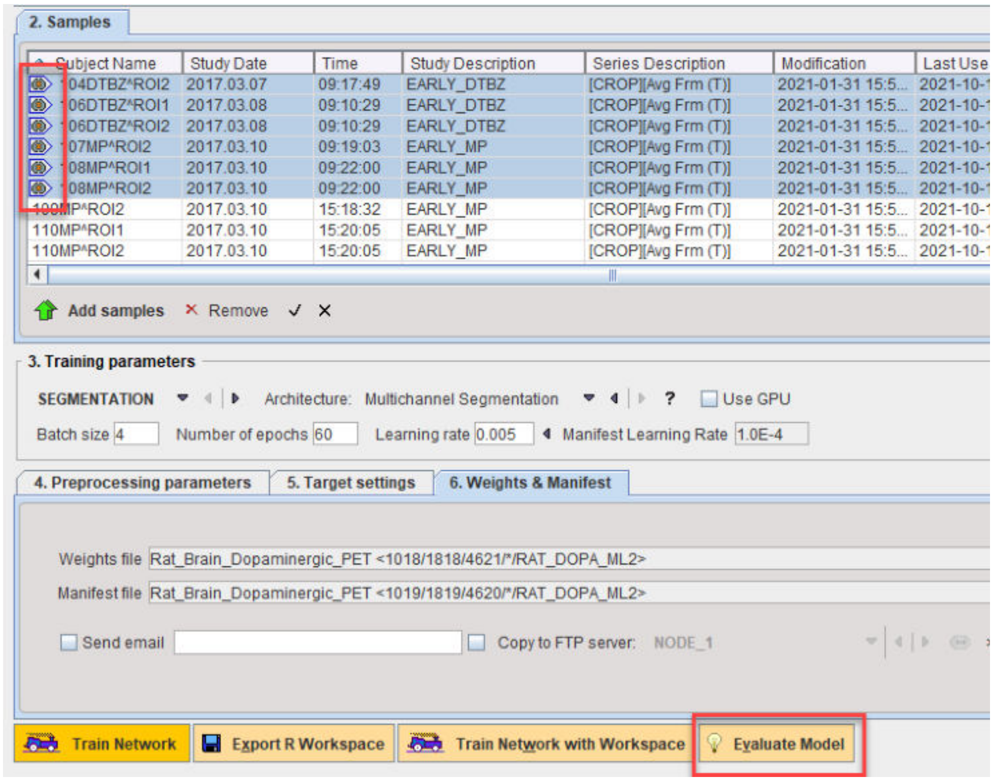
Component name	Subject name	Subject id	Series descr.	Modify time	Last Use	User	Arch	Project	Group
PMODSW-566-02	~[AI LEARNING SET]	AI LEARNING SE		2021-01-12 18:10:	2021-01-12 18:	User1	RAT_DOPA		
PMODSW-566-01	~[AI LEARNING SET]	AI LEARNING SE		2021-01-12 16:49:	2021-01-12 18:	User1	RAT_DOPA		

Create Learning Set + [Delete | Export | Rename] existing Import Description

2. Samples [4 of 10]

Subject Name	Study Date	Study Time	Series Date	Series Time	Study Description	Series Description	Modification	Last Use	Mod	nt
34DTBZ^ROI1	2017.10.11	09:16:15			EARLY_DTBZ	[CROP][Avg Frm (T)]	2021-01-09 17:36:(	2022-06-03 16:	PT	1
34DTBZ^ROI2	2017.10.11	09:16:15			EARLY_DTBZ	[CROP][Avg Frm (T)]	2021-01-09 17:36:(	2021-01-12 16:	PT	1
35DTBZ^ROI1	2017.10.11	09:18:50			EARLY_DTBZ	[CROP][Avg Frm (T)]	2021-01-09 17:35:5	2021-01-12 16:	PT	1
35DTBZ^ROI2	2017.10.11	09:18:50			EARLY_DTBZ	[CROP][Avg Frm (T)]	2021-01-09 17:35:5	2021-01-12 16:	PT	1
[illegible]TBZ^ROI1	2017.10.11	15:09:26			EARLY_DTBZ	[CROP][Avg Frm (T)]	2021-01-09 17:35:5	2021-01-12 16:	PT	1
237DTBZ^ROI1	2017.10.11	15:11:40			EARLY_DTBZ	[CROP][Avg Frm (T)]	2021-01-09 17:35:4	2021-01-12 16:	PT	1
237DTBZ^ROI2	2017.10.11	15:11:40			EARLY_DTBZ	[CROP][Avg Frm (T)]	2021-01-09 17:35:4	2021-01-12 16:	PT	1
238DTBZ^ROI1	2017.10.11	15:15:05			EARLY_DTBZ	[CROP][Avg Frm (T)]	2021-01-09 17:35:4	2021-01-12 16:	PT	1
238DTBZ^ROI2	2017.10.11	15:15:05			EARLY_DTBZ	[CROP][Avg Frm (T)]	2021-01-09 17:35:3	2021-01-12 17:	PT	1
240DTBZ^ROI2	2017.10.12	08:46:45			EARLY_DTBZ	[CROP][Avg Frm (T)]	2021-01-09 17:35:4	2021-01-12 16:	PT	1

The performance of the model on the Test Set can be calculated using Evaluate Model:


2. Samples

Subject Name	Study Date	Time	Study Description	Series Description	Modification	Last Use
04DTBZ^ROI2	2017.03.07	09:17:49	EARLY_DTBZ	[CROP][Avg Frm (T)]	2021-01-31 15:5...	2021-10-
06DTBZ^ROI1	2017.03.08	09:10:29	EARLY_DTBZ	[CROP][Avg Frm (T)]	2021-01-31 15:5...	2021-10-
06DTBZ^ROI2	2017.03.08	09:10:29	EARLY_DTBZ	[CROP][Avg Frm (T)]	2021-01-31 15:5...	2021-10-
07MP^ROI2	2017.03.10	09:19:03	EARLY_MP	[CROP][Avg Frm (T)]	2021-01-31 15:5...	2021-10-
08MP^ROI1	2017.03.10	09:22:00	EARLY_MP	[CROP][Avg Frm (T)]	2021-01-31 15:5...	2021-10-
08MP^ROI2	2017.03.10	09:22:00	EARLY_MP	[CROP][Avg Frm (T)]	2021-01-31 15:5...	2021-10-
100MP^ROI2	2017.03.10	15:18:32	EARLY_MP	[CROP][Avg Frm (T)]	2021-01-31 15:5...	2021-10-
110MP^ROI1	2017.03.10	15:20:05	EARLY_MP	[CROP][Avg Frm (T)]	2021-01-31 15:5...	2021-10-
110MP^ROI2	2017.03.10	15:20:05	EARLY_MP	[CROP][Avg Frm (T)]	2021-01-31 15:5...	2021-10-

3. Training parameters

SEGMENTATION Architecture: Multichannel Segmentation ? Use GPU

Batch size 4 Number of epochs 60 Learning rate 0.005 Manifest Learning Rate 1.0E-4

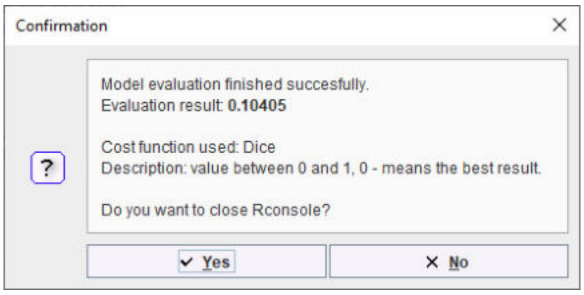
4. Preprocessing parameters | 5. Target settings | 6. Weights & Manifest

Weights file Rat_Brain_Dopaminergic_PET <1018/1818/4621/*/RAT_DOPA_ML2>

Manifest file Rat_Brain_Dopaminergic_PET <1019/1819/4620/*/RAT_DOPA_ML2>

Send email Copy to FTP server: NODE_1

Prediction will be run in R Console and each predicted segment compared to the reference associated with the sample. An average loss value for the test set samples will be returned.


Confirmation

Model evaluation finished succesfully.
Evaluation result: **0.10405**

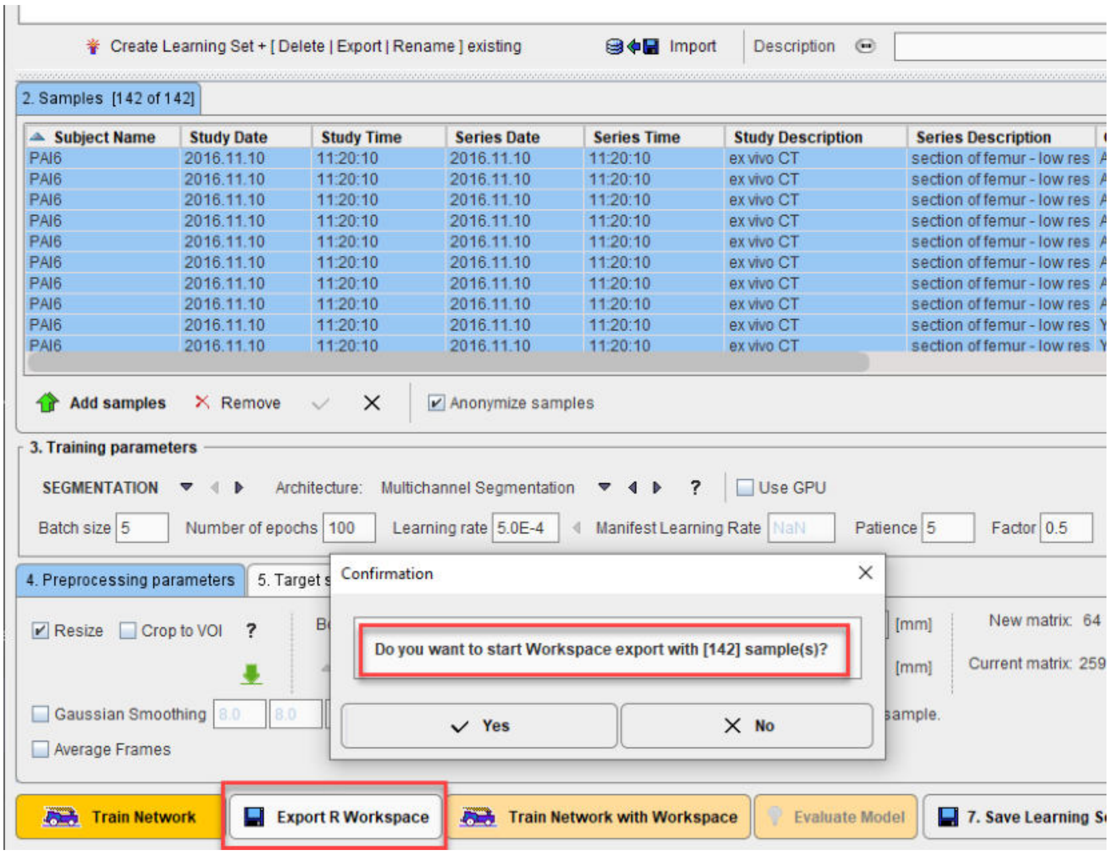
Cost function used: Dice
Description: value between 0 and 1, 0 - means the best result.

Do you want to close Rconsole?

4.1 Exporting an R workspace

Sample Selection

When exporting an R workspace for external training the selection in the **2. Samples** list is relevant. Only the selected samples will be exported and used for the training. An error message will be shown if only a single sample is selected during the export.


Create Learning Set + [Delete | Export | Rename] existing Import Description

2. Samples [142 of 142]

Subject Name	Study Date	Study Time	Series Date	Series Time	Study Description	Series Description	
PAI6	2016.11.10	11:20:10	2016.11.10	11:20:10	ex vivo CT	section of femur - low res	A
PAI6	2016.11.10	11:20:10	2016.11.10	11:20:10	ex vivo CT	section of femur - low res	A
PAI6	2016.11.10	11:20:10	2016.11.10	11:20:10	ex vivo CT	section of femur - low res	A
PAI6	2016.11.10	11:20:10	2016.11.10	11:20:10	ex vivo CT	section of femur - low res	A
PAI6	2016.11.10	11:20:10	2016.11.10	11:20:10	ex vivo CT	section of femur - low res	A
PAI6	2016.11.10	11:20:10	2016.11.10	11:20:10	ex vivo CT	section of femur - low res	A
PAI6	2016.11.10	11:20:10	2016.11.10	11:20:10	ex vivo CT	section of femur - low res	A
PAI6	2016.11.10	11:20:10	2016.11.10	11:20:10	ex vivo CT	section of femur - low res	A
PAI6	2016.11.10	11:20:10	2016.11.10	11:20:10	ex vivo CT	section of femur - low res	Y
PAI6	2016.11.10	11:20:10	2016.11.10	11:20:10	ex vivo CT	section of femur - low res	Y

Add samples Remove ☑ Anonymize samples

3. Training parameters

SEGMENTATION Architecture: Multichannel Segmentation ? ☐ Use GPU

Batch size 5 Number of epochs 100 Learning rate 5.0E-4 Manifest Learning Rate NaN Patience 5 Factor 0.5

4. Preprocessing parameters 5. Target s

☑ Resize ☐ Crop to VOI ?

☐ Gaussian Smoothing 8.0 8.0

☐ Average Frames

[mm] New matrix: 64
[mm] Current matrix: 259
ample.

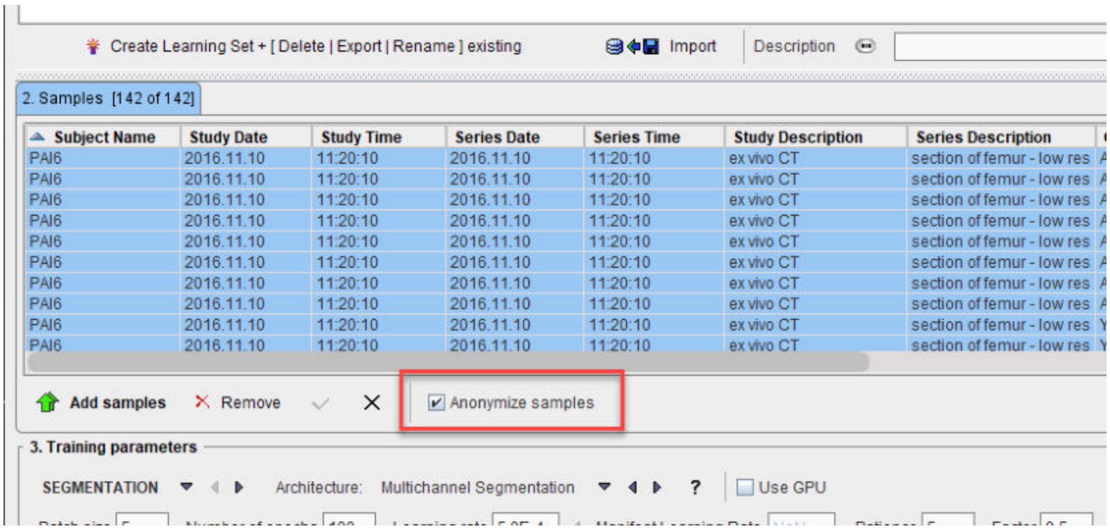
Confirmation

Do you want to start Workspace export with [142] sample(s)?

Yes No

Train Network Export R Workspace Train Network with Workspace Evaluate Model 7. Save Learning S

Note that the **Anonymize samples** option is useful when exporting an R workspace, ensuring that no subject information will be transferred to another workstation or cloud-computing infrastructure:


Create Learning Set + [Delete | Export | Rename] existing Import Description

2. Samples [142 of 142]

Subject Name	Study Date	Study Time	Series Date	Series Time	Study Description	Series Description	
PAI6	2016.11.10	11:20:10	2016.11.10	11:20:10	ex vivo CT	section of femur - low res	A
PAI6	2016.11.10	11:20:10	2016.11.10	11:20:10	ex vivo CT	section of femur - low res	A
PAI6	2016.11.10	11:20:10	2016.11.10	11:20:10	ex vivo CT	section of femur - low res	A
PAI6	2016.11.10	11:20:10	2016.11.10	11:20:10	ex vivo CT	section of femur - low res	A
PAI6	2016.11.10	11:20:10	2016.11.10	11:20:10	ex vivo CT	section of femur - low res	A
PAI6	2016.11.10	11:20:10	2016.11.10	11:20:10	ex vivo CT	section of femur - low res	A
PAI6	2016.11.10	11:20:10	2016.11.10	11:20:10	ex vivo CT	section of femur - low res	A
PAI6	2016.11.10	11:20:10	2016.11.10	11:20:10	ex vivo CT	section of femur - low res	A
PAI6	2016.11.10	11:20:10	2016.11.10	11:20:10	ex vivo CT	section of femur - low res	Y
PAI6	2016.11.10	11:20:10	2016.11.10	11:20:10	ex vivo CT	section of femur - low res	Y

Add samples Remove ☑ Anonymize samples

3. Training parameters

SEGMENTATION Architecture: Multichannel Segmentation ? ☐ Use GPU

R Workspace Export

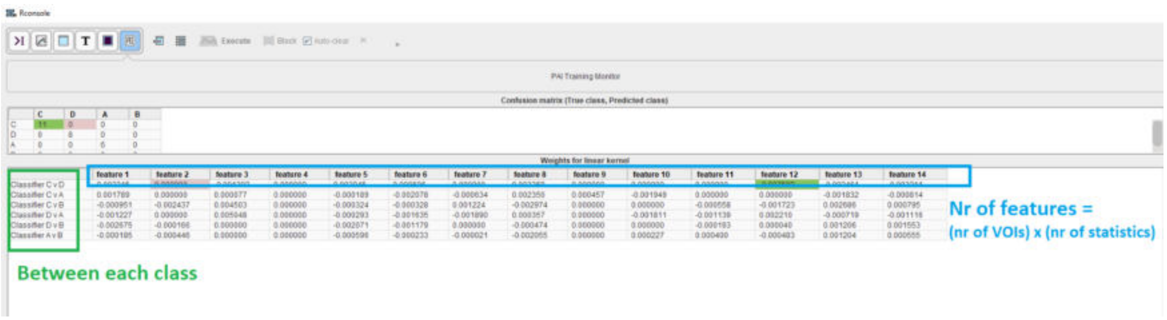
The **EXPORT** button opens the PMOD R Console, executes the preparations including data pre-processing, and opens a dialog window for saving the workspace. Typically **Save to File System** will be used to make it easily available for transfer to a different training environment.

4.2 Output Visualization for SVM Classification

A tabulated output of the weights by VOI and statistic (according to **Image reduction** options) is available following training with the Classification SVM architecture.

The confusion matrix is available for each SVM kernel.

For linear kernel a table of weights is available:



Confusion matrix (True class, Predicted class)

	C	D	A	B
C	🟩 11	🟥 0	0	0
D	0	8	0	0
A	0	0	6	0

Weights for linear kernel

	feature 1	feature 2	feature 3	feature 4	feature 5	feature 6	feature 7
Classifier C v D	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Classifier C v A	0.001789	0.000000	0.000077	0.000000	-0.000189	-0.002078	-0.000634
Classifier C v B	-0.000951	-0.002437	0.004503	0.000000	-0.000324	-0.000328	0.001224
Classifier D v A	-0.001227	0.000000	0.005048	0.000000	-0.000293	-0.001635	-0.001890
Classifier D v B	-0.002675	-0.000166	0.000000	0.000000	-0.002071	-0.001179	0.000000
Classifier A v B	-0.000185	-0.000446	0.000000	0.000000	-0.000598	-0.000233	-0.000021

	feature 8	feature 9	feature 10	feature 11	feature 12	feature 13	feature 14
Classifier C v D	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Classifier C v A	0.002356	0.000457	-0.001949	0.000000	0.000000	-0.001832	-0.000814
Classifier C v B	-0.002974	0.000000	0.000000	-0.000558	-0.001723	0.002686	0.000795
Classifier D v A	0.000357	0.000000	-0.001811	-0.001139	0.002210	-0.000719	-0.001116
Classifier D v B	-0.000474	0.000000	0.000000	-0.000193	0.000040	0.001206	0.001553
Classifier A v B	-0.002055	0.000000	0.000227	0.000400	-0.000483	0.001204	0.000555

PAI uses the SVC variant of SVM.

SVC implements the "one-versus-one" approach for multi-class classification. In total, n_classes * (n_classes - 1) / 2 classifiers are constructed and each one trains data from two classes. (https://scikit-learn.org/stable/modules/svm.html)

For example, when we have 4 different classes: 4*(4-3)/2 = 6.

The bigger an absolute value in a given row of the table, the more this feature distinguishes between the 2 classes.

For example for 2 VOIs and 3 statistics we have 6 features:

Feature 1 corresponds to VOI1, stats 1.

Feature 2 corresponds to VOI1, stats 2

Feature 3 corresponds to VOI1, stats 3

Feature 4 corresponds to VOI2, stats 1

Feature 4 corresponds to VOI2, stats 2

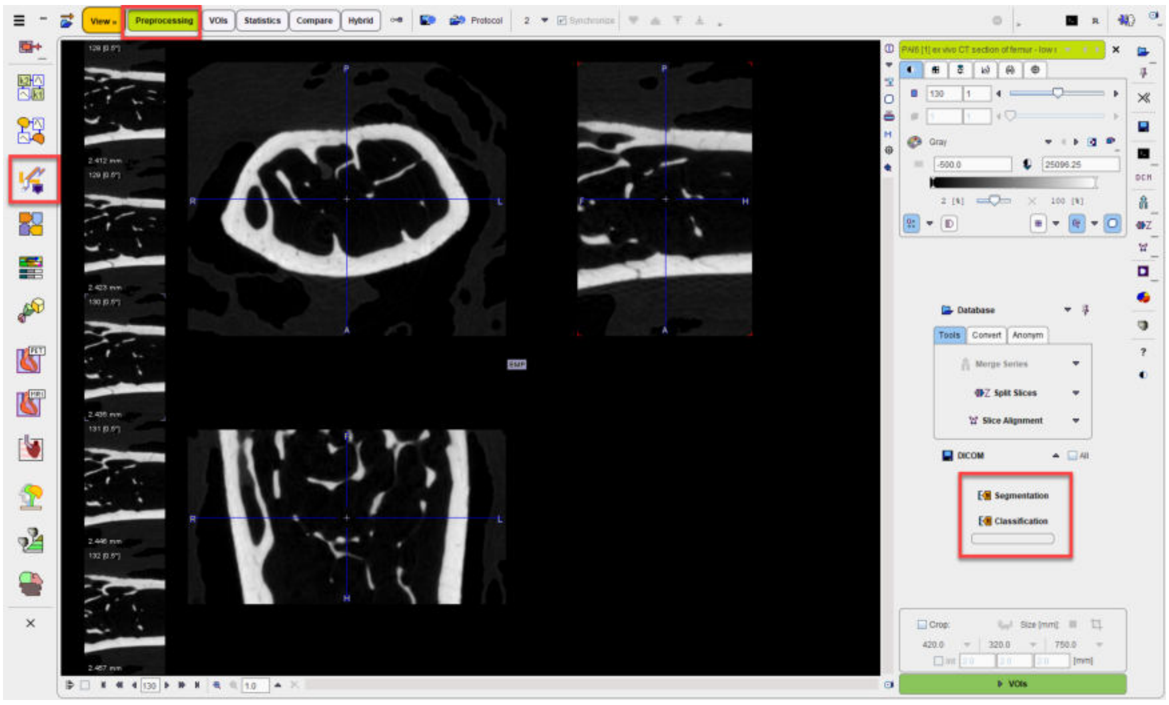
Feature 4 corresponds to VOI2, stats 3.

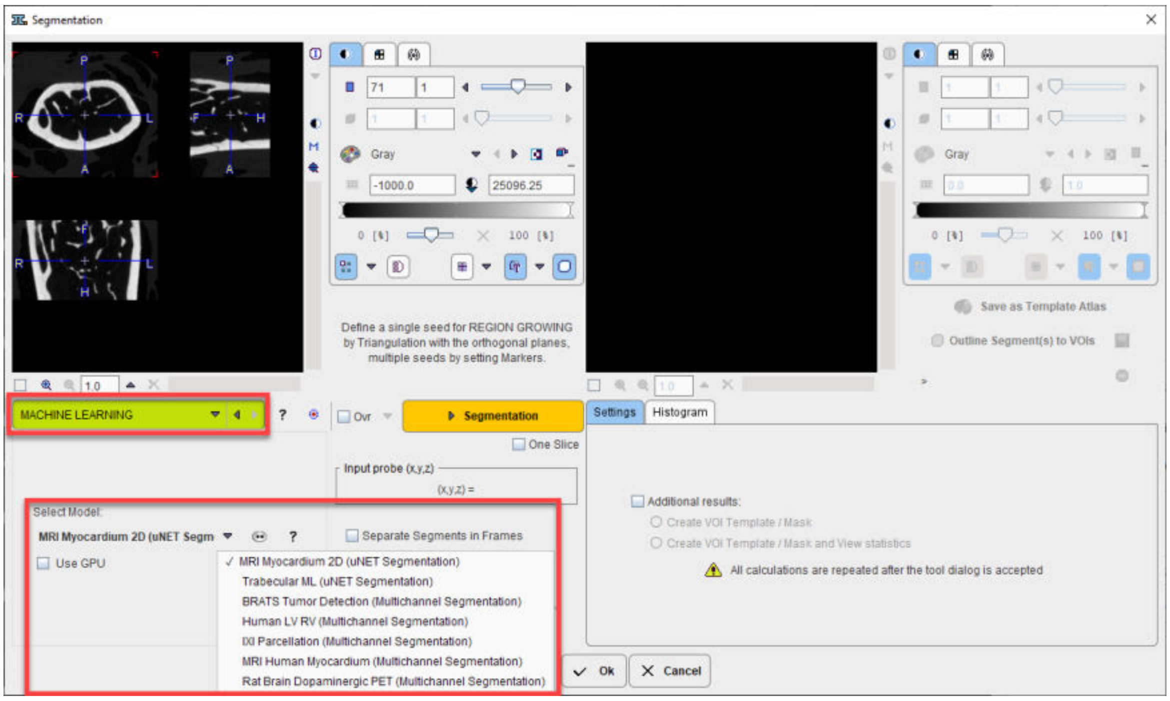
5 Use of Trained Neural Network for Prediction

Trained networks can be used in **View (including Pipeline Processing), Segment (PSEG), Neuro (PNEURO) and Cardiac MR (PCARDM)** tools for the segmentation or classification of input images with the same characteristics as the training images. Please apply the same preparations 41 including associations and cropping VOI definition as for the training samples.

5.1 In the View tool

A shortcut to the **Segmentation** interface with **MACHINE LEARNING** segmentation method is available on the **View** tab for the data currently loaded. Alternatively the **Segmentation** tool may be opened from the **Image Processing Tools** (for segment generation) or **VOI Tools** (for direct VOI generation without segment) and **MACHINE LEARNING** selected from the list of segmentation methods on the left:




Segmentation
MACHINE LEARNING
Segmentation
One Slice
Input probe (x,y,z)
Settings
Histogram
Define a single seed for REGION GROWING by Triangulation with the orthogonal planes, multiple seeds by setting Markers.
Save as Template Atlas
Outline Segment(s) to VOIs
Additional results:
Create VOI Template / Mask
Create VOI Template / Mask and View statistics
All calculations are repeated after the tool dialog is accepted
Separate Segments in Frames
Use GPU
Ok
Cancel

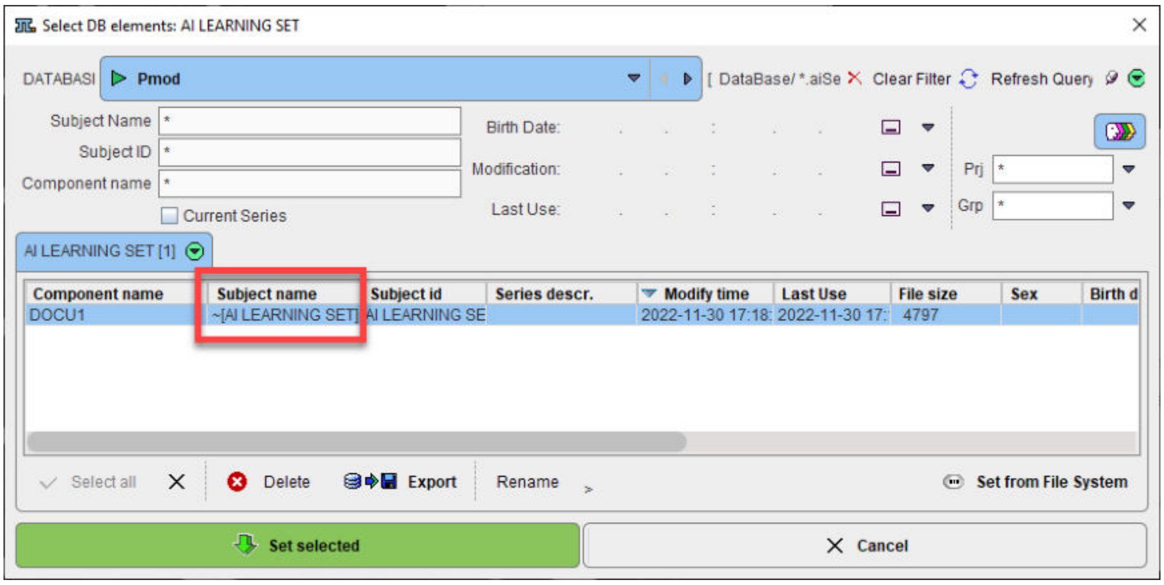
Select Model: MRI Myocardium 2D (uNET Segm

✓ MRI Myocardium 2D (uNET Segmentation)
Trabecular ML (uNET Segmentation)
BRATS Tumor Detection (Multichannel Segmentation)
Human LV RV (Multichannel Segmentation)
IXI Parcellation (Multichannel Segmentation)
MRI Human Myocardium (Multichannel Segmentation)
Rat Brain Dopaminergic PET (Multichannel Segmentation)

Use of an available/compatible GPU may be toggled on/off using the **Use GPU** checkbox.

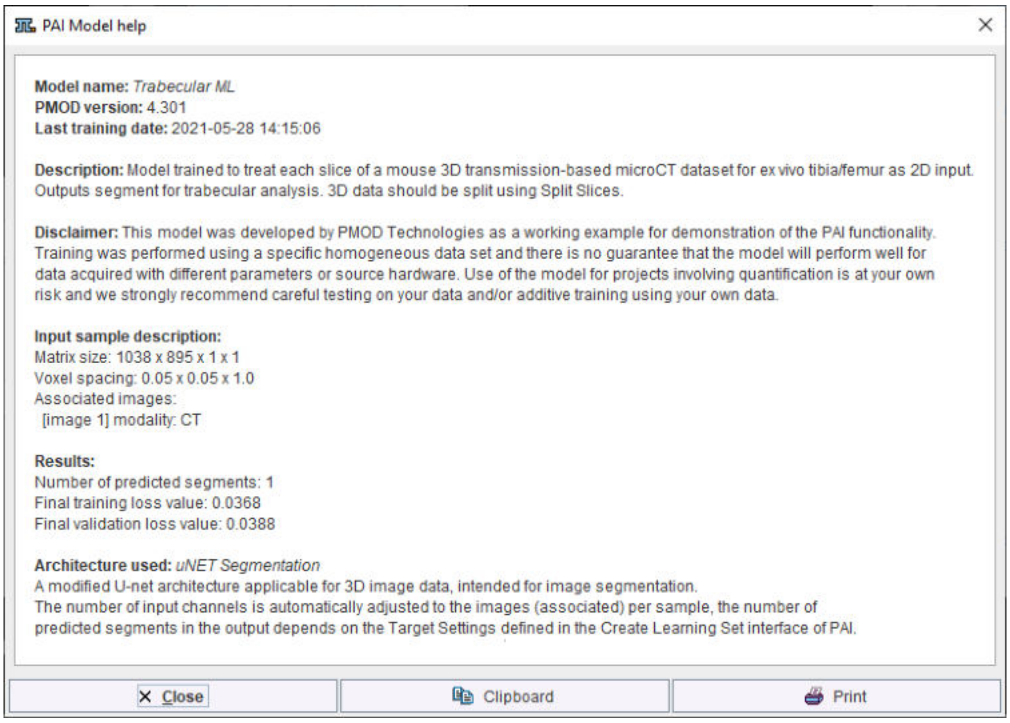
Following the selection of the **MACHINE LEARNING** segmentation method the trained model should be selected from the **Select Model** list. Trained models available in resources/pai are shown with naming according to the weights folder and architecture used.

Additionally, self-trained models saved in the database can be selected using the **AI Learning Set** browser:



Component name	Subject name	Subject id	Series descr.	Modify time	Last Use	File size	Sex	Birth d
DOCU1	~[AI LEARNING SET]	AI LEARNING SE		2022-11-30 17:18:	2022-11-30 17:	4797		

Further information about the trained model selected is available in the model help:


PAI Model help

Model name: *Trabecular ML*
PMOD version: 4.301
Last training date: 2021-05-28 14:15:06

Description: Model trained to treat each slice of a mouse 3D transmission-based microCT dataset for ex vivo tibia/femur as 2D input. Outputs segment for trabecular analysis. 3D data should be split using Split Slices.

Disclaimer: This model was developed by PMOD Technologies as a working example for demonstration of the PAI functionality. Training was performed using a specific homogeneous data set and there is no guarantee that the model will perform well for data acquired with different parameters or source hardware. Use of the model for projects involving quantification is at your own risk and we strongly recommend careful testing on your data and/or additive training using your own data.

Input sample description:
Matrix size: 1038 x 895 x 1 x 1
Voxel spacing: 0.05 x 0.05 x 1.0
Associated images:
[image 1] modality: CT

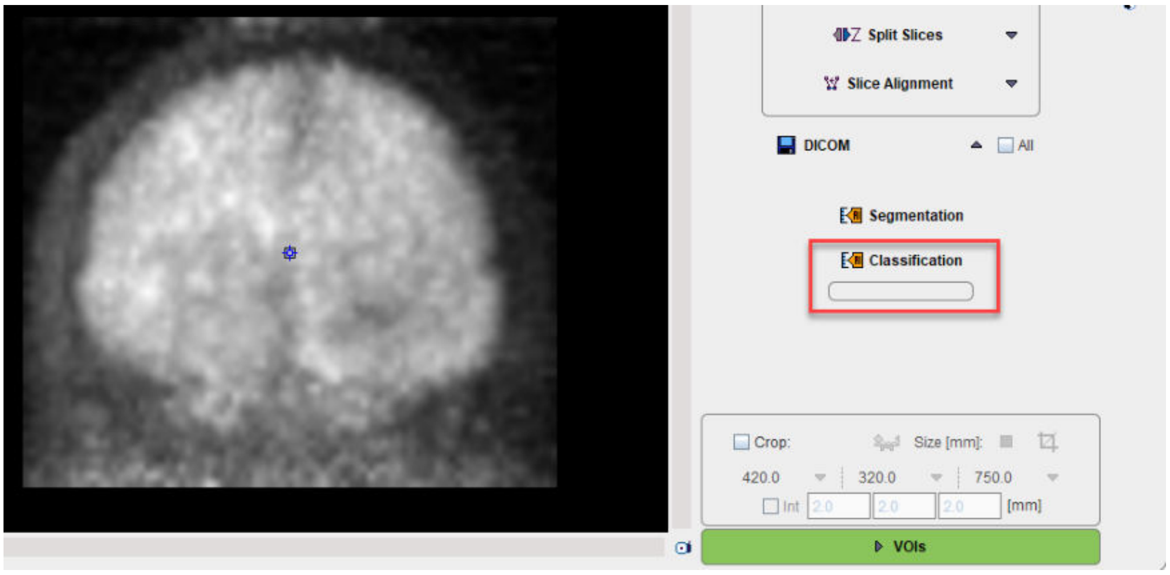
Results:
Number of predicted segments: 1
Final training loss value: 0.0368
Final validation loss value: 0.0388

Architecture used: *uNET Segmentation*
A modified U-net architecture applicable for 3D image data, intended for image segmentation.
The number of input channels is automatically adjusted to the images (associated) per sample, the number of predicted segments in the output depends on the Target Settings defined in the Create Learning Set interface of PAI.

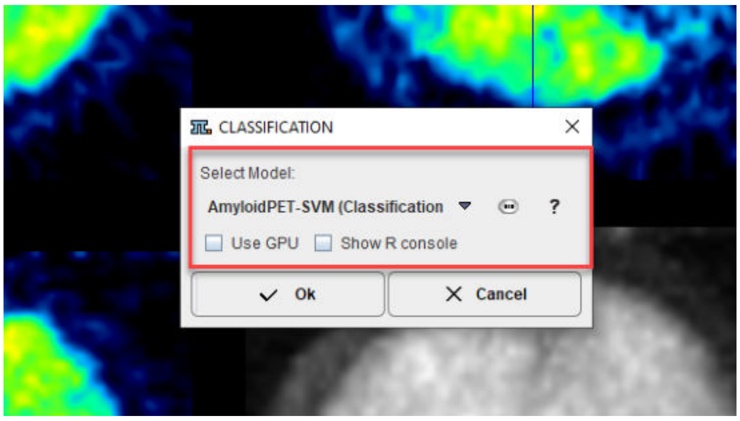
Note that the available text shown is controlled by the user for self-trained models during creation of the Learning Set (and architecture Python files if applicable).

The dimensions of input data to a model for prediction must match the dimensions used for training of the model. When a model was trained using 2D data, 3D datasets will be automatically split into 2D and the resulting segments reassembled into 3D volumes. Likewise, data with multiple frames (such as dynamic PET data) will be split/reassembled according to how the model was trained.

A shortcut to AI-based classification is available on the View tab for the data currently loaded.

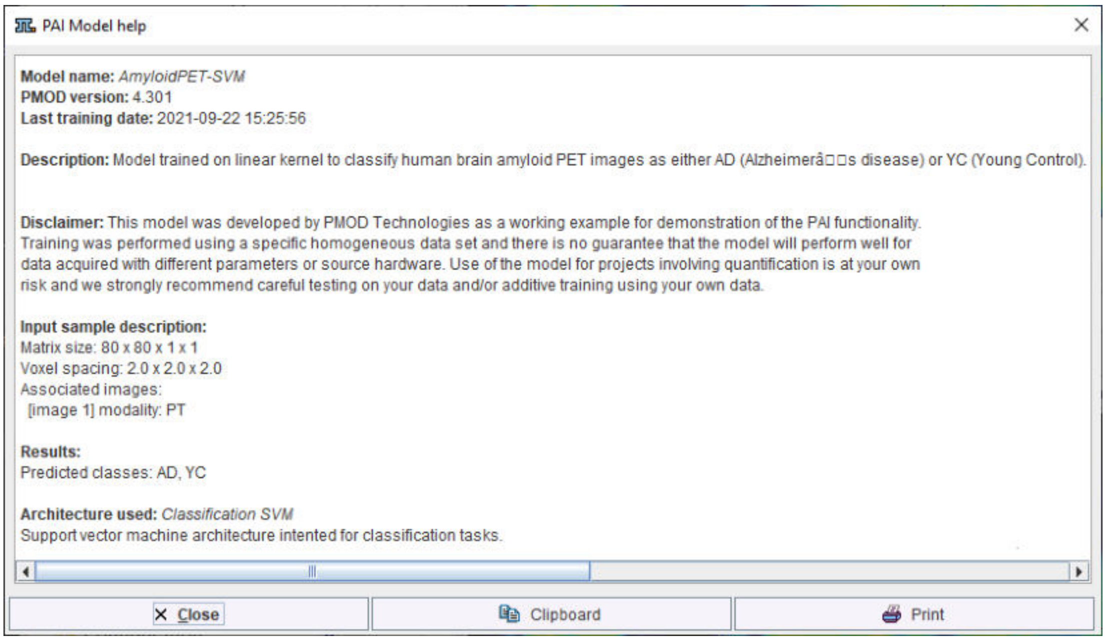


The model to be used must be selected from the menu.



Self-trained models may be selected using the **AI Learning Set** browser.

Additional information about the model is available using the PAI Model Help button.



Model name: *AmyloidPET-SVM*
PMOD version: 4.301
Last training date: 2021-09-22 15:25:56

Description: Model trained on linear kernel to classify human brain amyloid PET images as either AD (Alzheimerâ□□s disease) or YC (Young Control).

Disclaimer: This model was developed by PMOD Technologies as a working example for demonstration of the PAI functionality. Training was performed using a specific homogeneous data set and there is no guarantee that the model will perform well for data acquired with different parameters or source hardware. Use of the model for projects involving quantification is at your own risk and we strongly recommend careful testing on your data and/or additive training using your own data.

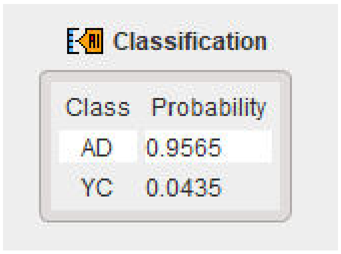
Input sample description:
Matrix size: 80 x 80 x 1 x 1
Voxel spacing: 2.0 x 2.0 x 2.0
Associated images:
[image 1] modality: PT

Results:
Predicted classes: AD, YC

Architecture used: *Classification SVM*
Support vector machine architecture intented for classification tasks.

Display of the R Console during classification is optional and is toggled using the **Show R console** checkbox.

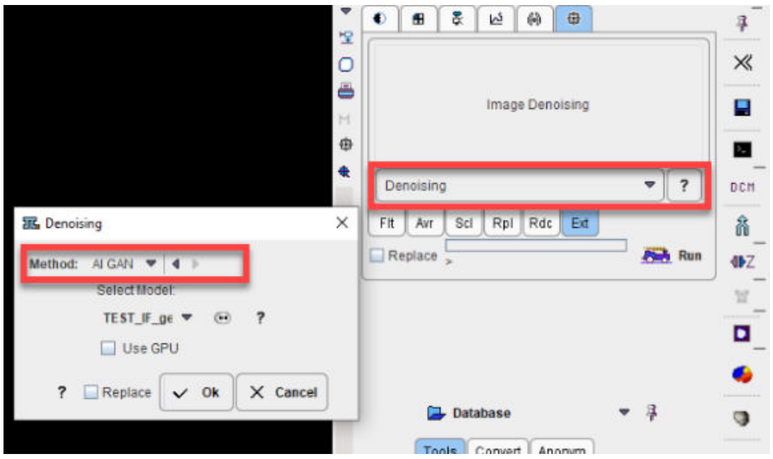
The result of classification is shown in tabulated form below the **Classification** shortcut.



Class	Probability
AD	0.9565
YC	0.0435

See the Case Study for Human Amyloid PET Classification 99 for further information

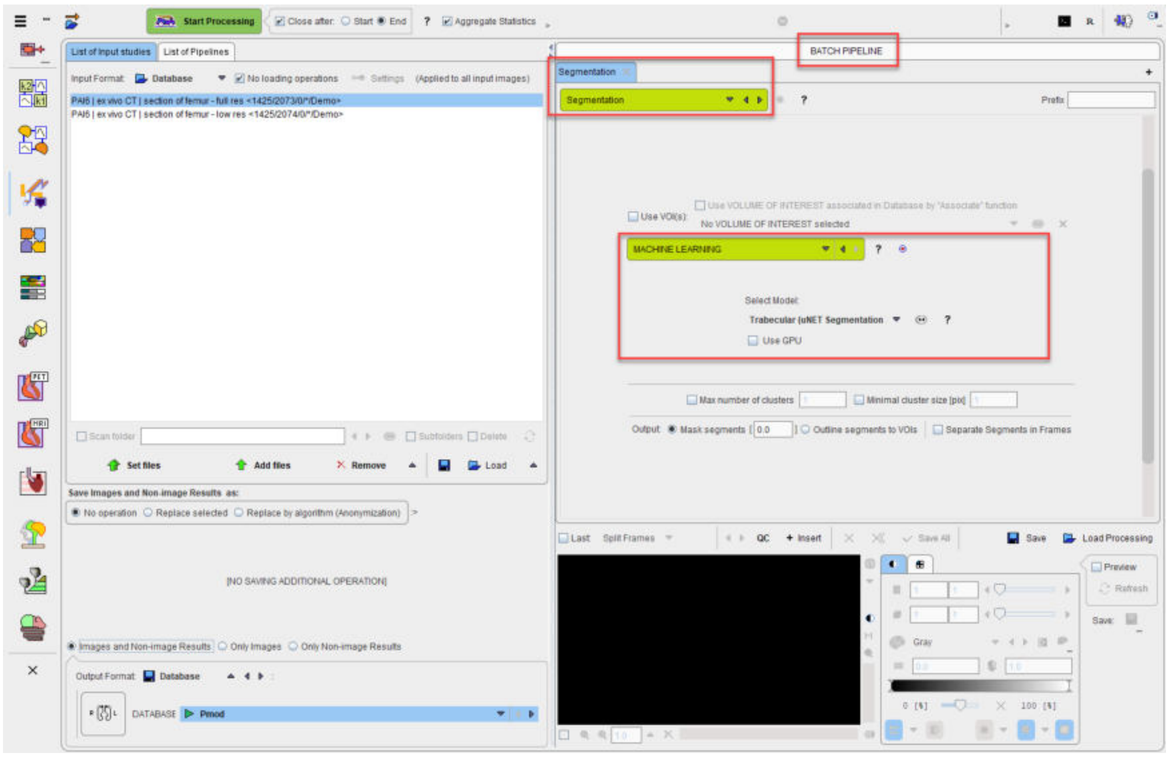
AI-based image denoising is under development using Generative Adversarial Networks (GAN). This functionality will be available as an option in the Denoising image processing tool (e.g. View tool):



An alternative method using Non-local Means is also available in Denoising image processing tool.

5.2 In Pipeline processing

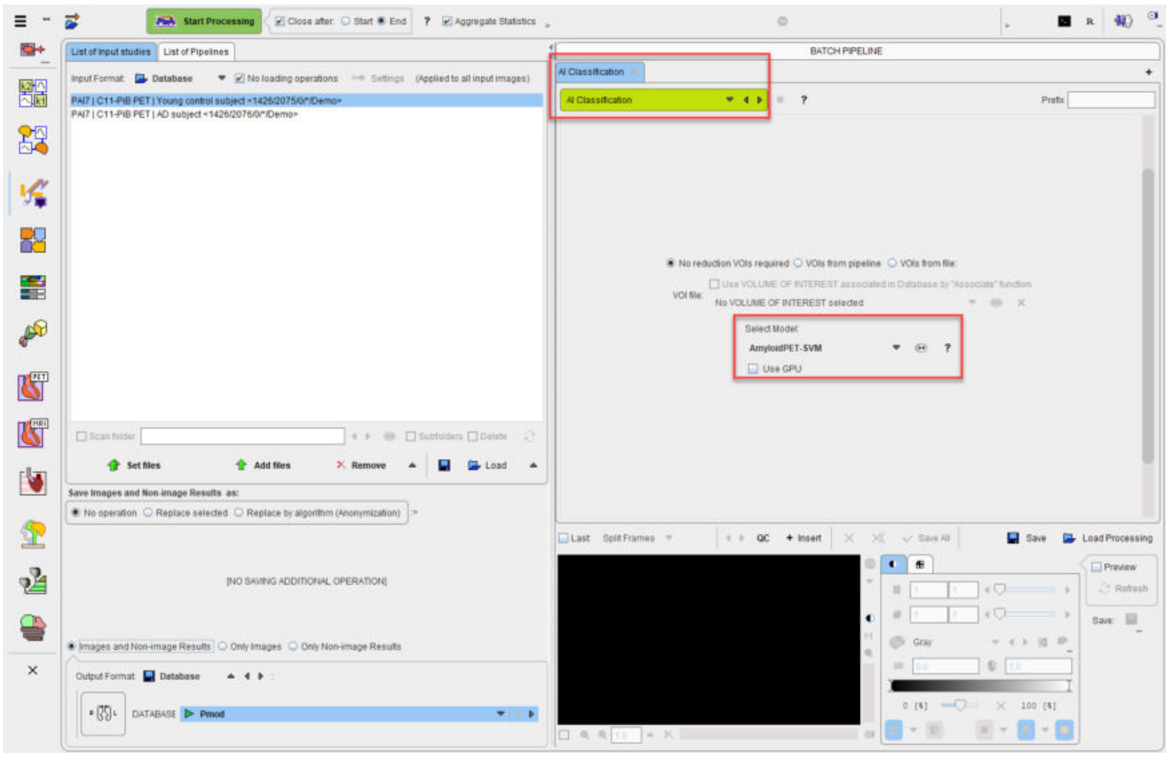
The **MACHINE LEARNING** segmentation method is also available in the **Segmentation** image processing tool within **Pipeline Processing**. This makes batch processing with AI-based segmentation available.



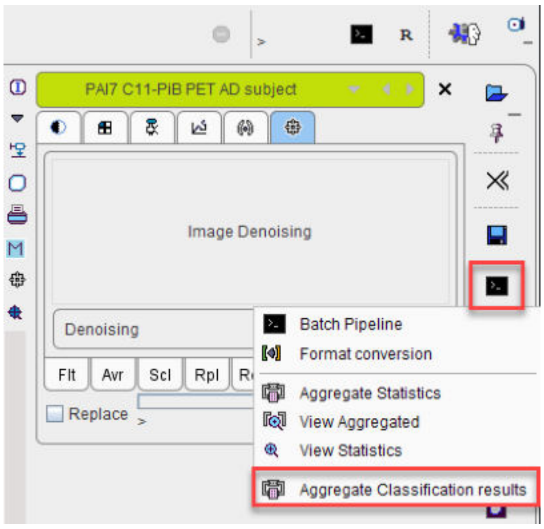
As for interactive segmentation, the model to be used must be selected from the list.

Note that the options to **Use VOIs**, control **Max number of clusters** or **Minimal cluster size**, and **Separate Segments in Frames**, are not applicable when **MACHINE LEARNING** is used.

Classification can also be performed in **Pipeline Processing**:



Classification results generated from Pipeline Processing may be aggregated to facilitate statistical analysis:



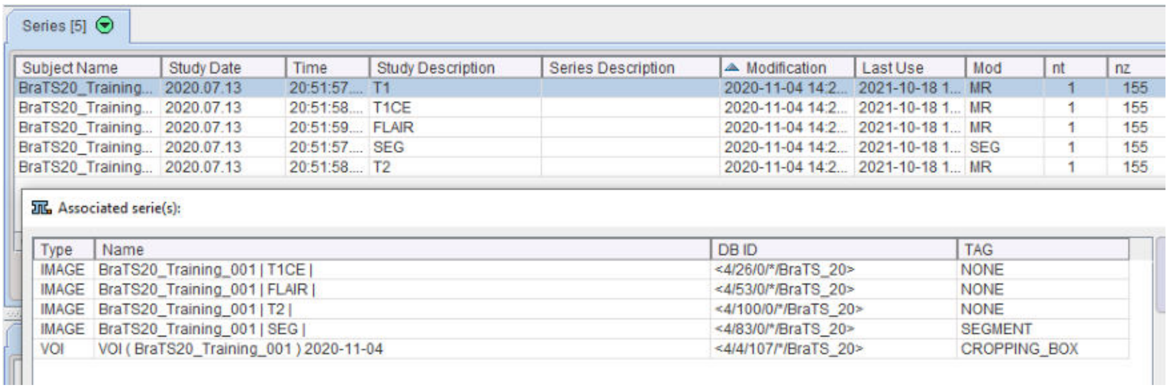
See the PBAS Documentation for further details on Pipeline Processing, Aggregation and use of the R Console.

5.3 In the PSEG tool

Note that the screenshots in this section illustrate PMOD version 4.3

Image Loading

PSEG does not support the loading of multiple input image series. This is not necessary for AI-based segmentation. In case of a segmentation requiring multiple input images, use the reference for the association list (e.g. T1 MR for the BraTS Tumor Segmentation[83] case study).



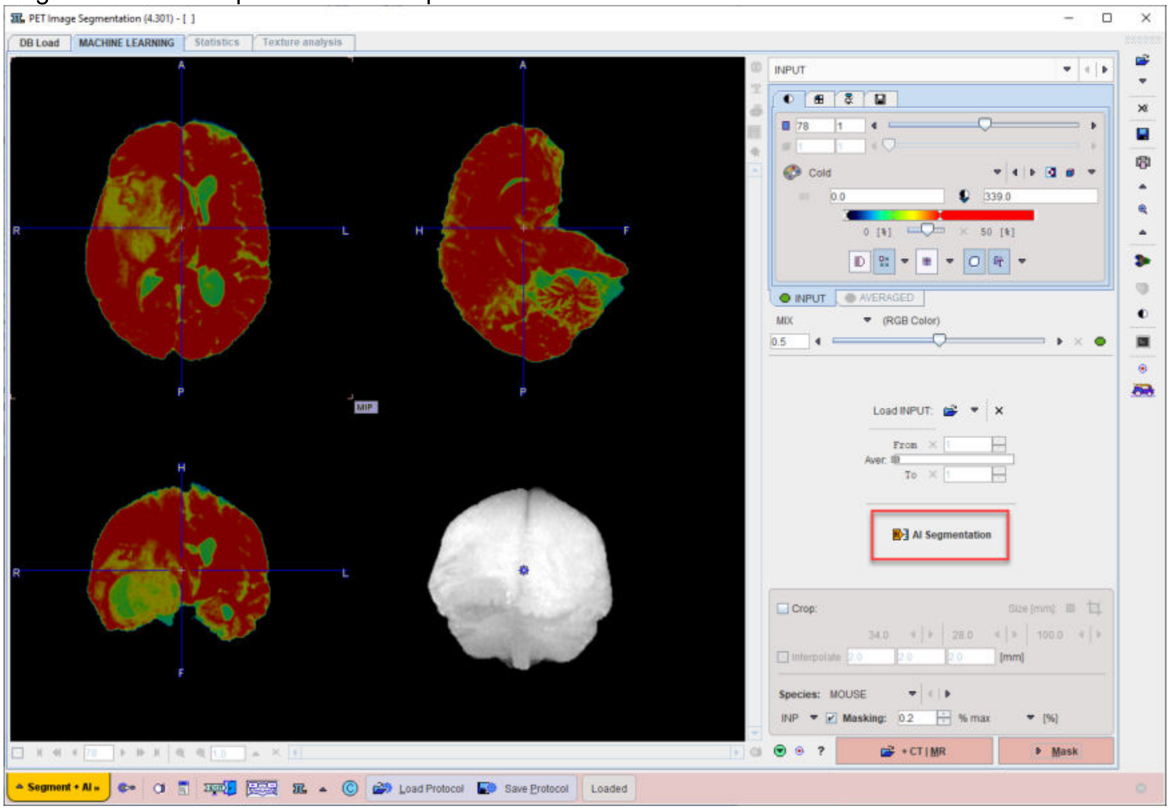
Series [5]

Subject Name	Study Date	Time	Study Description	Series Description	Modification	Last Use	Mod	nt	nz
BraTS20_Training...	2020.07.13	20:51:57....	T1		2020-11-04 14:2...	2021-10-18 1...	MR	1	155
BraTS20_Training...	2020.07.13	20:51:58....	T1CE		2020-11-04 14:2...	2021-10-18 1...	MR	1	155
BraTS20_Training...	2020.07.13	20:51:59....	FLAIR		2020-11-04 14:2...	2021-10-18 1...	MR	1	155
BraTS20_Training...	2020.07.13	20:51:57....	SEG		2020-11-04 14:2...	2021-10-18 1...	SEG	1	155
BraTS20_Training...	2020.07.13	20:51:58....	T2		2020-11-04 14:2...	2021-10-18 1...	MR	1	155

Associated serie(s):

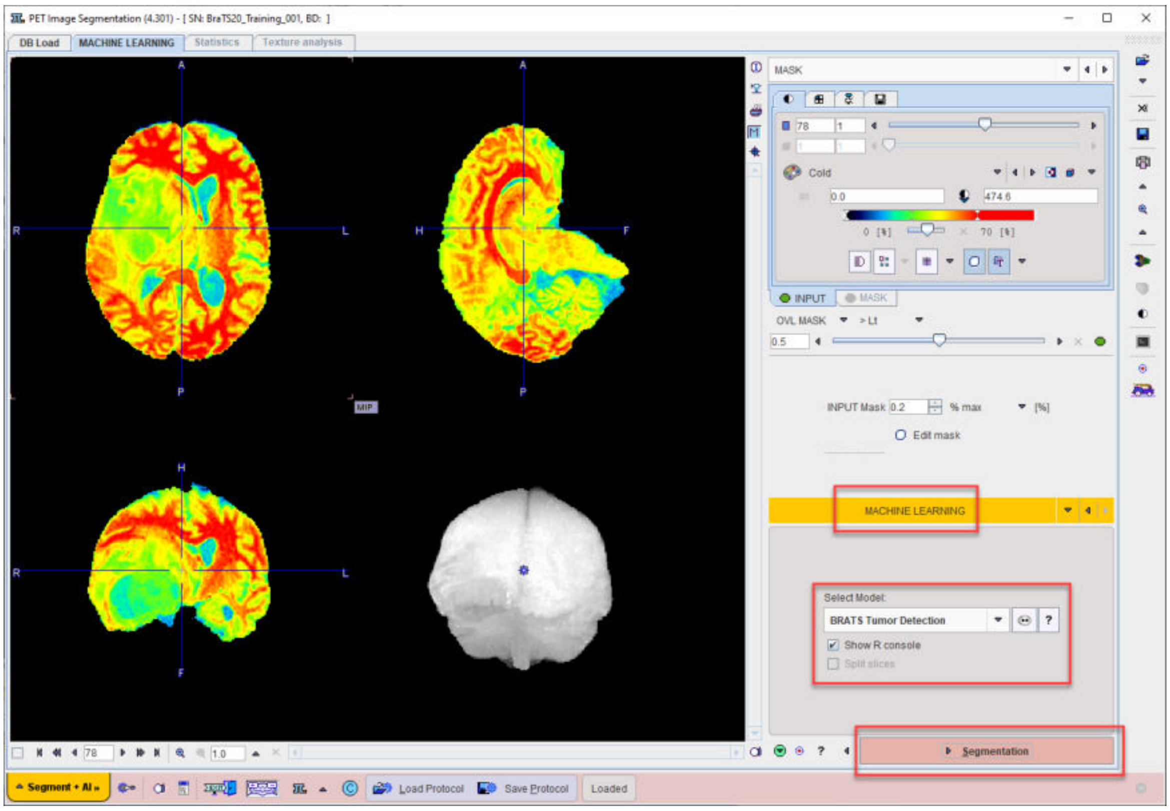
Type	Name	DB ID	TAG
IMAGE	BraTS20_Training_001 | T1CE |	<4/26/0/*/BraTS_20>	NONE
IMAGE	BraTS20_Training_001 | FLAIR |	<4/53/0/*/BraTS_20>	NONE
IMAGE	BraTS20_Training_001 | T2 |	<4/100/0/*/BraTS_20>	NONE
IMAGE	BraTS20_Training_001 | SEG |	<4/83/0/*/BraTS_20>	SEGMENT
VOI	VOI (BraTS20_Training_001) 2020-11-04	<4/4/107/*/BraTS_20>	CROPPING_BOX

Cropping or interpolation in the lower right are generally not required for the AI-based segmentation workflow. This will be done by the model depending on the preprocessing parameters used in the Learning Set. Cropping may be used when the image field-of-view is clearly larger than the scope of the model. For example, an MR image of the human head including neck may be cropped to brain only for the IXI Parcellation model (human brain MR deep nuclei segmentation[95] , see Case Study). **Masking** is not required for AI-based segmentation. We recommend using the shortcut AI Segmentation to skip to the next step in the workflow:

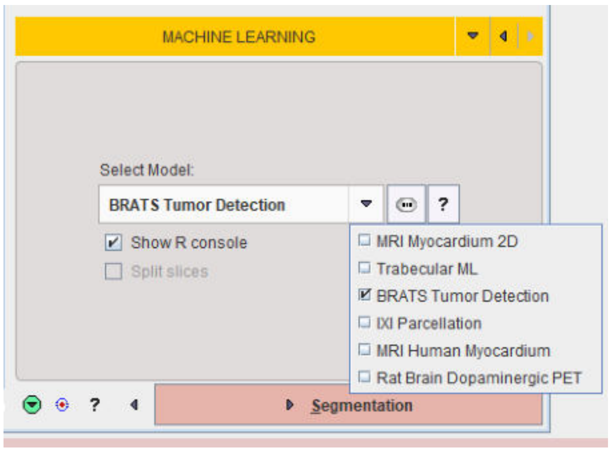


Model Selection

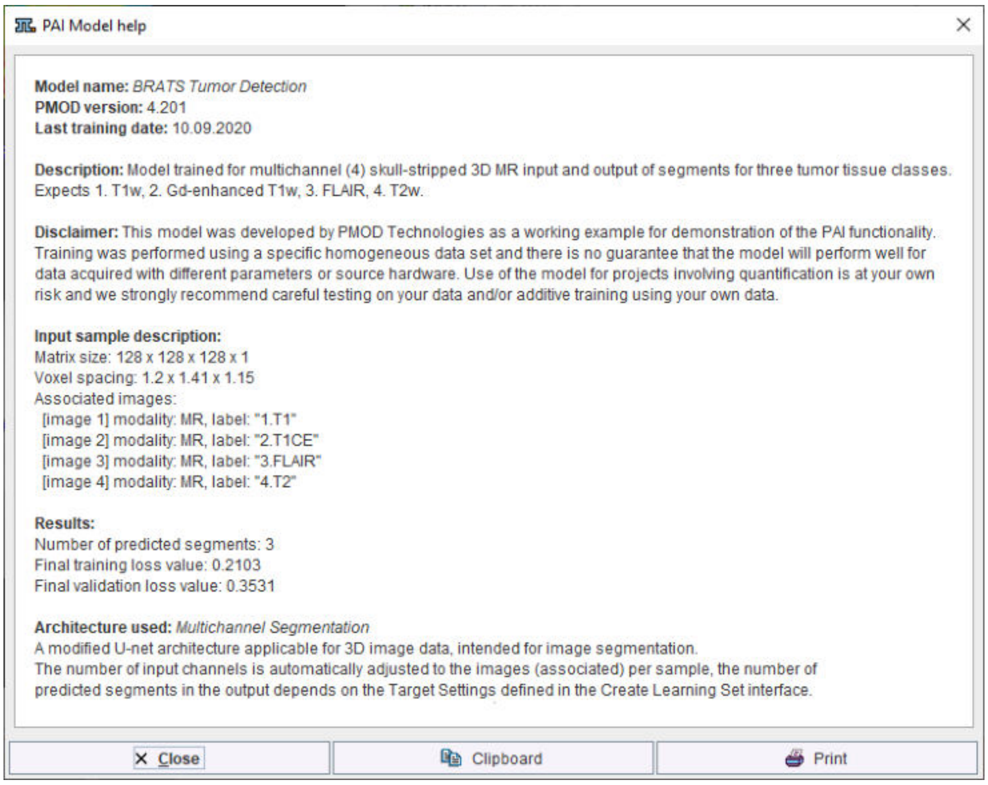
On the **MASK** page set the segmentation method is set to **MACHINE LEARNING**.



Choose the appropriate model for the segmentation task using the menu **Select Model**.



Self-trained models may be selected using the AI Learning Set browser. Note that a description of the model can be retrieved using the **PAI Model Help** button:



Model name: *BRATS Tumor Detection*
PMOD version: 4.201
Last training date: 10.09.2020

Description: Model trained for multichannel (4) skull-stripped 3D MR input and output of segments for three tumor tissue classes. Expects 1. T1w, 2. Gd-enhanced T1w, 3. FLAIR, 4. T2w.

Disclaimer: This model was developed by PMOD Technologies as a working example for demonstration of the PAI functionality. Training was performed using a specific homogeneous data set and there is no guarantee that the model will perform well for data acquired with different parameters or source hardware. Use of the model for projects involving quantification is at your own risk and we strongly recommend careful testing on your data and/or additive training using your own data.

Input sample description:
Matrix size: 128 x 128 x 128 x 1
Voxel spacing: 1.2 x 1.41 x 1.15
Associated images:
[image 1] modality: MR, label: "1.T1"
[image 2] modality: MR, label: "2.T1CE"
[image 3] modality: MR, label: "3.FLAIR"
[image 4] modality: MR, label: "4.T2"

Results:
Number of predicted segments: 3
Final training loss value: 0.2103
Final validation loss value: 0.3531

Architecture used: *Multichannel Segmentation*
A modified U-net architecture applicable for 3D image data, intended for image segmentation.
The number of input channels is automatically adjusted to the images (associated) per sample, the number of predicted segments in the output depends on the Target Settings defined in the Create Learning Set interface.

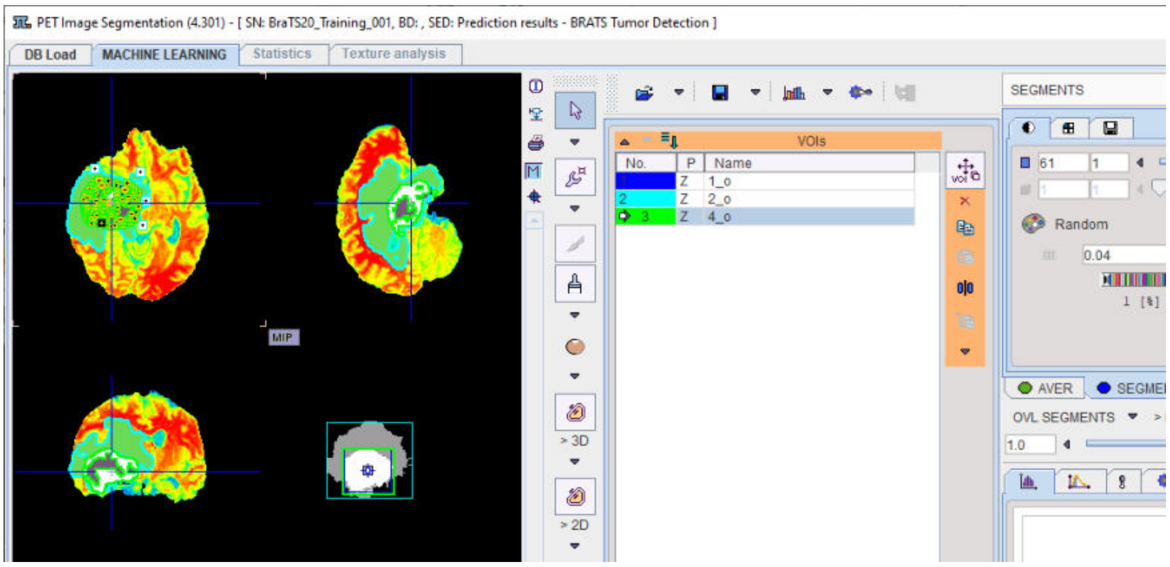
Note that the available text shown is controlled by the user for self-trained models during creation of the Learning Set (and architecture Python files if applicable).

Displaying the R Console during prediction is optional, according to the checkbox selection **Show R Console**.

Segmentation

Use the **Segmentation** workflow button to start prediction. The input data is transferred to the R Console and processed using the selected model.

The segmentation result is overlaid on the input series on the **SEGMENTS** page.


PET Image Segmentation (4.301) - [SN: BraTS20_Training_001, BD: , SED: Prediction results - BRATS Tumor Detection]
DB Load
MACHINE LEARNING
Statistics
Texture analysis
MIP
> 3D
> 2D
VOIs
SEGMENTS
61
1
Random
0.04
1 [%]
AVER
SEGME
OVL SEGMENTS
1.0

No.	P	Name
1	Z	1_o
2	Z	2_o
3	Z	4_o

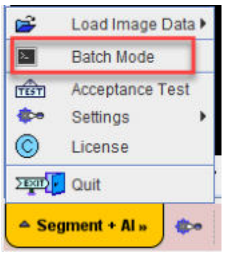
The segments are automatically converted into numbered VOIs. The interactive VOI labeling function is still available by right-clicking a segment in the image. This opens a dialog window which allows a VOI name to be selected from a list, or simply entered. Note that the VOI name from list functionality is also available via VOI Properties.

The segments label map itself can be saved using the save button in the taskbar to the right.

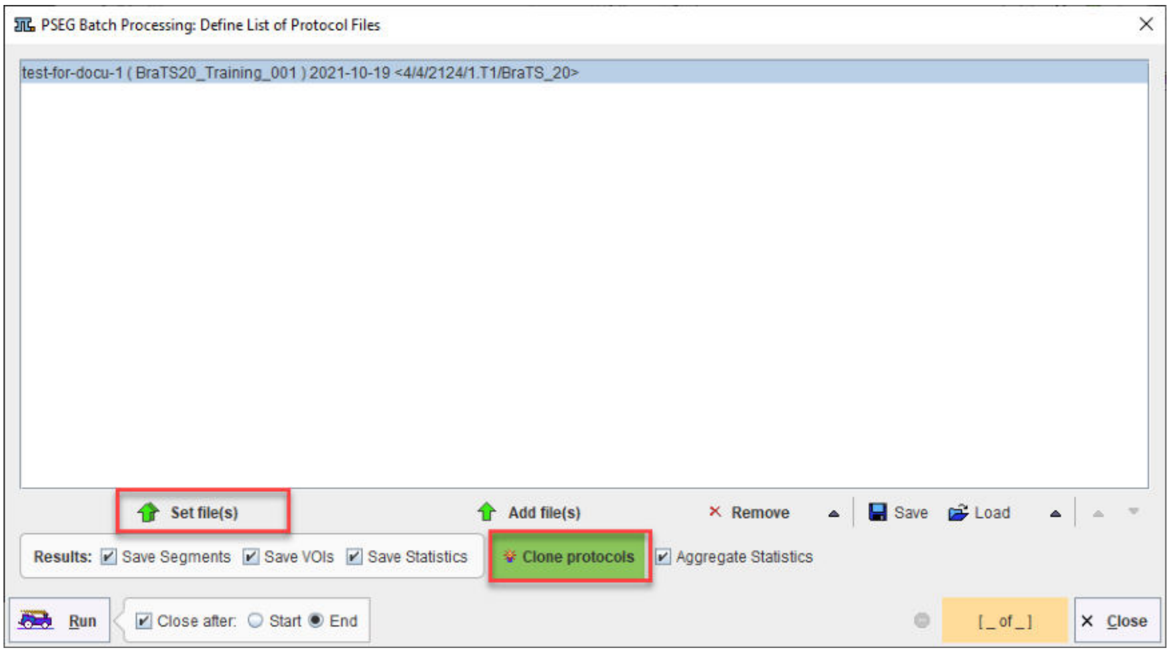
Please refer to the *PSEG User Guide* for details of the general segmentation functionality.

At the end of processing the protocol can be saved using **Save Protocol**. Using Load Protocol the workflow can be retrieved and re-executed. Additionally, the input series can be changed and the protocol used for analysis of the next sample. Protocols also provide access to batch processing.

Batch Processing



A saved protocol is required to start batch processing. This saved protocol provides the settings that will be used for the workflow across all samples. A set of saved protocols can be loaded using **Set Files**, or more commonly a single protocol describing the desired workflow can be cloned for all input samples using **Clone Protocols**.

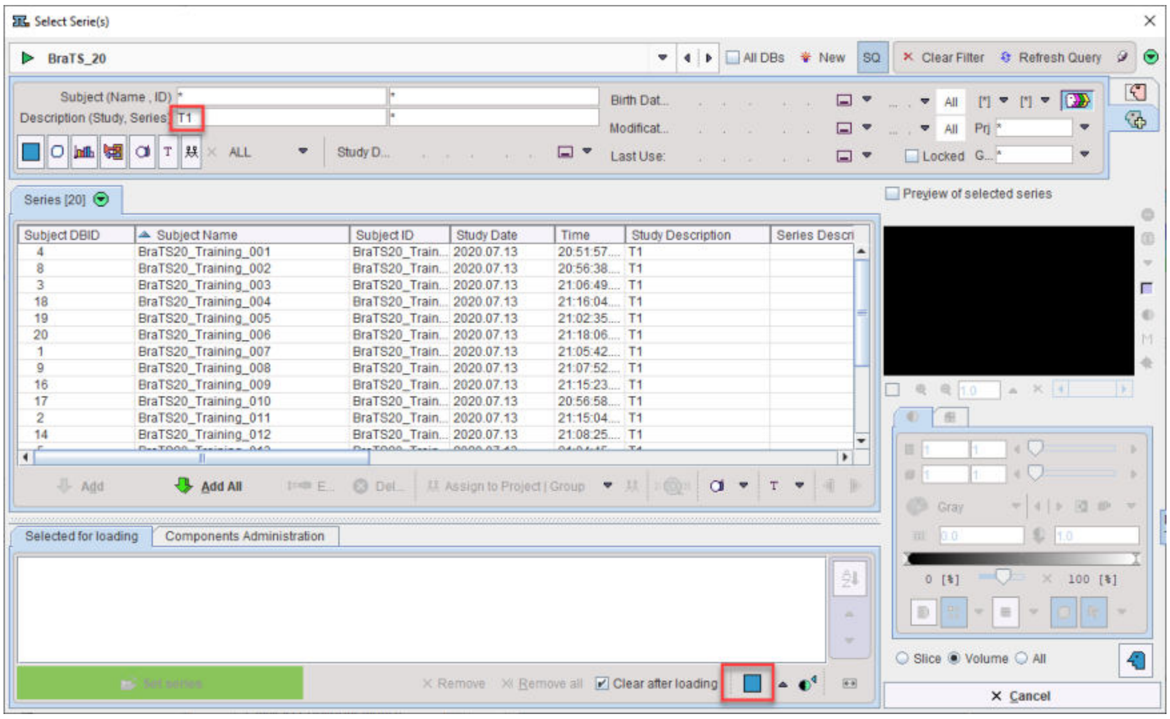


test-for-docu-1 (BraTS20_Training_001) 2021-10-19 <4/4/2124/1.T1/BraTS_20>

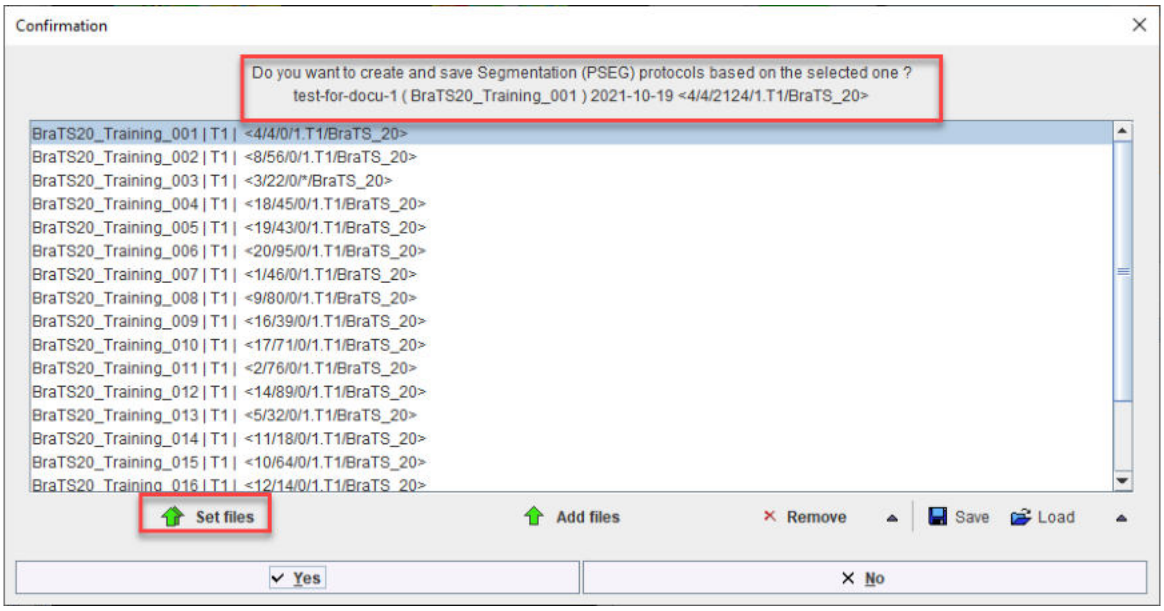
Results: ☑ Save Segments ☑ Save VOIs ☑ Save Statistics ☑ Aggregate Statistics

☑ Close after: ○ Start ● End

Clone Protocols opens a dialog window in which a list of input samples should be provided using **Set Files**. In the case of AI segmentation tasks that require multiple input series, the reference for the association list should be selected. PMOD will automatically retrieve the other associated series for processing. The flat view for the database and advanced database queries can be useful to select all input series for batch processing efficiently.

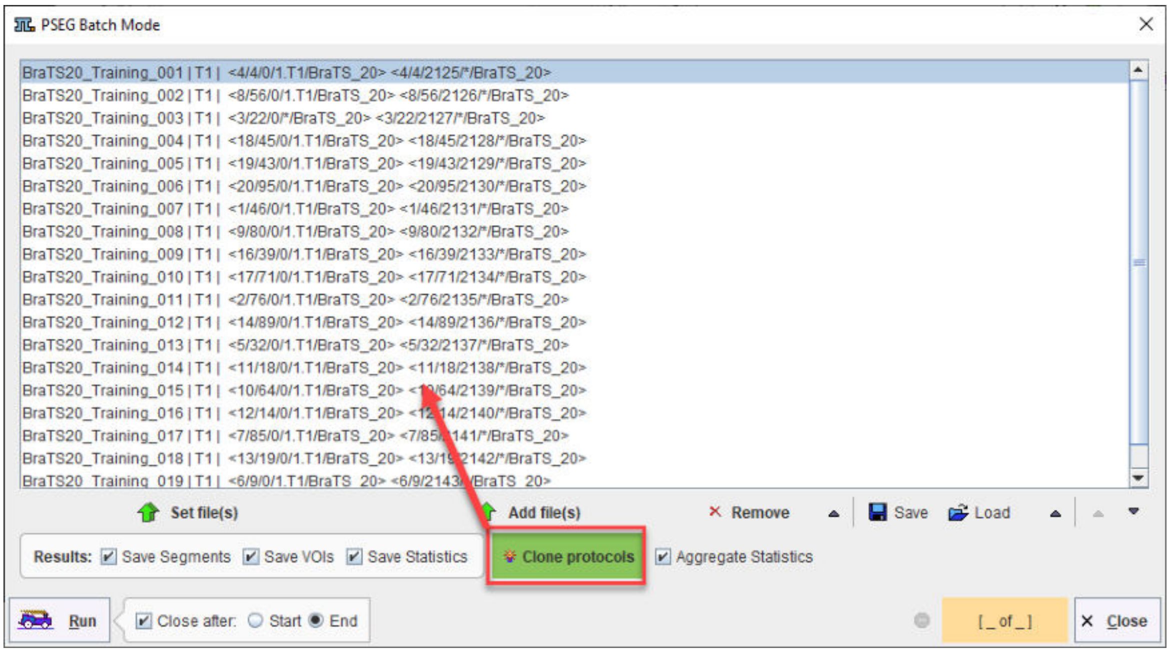


Subject DBID	Subject Name	Subject ID	Study Date	Time	Study Description	Series Descri
4	BraTS20_Training_001	BraTS20_Train...	2020.07.13	20:51:57....	T1	
8	BraTS20_Training_002	BraTS20_Train...	2020.07.13	20:56:38....	T1	
3	BraTS20_Training_003	BraTS20_Train...	2020.07.13	21:06:49....	T1	
18	BraTS20_Training_004	BraTS20_Train...	2020.07.13	21:16:04....	T1	
19	BraTS20_Training_005	BraTS20_Train...	2020.07.13	21:02:35....	T1	
20	BraTS20_Training_006	BraTS20_Train...	2020.07.13	21:18:06....	T1	
1	BraTS20_Training_007	BraTS20_Train...	2020.07.13	21:05:42....	T1	
9	BraTS20_Training_008	BraTS20_Train...	2020.07.13	21:07:52....	T1	
16	BraTS20_Training_009	BraTS20_Train...	2020.07.13	21:15:23....	T1	
17	BraTS20_Training_010	BraTS20_Train...	2020.07.13	20:56:58....	T1	
2	BraTS20_Training_011	BraTS20_Train...	2020.07.13	21:15:04....	T1	
14	BraTS20_Training_012	BraTS20_Train...	2020.07.13	21:08:25....	T1	



Do you want to create and save Segmentation (PSEG) protocols based on the selected one ?
test-for-docu-1 (BraTS20_Training_001) 2021-10-19 <4/4/2124/1.T1/BraTS_20>

```
BraTS20_Training_001 | T1 | <4/4/0/1.T1/BraTS_20>
BraTS20_Training_002 | T1 | <8/56/0/1.T1/BraTS_20>
BraTS20_Training_003 | T1 | <3/22/0/*/BraTS_20>
BraTS20_Training_004 | T1 | <18/45/0/1.T1/BraTS_20>
BraTS20_Training_005 | T1 | <19/43/0/1.T1/BraTS_20>
BraTS20_Training_006 | T1 | <20/95/0/1.T1/BraTS_20>
BraTS20_Training_007 | T1 | <1/46/0/1.T1/BraTS_20>
BraTS20_Training_008 | T1 | <9/80/0/1.T1/BraTS_20>
BraTS20_Training_009 | T1 | <16/39/0/1.T1/BraTS_20>
BraTS20_Training_010 | T1 | <17/71/0/1.T1/BraTS_20>
BraTS20_Training_011 | T1 | <2/76/0/1.T1/BraTS_20>
BraTS20_Training_012 | T1 | <14/89/0/1.T1/BraTS_20>
BraTS20_Training_013 | T1 | <5/32/0/1.T1/BraTS_20>
BraTS20_Training_014 | T1 | <11/18/0/1.T1/BraTS_20>
BraTS20_Training_015 | T1 | <10/64/0/1.T1/BraTS_20>
BraTS20_Training_016 | T1 | <12/14/0/1.T1/BraTS_20>
```



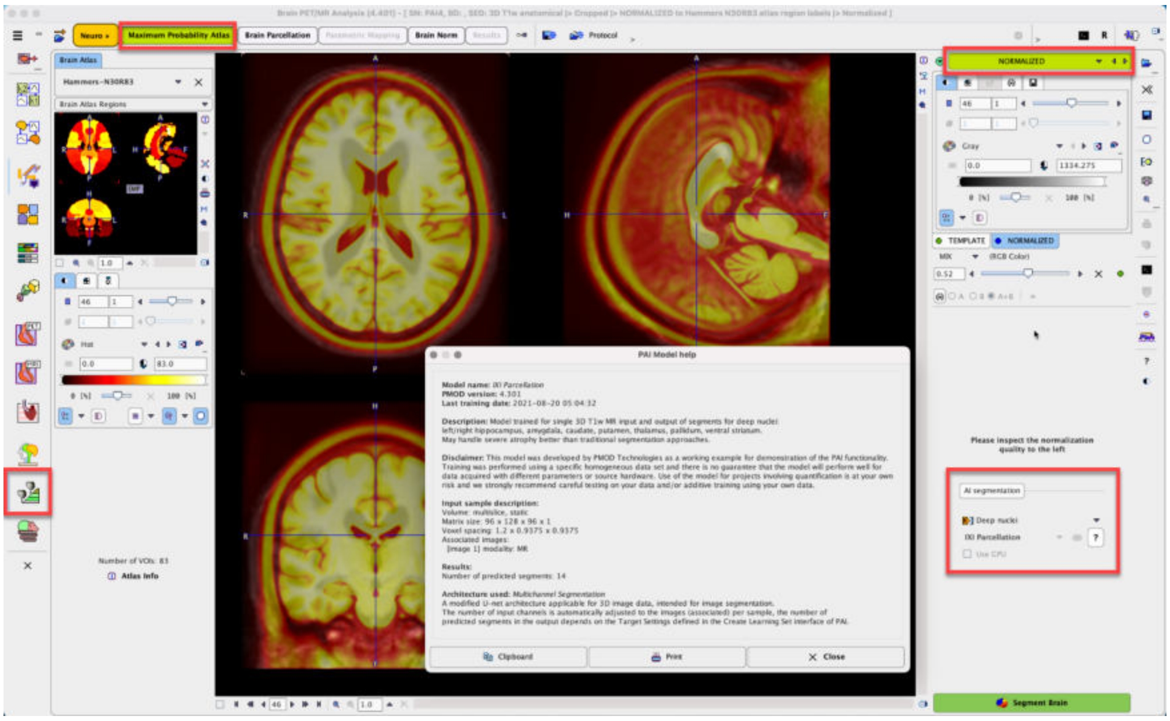
```
BraTS20_Training_001 | T1 | <4/4/0/1.T1/BraTS_20> <4/4/2125/*/BraTS_20>
BraTS20_Training_002 | T1 | <8/56/0/1.T1/BraTS_20> <8/56/2126/*/BraTS_20>
BraTS20_Training_003 | T1 | <3/22/0/*/BraTS_20> <3/22/2127/*/BraTS_20>
BraTS20_Training_004 | T1 | <18/45/0/1.T1/BraTS_20> <18/45/2128/*/BraTS_20>
BraTS20_Training_005 | T1 | <19/43/0/1.T1/BraTS_20> <19/43/2129/*/BraTS_20>
BraTS20_Training_006 | T1 | <20/95/0/1.T1/BraTS_20> <20/95/2130/*/BraTS_20>
BraTS20_Training_007 | T1 | <1/46/0/1.T1/BraTS_20> <1/46/2131/*/BraTS_20>
BraTS20_Training_008 | T1 | <9/80/0/1.T1/BraTS_20> <9/80/2132/*/BraTS_20>
BraTS20_Training_009 | T1 | <16/39/0/1.T1/BraTS_20> <16/39/2133/*/BraTS_20>
BraTS20_Training_010 | T1 | <17/71/0/1.T1/BraTS_20> <17/71/2134/*/BraTS_20>
BraTS20_Training_011 | T1 | <2/76/0/1.T1/BraTS_20> <2/76/2135/*/BraTS_20>
BraTS20_Training_012 | T1 | <14/89/0/1.T1/BraTS_20> <14/89/2136/*/BraTS_20>
BraTS20_Training_013 | T1 | <5/32/0/1.T1/BraTS_20> <5/32/2137/*/BraTS_20>
BraTS20_Training_014 | T1 | <11/18/0/1.T1/BraTS_20> <11/18/2138/*/BraTS_20>
BraTS20_Training_015 | T1 | <10/64/0/1.T1/BraTS_20> <10/64/2139/*/BraTS_20>
BraTS20_Training_016 | T1 | <12/14/0/1.T1/BraTS_20> <12/14/2140/*/BraTS_20>
BraTS20_Training_017 | T1 | <7/85/0/1.T1/BraTS_20> <7/85/2141/*/BraTS_20>
BraTS20_Training_018 | T1 | <13/19/0/1.T1/BraTS_20> <13/19/2142/*/BraTS_20>
BraTS20_Training_019 | T1 | <6/9/0/1.T1/BraTS_20> <6/9/2143/*/BraTS_20>
```

Results: [✓] Save Segments [✓] Save VOIs [✓] Save Statistics | Clone protocols | [✓] Aggregate Statistics

Once the desired list of protocols has been loaded or generated through cloning, batch processing can be launched using the **Run** button. Results will be saved in the same location as the input data - we strongly recommend using the database.

5.4 In the PNEURO tool

A neural network is available in the Neuro tool for replacement of the deep nuclei segments during segmentation of 3D T1-weighted MR images:



Model name: *IXI Parcellation*
PMOD version: 4.301
Last training date: 2021-08-20 05:04:32

Description: Model trained for single 3D T1w MR input and output of segments for deep nuclei:
left/right hippocampus, amygdala, caudate, putamen, thalamus, pallidum, ventral striatum.
May handle severe atrophy better than traditional segmentation approaches.

Disclaimer: This model was developed by PMOD Technologies as a working example for demonstration of the PAI functionality. Training was performed using a specific homogeneous data set and there is no guarantee that the model will perform well for data acquired with different parameters or source hardware. Use of the model for projects involving quantification is at your own risk and we strongly recommend careful testing on your data and/or additive training using your own data.

Input sample description:
Volume: multislice, static
Matrix size: 96 x 128 x 96 x 1
Voxel spacing: 1.2 x 0.9375 x 0.9375
Associated images:
[image 1] modality: MR

Results:
Number of predicted segments: 14

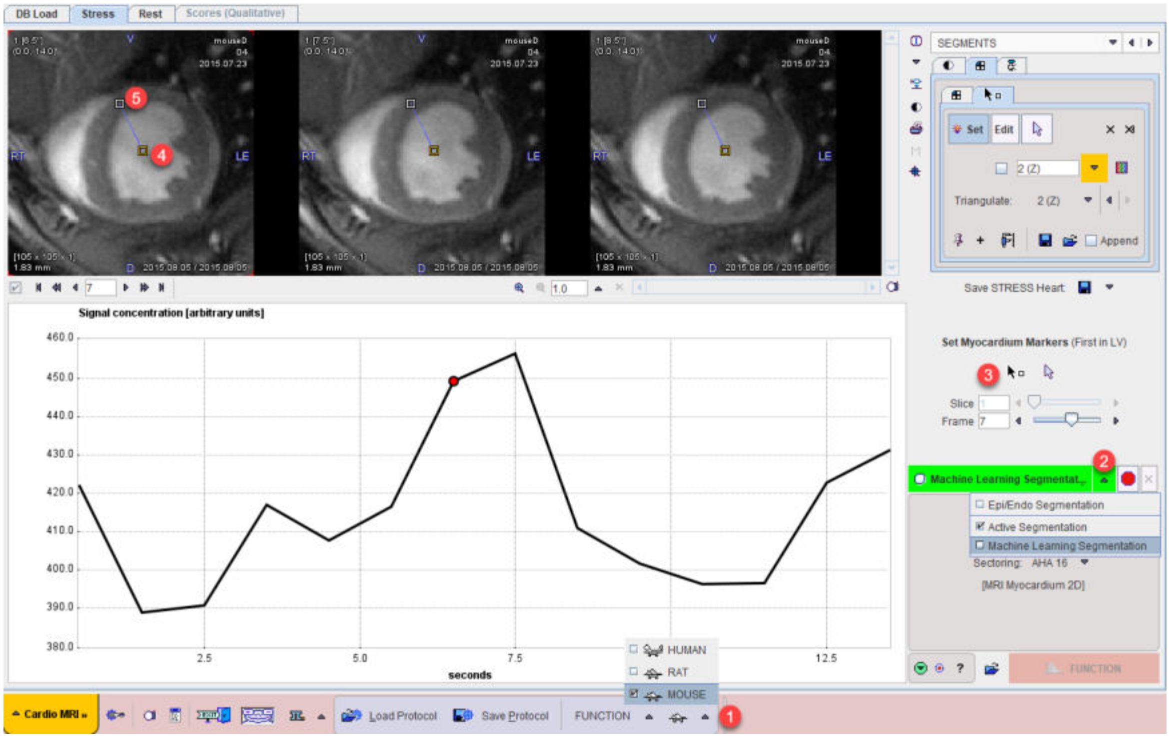
Architecture used: *Multichannel Segmentation*
A modified U-net architecture applicable for 3D image data, intended for image segmentation.
The number of input channels is automatically adjusted to the images (associated) per sample, the number of predicted segments in the output depends on the Target Settings defined in the Create Learning Set interface of PAI.

Please refer to the Neuro tool (PNEURO) documentation for more information.

5.5 In the PCARDM tool

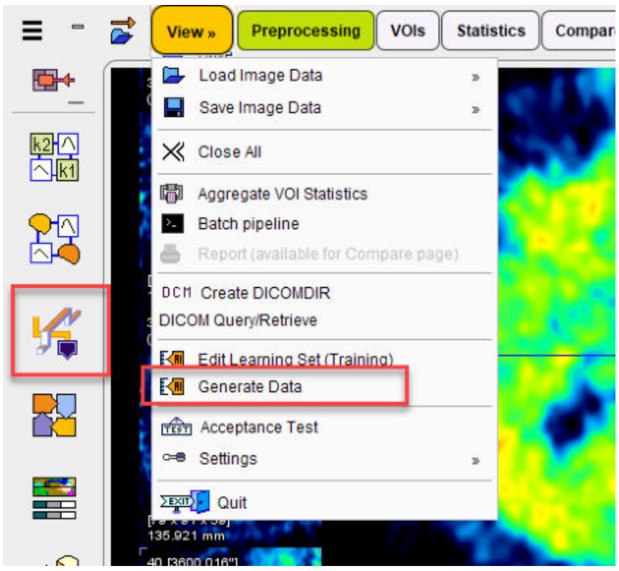
A neural network is available in the Cardiac MR tool (PCARDM) for segmentation of the left ventricle epi/endo contours for left ventricular function analysis. Models are available for human and mouse cardiac MR data.

Please refer to the PCARDM documentation for a detailed description of the workflow.

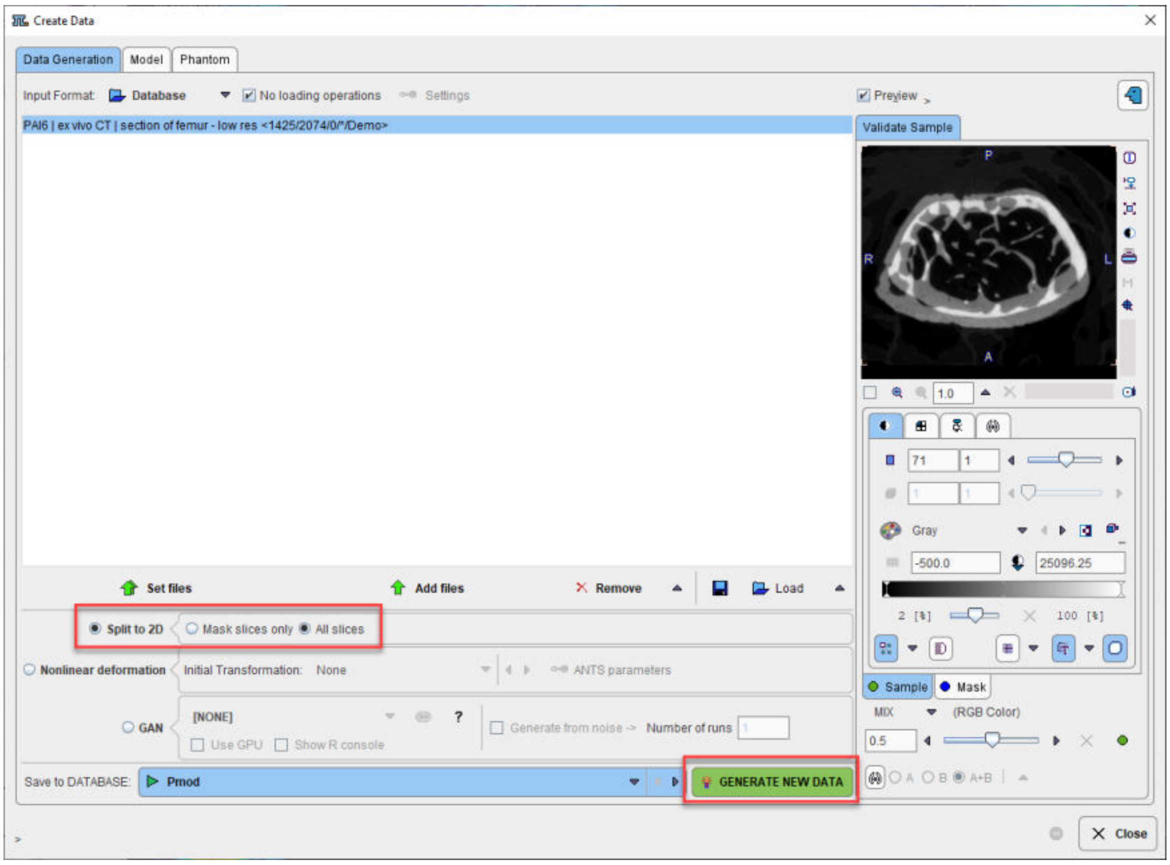


6 Data generation

A number of methods to generate additional data for model training are available in the Generate Data tool. It is available via the main tool menu in the View tool:



The most common methods are available on the main Data Generation tab:



Input files can be specified using the typical **Set Files** interface.

Split to 2D is useful to prepare individual 2D slices from a 3D dataset for training of a model that should work slicewise. This method can be tested and understood using the PAI6 mouse microCT section of femur example in our Demo database. The 142 slice low-resolution series can be taken from the Demo database and split into 142 individual 2D series into the Pmod database, ready to start a Learning Set. If a reference segmentation result (SEGMENT mask) is available and Associated with the input series, pairs of Associated image/mask will be generated.

Nonlinear deformation takes advantage of the iterative ANTS elastic matching method to save all iterations between two input images.

GAN uses Generative Adversarial Networks to generate entirely novel image data. For example, this may be an image representing typical cardiac MR function data, starting from automatically generated noise input images.

We strongly recommend the use of our Database functionality for input/output to take advantage of the Association functionality and create data that can be rapidly incorporated into Learning Sets.

4D data can also be generated using a geometric model and simulated data through the Model and Phantom tabs. Please see the documentation for our Geometric Models tool (PGEM), section Generation of Dynamic PET Phantom Images, for more information and contact us for further assistance.

7 *Case Studies - Application of PAI*

PAI and the several neural network architectures included in the resources have been tested using a range of case studies. These are described here.

7.1 Rat Brain Dopaminergic PET

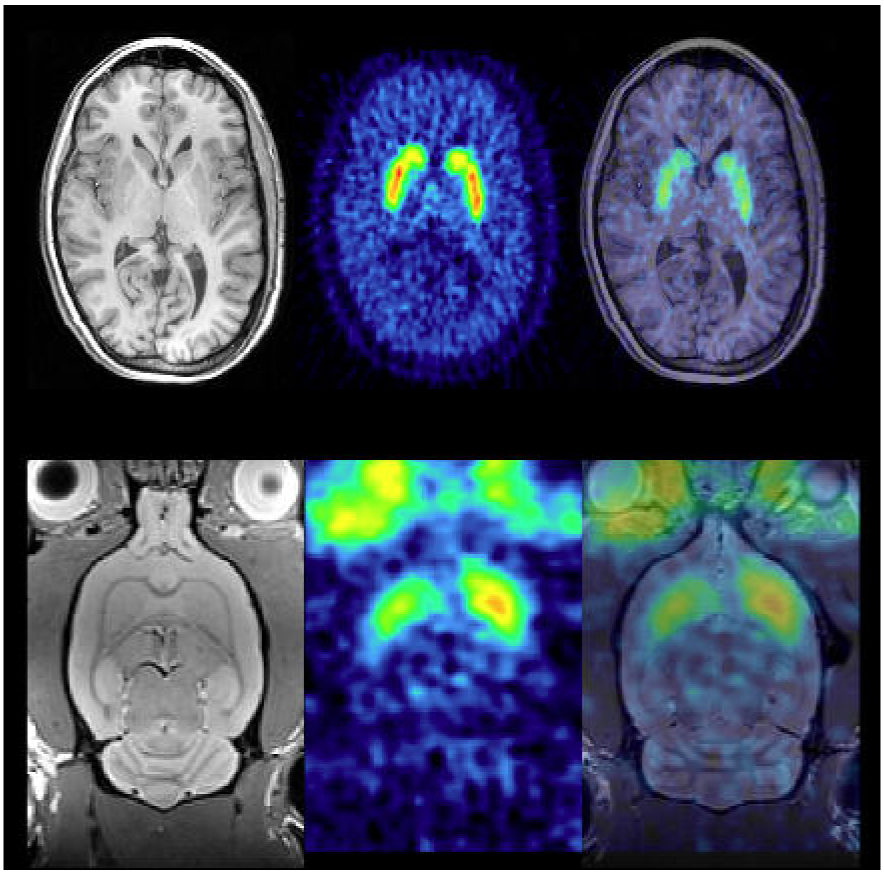
The **Multichannel Segmentation** architecture is designed to be used in new applications. It was initially tested for the 4 input series, 3 segment output MICCAI BraTS example case, and was successfully reapplied for segmentation of striatum and cerebellum in rat brain dopaminergic PET. The process of preparing the data to reapply the architecture, the training, and the evaluation of the outcome is described in this case study.

Example data to test the Rat Brain Dopaminergic PET model is available in our Demo database (Subject PAI5). The data used for the entire case study was kindly provided by Prof. Kristina Herfert at the Werner Siemens Imaging Center at the University of Tuebingen in Germany.

*To try the model for yourself we recommend use of the Segment tool (PSEG). Select the **early average** series as Input series. The **late average** series will be detected automatically according to the Association mechanism. The model selection must be Rat Brain Dopaminergic PET (Multichannel Segmentation). The model was trained with 3D data so no Split Slices/Frames is available or required. The resulting VOIs can be saved and used to extract TACs from the **Inveon dynamic PET** series (e.g. standalone analysis in View, or multimodal analysis combined with **t2_tse3d anatomical reference** MR in Fusion). Note that the PET data has already been coregistered to the MR including reslicing. The data used in this case study was all from Siemens Inveon PET and for tracers labeled with C-11 - the performance of the model may vary for data from different hardware and/or with different tracers.*

7.1.1 Sample Preparation

PET imaging of the dopaminergic system results in images with high tracer uptake in the basal ganglia:

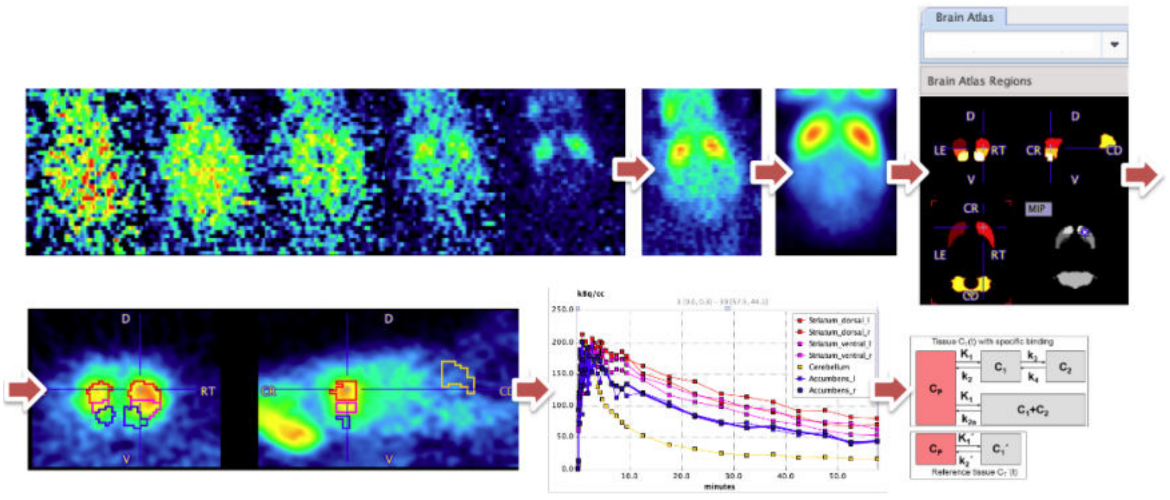


Fully quantitative analysis of PET with tracers for targets in the dopaminergic system typically uses time-activity curves from the striatum (in small animals; caudate and putamen in humans) and cerebellum (reference regions devoid of dopaminergic neurons/synapses). Dynamic PET studies typically cover a time range from 0-60 minutes after intravenous tracer injection.

In both small animals and humans, brain VOI atlases and tracer-specific templates such as those available in PMOD have long been used to achieve reproducible analysis and facilitate batch processing. The application of VOI atlases and templates used depend on the availability of anatomical imaging series to complement the PET data. In small animal dopaminergic PET studies it is common to have only the dynamic PET data. Matching to a PET template normally requires averaging in a time range that reveals specific binding and reduces image noise. Due to the limited tracer distribution researchers sometimes struggle to achieve a satisfactory result.

PMOD's **Rodent Brain analysis tool (PNROD)** provides a streamlined workflow for such analysis and with careful creation of a template image and selection of the averaging range works well for batch processing of rodent brain PET data. It was used to process 382 rat brain dynamic PET image series. The data comprised PET at a range of ages and with the tracers [^{11}C]-raclopride, [^{11}C]-methylphenidate and [^{11}C]-DTBZ. A template image specific to this study data was created from a subset of the data using standard functionality in **View** and **Fusion** tools.

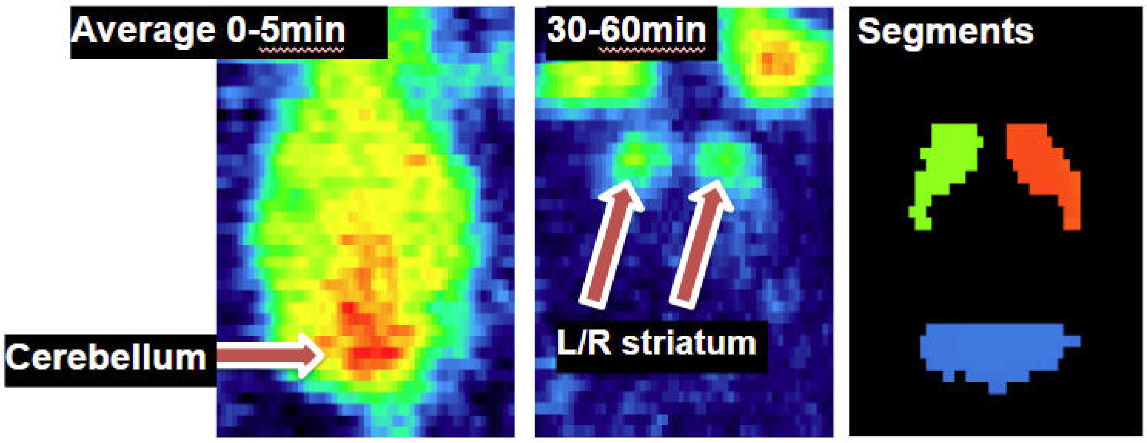
The traditional analysis workflow for this data is summarized below:



The dynamic PET data was averaged. This average was normalized to the template, allowing atlas VOIs for striatal regions and cerebellum to be created in the original PET image space. These VOIs were used to extract time-activity curves that were used for kinetic modeling. The Simplified Reference Tissue Model was used to calculate BP_{nd}.

PMOD's AI framework (PAI) offers an alternative approach to segmentation of such data. 382 input samples represents a reasonable number for training of an ML model, and the VOIs generated by PNROD represent gold-standard method reference segments.

The **Multichannel Segmentation** architecture presented a potential advantage over the traditional workflow in that multiple averages can be provided as input. The average used for PNROD processing was selected to provide a specific-binding signal in the striatum but also some remaining blood pool signal to represent a more general brain outline. The cerebellum is not well defined in this average. Therefore an early average image created from the first 5 minutes after tracer injection and a late average of 30-60 minutes after tracer injection were generated using **Pipeline Processing** and organized in a **Database**:



The VOI results from PNROD were converted into reference Segments using the **Mask By VOI Number** functionality available in the **View** tool and added to the **Database**.

The LATE and SEGMENT images were **Associated** with the EARLY image for each subject in the **Database**. The **Project** labels 1.Early and 2.Late were assigned to the EARLY and LATE images.

The **Database** was used to create a **Learning Set** for training with the **Multichannel Segmentation** architecture.

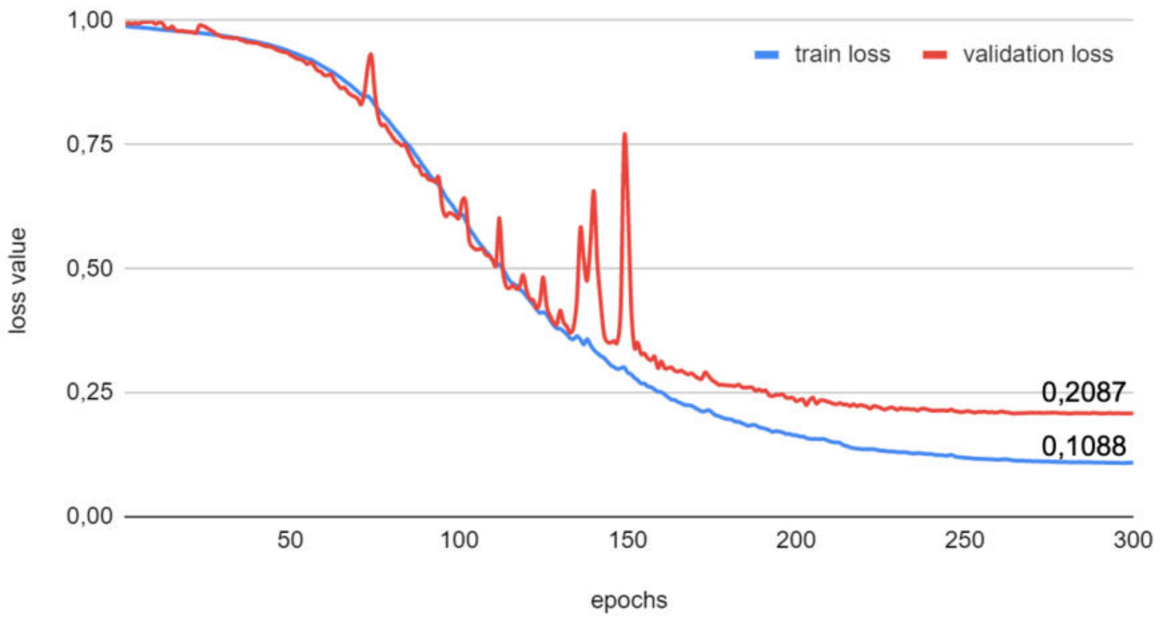
7.1.2 Training and Validation

The **Learning Set** was exported as an R workspace[62] and transferred to cloud computing infrastructure for training. PMOD can be installed on the virtual machine provided by the cloud computing provider and training launched using the R workspace.

The parameters used were:

- AWS p2.8xlarge, 8 GPU K80, memory 64 GB
- 382 samples (288 training, 72 validation, 22 testing)
- batch size 24
- Learning rate 0.005
- epochs 300

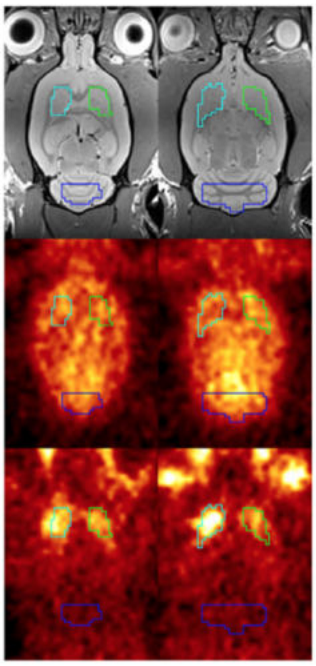
Training took 268 minutes at 0.07 seconds/sample. The loss values for training and validation were extracted from the **Manifest** and plotted:

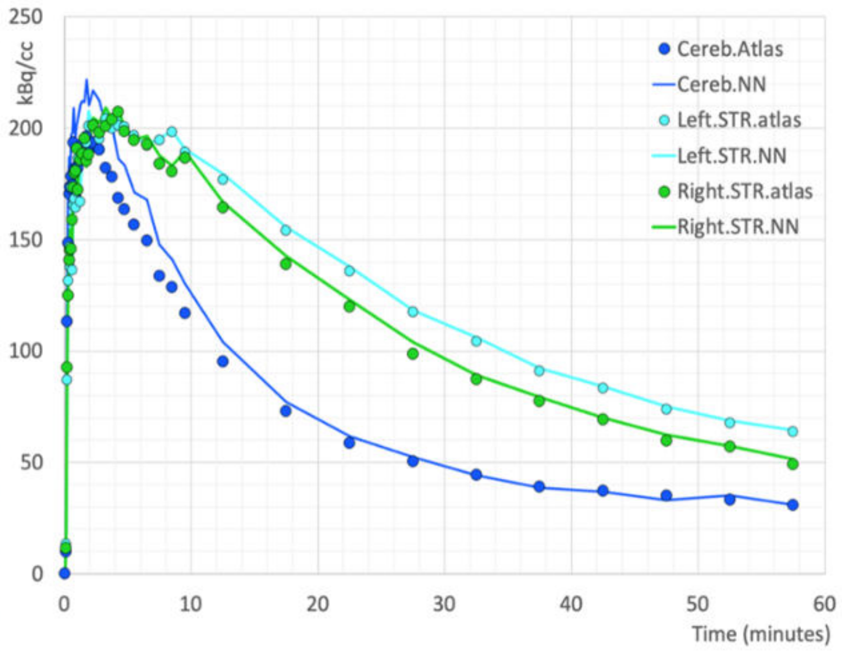


The plot illustrates how the loss value reduces with each epoch until a plateau is reached.

The **Learning Set**, **Weights** and **Manifest** were exported from the virtual machine and used to create a new model folder **Rat Brain Dopaminergic PET** in the weights subfolder of the **Multichannel Segmentation** folder in the PMOD installation */resources/pai* folder. After this **Deployment** it was possible to test the model performance of comparable rat brain [^{11}C]-methylphenidate data for which an anatomical MR image was available.

The resulting VOIs are shown below on the EARLY, LATE and anatomical reference MR images:





Time-activity curves were extracted from 8 dynamic PET series (4 RAC, 4 MP) using both sets of VOIs (AI-based, PNROD reference) and BPnd calculated using the SRTM model. The mean +/- SD BPnd using the AI-based VOIs was 1.29 +/- 0.67 and for PNROD VOIs 1.39 +/- 0.79.

Example data to test the Rat Brain Dopaminergic PET model is available in our Demo database (Subject PAI5). The data used for the entire case study was kindly provided by Prof. Kristina Herfert at the Werner Siemens Imaging Center at the University of Tuebingen in Germany.

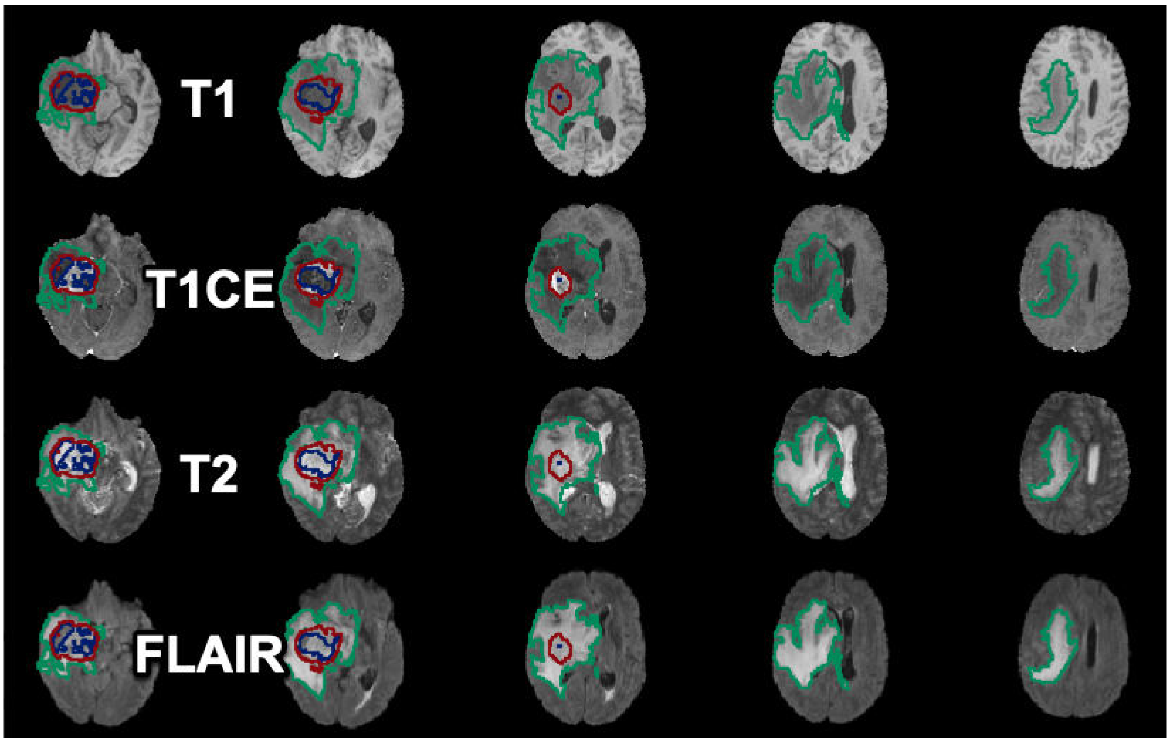
*To try the model for yourself we recommend use of the Segment tool (PSEG). Select the **early average** series as Input series. The **late average** series will be detected automatically according to the Association mechanism. The model selection must be Rat Brain Dopaminergic PET (Multichannel Segmentation). The model was trained with 3D data so no Split Slices/Frames is available or required. The resulting VOIs can be saved and used to extract TACs from the **Inveon dynamic PET** series (e.g. standalone analysis in View, or multimodal analysis combined with **t2_tse3d anatomical reference** MR in Fusion). Note that the PET data has already been coregistered to the MR including reslicing.The data used in this case study was all from Siemens Inveon PET and for tracers labeled with C-11 - the performance of the model may vary for data from different hardware and/or with different tracers.*

7.2 Brain Tumor Segmentation - MICCAI Challenge

PAI is provided with a neural network architecture designed for multichannel segmentation. This architecture expects one or more 3D input series and generates segmentation results with one or more segments/VOIs.

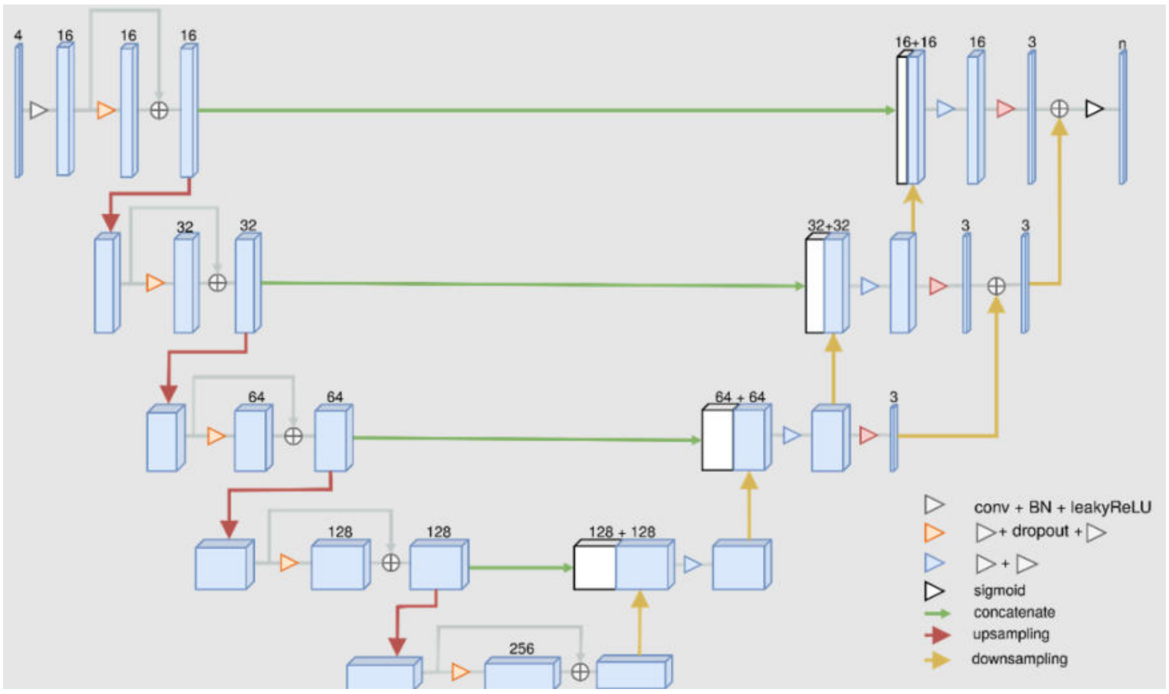
The initial application of this architecture was for a case called **Tumor Detection**. A trained model **BRATS Tumor Detection (Multichannel Segmentation)** is available in the **View** and **Segmentation** tools when PAI is licensed. It is based on the MICCAI Brain Tumor Segmentation (BraTS) Challenge: http://braintumorsegmentation.org. BraTS utilizes multi-institutional pre-operative

MRI scans and focuses on the segmentation of intrinsically heterogeneous (in appearance, shape, and histology) brain glioma tumors.

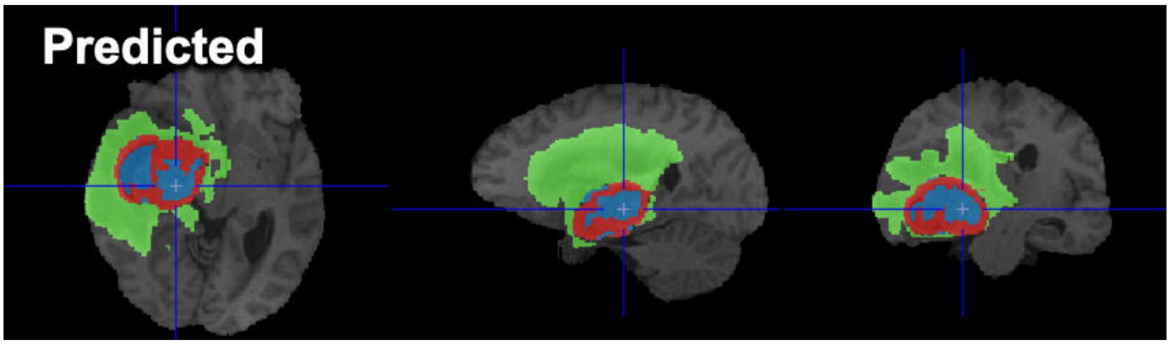


Training and testing of the **Tumor Detection** model in PAI was performed using the data from the [2020 BraTS Challenge]() containing 369 samples. Each sample consists of four MR images (native T1, post-Gd-contrast T1-weighted, T2 FLAIR, T2-weighted) and one image containing three reference segments as label numbers.

The **Multichannel Segmentation** architecture is a modified version of the convolutional neural network U-Net:



The output of the **Tumor Detection** model is a label image with three segments (label 1: non-enhancing tumor; label 2: peritumoral edema; label 4: Gd-enhancing tumor).



Example data to test the BRATS Tumor Segmentation model is available in our Demo database (Subject PAI1). This example is one of the subjects available in the data for the 2020 BraTS Challenge .

*To try the model for yourself we recommend use of the Segment tool (PSEG). Select the **T1** series as Input series. The **T1CE, FLAIR & T2** series will be detected automatically according to the Association mechanism. The model selection must be BRATS tumor segmentation (Multichannel Segmentation). The model was trained with 3D data so no Split Slices/Frames is available or required. The data used in this case study was all from the BraTS Challenge and is skull-stripped 3D MR - the performance of the model may vary for data from different hardware and/or with different contrast/sequences/pre-processing.*

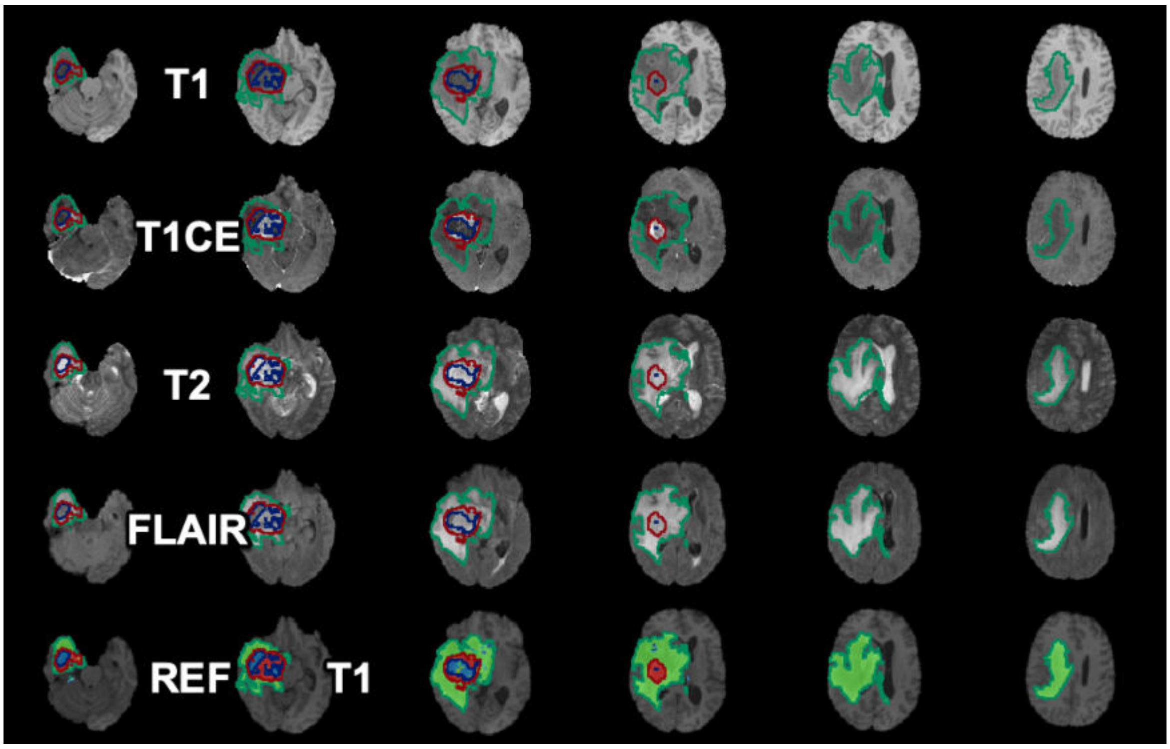
BraTS Reference:

Bakas, Spyridon & Reyes Jan. (2019). Identifying the Best Machine Learning Algorithms for Brain Tumor Segmentation, Progression Assessment, and Overall Survival Prediction in the BRATS Challenge. https://arxiv.org/pdf/1811.02629.pdf

7.2.1 Sample Preparation

Each sample from BraTS Challenge data consists of four MR images (native T1, post-Gd-contrast T1-weighted, T2 FLAIR, T2-weighted) and one image containing three reference segments as label numbers. These series were imported into a database and the five series per subject were associated. The reference segment series was labeled as SEGMENT in the Association interface.

As the data was already cropped and coregistered, no additional preprocessing was applied.



7.2.2 Training and Validation

A new Learning Set was prepared, containing the 369 available samples, configured for a segmentation task and using the Multichannel Segmentation architecture.

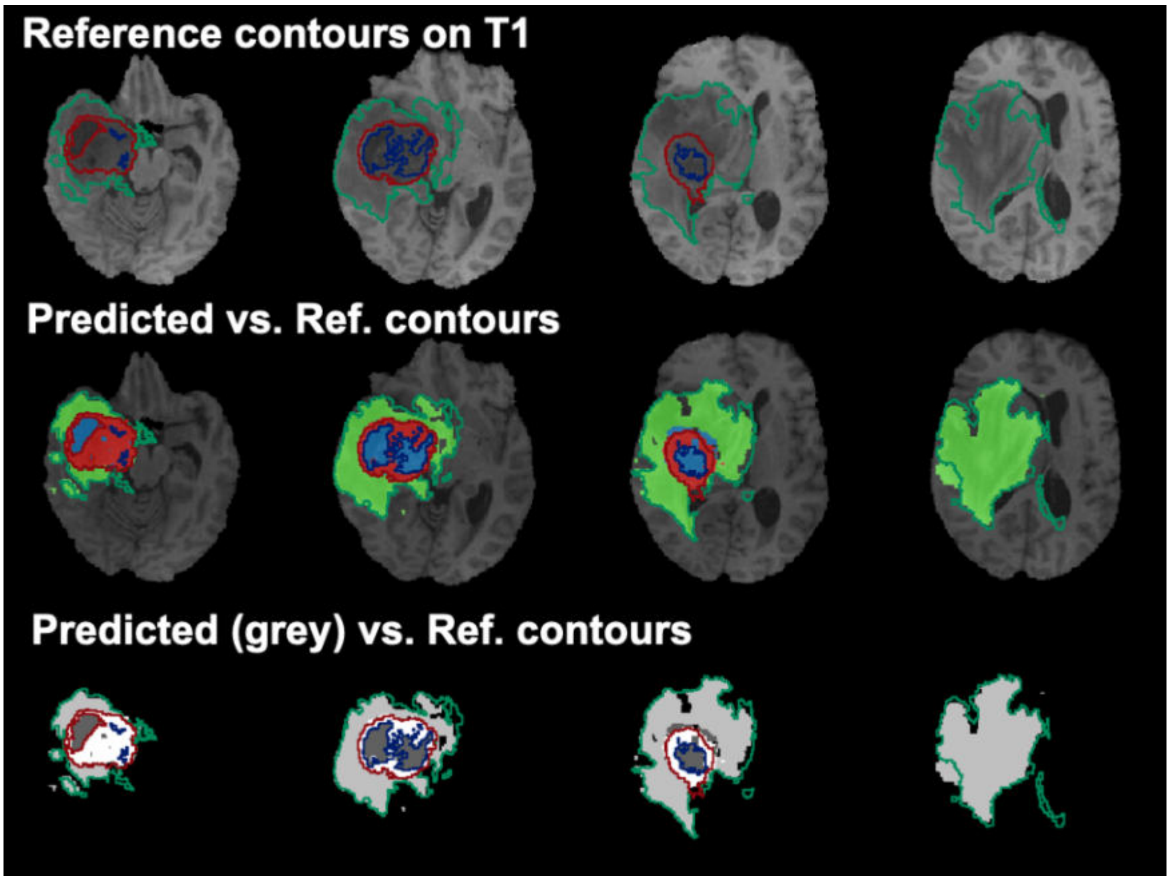
An R Workspace containing all samples was exported and used for training on commercial cloud computing infrastructure. Training was performed with the settings:

- AWS p3.8xlarge, 4 V100 GPU, 244 GB memory
- 369 samples (295 training, 74 validation)
- Crop to associated VOI
- 500 epochs
- batch size of 4
- Learning rate 0.0005

The final training set loss value was 0.2103 (1 - Dice coefficent) and the validation set loss value was 0.3531. Training took 810 minutes.

The weights and manifest resulting from the training were retrieved and used to add the training model to the /resources/pai folder for deployment on local installations.

An example of the results on one subject are illustrated below:


Reference contours on T1
Predicted vs. Ref. contours
Predicted (grey) vs. Ref. contours

Example data to test the BRATS Tumor Segmentation model is available in our Demo database (Subject PAI1). This example is one of the subjects available in the data for the 2020 BraTS Challenge .

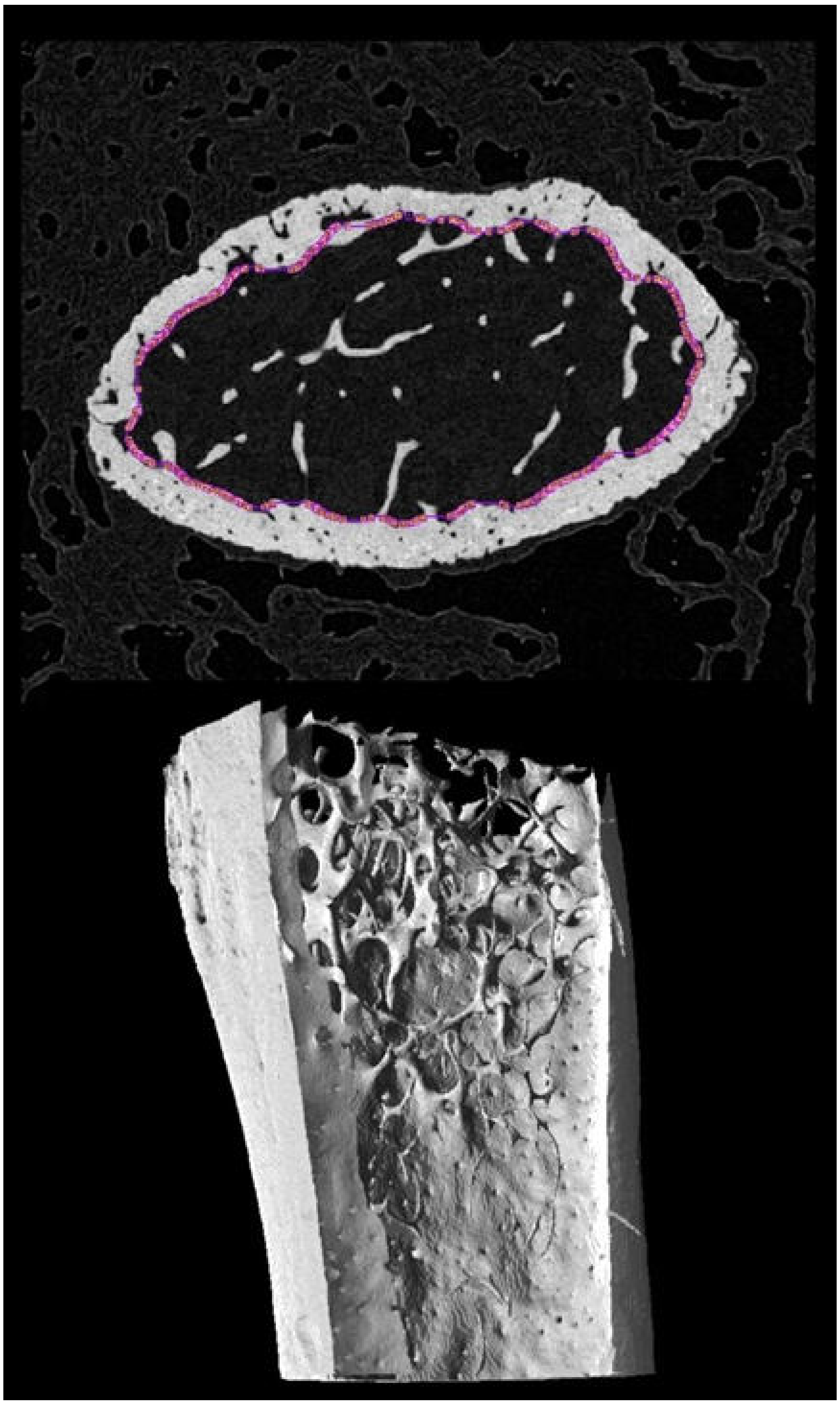
*To try the model for yourself we recommend use of the Segment tool (PSEG). Select the **T1** series as Input series. The **T1CE, FLAIR & T2** series will be detected automatically according to the Association mechanism. The model selection must be BRATS tumor segmentation (Multichannel Segmentation). The model was trained with 3D data so no Split Slices/Frames is available or required. The data used in this case study was all from the BraTS Challenge and is skull-stripped 3D MR - the performance of the model may vary for data from different hardware and/or with different contrast/sequences/pre-processing.*

7.3 Mouse Bone Trabecular Segmentation

A common application in bone research is analysis of bone remodeling in the trabecular of distal femur and proximal tibia in mice. MicroCT imaging of ex vivo femur and tibia is used to generate 3D datasets with pixel size in the range of 5 um. Analysis of trabecular bone is then performed in the metaphysis and requires an initial segmentation separating trabecular bone from cortical bone and the growth plate. Simple cropping of the 3D dataset is usually sufficient to avoid the growth plate and to define the division between metaphysis and diaphysis. Some researchers additionally distinguish between primary and secondary spongiosa.

Definition of the trabecular region to be analysed, including as much material as possible but without including cortical bone, can be laborious and even automated solutions need tedious clean up. AI-based segmentation could offer a rapid alternative and provide the starting “tissue volume” for further trabecular analysis.

We used two distal femur and eight proximal tibia microCT datasets to train a uNet architecture using two approaches. In a strategy to increase the amount of data available for training, we treated each 2D axial slice of the femur/tibia as a separate sample. One femur and all eight tibia datasets were provided with reference segments created in the Bruker microCT software CTan.



Example data to test the Mouse bone trabecular model is available in our Demo database (Subject PAI6). The data used for the case study was kindly provided by Dr. Phil Salmon from Bruker microCT.

To try the model for yourself we recommend use of the Segment tool (PSEG) or segmentation interface in View. Either the low res or full res series may be used for testing. The low res series has been downsampled but includes more slices in the 3D volume. The model selection must be

Trabecular ML (uNET Segmentation). The model was trained with 2D data so Split Slices is required. No further cropping of the demo data is required. The data used in this case study was from Bruker microCT - the performance of the model may vary for data from different hardware and/or with different reconstruction/pre-processing.

7.3.1 Sample Preparation

In the first approach - rapid model - a subset of 70 slices from two femurs were imported into a database along with their reference segments. The microCT and segment series were associated using automatic association. A learning set was prepared with all samples and segmentation using uNET segmentation architecture. 60 samples were assigned for training - to be split into 48 training and 12 validation - and 10 samples were labeled as Test samples for evaluation of the model.

In the second approach - detailed model - all slices from the one femur with full reference segments and seven of the eight tibia were imported into a database along with their reference segments. The microCT and segment series were associated using automatic association. In total 3509 2D samples were available for training. A learning set was prepared with all samples and segmentation using uNET segmentation architecture. 3509 samples were assigned for training - to be split into 2486 training and 622 validation - and 401 were labeled as Test samples for evaluation of the model.

7.3.2 Training and Validation

Rapid Model

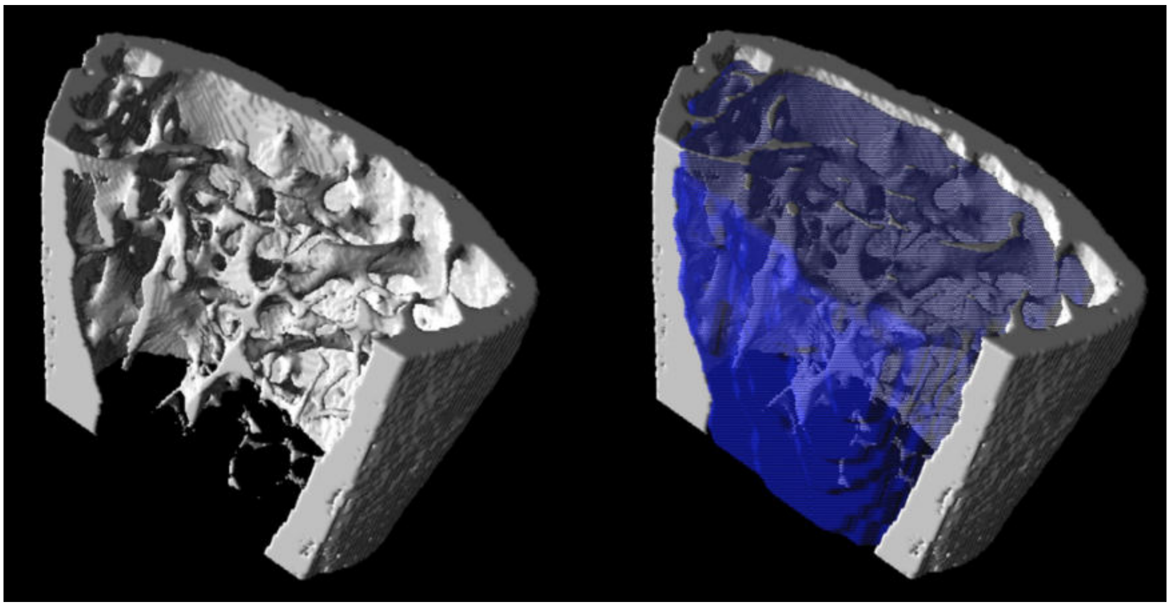
The original resolution of the samples was 5.5 and 2.8 um and for training all were resampled to 50 um. No additional cropping was applied and the image values were normalized using the z-score method. Training was performed as follows:

notebook computer with 8 core 2.3 GHz Intel i7 and 32 GB RAM using only CPU

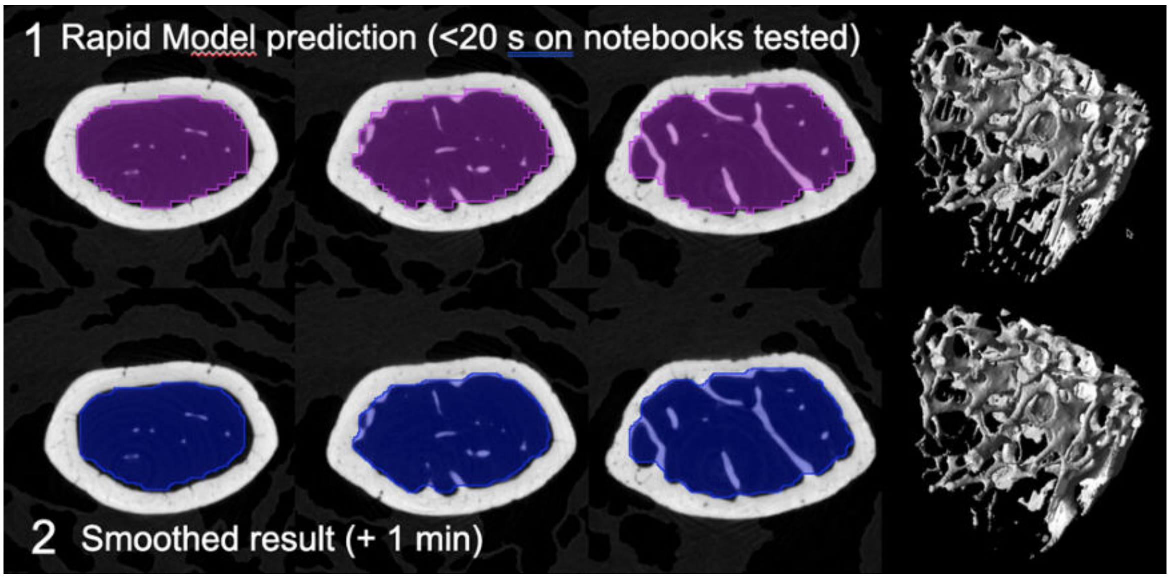
- 70 samples (48 training, 12 validation, 10 testing)
- batch size 12
- learning rate 0.0005
- 300 epochs

Training took 151 seconds. The training loss value was 0.0368 and the validation set loss value was 0.0388. Testing was performed on the 10 additional samples using the Evaluate Model functionality with a loss value of 0.1037.

The model was tested on the second femur dataset with reduced pixel size of 0.0112 mm, and on the eighth tibia dataset with native 0.005 mm pixel size. Both datasets were loaded as 3D volumes and prediction was run with the Split Slices option. The trabecular region was correctly segmented in < 20 seconds on a notebook similar to that used for training.

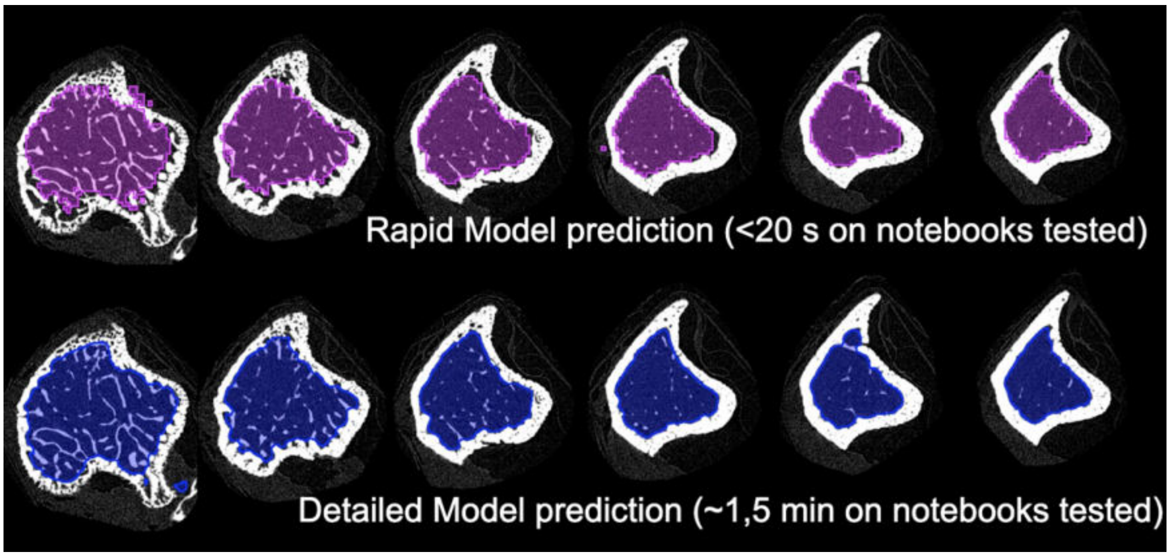


Due to training with resampling to 50 um the resulting segments displayed clear staircase artefacts. There was some overlap with cortical bone. A segment smoothing approach was investigated to remove the staircase artefacts. The segment was transferred from the Segment tool to View and smoothed with a Gaussian filter at 0.05 x 0.05 x 0.05 mm FWHM. Automatic isocontouring at 50% (Max-min) threshold was used to create a new VOI.


1 Rapid Model prediction (<20 s on notebooks tested)
2 Smoothed result (+ 1 min)

Smoothing visually improved the VOI but some overlap with cortical bone remained.

On tibia segmentation also took < 20 seconds on the notebook tested. Similar staircase artefacts and some overlap with cortical bone were present.


Rapid Model prediction (<20 s on notebooks tested)
Detailed Model prediction (~1,5 min on notebooks tested)

Detailed Model

The original resolution of the samples used was 5 and 5.5 um and for training all samples were resampled to 10 um. No additional cropping was applied and the image values were normalized using the z-score method. Training was performed as follows:

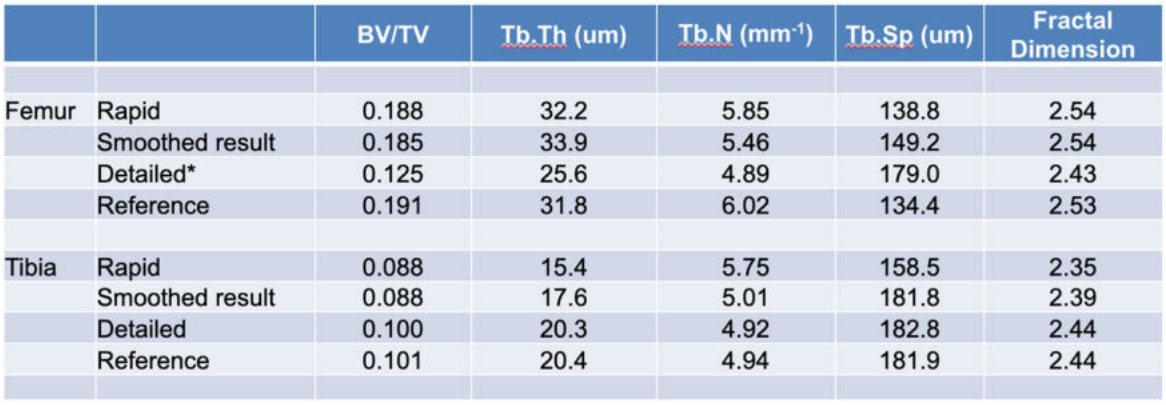
- AWS p2.8xlarge, 96 GB GPU, 488 GB RAM
- 3509 samples (2486 training, 622 validation, 401 testing)
- batch size 60
- learning rate 0.0005
- 10 epochs

Training took 600 seconds. The training loss value was 0.0546 and the validation set loss value was 0.0545. Testing was performed on all slices of the eighth tibia dataset using the Evaluate Model functionality with a loss value of 0.0580.

The trained model was tested on the same femur and tibia datasets as the rapid model. Both datasets were loaded as 3D volumes and prediction was run with the Split Slices option. Prediction on the tibia in native pixel size took approximately 90 seconds on the notebook tested. The contour closely followed the endosteum (inner tibial surface) with some artefacts present close to the growth plate. On femur several artefacts were present that resulted in inner holes in the trabecular VOI. Manual editing was used to close the holes and allow morphometric analysis.

Morphometric analysis

The segments generated by both models and smoothed versions of the rapid model segments were used for further trabecular bone analysis. A single threshold of 2000 Hounsfield units (HU) was used to segment trabecular bone voxels within the model segments. The parameters BV/TV (bone / tissue volume), trabecular thickness, trabecular number, trabecular separation, and fractal dimension were calculated using the volume and surface area statistics according to the plate model.



		BV/TV	Tb.Th (um)	Tb.N (mm^{-1})	Tb.Sp (um)	Fractal Dimension
Femur	Rapid	0.188	32.2	5.85	138.8	2.54
	Smoothed result	0.185	33.9	5.46	149.2	2.54
	Detailed*	0.125	25.6	4.89	179.0	2.43
	Reference	0.191	31.8	6.02	134.4	2.53
Tibia	Rapid	0.088	15.4	5.75	158.5	2.35
	Smoothed result	0.088	17.6	5.01	181.8	2.39
	Detailed	0.100	20.3	4.92	182.8	2.44
	Reference	0.101	20.4	4.94	181.9	2.44

Morphological analysis revealed that the rapid model yielded comparable results to the reference segments for the femur tested, but resulted in marked deviation from the reference segments for the tibia tested. Smoothing of the segments generated by the rapid model did not universally improve the result. The detailed model resulted in near replication of the reference results for tibia, but did not produce good results for the femur. We believe that this was likely a result of overtraining the model towards tibia (seven tibia vs. one femur in training set).

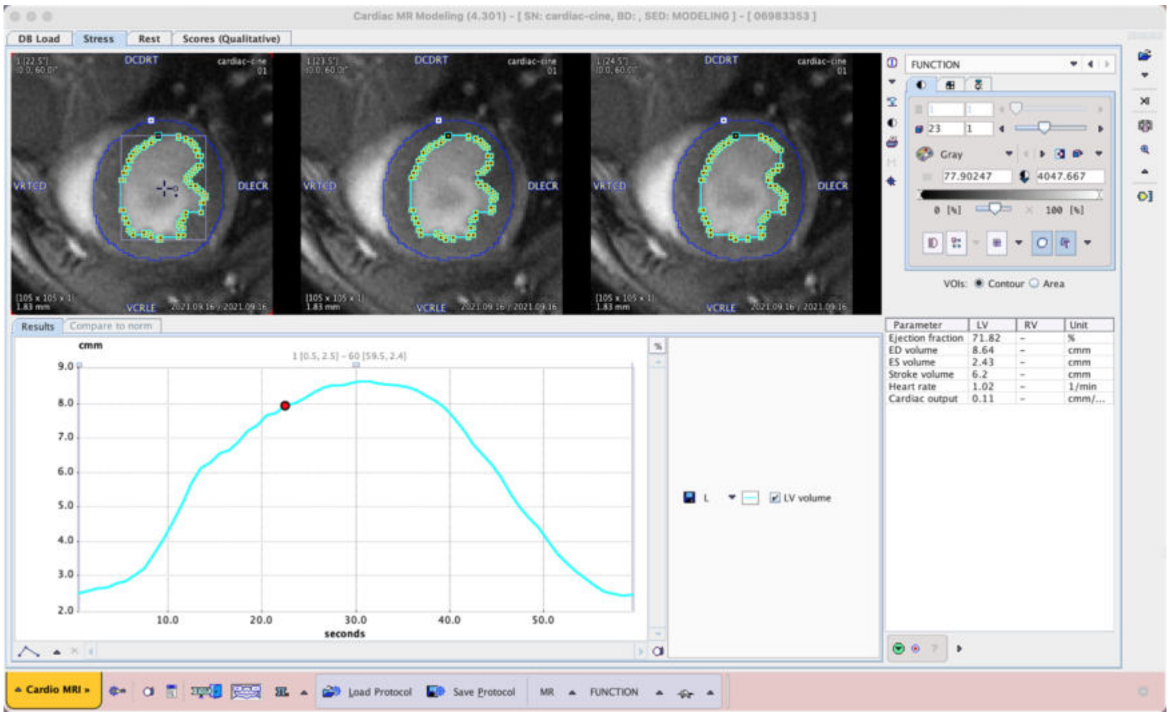
Use of either model for study-level trabecular analysis appears feasible. Erosion of the tissue volume segment resulting from PAI by several voxels in 2D mode would likely be beneficial to further ensure that cortical bone is excluded.

Example data to test the Mouse bone trabecular model is available in our Demo database (Subject PAI6). The data used for the case study was kindly provided by Dr. Phil Salmon from Bruker microCT.

To try the model for yourself we recommend use of the Segment tool (PSEG) or segmentation interface in View. Either the low res or full res series may be used for testing. The low res series has been downsampled but includes more slices in the 3D volume. The model selection must be Trabecular ML (uNET Segmentation). The model was trained with 2D data so Split Slices is required. No further cropping of the demo data is required. The data used in this case study was from Bruker microCT - the performance of the model may vary for data from different hardware and/or with different reconstruction/pre-processing.

7.4 Mouse & Human Cardiac MR Cine Left Ventricle Segmentation

Cardiac MR cine acquisitions are widely used to assess left ventricular function, using the evolution of left ventricle volume over the cardiac cycle to calculate parameters such as systolic/diastolic volume, ejection fraction and cardiac output (if frame timing is provided corresponding to the actual duration of the cardiac cycle).



Parameter	LV	RV	Unit
Ejection fraction	71.82	-	%
ED volume	8.64	-	cmm
ES volume	2.43	-	cmm
Stroke volume	6.2	-	cmm
Heart rate	1.02	-	1/min
Cardiac output	0.11	-	cmm/...

Manual delineation of the left ventricle is subjective and can be time consuming in case of many cine frames and/or multislice acquisitions. Automatic methods may be limited in success due to variable acquisition parameters, image contrast and noise, as well as artefacts due to swirling blood in the ventricle.

We hypothesised that AI-based segmentation could provide a useful alternative to these methods.

Example data to test the MRI Myocardium 2D and MRI Human Myocardium models is available in our Demo database (Subjects PAI2 and PAI3). The data used for the case study was provided by Bruker colleagues and in a private collaboration.

*To try the models for yourself we recommend use of the **Cardiac MR Modeling tool** (**PCARDM**) FUNCTION workflow. See the specific documentation for PCARDM. The model selection is determined by the species selection in PCARDM. The models were trained with 2D data so Split Slices and Split Frames are required. The recommended **Heart Box** for example PAI2 is 10 x 10 mm, and for PAI3 100 x 100 mm. The data used in these cases was for a limited range of MR sequences - the performance of the model may vary for data from different hardware and/or with different contrast/pre-processing.*

7.4.1 Sample Preparation

Mouse data

Mouse bright-blood cardiac cine data was collected from internal sources and contained conventional prospectively-gated as well as Bruker IntraGate and IntraGate UTE (ultrashort echo time) time. The mixture of source data resulted in a range of pixel size, number of cardiac cycle frames, as well as blood contrast/artefacts. All series were split by frames and slices to be treated as individual 2D data. Each series was rotated such that the right ventricle was displayed on the left of screen and cropped using a 10 x 10 mm box. Reference segments for myocardial epi- and endo- contours were created manually using the paintbrush VOI tools and masking functionality. Images and reference segments were associated in a database.

Human data

Human bright-blood cardiac cine data was shared in a private collaboration. 10 subjects were available.

The subjects had between 14-18 slice coverage of the myocardium and 25 frames of the cardiac cycle. These were split by frames/slices to be treated as individual 2D data. Reference segments for myocardial epi- and endo- contours were manually drawn using VOI tools and images associated with reference segments in a database.

7.4.2 Training and Validation

Mouse data

All samples were added to a learning set and the uNET segmentation architecture selected.

Prediction was tested on several additional examples. Preprocessing to rotate new data such that the right ventricle is on left of screen was required for successful prediction. Dark artefacts in the blood can lead to segmentation artefacts that require manual correction before cardiac function can be accurately assessed.

Human data

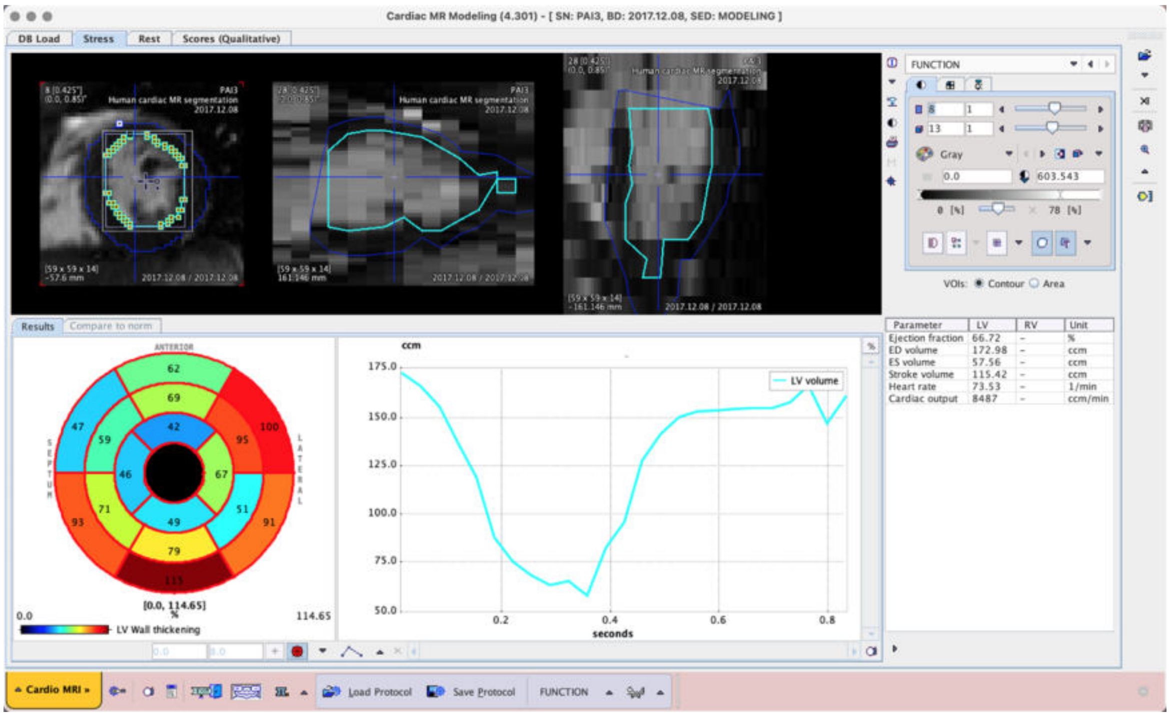
All samples were added to a learning set and the Multichannel Segmentation architecture selected.

Training was performed as follows:

- notebook Intel i7 8 x 2.3 GHz
- 32 GB memory
- no additional cropping
- 3225 samples (2580 training, 645 validation)
- 106 epochs (aborted)
- batch size 50
- learning rate 0.005

Training took 63 minutes.

Prediction could only be tested on data that was already used in training. Acceptable segmentation results and function analysis was achieved, but the model may be overtrained for this data. The example below shows a result in PCARDM using the MACHINE LEARNING segmentation option that utilises the model trained in this case study.



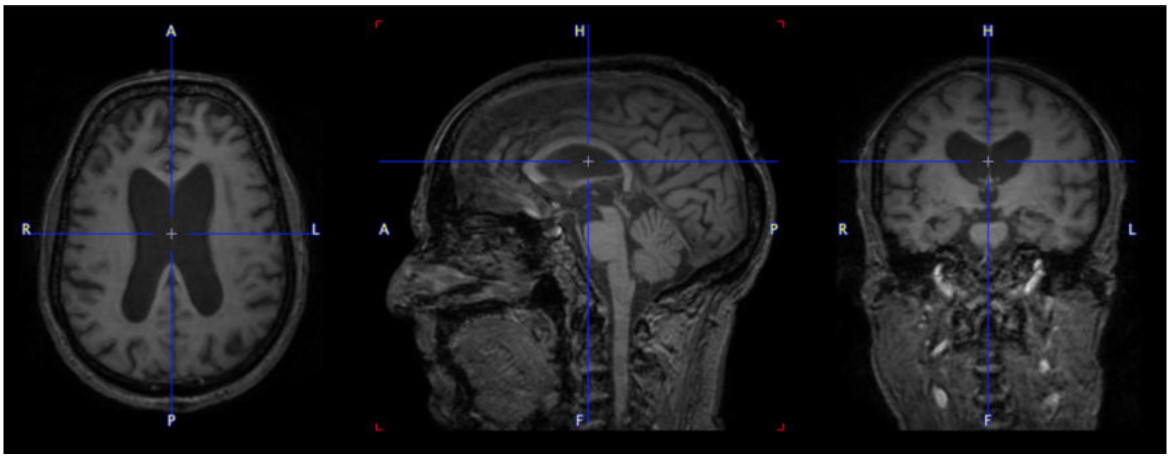
Parameter	LV	RV	Unit
Ejection fraction	66.72	-	%
ED volume	172.98	-	ccm
ES volume	57.56	-	ccm
Stroke volume	115.42	-	ccm
Heart rate	73.53	-	1/min
Cardiac output	8487	-	ccm/min

Example data to test the MRI Myocardium 2D and MRI Human Myocardium models is available in our Demo database (Subjects PAI2 and PAI3). The data used for the case study was provided by Bruker colleagues and in a private collaboration.

*To try the models for yourself we recommend use of the **Cardiac MR Modeling tool** (**PCARDM**) FUNCTION workflow. See the specific documentation for PCARDM. The model selection is determined by the species selection in PCARDM. The models were trained with 2D data so Split Slices and Split Frames are required. The recommended **Heart Box** for example PAI2 is 10 x 10 mm, and for PAI3 100 x 100 mm. The data used in these cases was for a limited range of MR sequences - the performance of the model may vary for data from different hardware and/or with different contrast/pre-processing.*

7.5 Human Deep Nuclei Segmentation

Accurate segmentation of sub-cortical brain regions (the deep nuclei - caudate, putamen, ventral striatum, thalamus, hippocampus, amygdala) from high resolution MR data such as 3D T1-weighted sequences (e.g. Siemens MPRAGE, GE FSPGR) is a requirement in the study of diseases such as Parkinson's using imaging. The segmented VOIs may be used for volumetric analysis of MRI data in the input image space or also for PET/SPECT quantification. Traditional methods of achieving this through spatial normalization to brain templates can fail or be inaccurate in some cases, particularly when there is pronounced brain atrophy. PMOD's PNEURO Parcellation workflow is specifically designed to provide increased accuracy for deep nuclei segmentation but requires additional processing time compared to template-based normalization and can still struggle in cases with severe atrophy.



We hypothesized that AI-based segmentation could be used as an alternative for cases with severe atrophy or where traditional methods struggled.

To test this hypothesis we used data from the IXI database of 3D T1-weighted MR:

https://brain-development.org/ixi-dataset/

Example data to test the IXI Parcellation model is available in our Demo database (Subject PAI4). The data used for the case study was extracted from the IXI dataset: https://brain-development.org/ixi-dataset/

*To try the model for yourself we recommend use of the **Segment tool (PSEG)**. The PAI4 example has the required orientation and any new data used for testing should match this. Cropping is not strictly required but is recommended to reduce the field-of-view to the brain. The model selection is **IXI Parcellation (Multichannel Segmentation)**. The model was trained with 3D data so Split Slices/Frames is not available/required. The data used in these cases was all T1-weighted 3D MR - the performance of the model may vary for data from different hardware and/or with different contrast/pre-processing.*

7.5.1 Sample Preparation

298 3D T1-weighted series from the IXI database were imported into a PMOD database for generation of reference segments using the PNEURO Parcellation workflow in batch processing mode.

The VOIs for bilateral caudate, putamen, ventral striatum, thalamus, hippocampus, amygdala, were reviewed by an experienced reviewer and any with substantial deviations were rejected for use in AI model training. Rejected samples were kept for later use as test samples.

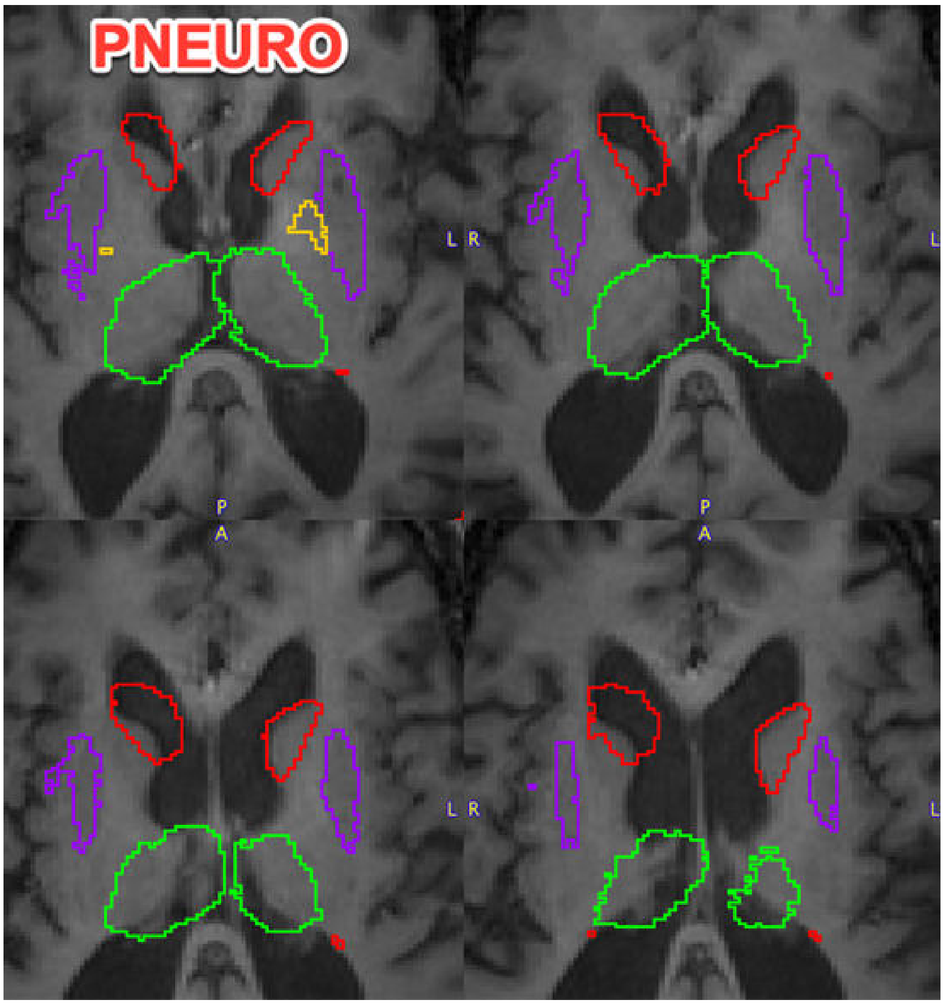
7.5.2 Training and Validation

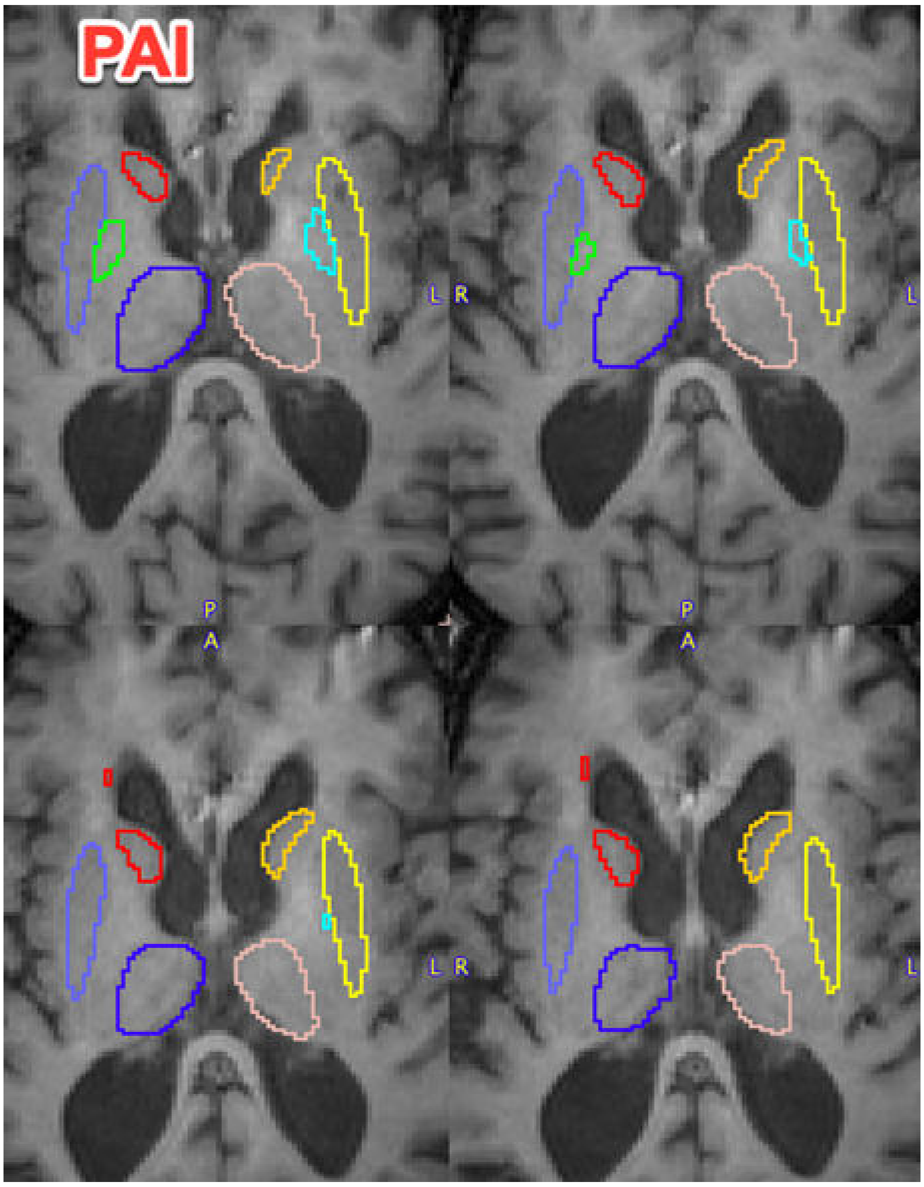
A Learning Set was created and the Multichannel Segmentation architecture selected. Training was performed as follows:

- AWS p2.8xlarge, 8 K80 GPU, 64 GB memory
- No cropping
- 200 samples (160 training, 40 validation)
- 200 epochs (aborted)
- batch size 16
- learning rate 0.005

Training took 17.5 hours.

Prediction was tested on samples excluded from training due to failed segmentation in PNEURO. The trained model was less sensitive to atrophy and large variations in ventricular anatomy than PNEURO and yielded acceptable deep nuclei VOIs. The example below illustrates the problems that can be encountered in PNEURO due to severe atrophy and successful result generated by the trained model:


PNEURO
L
R
P
A
L
R
P
A



A small artefact can be noted anterior to the right caudate. This could be cleaned up using the manual VOI tools and may be removed following further training of the model with additional samples.

The model only returns VOIs in the original input image space, while in successful cases PNEURO allows spatial transformation between input and atlas space.

Example data to test the IXI Parcellation model is available in our Demo database (Subject PAI4). The data used for the case study was extracted from the IXI dataset: https://brain-development.org/ixi-dataset/

To try the model for yourself we recommend use of the ***Segment tool (PSEG)****. The PAI4 example has the required orientation and any new data used for testing should match this. Cropping is not strictly required but is recommended to reduce the field-of-view to the brain. The model selection is* ***IXI Parcellation (Multichannel Segmentation)****. The model was trained with 3D data so Split Slices/Frames is not available/required. The data used in these cases was all T1-weighted 3D MR - the performance of the model may vary for data from different hardware and/or with different contrast/pre-processing.*

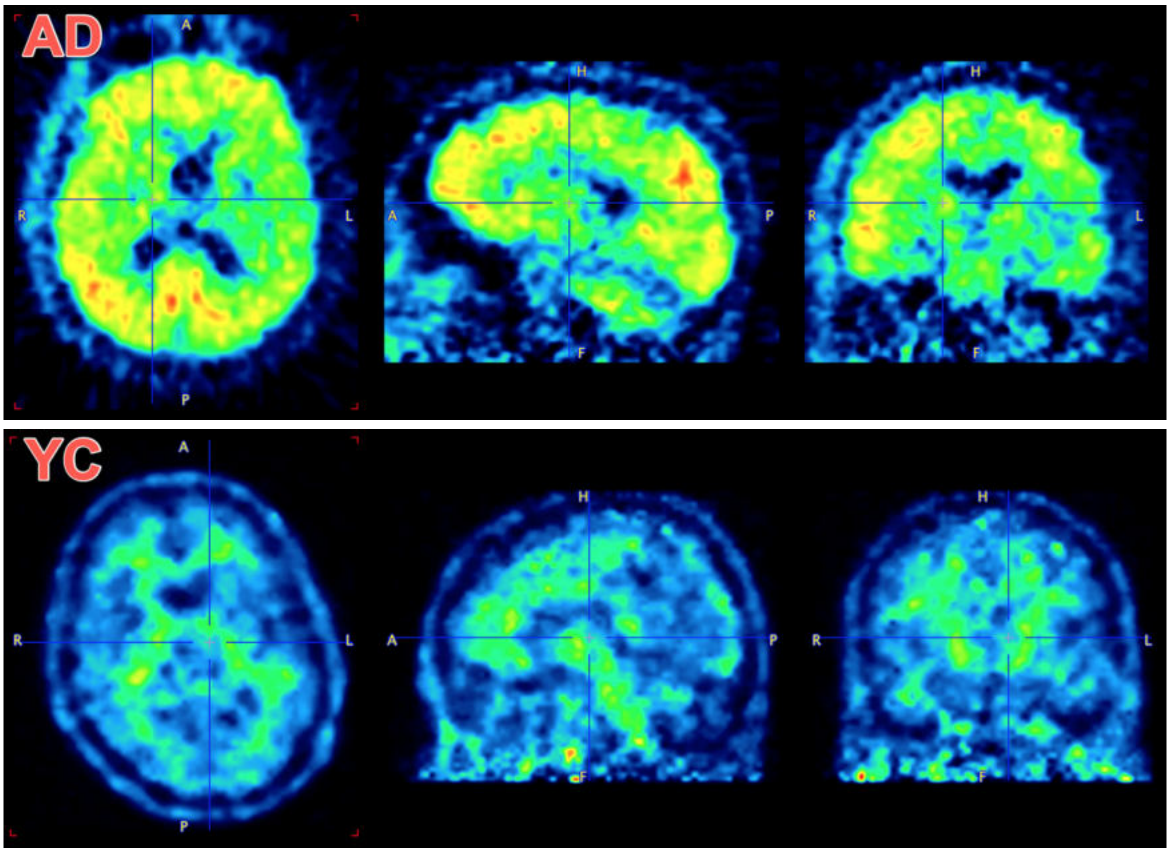
7.6 Human Amyloid PET Classification

Many studies have been performed using amyloid PET in the search for a biomarker predicting progression to ALzheimer's/dementia, and several databases of imaging data for amyloid PET tracers are available online (often along with anatomical MR and FDG PET). Visually, there are pronounced differences between the distribution of the amyloid PET tracer 11C-PiB in subjects without significant amyloid accumulation in cortical regions and in subjects with large accumulation. In subjects without significant accumulation, tracer uptake is clear in white matter with very low signal in grey matter/cortical regions. In subjects with significant amyloid accumulation the tracer is more uniformly distributed throughout the brain with grey matter uptake approaching/equalling white matter. Note that healthy subjects may still have amyloid accumulation and amyloid accumulation is not clear in all cases of Alzheimer's that undergo imaging.

We used an online database of 11C-PiB PET to illustrate the functionality of our AI classification using the SVM architecture:

http://www.gaain.org/centiloid-project (Standard PiB data)

This database contains amyloid PET data for subjects with Alzheimer's disease (AD) and amyloid accumulation, and comparative data from young controls (YC) with low/negligible amyloid accumulation.



Example data to test the Amyloid PET SVM model is available in our Demo database (Subject PAI7). The data used for the case study was downloaded from the GAAIN project website: http://www.gaain.org/centiloid-project

*To try the model for yourself the workflow in the **View** tool is required. If new data is tested, cropping should be applied to reduce the field-of-view to the brain. The model selected should be AmyloidPET-SVM (Classification SVM). The data used in these cases was all cropped 3D PET for late average 11C-PiB from unknown PET hardware - the performance of the model may vary for data from different hardware and/or with different tracers/reconstruction/pre-processing.*

7.6.1 Sample Preparation

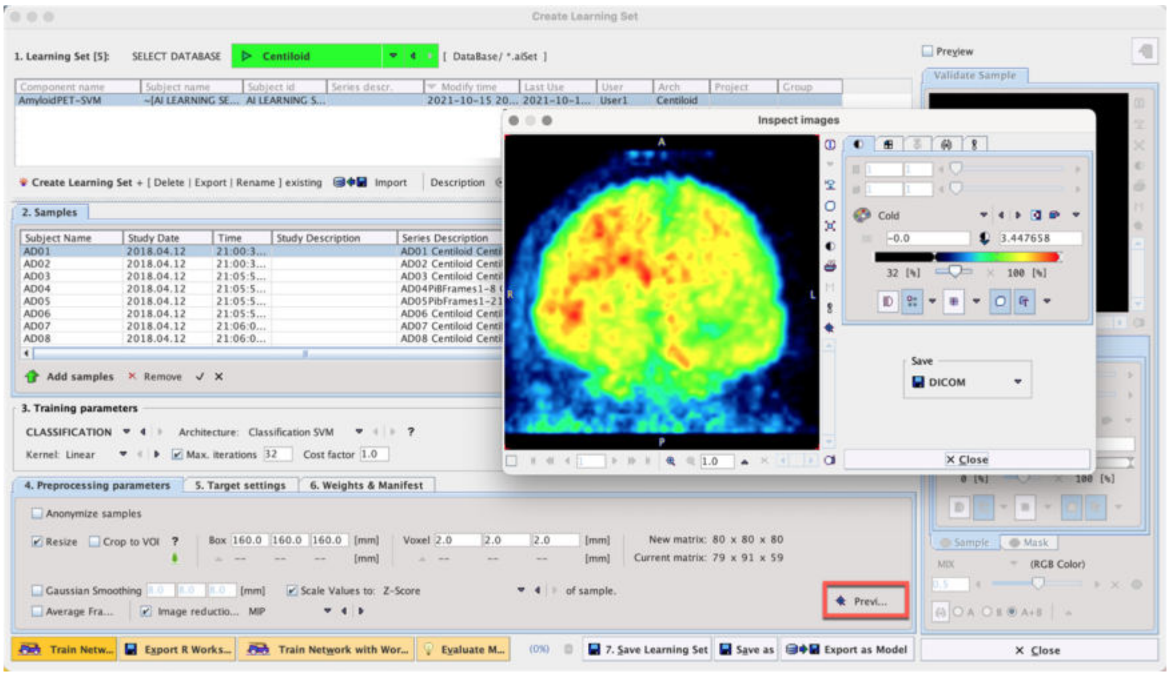
The 79 available datasets (45 AD, 34 YC) were imported into a database. Manual cropping was performed to more precisely limit the field-of-view to the brain. No other preprocessing was applied. The resulting image series varied in matrix size (x 67-175, y 79-227, z 35-104). AD were assigned the Group label “AD” and YC the Group label “YC” to define the classes to be trained and later predicted.

7.6.2 Training and Validation

A Learning Set was created and the Classification SVM architecture selected. The linear kernel was used with a maximum 32 iterations and Cost factor = 1.0.

Training was performed as follows:

- Windows workstation Intel i7-2600K 3.4GHz (no GPU use possible for SVM)
- 24 GB RAM
- 79 samples (45 class = AD, 34 class = YC)
- The samples were Resized to Box size 160 x 160 x 160 mm and Voxel size 2.0 x 2.0 x 2.0 mm, resulting in a new standardized image matrix of 80 x 80 x 80 for all samples.
- The image values were normalized using the z-score method
- Image reduction was applied using MIP (maximum intensity projection)



1. Learning Set [5]: SELECT DATABASE Centiloid [DataBase/ *.aiSet]

Component name	Subject name	Subject id	Series descr.	Modify time	Last Use	User	Arch	Project	Group
AmyloidPET-SVM	~[AI LEARNING SE...	AI LEARNING S...		2021-10-15 20...	2021-10-1...	User1	Centiloid		

2. Samples

Subject Name	Study Date	Time	Study Description	Series Description
AD01	2018.04.12	21:00:3...		AD01 Centiloid Centi
AD02	2018.04.12	21:00:3...		AD02 Centiloid Centi
AD03	2018.04.12	21:05:5...		AD03 Centiloid Centi
AD04	2018.04.12	21:05:5...		AD04PiBFrames1-8 (
AD05	2018.04.12	21:05:5...		AD05PibFrames1-21
AD06	2018.04.12	21:05:5...		AD06 Centiloid Centi
AD07	2018.04.12	21:06:0...		AD07 Centiloid Centi
AD08	2018.04.12	21:06:0...		AD08 Centiloid Centi

3. Training parameters

CLASSIFICATION Architecture: Classification SVM

Kernel: Linear [✓] Max. iterations 32 Cost factor 1.0

4. Preprocessing parameters | 5. Target settings | 6. Weights & Manifest

[ ] Anonymize samples

[✓] Resize [ ] Crop to VOI Box 160.0 160.0 160.0 [mm] Voxel 2.0 2.0 2.0 [mm] New matrix: 80 x 80 x 80 Current matrix: 79 x 91 x 59

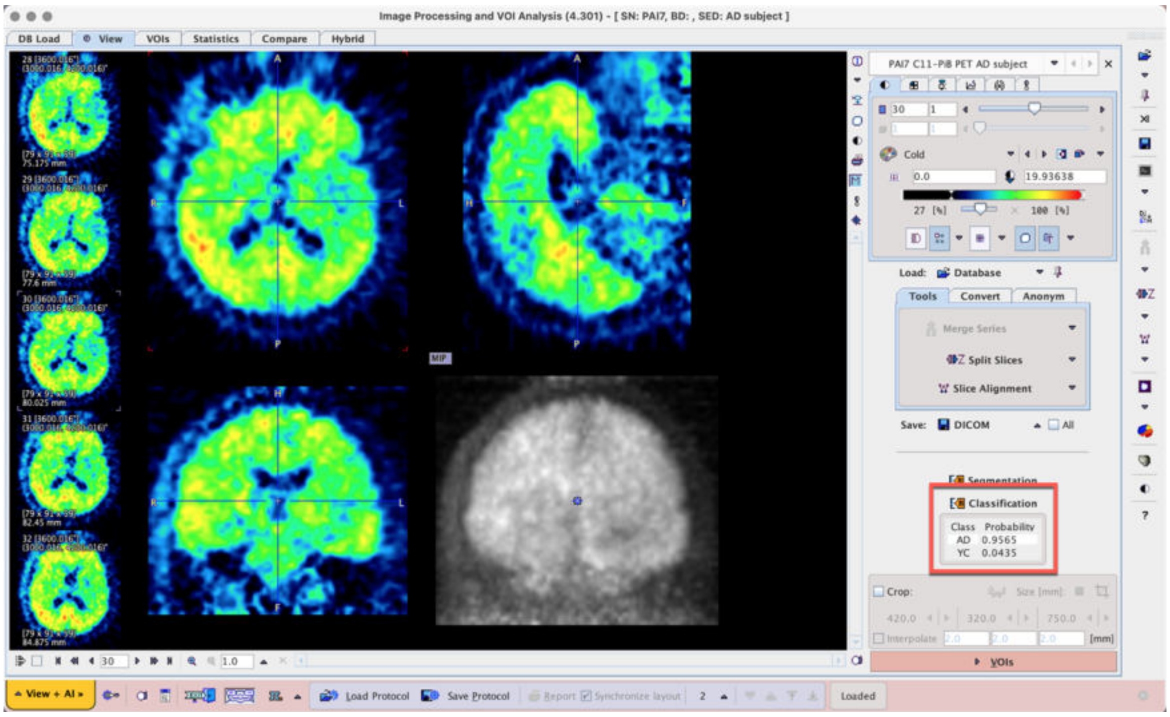
[ ] Gaussian Smoothing [mm] [✓] Scale Values to: Z-Score of sample.

[ ] Average Fra... [✓] Image reductio... MIP

-0.0 3.447658 32 [%] 100 [%]

Training took less than 5 minutes.

Randomly selected AD and YC samples were selected from the same dataset to test Prediction. The 3D input data is loaded and the MIP is calculated as part of classification. High probabilities for the expected class were returned.



Class	Probability
AD	0.9565
YC	0.0435

Example data to test the Amyloid PET SVM model is available in our Demo database (Subject PAI7). The data used for the case study was downloaded from the GAAIN project website: http://www.gaain.org/centiloid-project

To try the model for yourself the workflow in the ***View*** *tool is required. If new data is tested, cropping should be applied to reduce the field-of-view to the brain. The model selected should be AmyloidPET-SVM (Classification SVM). The data used in these cases was all cropped 3D PET for late average 11C-PiB from unknown PET hardware - the performance of the model may vary for data from different hardware and/or with different tracers/reconstruction/pre-processing.*

8 Appendix

8.1 Exporting an R Workspace and Training in a Cloud Computing Environment

Training an ML model can be a resource- and time-consuming task. Therefore it is often reasonable to perform this task using remote computing power. That can be either a server within the same institution or so-called Cloud Computing services (e.g. Amazon Web Services, Microsoft Azure). The architecture can be optimized for Machine Learning calculations. In both cases PMOD's PSEG tool provides a convenient way to transfer the data to the computing unit via an R Workspace. Such a solution guarantees that the time required on the computing machine will be spent entirely on the training process (since the data stored in the R Workspace can be already preprocessed in PSEG). Software setup can also be simplified, as there is no need to install PMOD on the computing unit.

Please note that the short guide below assumes that appropriate versions of R (with the required packages) and TensorFlow libraries are provided and configured on the computing unit (See Installation of R and Python for UNIX platforms).

Remote training following Export of an R workspace (overview):

1. Local machine: Create / load the Learning Set in the PSEG tool according to the Documentation above
2. Remember to configure all training parameters, as if preparing for a standard local training session
3. Select "Export R Workspace"
4. After the export is finished please select a path and a name for the Workspace
5. Transfer the R Workspace and "pm.ai.tar.gz" to the computing unit ("pm.ai.tar.gz" is an R package created by PMOD. It is required to run machine learning-based processing. You can find it in your PMOD installation folder in the subfolder: "Pmod4.4/resources/extlibs/r/lib/")
6. Start R on the computing unit
7. Run the training process in the R environment:

```
install.packages(“~/.../pm.ai.tar.gz”)

library(“pm.ai”)

library(“keras”)

library(“stringr”)

load.workspace(“~/.../Workspace.RData”)

pm.ai.learn()
```

To activate the tensorflow 2.3 environment, use the following terminal command "source activate tensorflow2_latest_p37".

To monitor the GPU card usage, use the terminal command "watch -n0.1 nvidia-smi". This allows you to see the resources used at an interval of 0.1 second (the time interval can be changed).

The “htop” command allows you to see the Memory and CPU usage.

9 PMOD Disclaimer

PMOD is a software
FOR RESEARCH USE ONLY (RUO)
and must not be used for diagnosis or treatment of patients.

10 PMOD Copyright Notice

PMOD Technologies LLC
Industriestrasse 26
8117 Fällanden
Switzerland
support@pmod.com
http://www.pmod.com