PMOD Artificial Intelligence Framework (PAI)

USER MANUAL
Version 4.2

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

1 Introduction

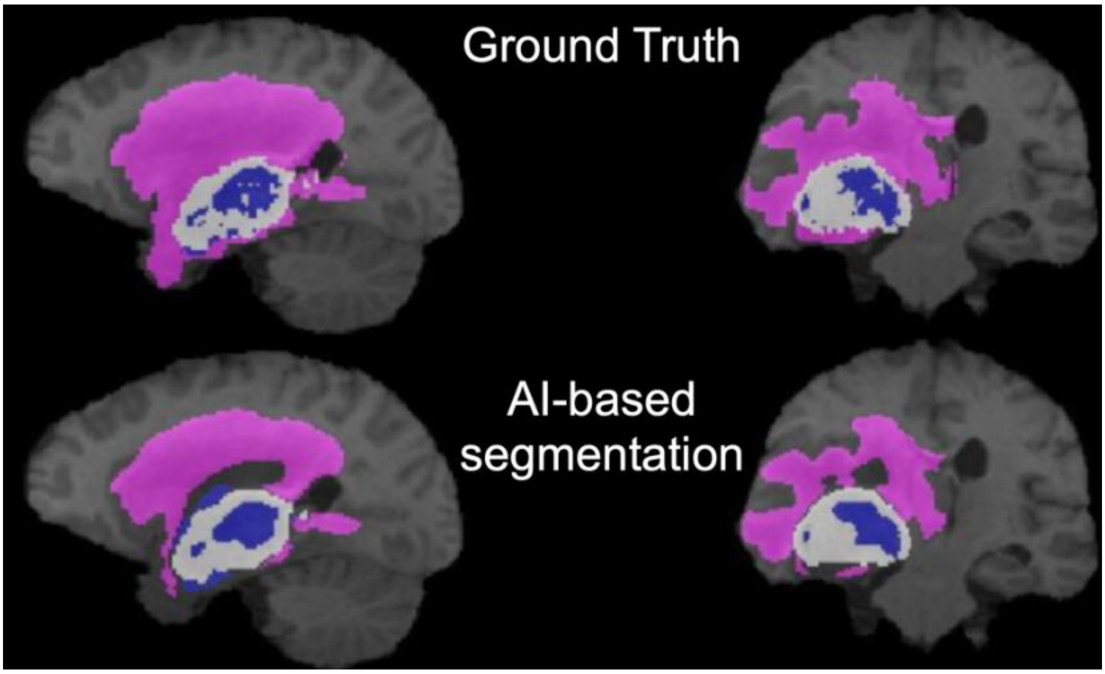
Computer algorithms based on Artificial Intelligence (AI) have become a success story and are now part of daily life. Machine Learning (ML) is a subset of AI in which humans provide the input data and expected results, and the computer determines “rules” with which it can process the data to approach 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.

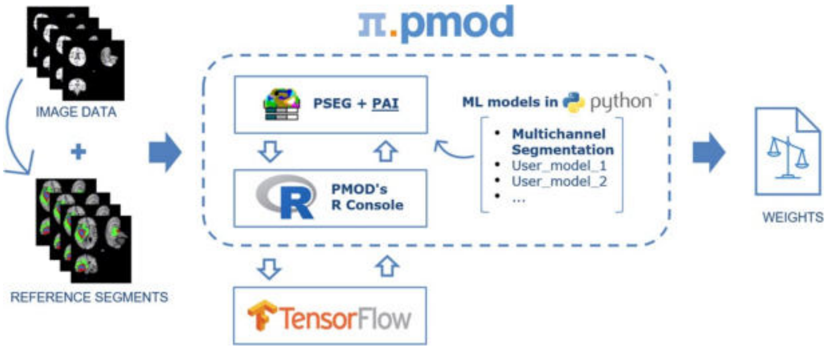


However, training a convolutional neural network to perform such a segmentation task requires a substantial amount of data, time and effort.

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.

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 result (one or several segments, in the format of label maps that can be associated with the input images).

Training

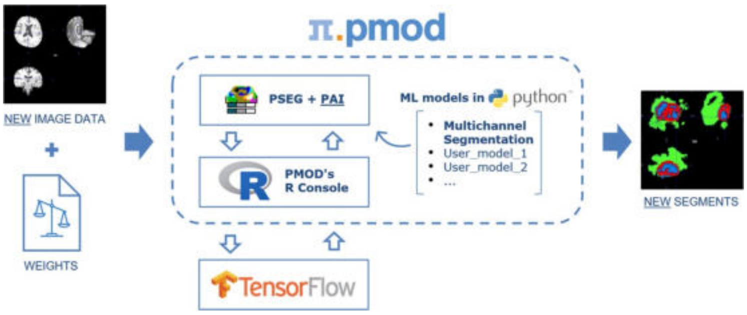
Training is performed in TensorFlow using a Learning Set and an ML model. There are different mechanisms available, which will be explained below. Basically, training can be performed locally in PSEG, or delegated to a more powerful infrastructure such as Cloud-Computing.

Training Result

The result of training is a "trained model" - a set of **Weights** for the layers in 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). 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).

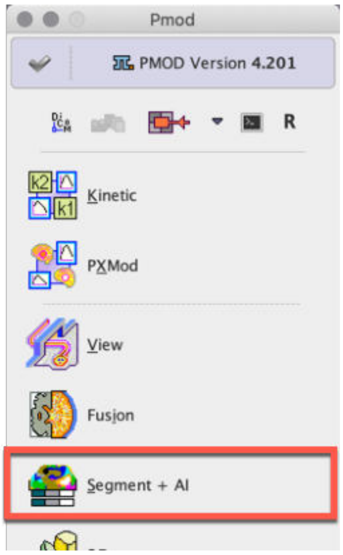
Prediction

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

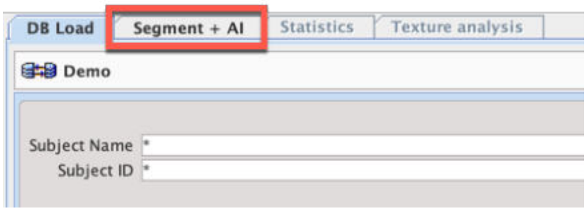


Implementation in PSEG

All PAI functionality is integrated in PSEG. If PAI has been licensed, a **+ AI** indication appears next to **PSEG** in the main PMOD ToolBox:

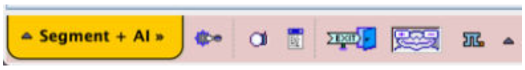


Similarly, in PSEG the **Segment** tab becomes **Segment + AI**



Subject Name *
Subject ID *

and the menu button as well:



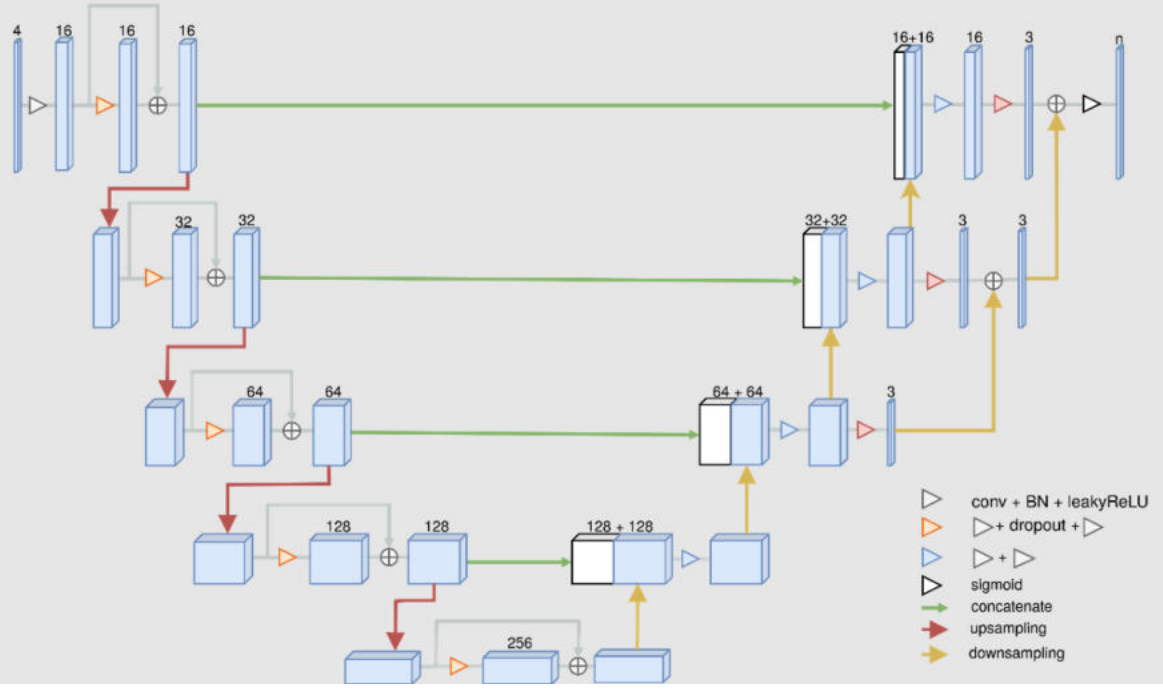
1.3 Example included in Distribution

PAI is provided with a demonstration 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 **Tumor Detection** is available in the **Segmentation** tool when PAI is licensed, using the **Multichannel Segmentation** architecture. 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).

A second experimental model based on the **Multichannel Segmentation** architecture is included with PMOD 4.203. This model is called **Rat Brain Dopaminergic PET**. It requires early and late average images derived from PET with tracers such as [^{11}C]-raclopride and [^{11}C]-methylphenidate, and the output is a label image with three segments (label 1: cerebellum; label 2: left striatum; label 3: right striatum). The model was trained using 382 rat brain dopaminergic system dynamic PET datasets. A description of the preparation of the data and process to train and evaluate the model is included later in this documentation as a case study.

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

2 Installation of PAI Infrastructure

PAI requires the following elements:

- PMOD version 4.2 with PSEG and PAI licensed (please use the latest build available in our download area)
- Local installation of R with the required packages. PMOD uses the R functionality via the Rserve library. 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 (version 2.3 or newer) 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 2.4:
   a. Drivers 11.0: https://www.nvidia.com/download/index.aspx?lang=en-us
   b. CUDA Toolkit 11.0: https://developer.nvidia.com/cuda-toolkit-archive
   c. cuDNN 8.0.4 for CUDA 11.0: 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:
   ```
   python -m pip install --upgrade pip
   ```
6. Install TensorFlow by entering in a command terminal:
   ```
   python -m pip install --upgrade tensorflow
   ```
7. Check that `tensorflow` appears in the list of installed packages:
   ```
   python -m pip 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.

> Note: GPU support for Windows (point 3. above) was tested for TensorFlow 2.4.

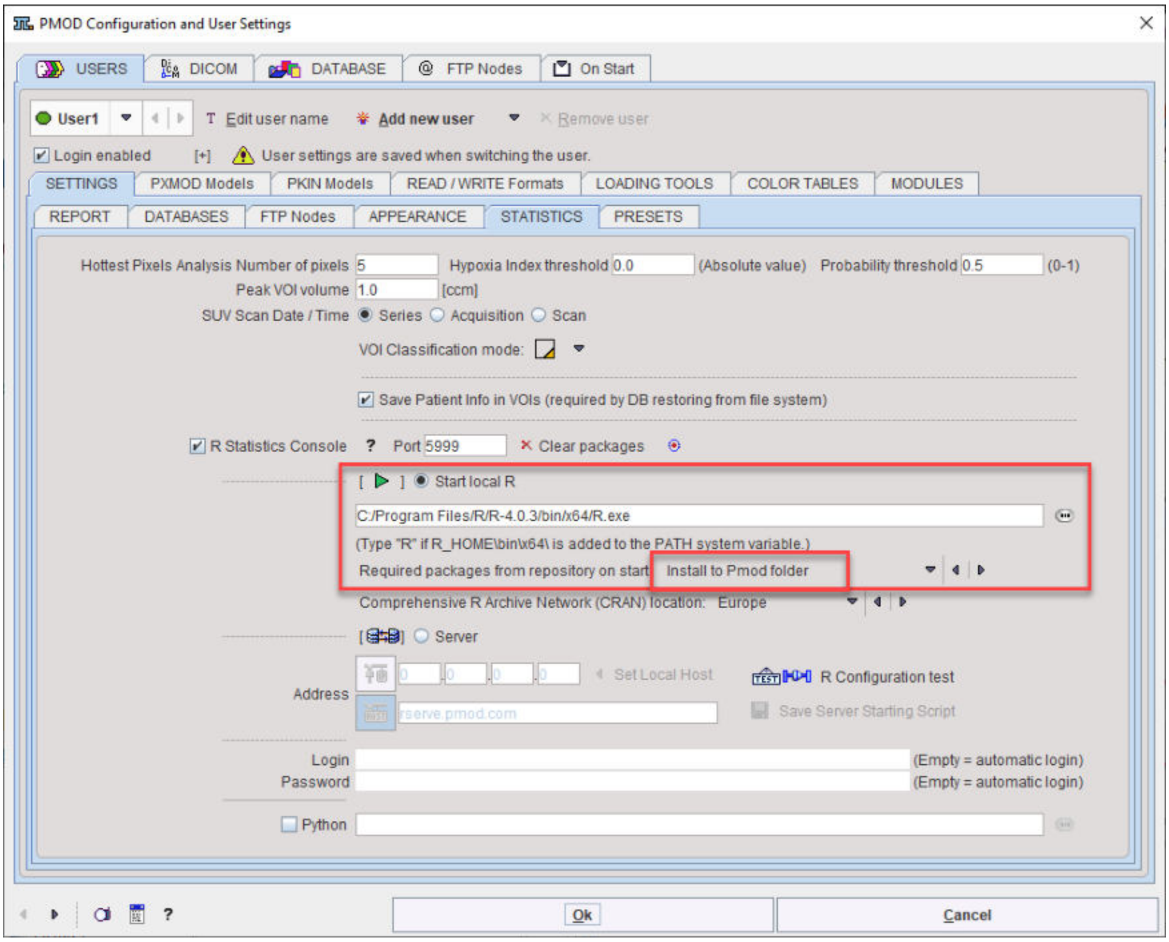
2.1.2 R Installation

Please download and install the most recent R version for Windows from https://cran.r-project.org/ (note that our testing to date has used version 4.0.3).

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 11 below.

2.1.2.1 Default R Configuration

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



PMOD Configuration and User Settings

USERS | DICOM | DATABASE | FTP Nodes | On Start

User1 | Edit user name | Add new user | Remove user

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

SETTINGS | PXMOD Models | PKIN Models | READ / WRITE Formats | LOADING TOOLS | COLOR TABLES | MODULES

REPORT | DATABASES | FTP Nodes | APPEARANCE | STATISTICS | PRESETS

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 ? Port 5999 ✕ Clear packages

[▶] ● Start local R

C:/Program Files/R/R-4.0.3/bin/x64/R.exe

(Type "R" if R_HOME\bin\x64\ is added to the PATH system variable.)

Required packages from repository on start: Install to Pmod folder

Comprehensive R Archive Network (CRAN) location: Europe

○ Server

Address 0 . 0 . 0 . 0 Set Local Host R Configuration test

rserve.pmod.com Save Server Starting Script

Login (Empty = automatic login)

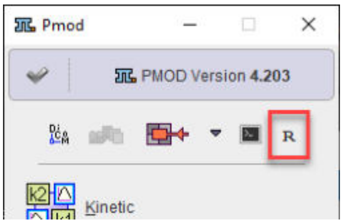
Password (Empty = automatic login)

□ Python

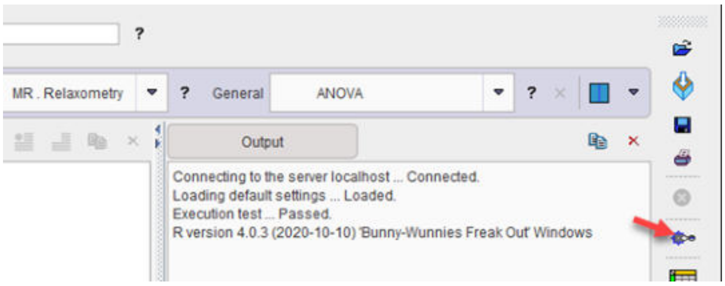
Ok | Cancel

On the **STATISTICS** tab select **Start local R** to ensure local execution 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.

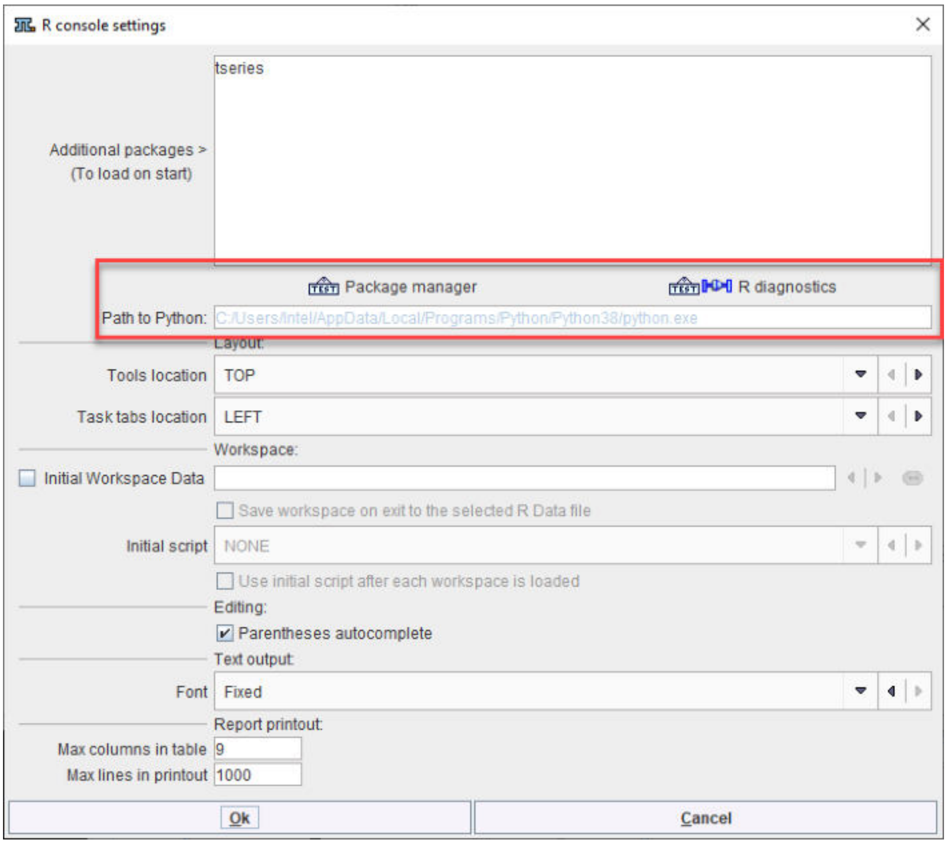


Then click on the **R** icon to open the **PMOD Console**. The required packages are downloaded and installed, followed by an execution test and printing of the R version information:



```
Connecting to the server localhost ... Connected.
Loading default settings ... Loaded.
Execution test ... Passed.
R version 4.0.3 (2020-10-10) 'Bunny-Wunnies Freak Out' Windows
```

The settings button indicated above opens the dialog window below.



Additional packages > (To load on start): tseries

Path to Python: C:/Users/intel/AppData/Local/Programs/Python/Python38/python.exe

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:

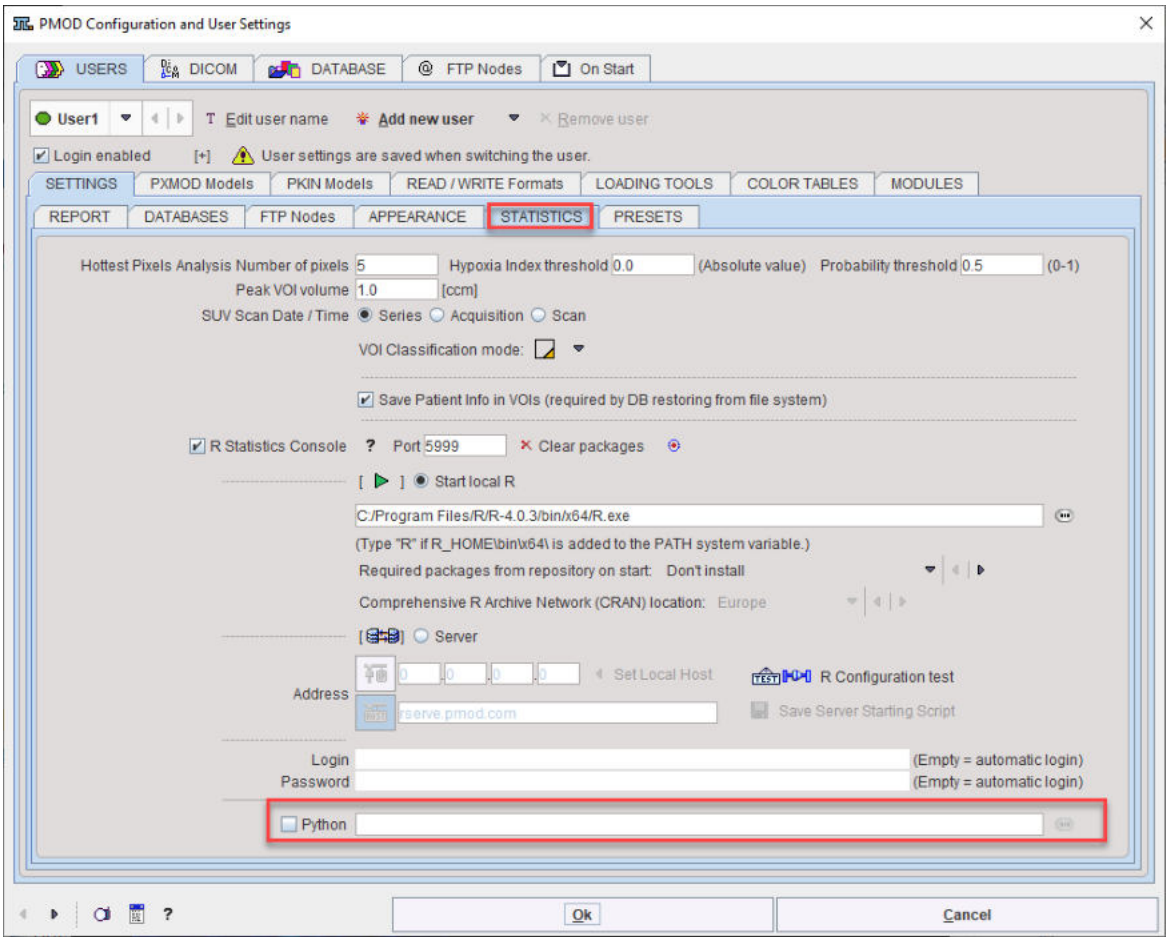
Font: Fixed

Report printout:

Max columns in table: 9

Max lines in printout: 1000

Please verify that the **Path to Python** corresponds to your local installation of Python 3.8. If it does not, 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 ? Port 5999

[▶] ● Start local R

C:/Program Files/R/R-4.0.3/bin/x64/R.exe

(Type "R" if R_HOME\bin\x64\ is added to the PATH system variable.)

Required packages from repository on start: Don't install

Comprehensive R Archive Network (CRAN) location: Europe

○ Server

Address 0 . 0 . 0 . 0

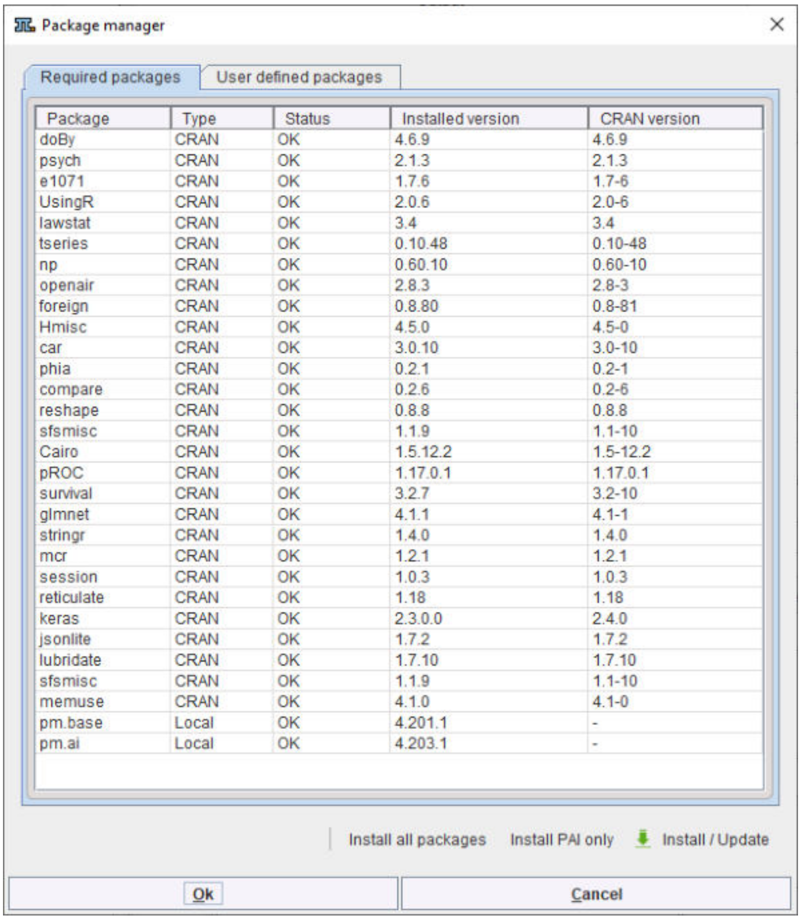
rserve.pmod.com

Login (Empty = automatic login)

Password (Empty = automatic login)

□ Python

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



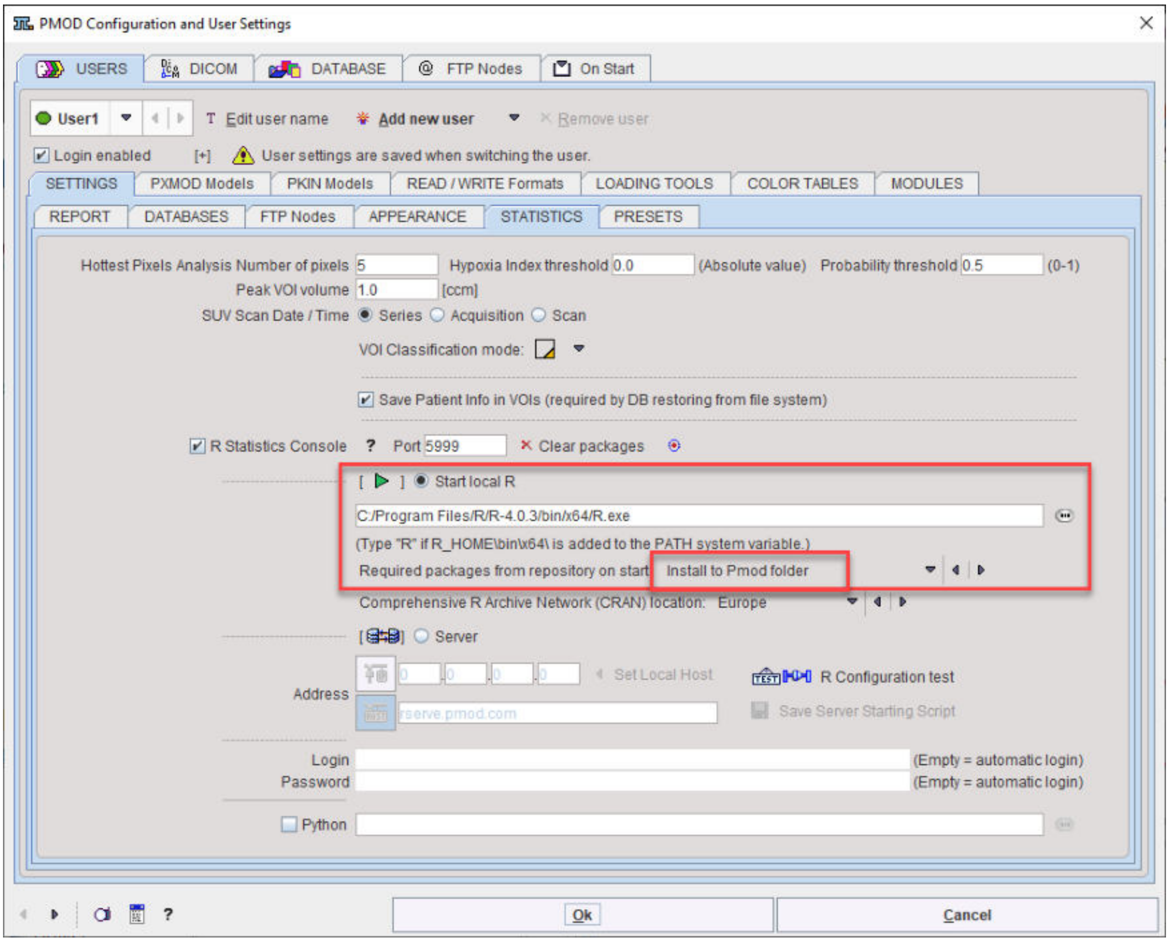
Package	Type	Status	Installed version	CRAN version
doBy	CRAN	OK	4.6.9	4.6.9
psych	CRAN	OK	2.1.3	2.1.3
e1071	CRAN	OK	1.7.6	1.7-6
UsingR	CRAN	OK	2.0.6	2.0-6
lawstat	CRAN	OK	3.4	3.4
tseries	CRAN	OK	0.10.48	0.10-48
np	CRAN	OK	0.60.10	0.60-10
openair	CRAN	OK	2.8.3	2.8-3
foreign	CRAN	OK	0.8.80	0.8-81
Hmisc	CRAN	OK	4.5.0	4.5-0
car	CRAN	OK	3.0.10	3.0-10
phia	CRAN	OK	0.2.1	0.2-1
compare	CRAN	OK	0.2.6	0.2-6
reshape	CRAN	OK	0.8.8	0.8.8
sfsmisc	CRAN	OK	1.1.9	1.1-10
Cairo	CRAN	OK	1.5.12.2	1.5-12.2
pROC	CRAN	OK	1.17.0.1	1.17.0.1
survival	CRAN	OK	3.2.7	3.2-10
glmnet	CRAN	OK	4.1.1	4.1-1
stringr	CRAN	OK	1.4.0	1.4.0
mcr	CRAN	OK	1.2.1	1.2.1
session	CRAN	OK	1.0.3	1.0.3
reticulate	CRAN	OK	1.18	1.18
keras	CRAN	OK	2.3.0.0	2.4.0
jsonlite	CRAN	OK	1.7.2	1.7.2
lubridate	CRAN	OK	1.7.10	1.7.10
sfsmisc	CRAN	OK	1.1.9	1.1-10
memuse	CRAN	OK	4.1.0	4.1-0
pm.base	Local	OK	4.201.1	-
pm.ai	Local	OK	4.203.1	-

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.



☑ 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 ? Port 5999

● Start local R

C:/Program Files/R/R-4.0.3/bin/x64/R.exe

(Type "R" if R_HOME\bin\x64\ is added to the PATH system variable.)

Required packages from repository on start Install to Pmod folder

Comprehensive R Archive Network (CRAN) location: Europe

○ Server

Address 0 0 0 0

rserve.pmod.com

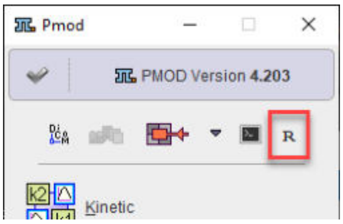
Login (Empty = automatic login)

Password (Empty = automatic login)

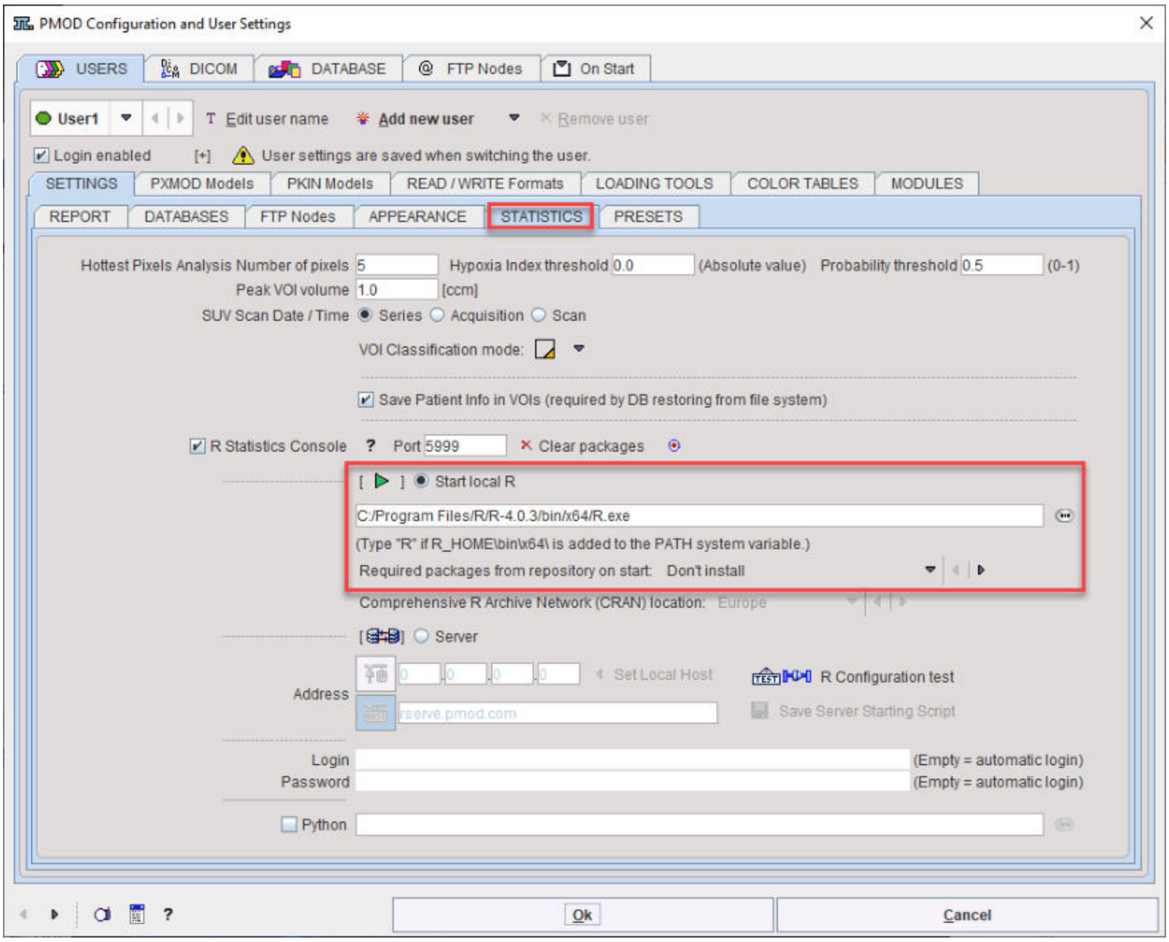
□ Python

On the **STATISTICS** tab select **Start local R** to ensure local execution and verify that the path to the local R installation is correct. Select **Install to Pmod folder** to install the base required packages on PMOD restart.

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



Return to the **Configuration** facility from the main ToolBox and change the R package installation selection to Don't install:



PMOD Configuration and User Settings

USERS | DICOM | DATABASE | FTP Nodes | On Start

User1 | Edit user name | Add new user | Remove user

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

SETTINGS | PXMOD Models | PKIN Models | READ / WRITE Formats | LOADING TOOLS | COLOR TABLES | MODULES

REPORT | DATABASES | FTP Nodes | APPEARANCE | STATISTICS | PRESETS

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 ? Port 5999 ✕ Clear packages

[▶] ● Start local R

C:/Program Files/R/R-4.0.3/bin/x64/R.exe

(Type "R" if R_HOME\bin\x64\ is added to the PATH system variable.)

Required packages from repository on start: Don't install

Comprehensive R Archive Network (CRAN) location: Europe

○ Server

Address 0 . 0 . 0 . 0 Set Local Host R Configuration test

rserve.pmod.com Save Server Starting Script

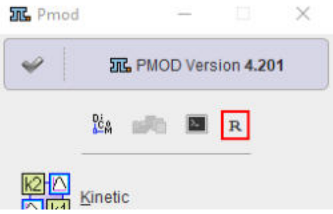
Login (Empty = automatic login)

Password (Empty = automatic login)

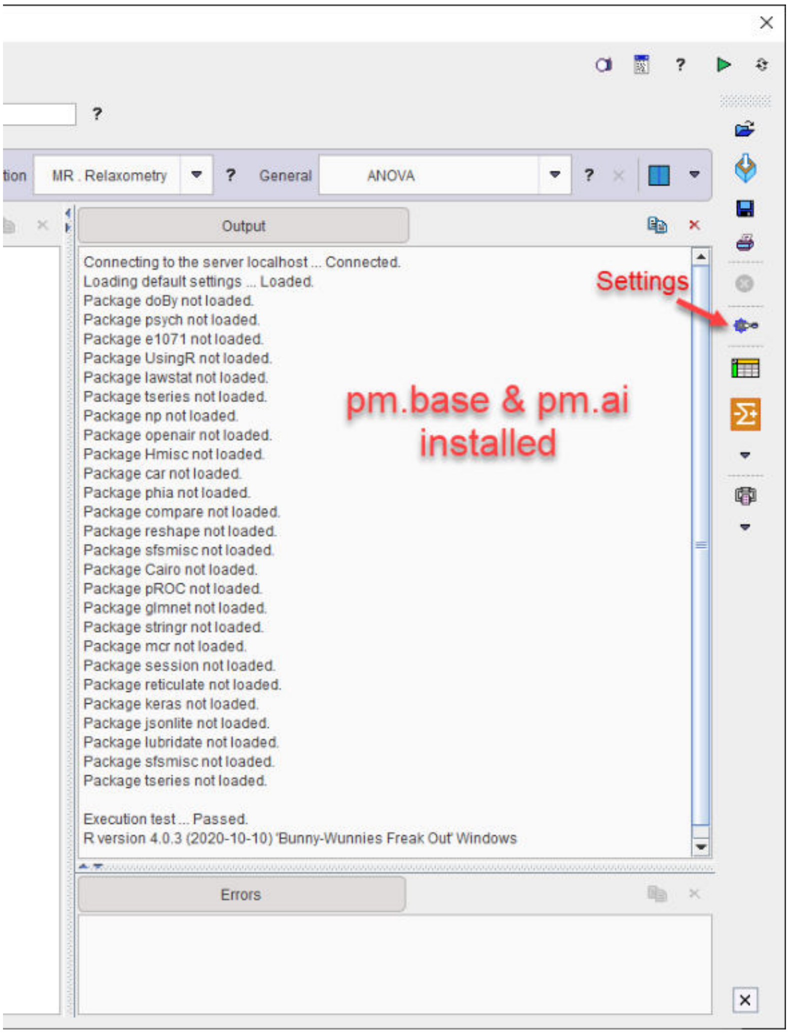
☐ Python

Ok | Cancel

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:

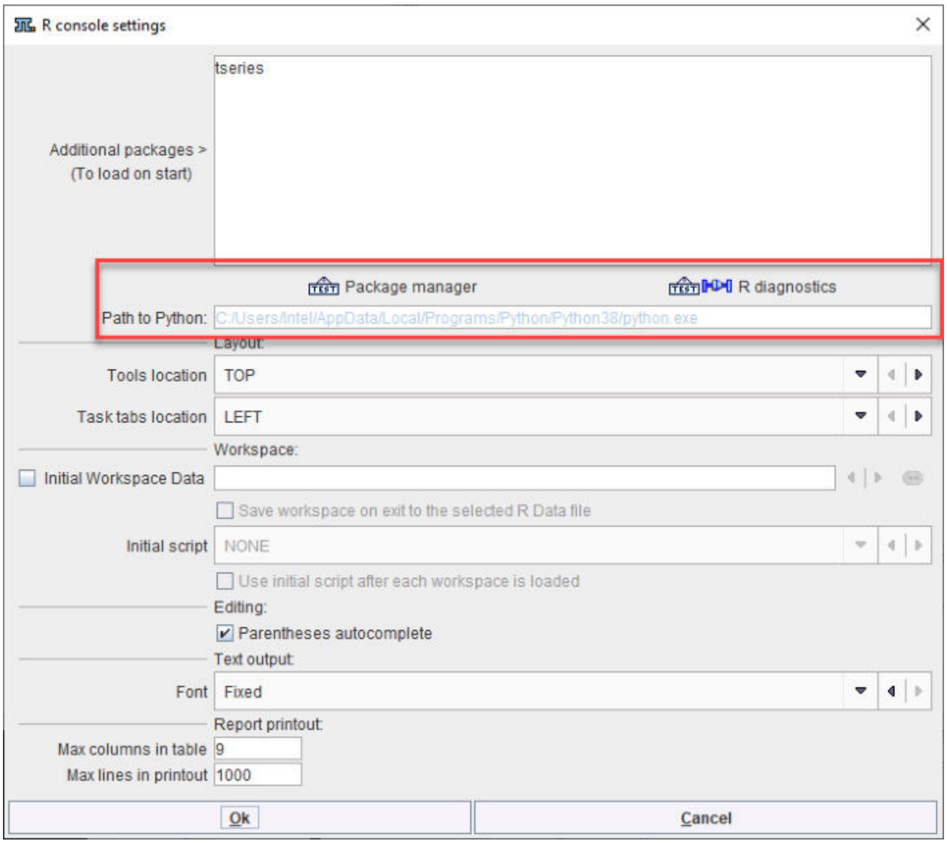


Settings

pm.base & pm.ai installed

```
Connecting to the server localhost ... Connected.
Loading default settings ... 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 session not loaded.
Package reticulate not loaded.
Package keras not loaded.
Package jsonlite not loaded.
Package lubridate not loaded.
Package sfsmisc not loaded.
Package tseries not loaded.

Execution test ... Passed.
R version 4.0.3 (2020-10-10) 'Bunny-Wunnies Freak Out' Windows
```

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



R console settings

Additional packages > (To load on start): tseries

Path to Python: C:/Users/Intel/AppData/Local/Programs/Python/Python38/python.exe

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:

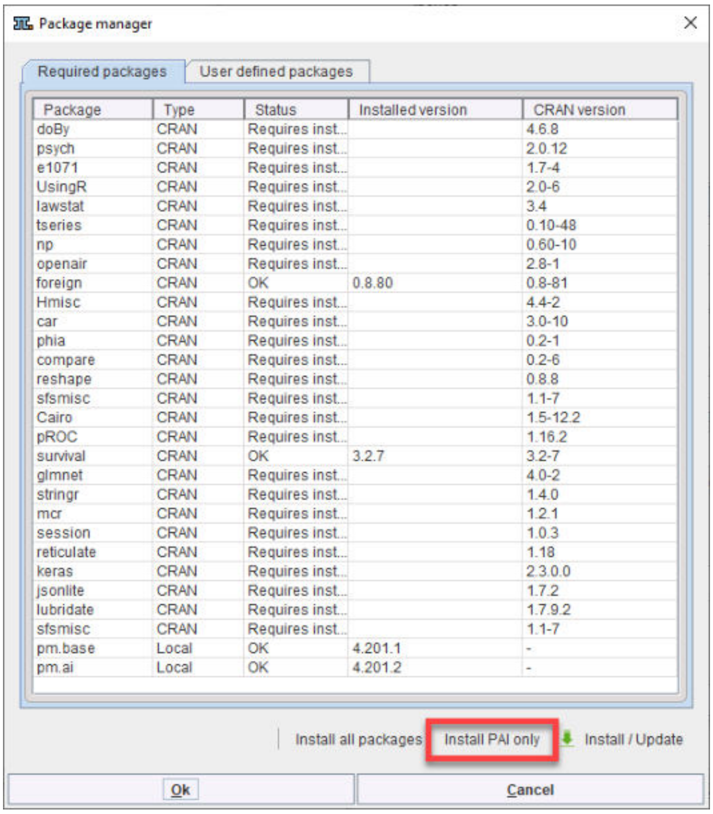
Font: Fixed

Report printout:

Max columns in table: 9

Max lines in printout: 1000

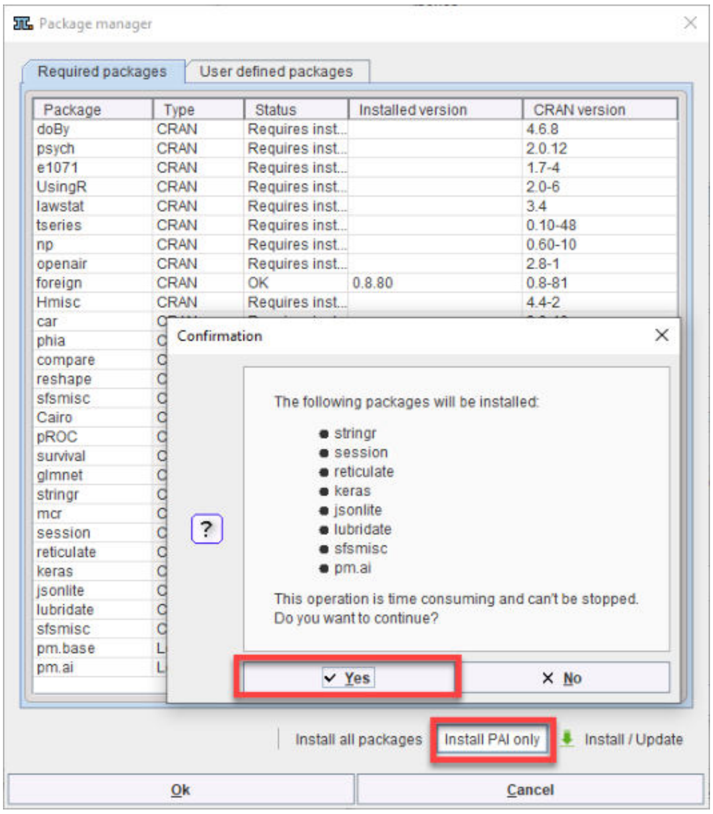
Please verify the **Path to Python** and open the **Package Manager**. Most packages will have **Requires installation** in the **Status** column



Package manager

Package	Type	Status	Installed version	CRAN version
doBy	CRAN	Requires inst...		4.6.8
psych	CRAN	Requires inst...		2.0.12
e1071	CRAN	Requires inst...		1.7-4
UsingR	CRAN	Requires inst...		2.0-6
lawstat	CRAN	Requires inst...		3.4
tseries	CRAN	Requires inst...		0.10-48
np	CRAN	Requires inst...		0.60-10
openair	CRAN	Requires inst...		2.8-1
foreign	CRAN	OK	0.8.80	0.8-81
Hmisc	CRAN	Requires inst...		4.4-2
car	CRAN	Requires inst...		3.0-10
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...		1.1-7
Cairo	CRAN	Requires inst...		1.5-12.2
pROC	CRAN	Requires inst...		1.16.2
survival	CRAN	OK	3.2.7	3.2-7
glmnet	CRAN	Requires inst...		4.0-2
stringr	CRAN	Requires inst...		1.4.0
mcr	CRAN	Requires inst...		1.2.1
session	CRAN	Requires inst...		1.0.3
reticulate	CRAN	Requires inst...		1.18
keras	CRAN	Requires inst...		2.3.0.0
jsonlite	CRAN	Requires inst...		1.7.2
lubridate	CRAN	Requires inst...		1.7.9.2
sfsmisc	CRAN	Requires inst...		1.1-7
pm.base	Local	OK	4.201.1	-
pm.ai	Local	OK	4.201.2	-

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



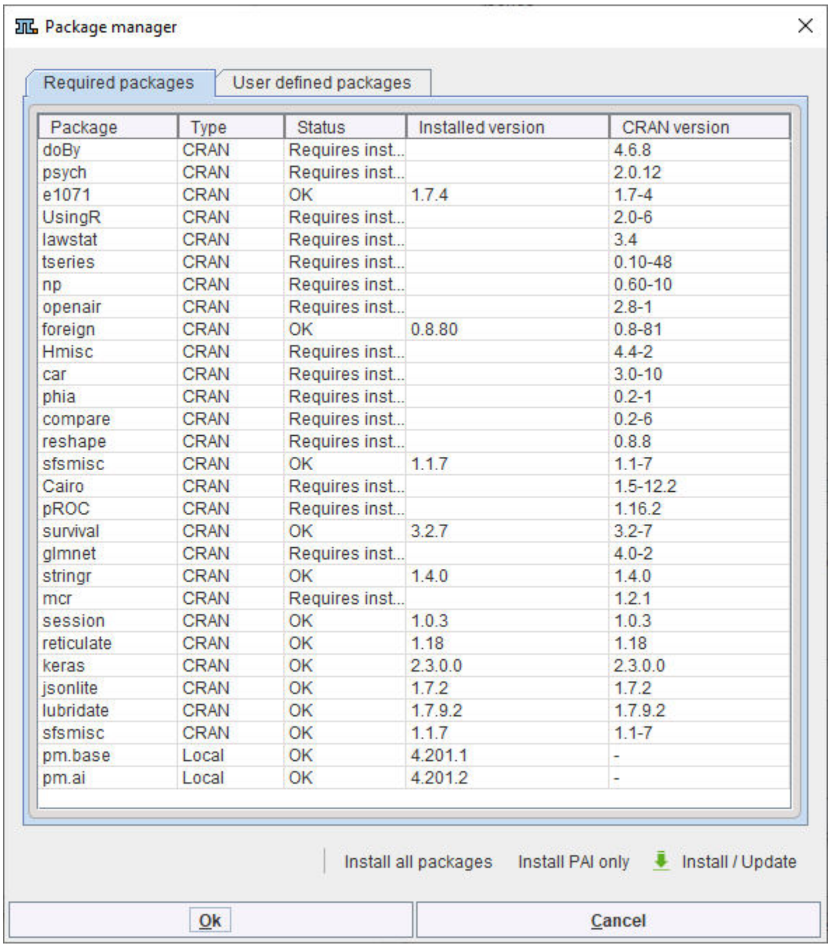
Package	Type	Status	Installed version	CRAN version
doBy	CRAN	Requires inst...		4.6.8
psych	CRAN	Requires inst...		2.0.12
e1071	CRAN	Requires inst...		1.7-4
UsingR	CRAN	Requires inst...		2.0-6
lawstat	CRAN	Requires inst...		3.4
tseries	CRAN	Requires inst...		0.10-48
np	CRAN	Requires inst...		0.60-10
openair	CRAN	Requires inst...		2.8-1
foreign	CRAN	OK	0.8.80	0.8-81
Hmisc	CRAN	Requires inst...		4.4-2
car	C			
phia	C			
compare	C			
reshape	C			
sfsmisc	C			
Cairo	C			
pROC	C			
survival	C			
glmnet	C			
stringr	C			
mcr	C			
session	C			
reticulate	C			
keras	C			
jsonlite	C			
lubridate	C			
sfsmisc	C			
pm.base	L			
pm.ai	L			

The following packages will be installed:

- stringr
- session
- reticulate
- keras
- jsonlite
- lubridate
- sfsmisc
- pm.ai

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

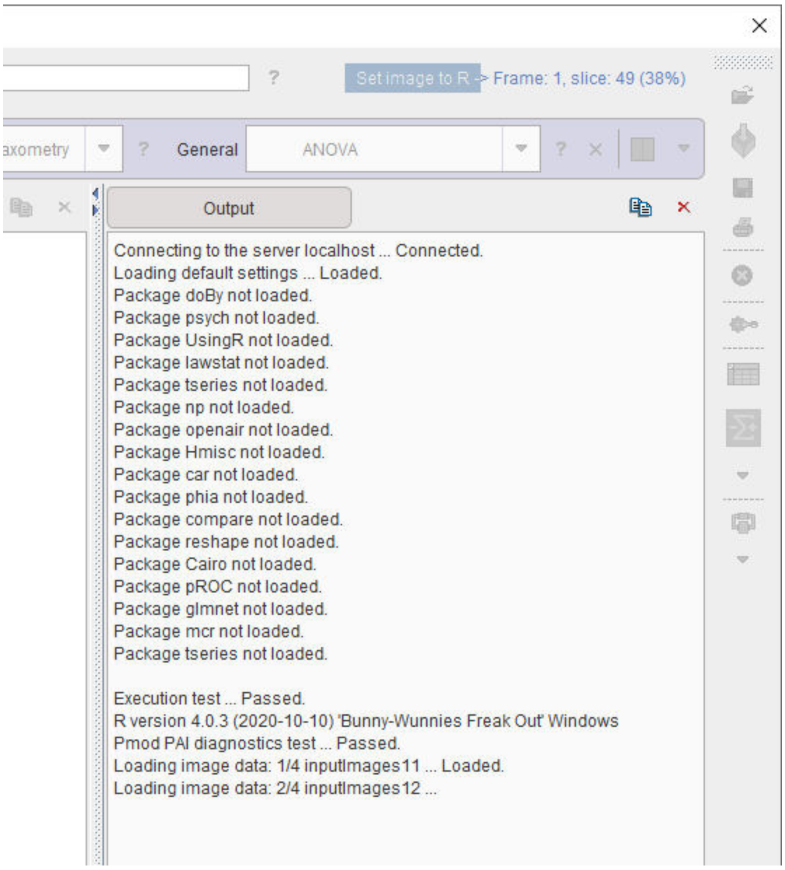
Following installation the **Status** entries will be updated:



Package	Type	Status	Installed version	CRAN version
doBy	CRAN	Requires inst...		4.6.8
psych	CRAN	Requires inst...		2.0.12
e1071	CRAN	OK	1.7.4	1.7-4
UsingR	CRAN	Requires inst...		2.0-6
lawstat	CRAN	Requires inst...		3.4
tseries	CRAN	Requires inst...		0.10-48
np	CRAN	Requires inst...		0.60-10
openair	CRAN	Requires inst...		2.8-1
foreign	CRAN	OK	0.8.80	0.8-81
Hmisc	CRAN	Requires inst...		4.4-2
car	CRAN	Requires inst...		3.0-10
phia	CRAN	Requires inst...		0.2-1
compare	CRAN	Requires inst...		0.2-6
reshape	CRAN	Requires inst...		0.8.8
sfsmisc	CRAN	OK	1.1.7	1.1-7
Cairo	CRAN	Requires inst...		1.5-12.2
pROC	CRAN	Requires inst...		1.16.2
survival	CRAN	OK	3.2.7	3.2-7
glmnet	CRAN	Requires inst...		4.0-2
stringr	CRAN	OK	1.4.0	1.4.0
mcr	CRAN	Requires inst...		1.2.1
session	CRAN	OK	1.0.3	1.0.3
reticulate	CRAN	OK	1.18	1.18
keras	CRAN	OK	2.3.0.0	2.3.0.0
jsonlite	CRAN	OK	1.7.2	1.7.2
lubridate	CRAN	OK	1.7.9.2	1.7.9.2
sfsmisc	CRAN	OK	1.1.7	1.1-7
pm.base	Local	OK	4.201.1	-
pm.ai	Local	OK	4.201.2	-

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

> Note: When a prediction will be launched in PSEG at a later timepoint, the R Console will report the packages that were not installed, but the **Execution test** will still pass as illustrated below.

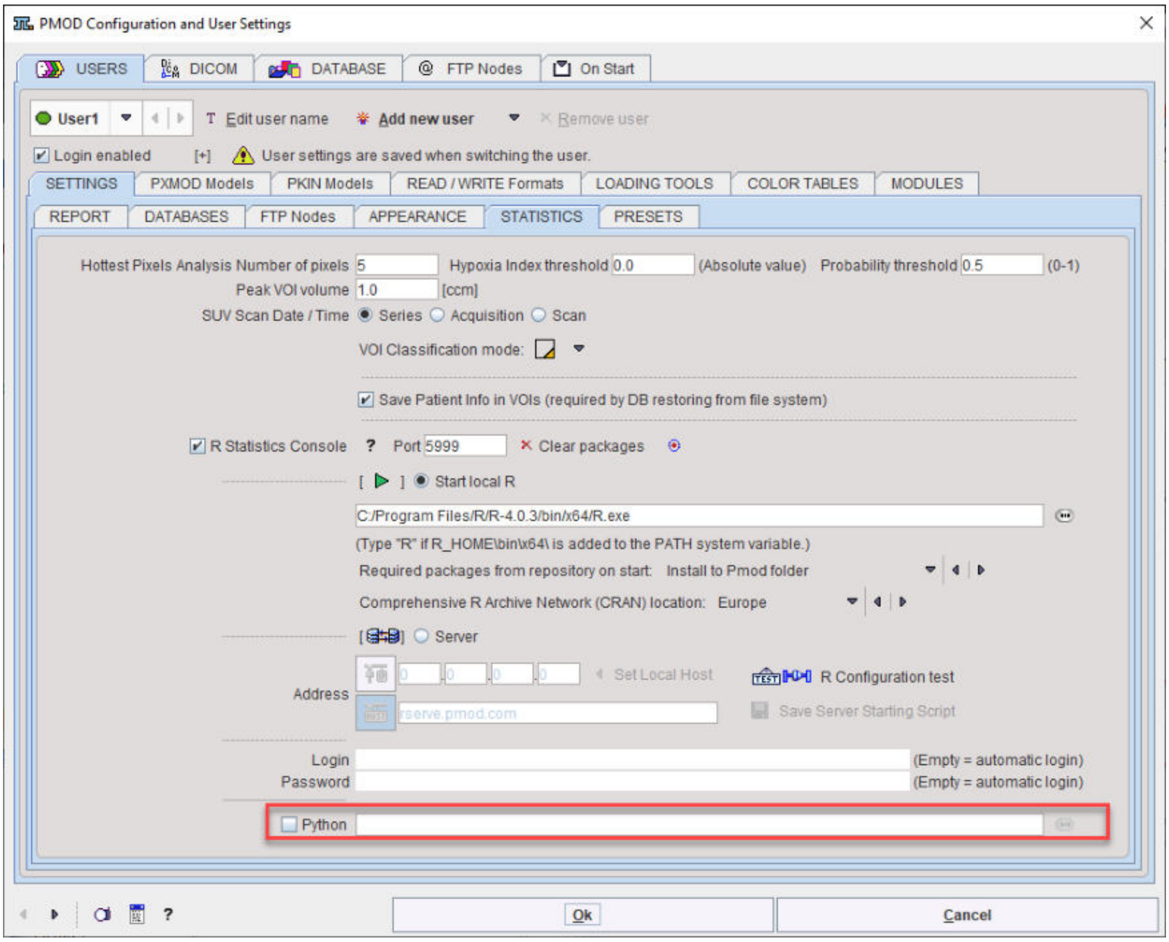


```
Connecting to the server localhost ... Connected.
Loading default settings ... Loaded.
Package doBy not loaded.
Package psych not loaded.
Package UsingR not loaded.
Package lawstat not loaded.
Package tseries not loaded.
Package np not loaded.
Package openair not loaded.
Package Hmisc not loaded.
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.
Package tseries not loaded.

Execution test ... Passed.
R version 4.0.3 (2020-10-10) 'Bunny-Wunnies Freak Out' Windows
Pmod PAI diagnostics test ... Passed.
Loading image data: 1/4 inputImages11 ... Loaded.
Loading image data: 2/4 inputImages12 ...
```

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:



PMOD Configuration and User Settings

USERS | DICOM | DATABASE | FTP Nodes | On Start

User1 | Edit user name | Add new user | Remove user

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

SETTINGS | PXMOD Models | PKIN Models | READ / WRITE Formats | LOADING TOOLS | COLOR TABLES | MODULES

REPORT | DATABASES | FTP Nodes | APPEARANCE | STATISTICS | PRESETS

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 ? Port 5999 Clear packages
[▶] ● Start local R
C:/Program Files/R/R-4.0.3/bin/x64/R.exe
(Type "R" if R_HOME\bin\x64\ is added to the PATH system variable.)
Required packages from repository on start: Install to Pmod folder
Comprehensive R Archive Network (CRAN) location: Europe
○ Server
Address 0 0 0 0 Set Local Host R Configuration test
rserve.pmod.com Save Server Starting Script
Login (Empty = automatic login)
Password (Empty = automatic login)

☐ Python

Ok | Cancel

Restart PMOD after setting the path to Python 3.8.

2.2 MacOS

MacOS Catalina is recommended for best compatibility with Rserve.

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/

Install TensorFlow using the Terminal command:

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

Check that tensorflow appears in the list of installed packages:

```
python3 -m pip list
```

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).

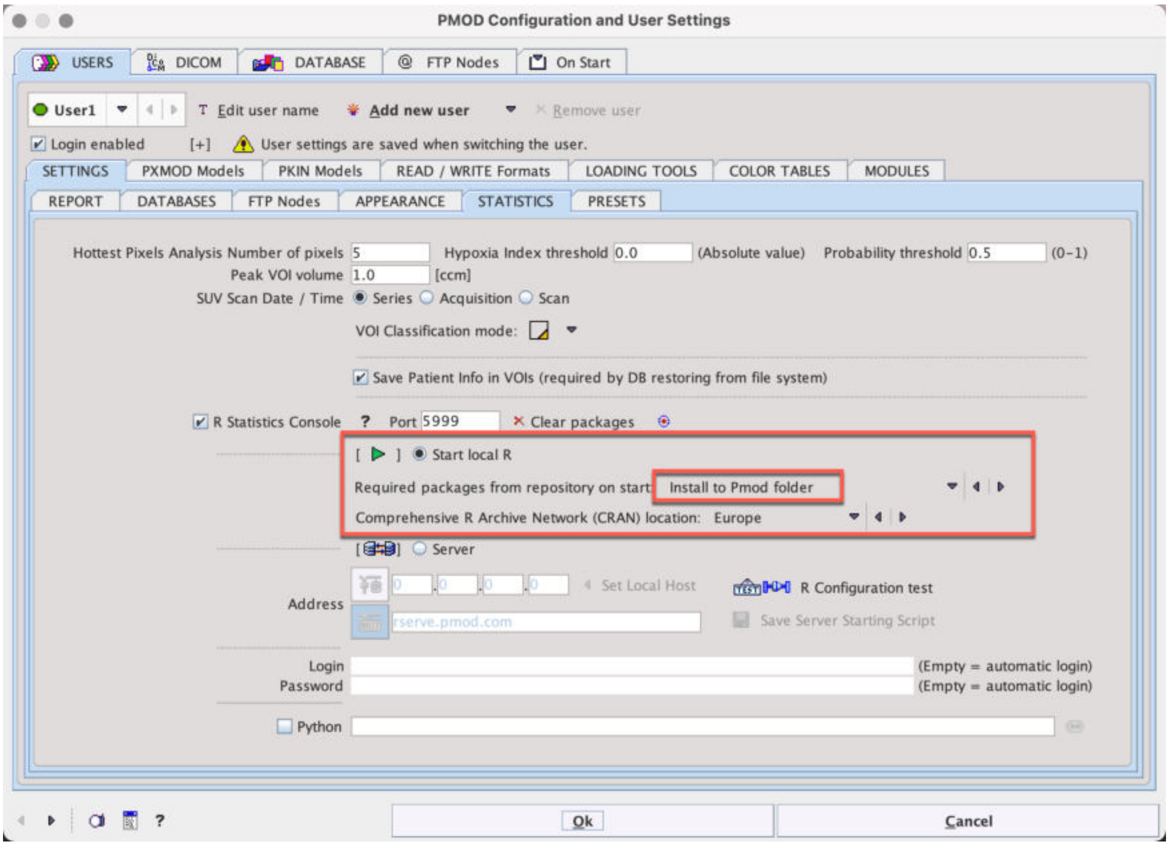
2.2.2 R Installation

Please download and install the most recent R version for MacOS from https://cran.r-project.org/ (note that our testing to date has used version 4.0.3).

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[11] below.

2.2.2.1 Default R Configuration

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



PMOD Configuration and User Settings

USERS | DICOM | DATABASE | FTP Nodes | On Start

User1 | Edit user name | Add new user | Remove user

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

SETTINGS | PXMOD Models | PKIN Models | READ / WRITE Formats | LOADING TOOLS | COLOR TABLES | MODULES

REPORT | DATABASES | FTP Nodes | APPEARANCE | STATISTICS | PRESETS

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 ? Port 5999 ✕ Clear packages

[▶] ● Start local R

Required packages from repository on start: Install to Pmod folder

Comprehensive R Archive Network (CRAN) location: Europe

[ ] ○ Server

Address 0 . 0 . 0 . 0 Set Local Host R Configuration test

rserve.pmod.com Save Server Starting Script

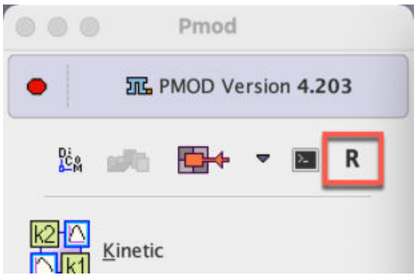
Login (Empty = automatic login)

Password (Empty = automatic login)

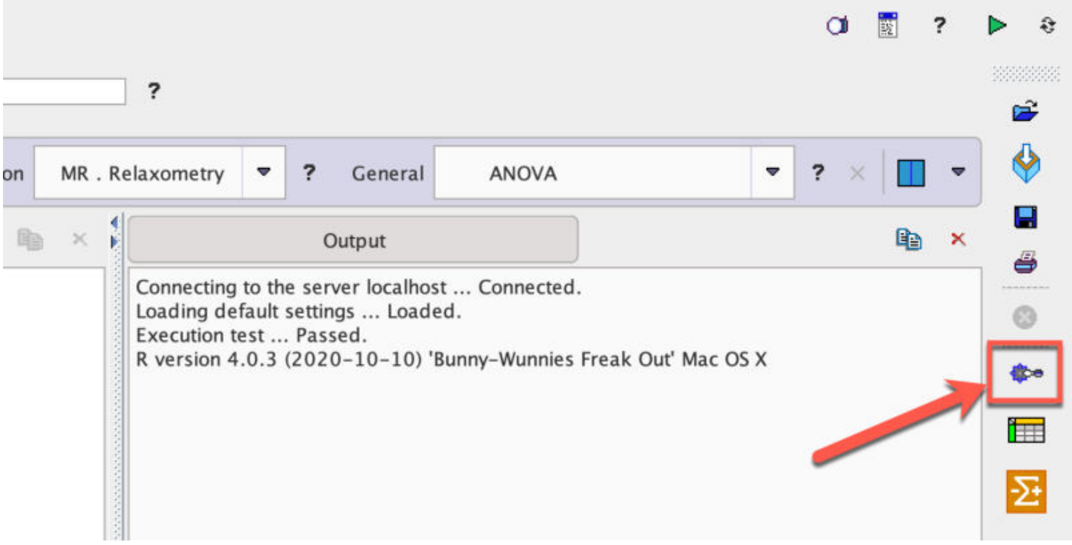
□ Python

On the **STATISTICS** tab select **Start local R** to ensure local execution. 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.

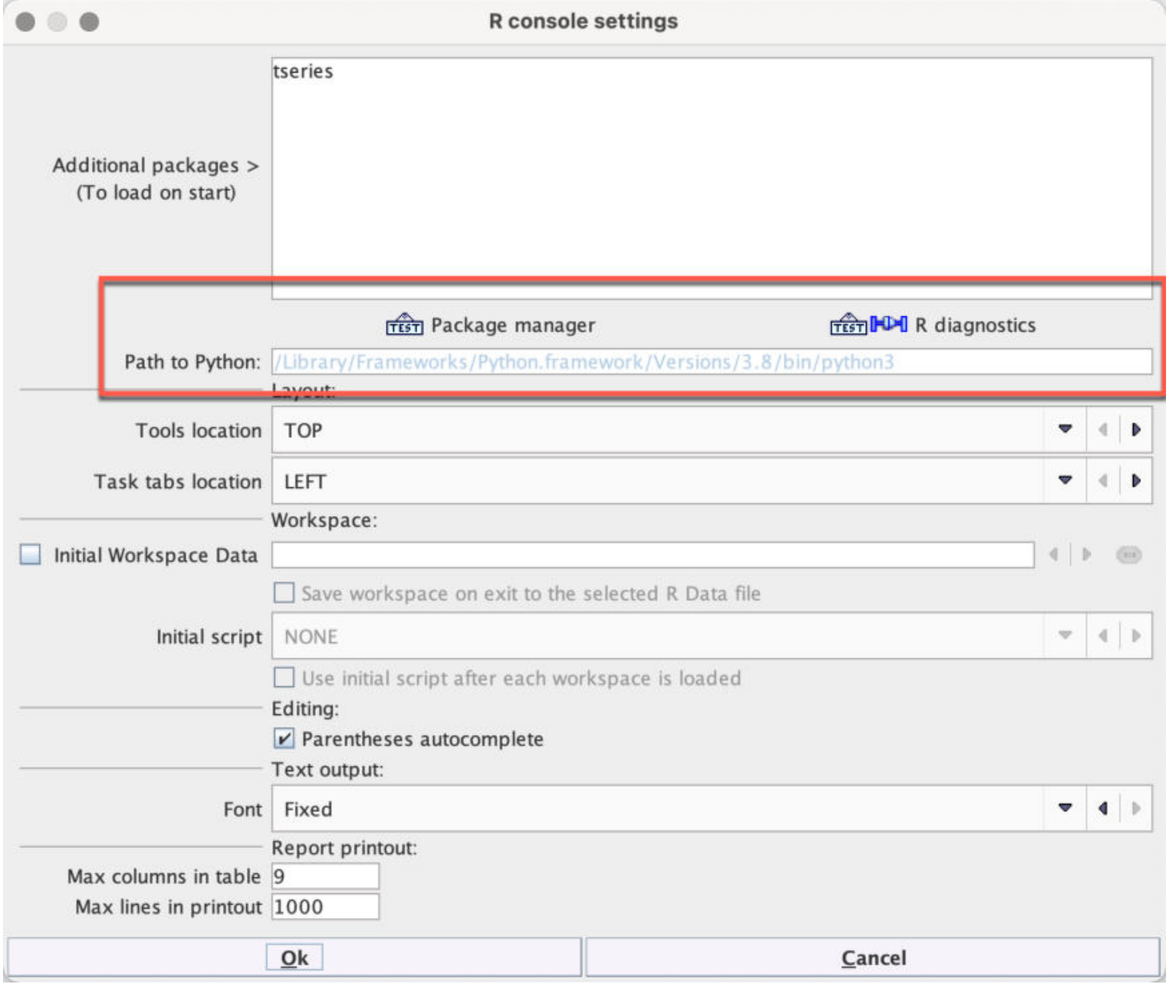


Then click on the **R** icon to open the **PMOD Console**. The required packages are downloaded and installed, followed by an execution test and printing of the R version information:



```
Connecting to the server localhost ... Connected.
Loading default settings ... Loaded.
Execution test ... Passed.
R version 4.0.3 (2020-10-10) 'Bunny-Wunnies Freak Out' Mac OS X
```

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



Additional packages > (To load on start): tseries

Path to Python: /Library/Frameworks/Python.framework/Versions/3.8/bin/python3

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:

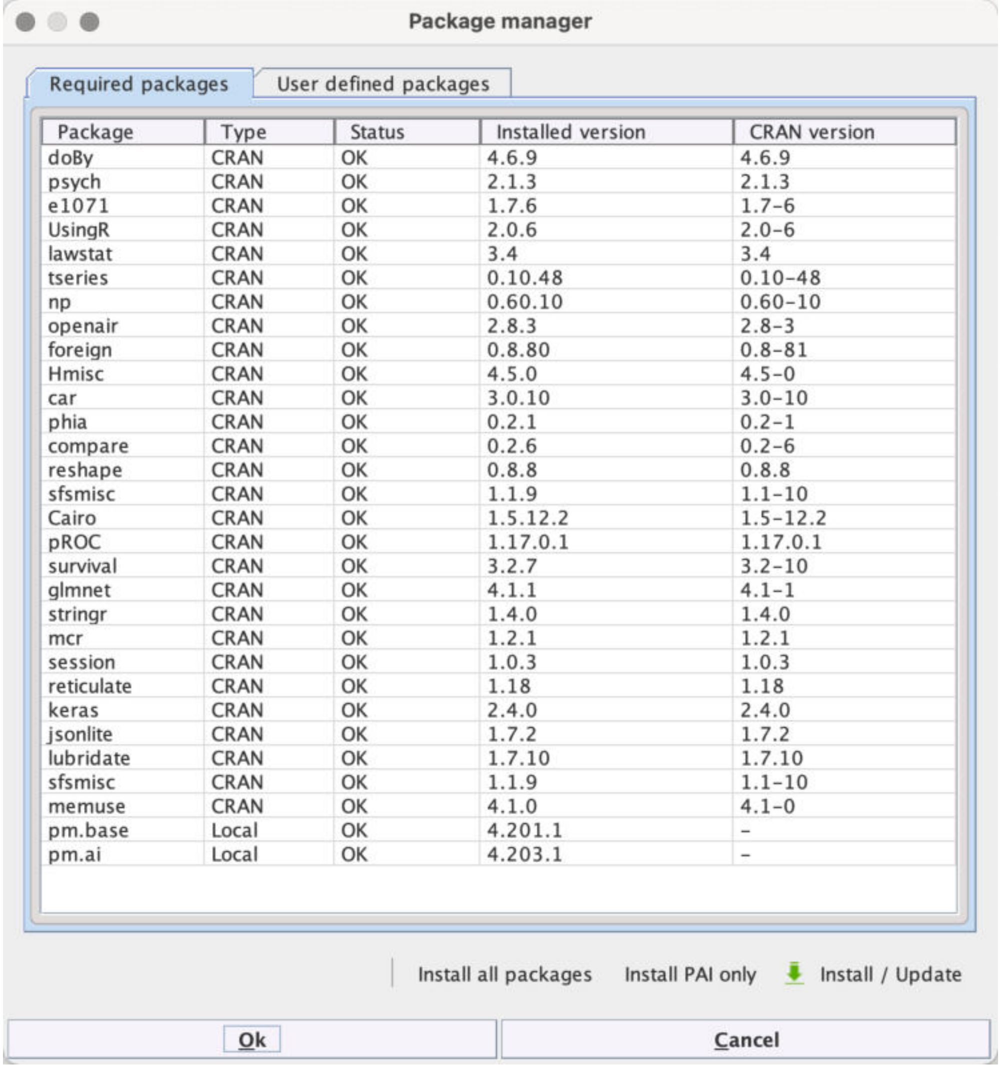
Font: Fixed

Report printout:

Max columns in table: 9

Max lines in printout: 1000

Please verify the **Path to Python** and open the **Package Manager**. All packages should have status **OK**.



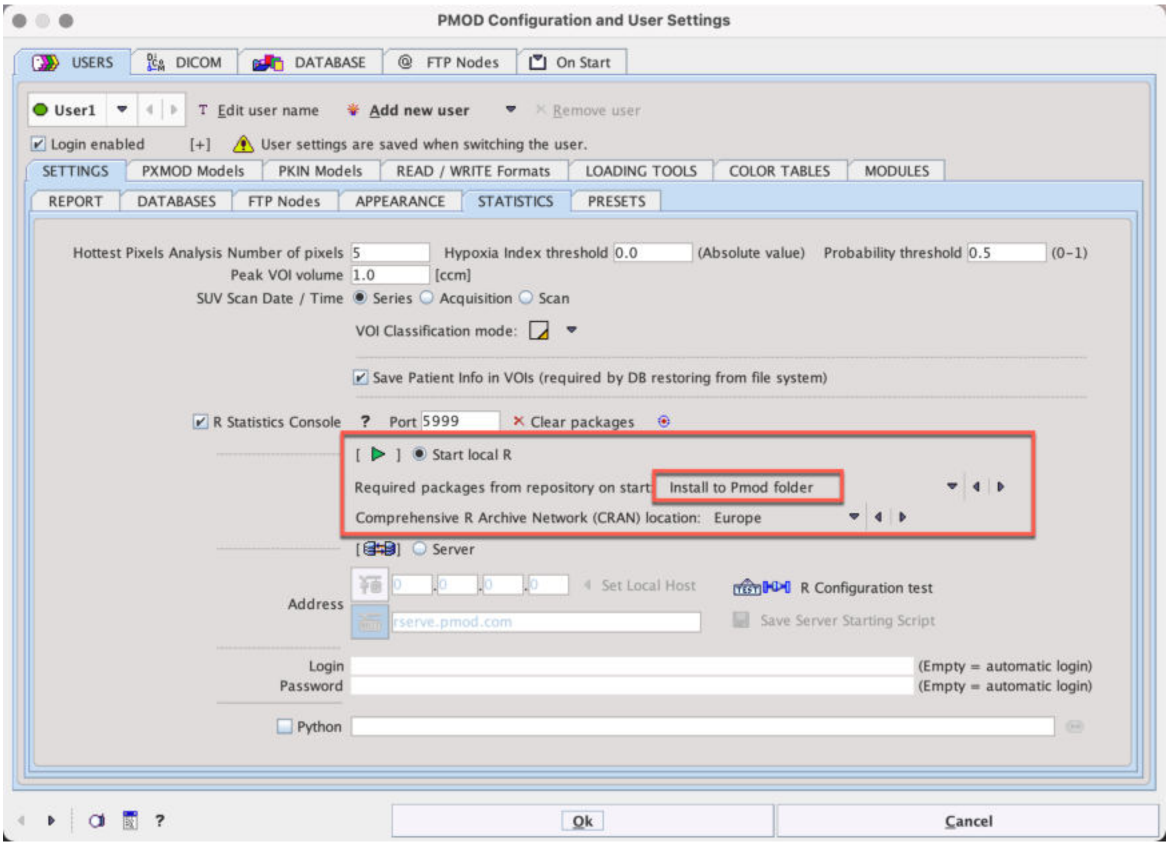
Package	Type	Status	Installed version	CRAN version
doBy	CRAN	OK	4.6.9	4.6.9
psych	CRAN	OK	2.1.3	2.1.3
e1071	CRAN	OK	1.7.6	1.7-6
UsingR	CRAN	OK	2.0.6	2.0-6
lawstat	CRAN	OK	3.4	3.4
tseries	CRAN	OK	0.10.48	0.10-48
np	CRAN	OK	0.60.10	0.60-10
openair	CRAN	OK	2.8.3	2.8-3
foreign	CRAN	OK	0.8.80	0.8-81
Hmisc	CRAN	OK	4.5.0	4.5-0
car	CRAN	OK	3.0.10	3.0-10
phia	CRAN	OK	0.2.1	0.2-1
compare	CRAN	OK	0.2.6	0.2-6
reshape	CRAN	OK	0.8.8	0.8.8
sfsmisc	CRAN	OK	1.1.9	1.1-10
Cairo	CRAN	OK	1.5.12.2	1.5-12.2
pROC	CRAN	OK	1.17.0.1	1.17.0.1
survival	CRAN	OK	3.2.7	3.2-10
glmnet	CRAN	OK	4.1.1	4.1-1
stringr	CRAN	OK	1.4.0	1.4.0
mcr	CRAN	OK	1.2.1	1.2.1
session	CRAN	OK	1.0.3	1.0.3
reticulate	CRAN	OK	1.18	1.18
keras	CRAN	OK	2.4.0	2.4.0
jsonlite	CRAN	OK	1.7.2	1.7.2
lubridate	CRAN	OK	1.7.10	1.7.10
sfsmisc	CRAN	OK	1.1.9	1.1-10
memuse	CRAN	OK	4.1.0	4.1-0
pm.base	Local	OK	4.201.1	-
pm.ai	Local	OK	4.203.1	-

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.



PMOD Configuration and User Settings

☑ 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 ? Port 5999 × Clear packages

[▶] ● Start local R

Required packages from repository on start: Install to Pmod folder

Comprehensive R Archive Network (CRAN) location: Europe

[ ] ○ Server

Address 0 . 0 . 0 . 0 Set Local Host R Configuration test

rserve.pmod.com Save Server Starting Script

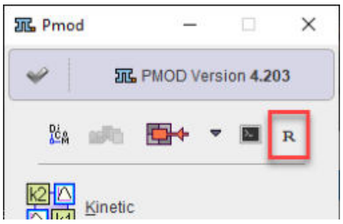
Login (Empty = automatic login)

Password (Empty = automatic login)

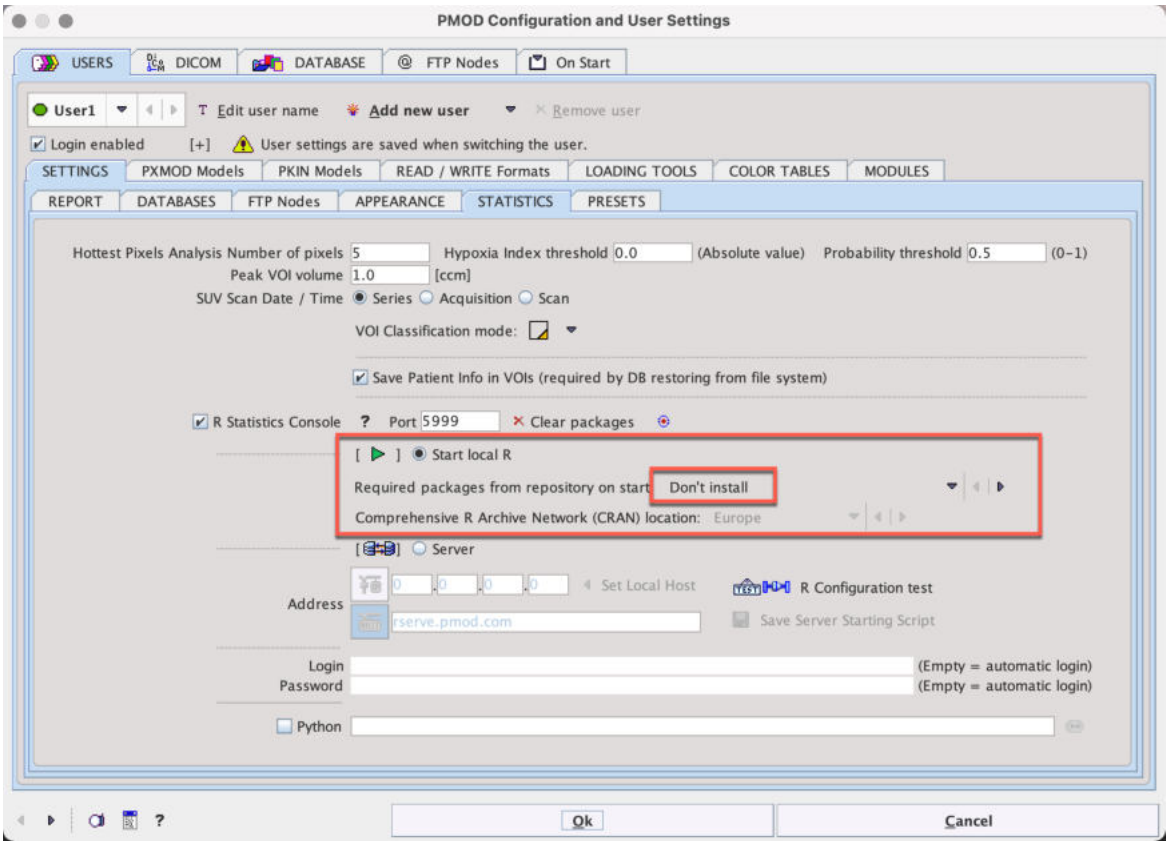
☐ Python

On the **STATISTICS** tab select **Start local R** to ensure local execution and verify that the path to the local R installation is correct. Select **Install to Pmod folder** to install the base required packages on PMOD restart.

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



Return to the **Configuration** facility from the main ToolBox and change the R package installation selection to Don't install:



PMOD Configuration and User Settings

User1

☑ 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 ? Port 5999 ✕ Clear packages

[▶] ● Start local R

Required packages from repository on start Don't install

Comprehensive R Archive Network (CRAN) location: Europe

○ Server

Address 0 .0 .0 .0 Set Local Host R Configuration test

rserve.pmod.com Save Server Starting Script

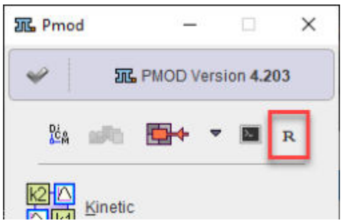
Login (Empty = automatic login)

Password (Empty = automatic login)

□ Python

On the **STATISTICS** tab select **Start local R** to ensure local execution and verify that the path to the local R installation is correct. Select **Don't install** for the R packages as illustrated above.

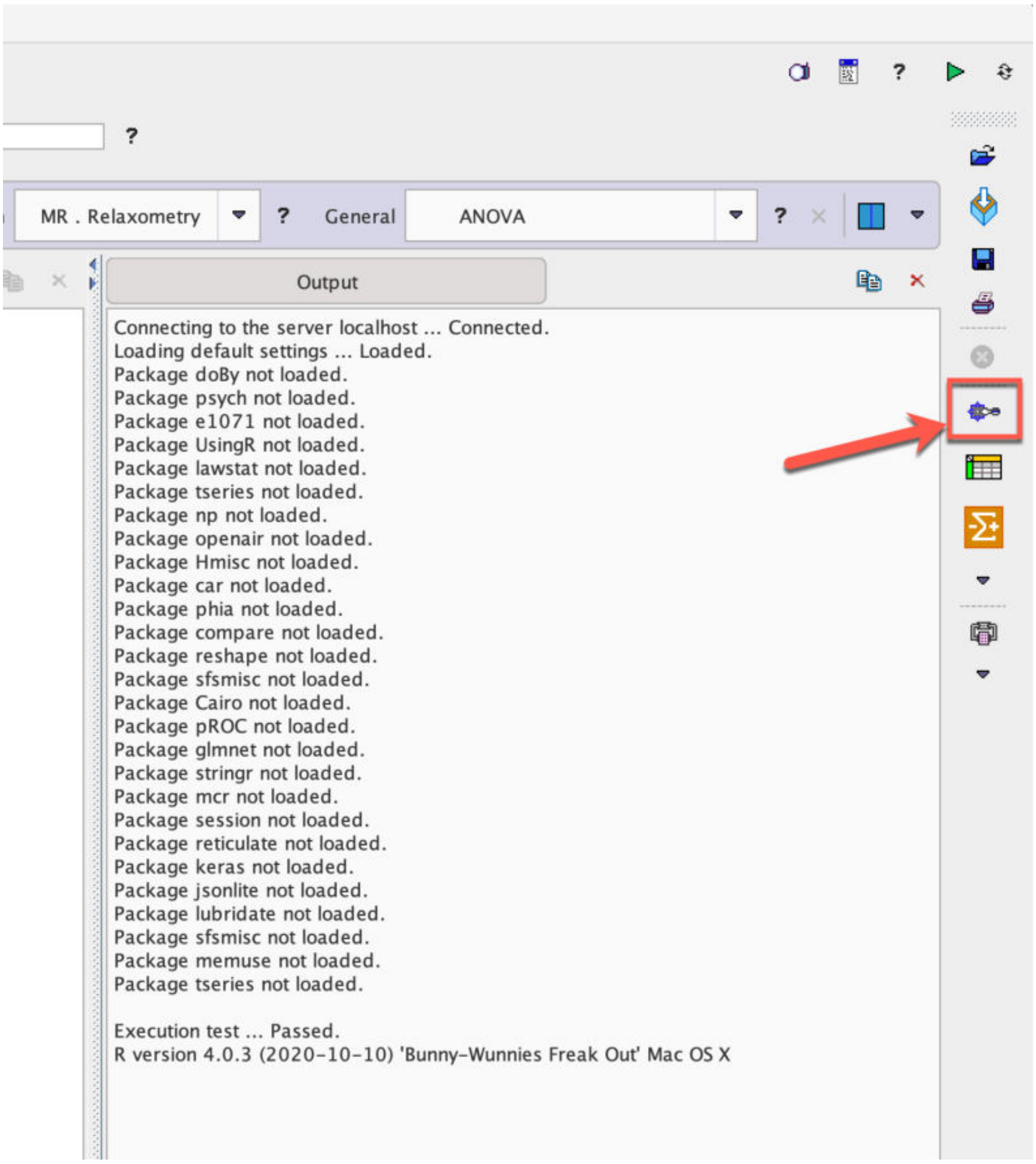
Restart PMOD and wait for the **R** icon on the main ToolBox to become active.



Pmod

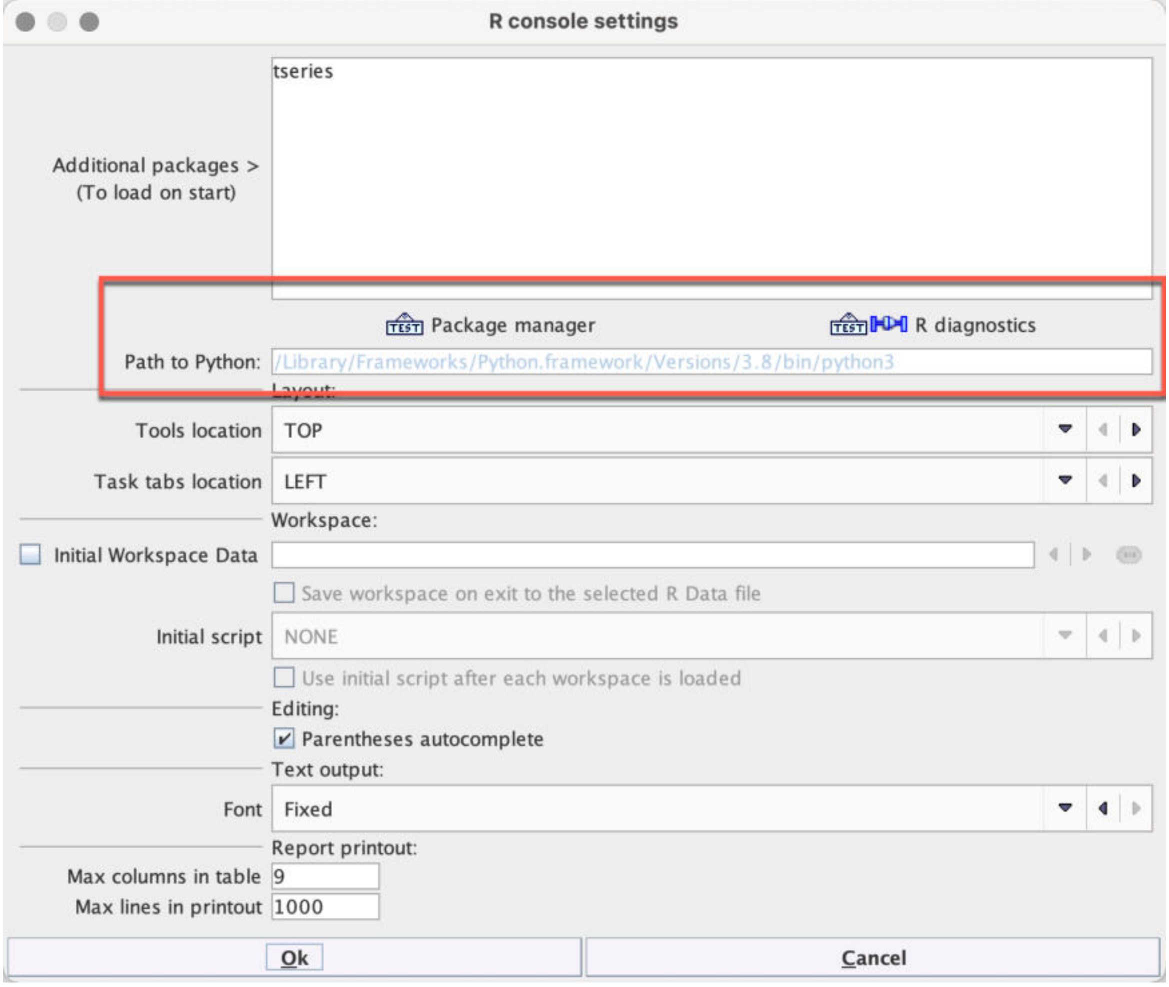
PMOD Version 4.203

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:



```
Connecting to the server localhost ... Connected.
Loading default settings ... 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 session not loaded.
Package reticulate not loaded.
Package keras not loaded.
Package jsonlite not loaded.
Package lubridate not loaded.
Package sfsmisc not loaded.
Package memuse not loaded.
Package tseries not loaded.

Execution test ... Passed.
R version 4.0.3 (2020-10-10) 'Bunny-Wunnies Freak Out' Mac OS X
```

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



R console settings

Additional packages > (To load on start): tseries

Path to Python: /Library/Frameworks/Python.framework/Versions/3.8/bin/python3

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:

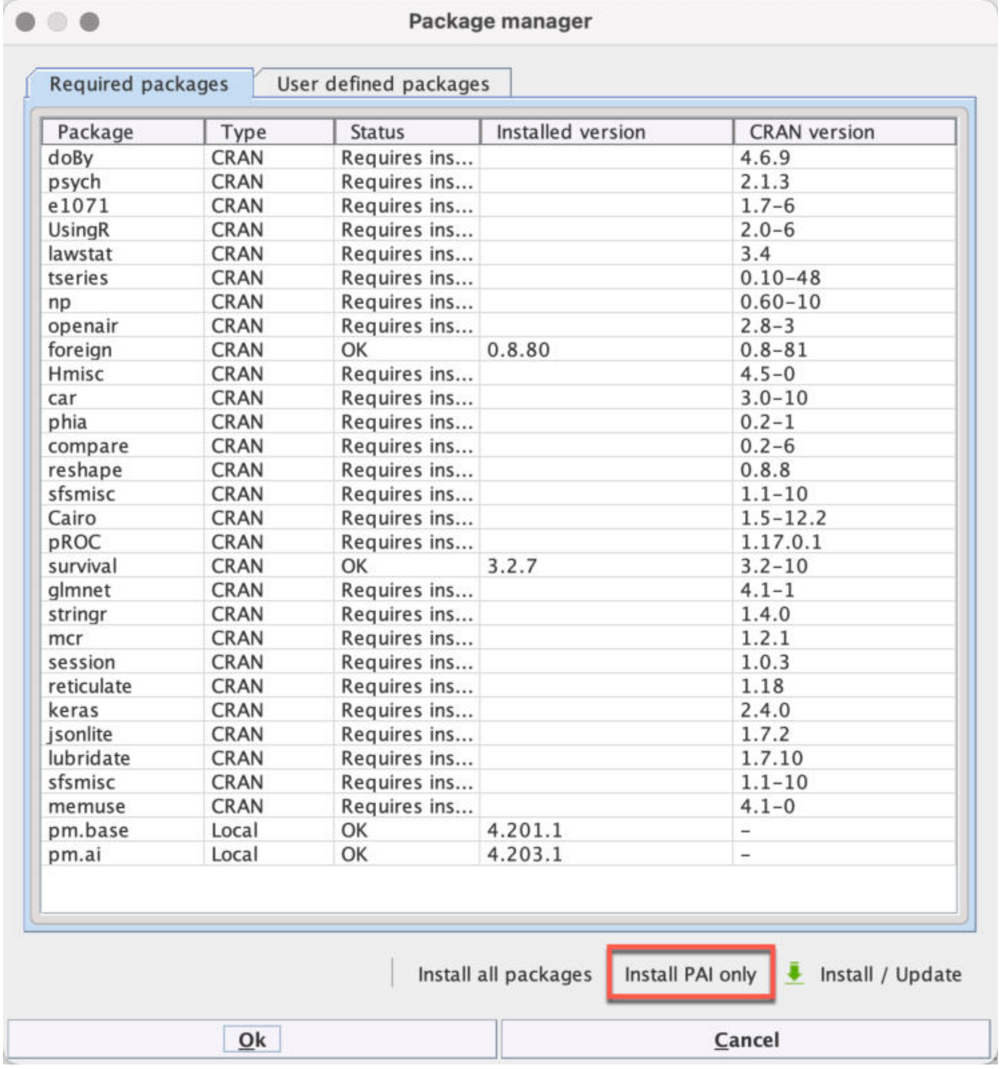
Font: Fixed

Report printout:

Max columns in table: 9

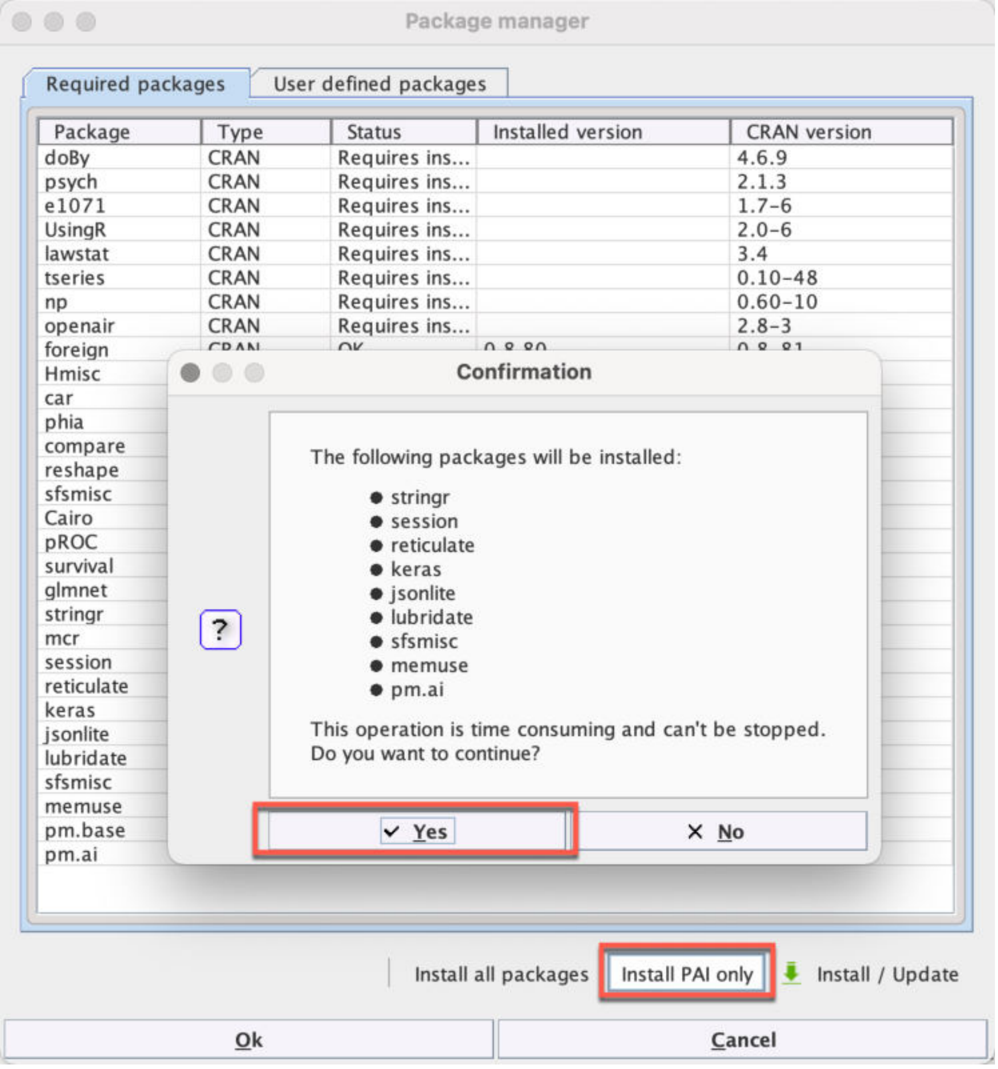
Max lines in printout: 1000

Please verify the **Path to Python** and open the **Package Manager**. Most packages will have **Requires installation** in the **Status** column



Package	Type	Status	Installed version	CRAN version
doBy	CRAN	Requires ins...		4.6.9
psych	CRAN	Requires ins...		2.1.3
e1071	CRAN	Requires ins...		1.7-6
UsingR	CRAN	Requires ins...		2.0-6
lawstat	CRAN	Requires ins...		3.4
tseries	CRAN	Requires ins...		0.10-48
np	CRAN	Requires ins...		0.60-10
openair	CRAN	Requires ins...		2.8-3
foreign	CRAN	OK	0.8.80	0.8-81
Hmisc	CRAN	Requires ins...		4.5-0
car	CRAN	Requires ins...		3.0-10
phia	CRAN	Requires ins...		0.2-1
compare	CRAN	Requires ins...		0.2-6
reshape	CRAN	Requires ins...		0.8.8
sfsmisc	CRAN	Requires ins...		1.1-10
Cairo	CRAN	Requires ins...		1.5-12.2
pROC	CRAN	Requires ins...		1.17.0.1
survival	CRAN	OK	3.2.7	3.2-10
glmnet	CRAN	Requires ins...		4.1-1
stringr	CRAN	Requires ins...		1.4.0
mcr	CRAN	Requires ins...		1.2.1
session	CRAN	Requires ins...		1.0.3
reticulate	CRAN	Requires ins...		1.18
keras	CRAN	Requires ins...		2.4.0
jsonlite	CRAN	Requires ins...		1.7.2
lubridate	CRAN	Requires ins...		1.7.10
sfsmisc	CRAN	Requires ins...		1.1-10
memuse	CRAN	Requires ins...		4.1-0
pm.base	Local	OK	4.201.1	-
pm.ai	Local	OK	4.203.1	-

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



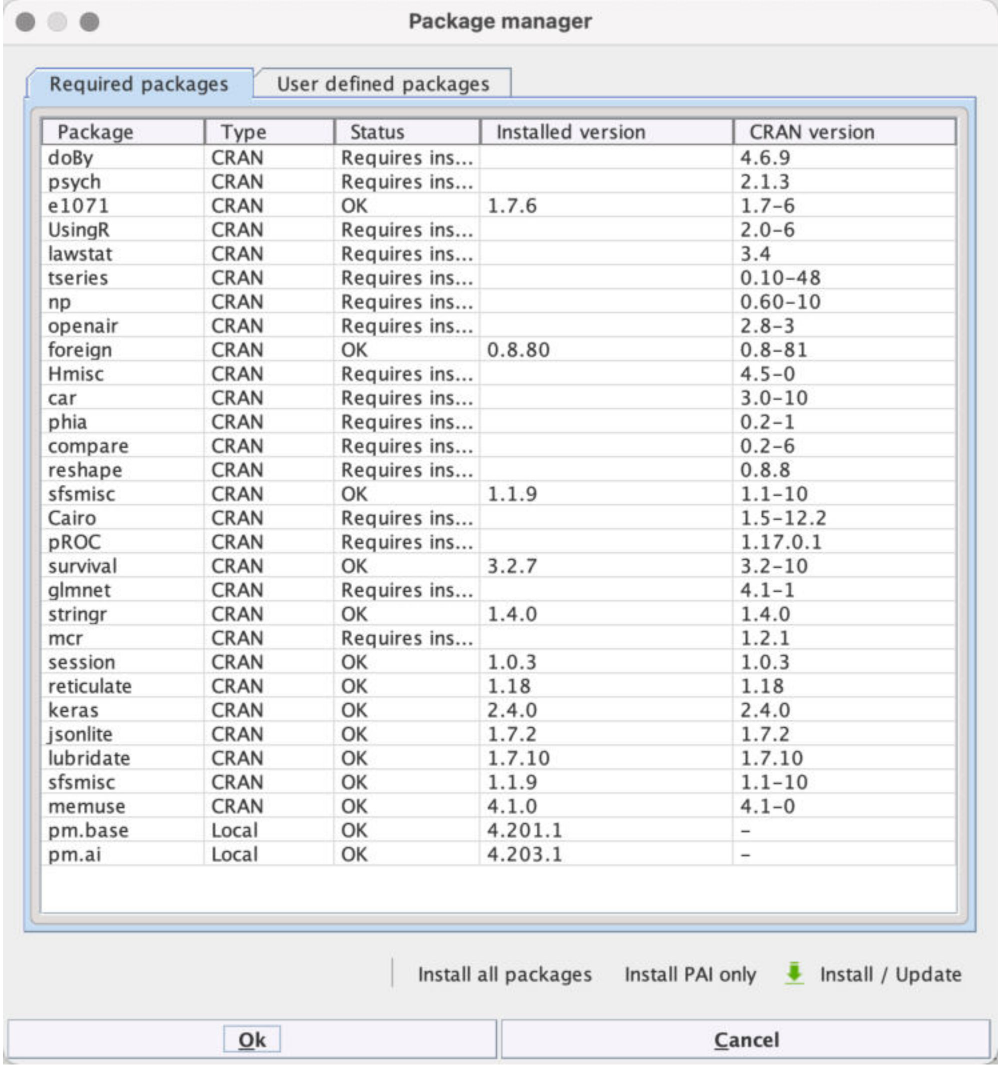
Package	Type	Status	Installed version	CRAN version
doBy	CRAN	Requires ins...		4.6.9
psych	CRAN	Requires ins...		2.1.3
e1071	CRAN	Requires ins...		1.7-6
UsingR	CRAN	Requires ins...		2.0-6
lawstat	CRAN	Requires ins...		3.4
tseries	CRAN	Requires ins...		0.10-48
np	CRAN	Requires ins...		0.60-10
openair	CRAN	Requires ins...		2.8-3
foreign	CRAN	OK	0.8-80	0.8-81
Hmisc				
car				
phia				
compare				
reshape				
sfsmisc				
Cairo				
pROC				
survival				
glmnet				
stringr				
mcr				
session				
reticulate				
keras				
jsonlite				
lubridate				
sfsmisc				
memuse				
pm.base				
pm.ai				

The following packages will be installed:

- stringr
- session
- reticulate
- keras
- jsonlite
- lubridate
- sfsmisc
- memuse
- pm.ai

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

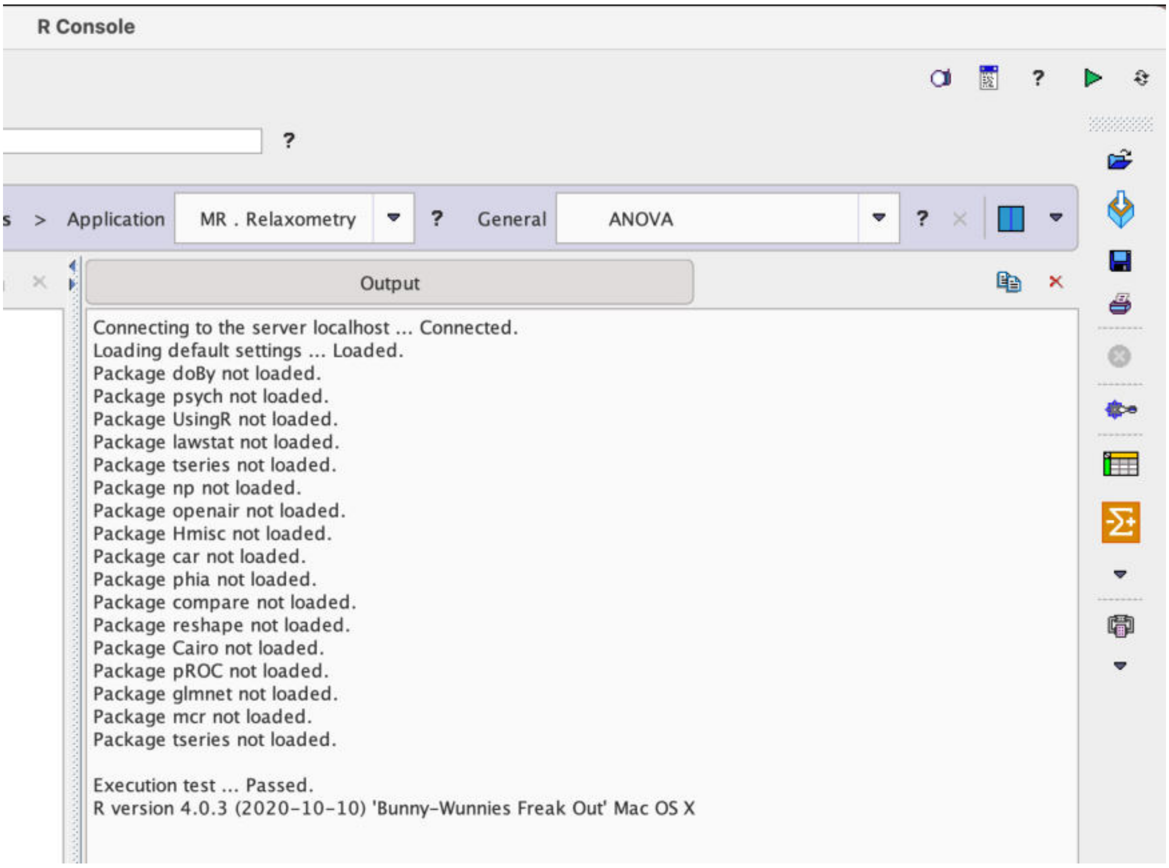
Following installation the **Status** entries will be updated:



Package	Type	Status	Installed version	CRAN version
doBy	CRAN	Requires ins...		4.6.9
psych	CRAN	Requires ins...		2.1.3
e1071	CRAN	OK	1.7.6	1.7-6
UsingR	CRAN	Requires ins...		2.0-6
lawstat	CRAN	Requires ins...		3.4
tseries	CRAN	Requires ins...		0.10-48
np	CRAN	Requires ins...		0.60-10
openair	CRAN	Requires ins...		2.8-3
foreign	CRAN	OK	0.8.80	0.8-81
Hmisc	CRAN	Requires ins...		4.5-0
car	CRAN	Requires ins...		3.0-10
phia	CRAN	Requires ins...		0.2-1
compare	CRAN	Requires ins...		0.2-6
reshape	CRAN	Requires ins...		0.8.8
sfsmisc	CRAN	OK	1.1.9	1.1-10
Cairo	CRAN	Requires ins...		1.5-12.2
pROC	CRAN	Requires ins...		1.17.0.1
survival	CRAN	OK	3.2.7	3.2-10
glmnet	CRAN	Requires ins...		4.1-1
stringr	CRAN	OK	1.4.0	1.4.0
mcr	CRAN	Requires ins...		1.2.1
session	CRAN	OK	1.0.3	1.0.3
reticulate	CRAN	OK	1.18	1.18
keras	CRAN	OK	2.4.0	2.4.0
jsonlite	CRAN	OK	1.7.2	1.7.2
lubridate	CRAN	OK	1.7.10	1.7.10
sfsmisc	CRAN	OK	1.1.9	1.1-10
memuse	CRAN	OK	4.1.0	4.1-0
pm.base	Local	OK	4.201.1	-
pm.ai	Local	OK	4.203.1	-

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

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

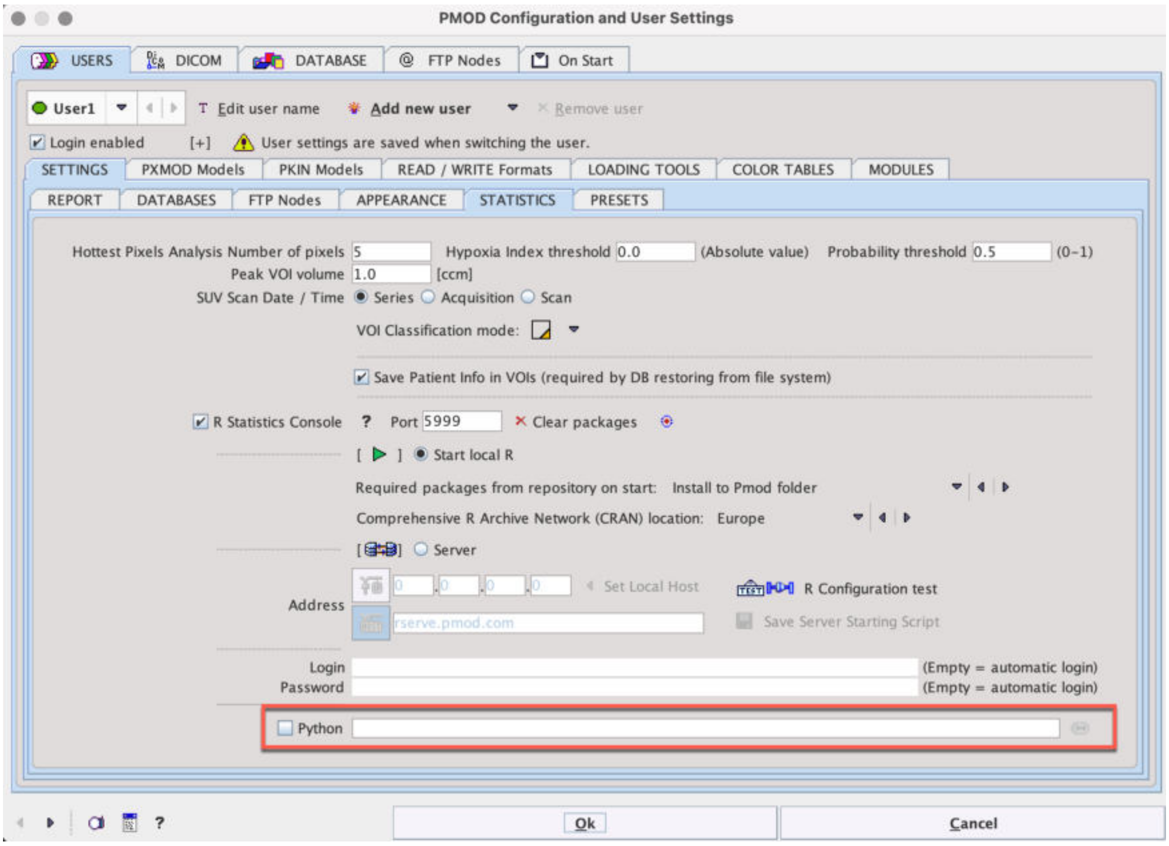


```
Connecting to the server localhost ... Connected.
Loading default settings ... Loaded.
Package doBy not loaded.
Package psych not loaded.
Package UsingR not loaded.
Package lawstat not loaded.
Package tseries not loaded.
Package np not loaded.
Package openair not loaded.
Package Hmisc not loaded.
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.
Package tseries not loaded.

Execution test ... Passed.
R version 4.0.3 (2020-10-10) 'Bunny-Wunnies Freak Out' Mac OS X
```

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:



User1

☑ 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 ? Port 5999

[▶] ● Start local R

Required packages from repository on start: Install to Pmod folder

Comprehensive R Archive Network (CRAN) location: Europe

○ Server

Address 0 0 0 0

rserve.pmod.com

Login (Empty = automatic login)

Password (Empty = automatic login)

□ Python

Restart PMOD after setting the path to Python 3.8.

2.3 Linux Platforms

2.3.1 R Installation

PAI requires R version 4.0.3 or higher. 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("session")

install.packages("keras")
install.packages("Rserve",,"http://rforge.net")
```

Rserve Configuration

Create the configuration file “/etc/Rserv.conf”.

Create the following content in the configuration file (enables TCP communication with R using port 5999):

```
remote enable port 5999

fileio disable

encoding utf8
```

Ensure that the port specified in the configuration file is not used by other applications and is enabled in the firewall. Alternatively a configuration file with user authentication can be used as follows:

```
remote enable

port 5999

fileio disable

encoding utf8

pwdfile /etc/Rserv.pwd

auth required

plaintext disable
```

The password file Rserv.pwd contains the defined users and passwords. It is a plain text file with tab-separated values:

```
myUser1 myPassword1

myUser2 myPassword2
```

Load and start Rserve from R:

```
library("Rserve");

Rserve();
```

Note: depending on how R is being run, it may require additional parameters. In that case the parameter should be passed in the args argument such as:

```
Rserve(args=’--no-save’);
```

2.3.2 Python and TensorFlow Installation

Python version 3.8 is required. Install Python and TensorFlow as follows:

```
sudo apt update sudo apt upgrade

sudo apt install build-essential
sudo apt install python3

sudo apt install python3-pip

pip3 install --upgrade pip

pip3 install tensorflow
```

Note: TensorFlow can alternatively be installed in Python’s virtual environment.

3 *Preparation of Training Data and Neural Network*

3.1 Data Preparation

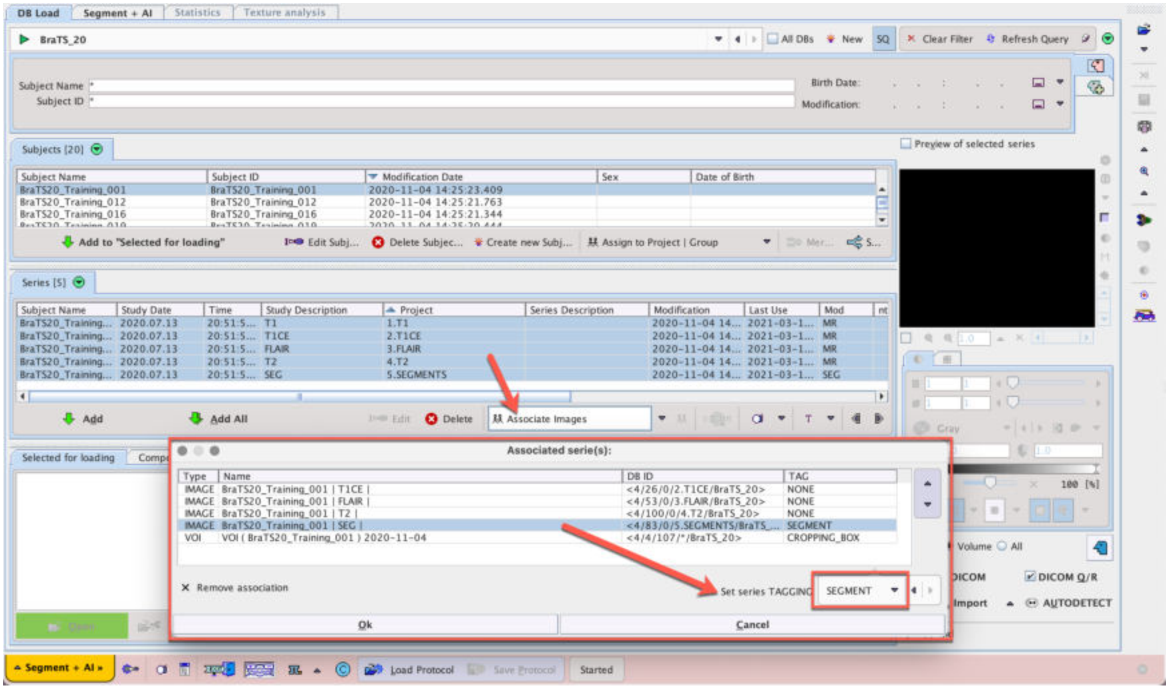
Image Import

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 the segmentation reference result 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 [20]

Subject Name	Subject ID	Modification Date	Sex	Date of Birth
BraTS20_Training_001	BraTS20_Training_001	2020-11-04 14:25:23.409		
BraTS20_Training_012	BraTS20_Training_012	2020-11-04 14:25:21.763		
BraTS20_Training_016	BraTS20_Training_016	2020-11-04 14:25:21.344		

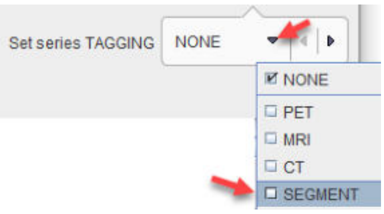
Series [5]

Subject Name	Study Date	Time	Study Description	Project	Series Description	Modification	Last Use	Mod
BraTS20_Training...	2020.07.13	20:51:5...	T1	1.T1		2020-11-04 14...	2021-03-1...	MR
BraTS20_Training...	2020.07.13	20:51:5...	T1CE	2.T1CE		2020-11-04 14...	2021-03-1...	MR
BraTS20_Training...	2020.07.13	20:51:5...	FLAIR	3.FLAIR		2020-11-04 14...	2021-03-1...	MR
BraTS20_Training...	2020.07.13	20:51:5...	T2	4.T2		2020-11-04 14...	2021-03-1...	MR
BraTS20_Training...	2020.07.13	20:51:5...	SEG	5.SEGMENTS		2020-11-04 14...	2021-03-1...	SEG

Associated serie(s):

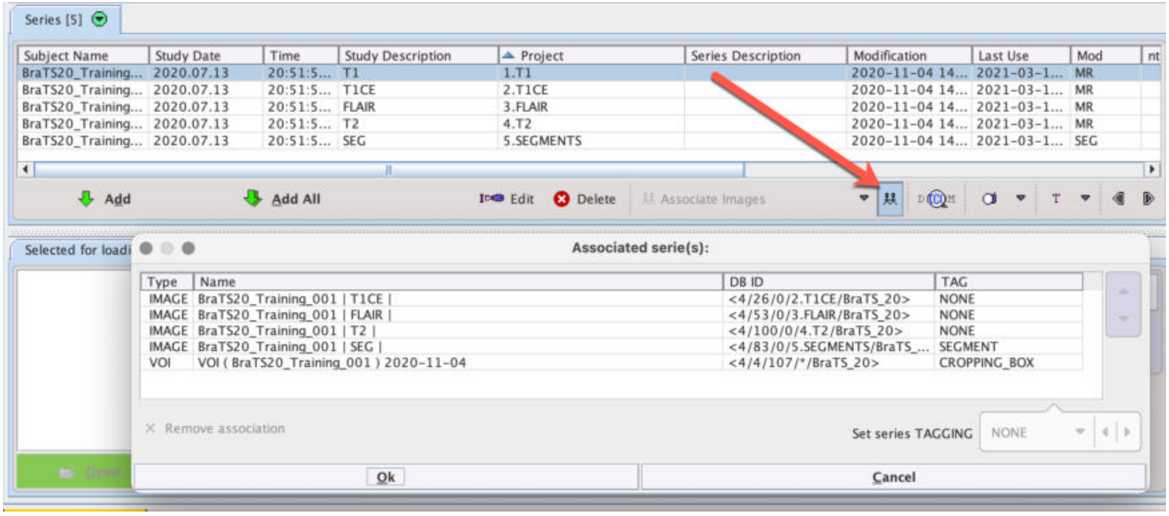
Type	Name	DB ID	TAG
IMAGE	BraTS20_Training_001 | T1CE |	<4/26/0/2.T1CE/BraTS_20>	NONE
IMAGE	BraTS20_Training_001 | FLAIR |	<4/53/0/3.FLAIR/BraTS_20>	NONE
IMAGE	BraTS20_Training_001 | T2 |	<4/100/0/4.T2/BraTS_20>	NONE
IMAGE	BraTS20_Training_001 | SEG |	<4/83/0/5.SEGMENTS/BraTS_...	SEGMENT
VOI	VOI (BraTS20_Training_001) 2020-11-04	<4/4/107/*/BraTS_20>	CROPPING_BOX

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	Time	Study Description	Project	Series Description	Modification	Last Use	Mod	nt
BraTS20_Training...	2020.07.13	20:51:5...	T1	1.T1		2020-11-04 14...	2021-03-1...	MR	
BraTS20_Training...	2020.07.13	20:51:5...	T1CE	2.T1CE		2020-11-04 14...	2021-03-1...	MR	
BraTS20_Training...	2020.07.13	20:51:5...	FLAIR	3.FLAIR		2020-11-04 14...	2021-03-1...	MR	
BraTS20_Training...	2020.07.13	20:51:5...	T2	4.T2		2020-11-04 14...	2021-03-1...	MR	
BraTS20_Training...	2020.07.13	20:51:5...	SEG	5.SEGMENTS		2020-11-04 14...	2021-03-1...	SEG	

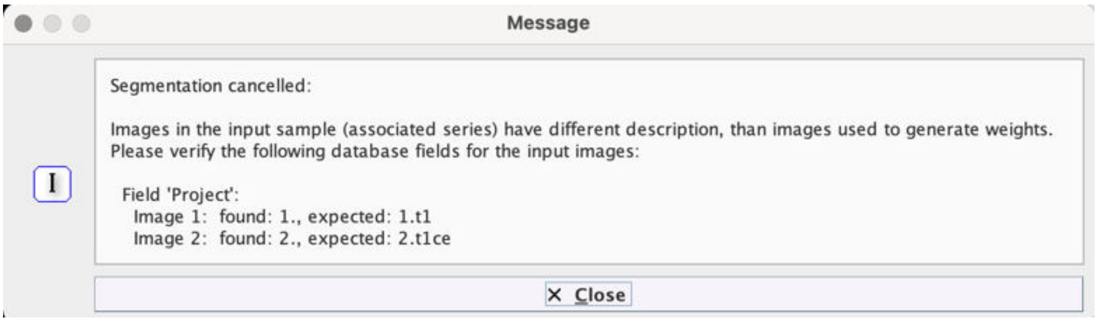
Associated serie(s):

Type	Name	DB ID	TAG
IMAGE	BraTS20_Training_001 | T1CE |	<4/26/0/2.T1CE/BraTS_20>	NONE
IMAGE	BraTS20_Training_001 | FLAIR |	<4/53/0/3.FLAIR/BraTS_20>	NONE
IMAGE	BraTS20_Training_001 | T2 |	<4/100/0/4.T2/BraTS_20>	NONE
IMAGE	BraTS20_Training_001 | SEG |	<4/83/0/5.SEGMENTS/BraTS_...	SEGMENT
VOI	VOI (BraTS20_Training_001) 2020-11-04	<4/4/107/*/BraTS_20>	CROPPING_BOX

Adding a Descriptive Variable for Training (Project Description)

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:



Message

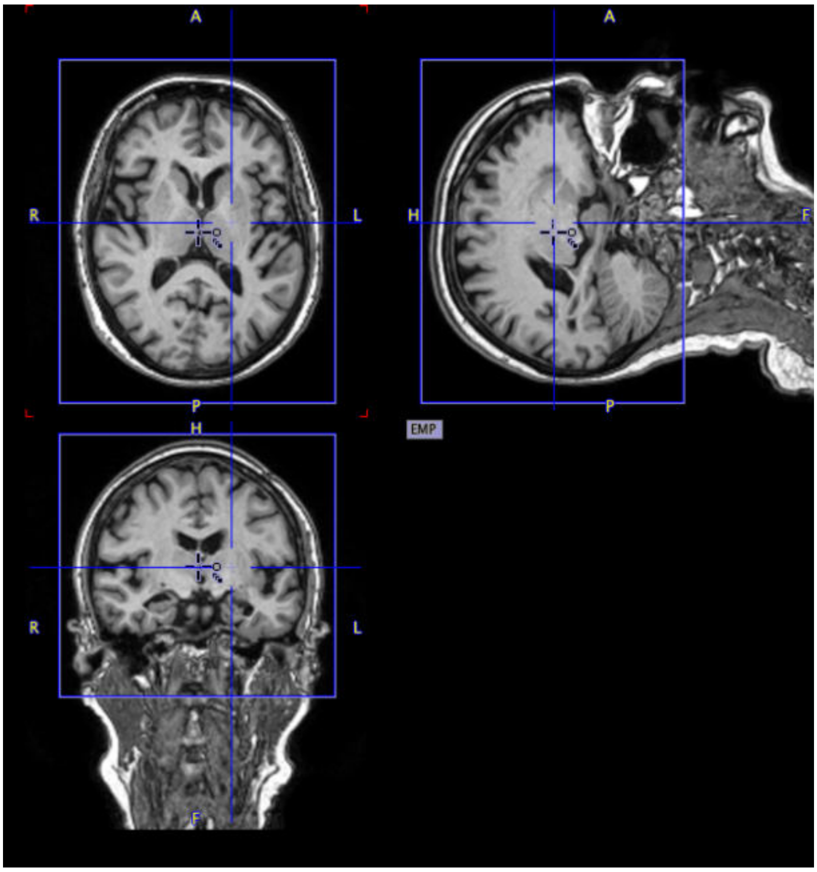
Segmentation cancelled:

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:

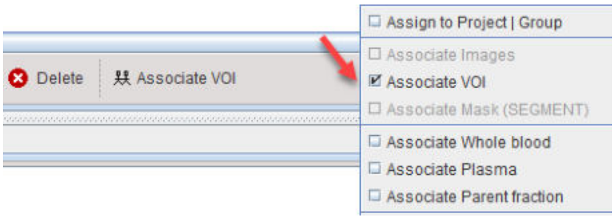
Field 'Project':
Image 1: found: 1., expected: 1.t1
Image 2: found: 2., expected: 2.t1ce

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.



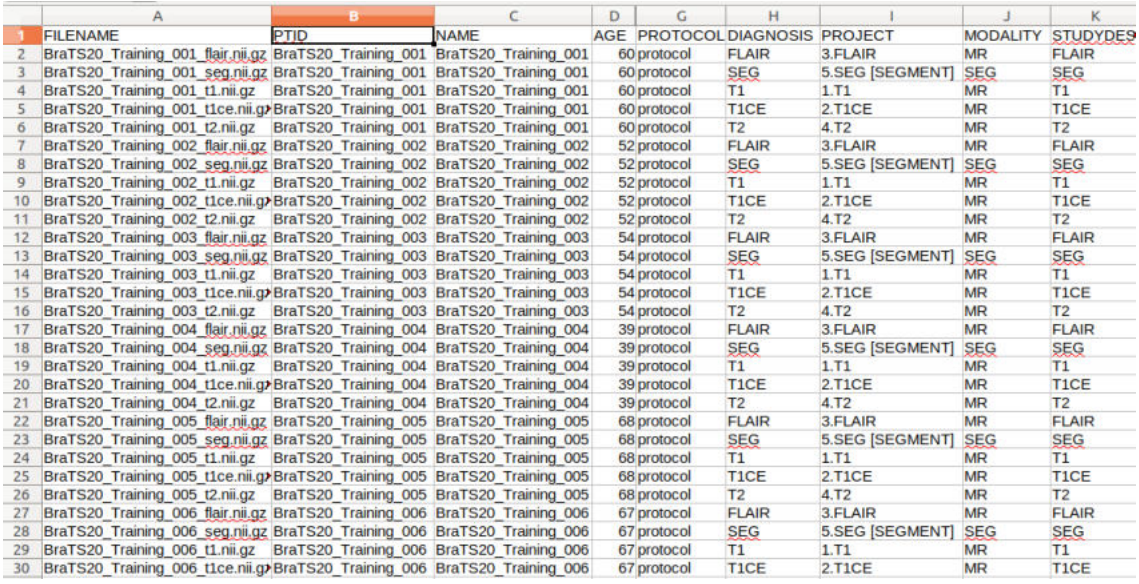
To achieve this, open the input image, create a suitable VOI such as a box, position it properly and save it to the database. Then select the input image in the **Series** list on the DB Load page, followed by **Associate VOI** from the same menu where the images were associated.



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

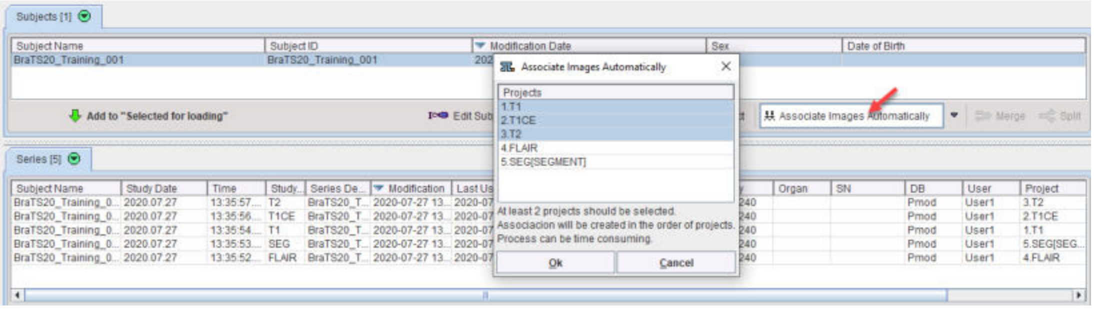
Automatic Association Creation

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 the **Incoming Folder** method. 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
BraTS20_Training_001	BraTS20_Training_001	202		

Associate Images Automatically

Projects
1.T1
2.T1CE
3.T2
4.FLAIR
5.SEG[SEGMENT]

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

Subject Name	Study Date	Time	Study..	Series De...	Modification	Last Us		Organ	SN	DB	User	Project
BraTS20_Training_0...	2020.07.27	13:35:57....	T2	BraTS20_T...	2020-07-27 13...	2020-07	240			Pmod	User1	3.T2
BraTS20_Training_0...	2020.07.27	13:35:56....	T1CE	BraTS20_T...	2020-07-27 13...	2020-07	240			Pmod	User1	2.T1CE
BraTS20_Training_0...	2020.07.27	13:35:54....	T1	BraTS20_T...	2020-07-27 13...	2020-07	240			Pmod	User1	1.T1
BraTS20_Training_0...	2020.07.27	13:35:53....	SEG	BraTS20_T...	2020-07-27 13...	2020-07	240			Pmod	User1	5.SEG[SEG...
BraTS20_Training_0...	2020.07.27	13:35:52....	FLAIR	BraTS20_T...	2020-07-27 13...	2020-07	240			Pmod	User1	4.FLAIR

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 above 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.

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.

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.
- “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
- 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:

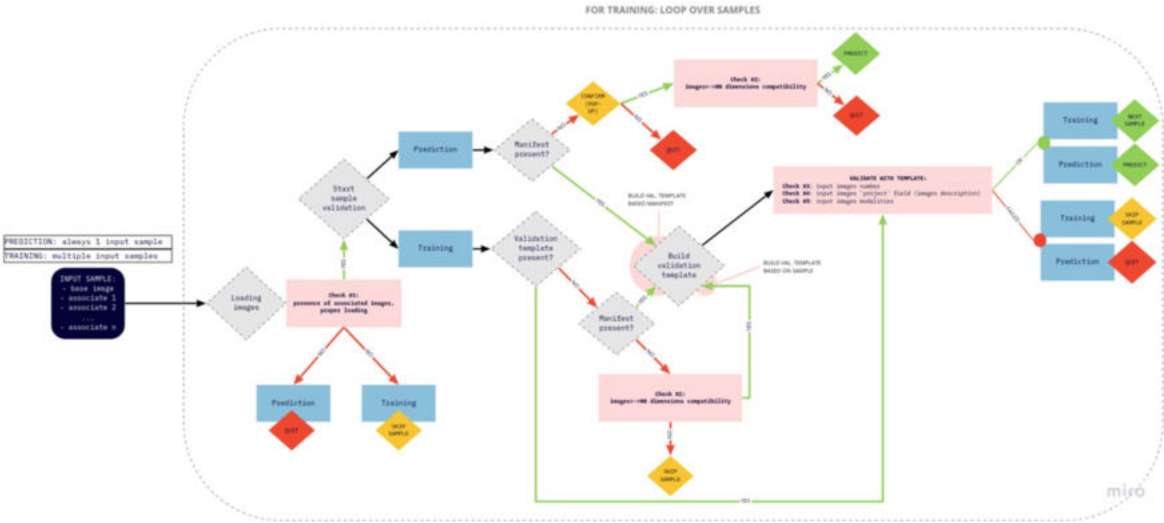
- Prediction: Cancel processing and inform the user
- Re-training: Skip the sample and print the information in the log

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 using training parameters different from previous training sessions.

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
YES
NO
NO
Prediction
QUIT
Training
SKIP SAMPLE
Start sample validation
Prediction
Training
Manifest present?
NO
CONFIRM (POP-UP)
YES
NO
QUIT
Check #2: images<->NN dimensions compatibility
YES
PREDICT
NO
QUIT
YES
BUILD VAL. TEMPLATE BASED MANIFEST
Validation template present?
NO
Manifest present?
YES
NO
Build validation template
BUILD VAL. TEMPLATE BASED ON SAMPLE
Check #2: images<->NN dimensions compatibility
YES
NO
SKIP SAMPLE
YES
VALIDATE WITH TEMPLATE:
Check #3: input images number
Check #4: input images 'project' field (images description)
Check #5: input images modalities
OK
FAILED
Training
NEXT SAMPLE
Prediction
PREDICT
Training
SKIP SAMPLE
Prediction
QUIT
miro

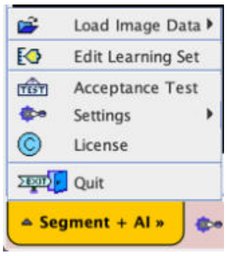
3.3 Learning Sets

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.

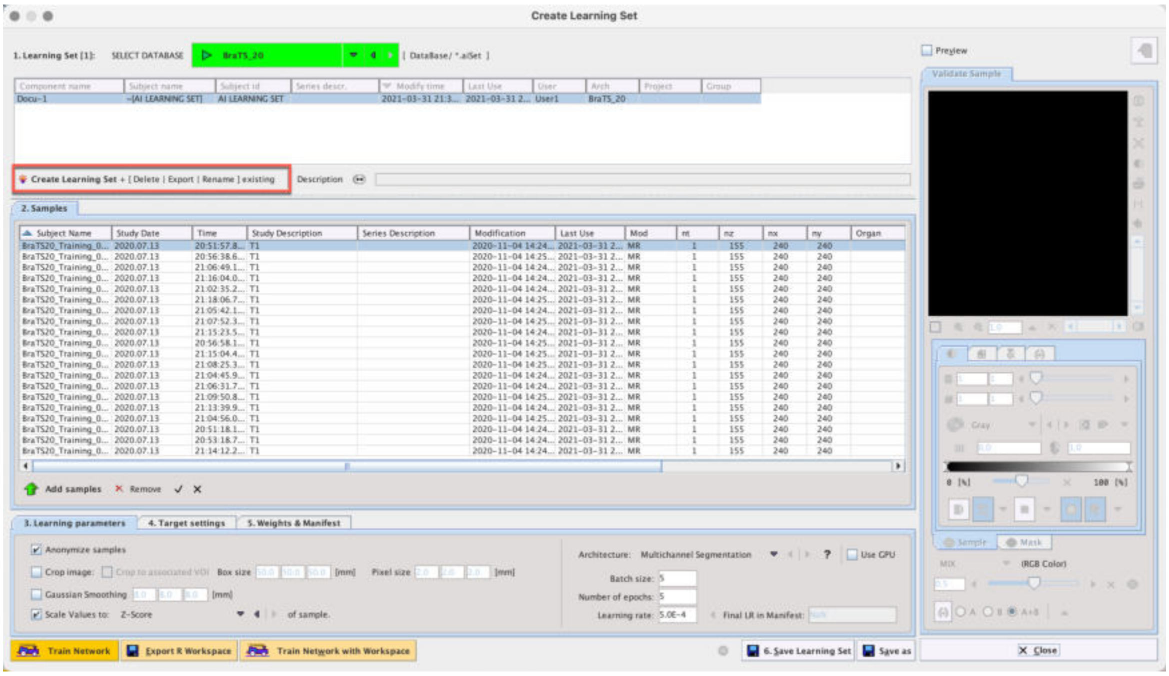
> Note that any necessary data preparations 34 should been done before creating or extending a **Leaning Set**.

Learning Set Dialog Window

To create a **Learning Set** or start a training run, select **Edit Learning Set** from the **Segment + AI** menu:



A dialog window appears listing 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.



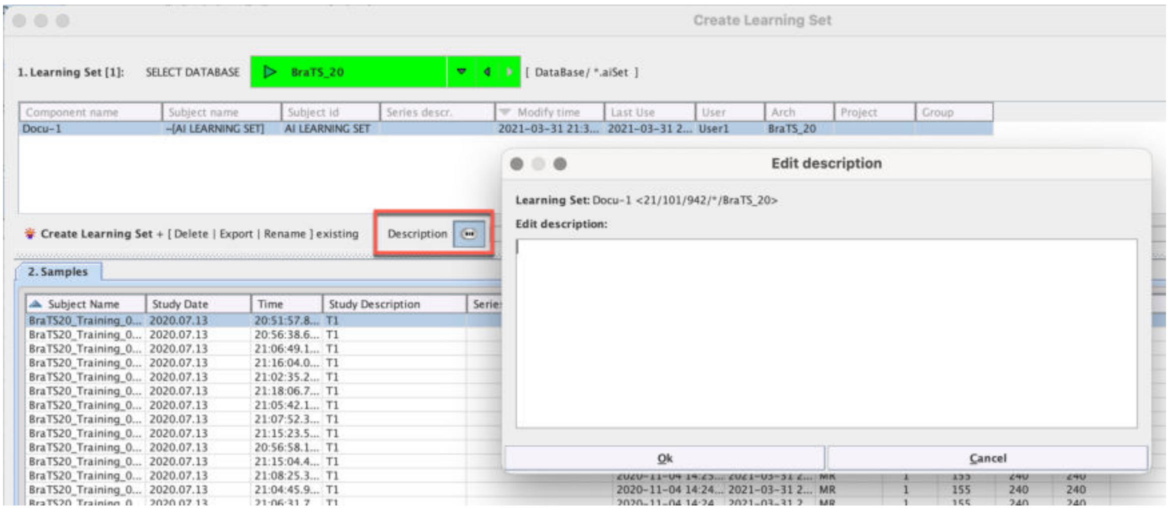
Component name	Subject name	Subject id	Series descr.	Modify time	Last Use	User	Arch	Project	Group
Docu-1	-[AI LEARNING SET]	AI LEARNING SET		2021-03-31 21:3...	2021-03-31 2...	User1	BraTS_20		

Subject Name	Study Date	Time	Study Description	Series Description	Modification	Last Use	Mod	nt	nz	nx	ny	Organ
BraTS20_Training_0...	2020.07.13	20:51:57.8...	T1		2020-11-04 14:24...	2021-03-31 2...	MR	1	155	240	240	
BraTS20_Training_0...	2020.07.13	20:56:38.6...	T1		2020-11-04 14:25...	2021-03-31 2...	MR	1	155	240	240	
BraTS20_Training_0...	2020.07.13	21:06:49.1...	T1		2020-11-04 14:24...	2021-03-31 2...	MR	1	155	240	240	
BraTS20_Training_0...	2020.07.13	21:16:04.0...	T1		2020-11-04 14:24...	2021-03-31 2...	MR	1	155	240	240	
BraTS20_Training_0...	2020.07.13	21:02:35.2...	T1		2020-11-04 14:24...	2021-03-31 2...	MR	1	155	240	240	
BraTS20_Training_0...	2020.07.13	21:18:06.7...	T1		2020-11-04 14:25...	2021-03-31 2...	MR	1	155	240	240	
BraTS20_Training_0...	2020.07.13	21:05:42.1...	T1		2020-11-04 14:24...	2021-03-31 2...	MR	1	155	240	240	
BraTS20_Training_0...	2020.07.13	21:07:52.3...	T1		2020-11-04 14:25...	2021-03-31 2...	MR	1	155	240	240	
BraTS20_Training_0...	2020.07.13	21:15:23.5...	T1		2020-11-04 14:24...	2021-03-31 2...	MR	1	155	240	240	
BraTS20_Training_0...	2020.07.13	20:56:58.1...	T1		2020-11-04 14:25...	2021-03-31 2...	MR	1	155	240	240	
BraTS20_Training_0...	2020.07.13	21:15:04.4...	T1		2020-11-04 14:25...	2021-03-31 2...	MR	1	155	240	240	
BraTS20_Training_0...	2020.07.13	21:08:25.3...	T1		2020-11-04 14:25...	2021-03-31 2...	MR	1	155	240	240	
BraTS20_Training_0...	2020.07.13	21:04:45.9...	T1		2020-11-04 14:24...	2021-03-31 2...	MR	1	155	240	240	
BraTS20_Training_0...	2020.07.13	21:06:31.7...	T1		2020-11-04 14:24...	2021-03-31 2...	MR	1	155	240	240	
BraTS20_Training_0...	2020.07.13	21:09:50.8...	T1		2020-11-04 14:25...	2021-03-31 2...	MR	1	155	240	240	
BraTS20_Training_0...	2020.07.13	21:13:39.9...	T1		2020-11-04 14:24...	2021-03-31 2...	MR	1	155	240	240	
BraTS20_Training_0...	2020.07.13	21:04:56.0...	T1		2020-11-04 14:25...	2021-03-31 2...	MR	1	155	240	240	
BraTS20_Training_0...	2020.07.13	20:51:18.1...	T1		2020-11-04 14:24...	2021-03-31 2...	MR	1	155	240	240	
BraTS20_Training_0...	2020.07.13	20:53:18.7...	T1		2020-11-04 14:24...	2021-03-31 2...	MR	1	155	240	240	
BraTS20_Training_0...	2020.07.13	21:14:12.2...	T1		2020-11-04 14:24...	2021-03-31 2...	MR	1	155	240	240	

☑ Anonymize samples
□ Crop image: □ Crop to associated VOI Box size 50.0 50.0 50.0 [mm] Pixel size 2.0 2.0 2.0 [mm]
□ Gaussian Smoothing 1.0 6.0 6.0 [mm]
☑ Scale Values to: Z-Score of sample.

Architecture: Multichannel Segmentation □ Use GPU
Batch size: 5
Number of epochs: 5
Learning rate: 5.0E-4
Final LR in Manifest:

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



Learning Set: Docu-1 <21/101/942/*/BraTS_20>

Edit description:

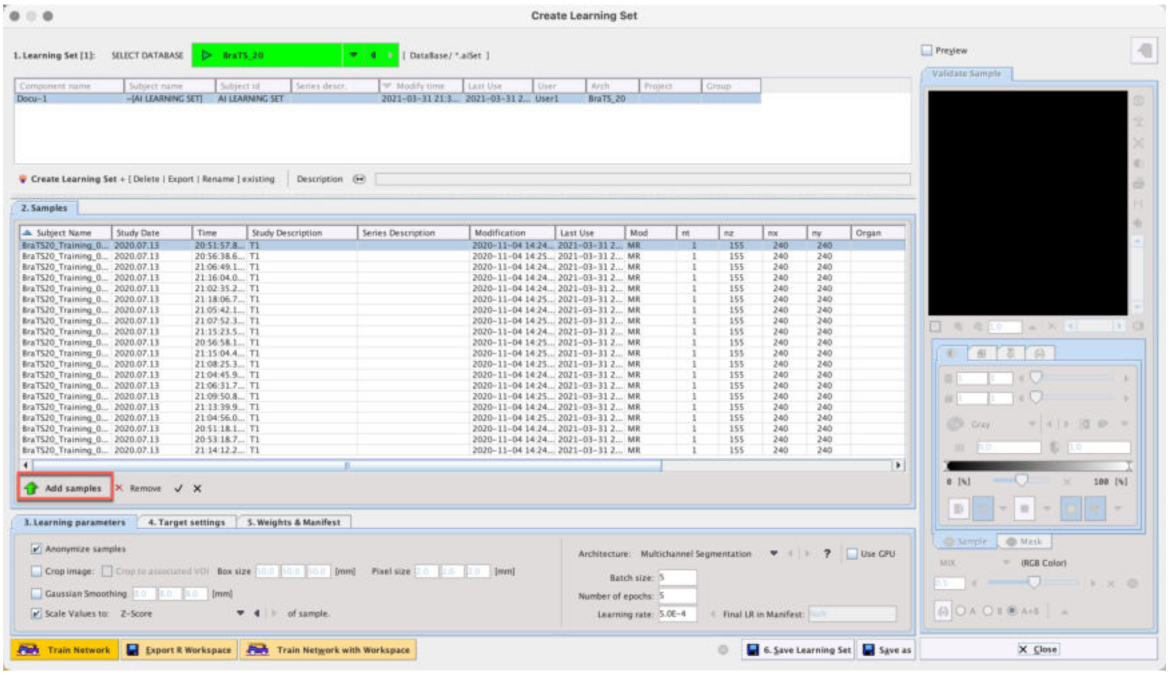
Component name	Subject name	Subject id	Series descr.	Modify time	Last Use	User	Arch	Project	Group
Docu-1	-[AI LEARNING SET]	AI LEARNING SET		2021-03-31 21:3...	2021-03-31 2...	User1	BraTS_20		

Subject Name	Study Date	Time	Study Description
BraTS20_Training_0...	2020.07.13	20:51:57.8...	T1
BraTS20_Training_0...	2020.07.13	20:56:38.6...	T1
BraTS20_Training_0...	2020.07.13	21:06:49.1...	T1
BraTS20_Training_0...	2020.07.13	21:16:04.0...	T1
BraTS20_Training_0...	2020.07.13	21:02:35.2...	T1
BraTS20_Training_0...	2020.07.13	21:18:06.7...	T1
BraTS20_Training_0...	2020.07.13	21:05:42.1...	T1
BraTS20_Training_0...	2020.07.13	21:07:52.3...	T1
BraTS20_Training_0...	2020.07.13	21:15:23.5...	T1
BraTS20_Training_0...	2020.07.13	20:56:58.1...	T1
BraTS20_Training_0...	2020.07.13	21:15:04.4...	T1
BraTS20_Training_0...	2020.07.13	21:08:25.3...	T1
BraTS20_Training_0...	2020.07.13	21:04:45.9...	T1

Leaning Set Creation

Create Learning Set opens a dialog window which allows an empty learning set to be created and saved to the database (as component with extension **.aiSet**).

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



Create Learning Set

1. Learning Set [1]: SELECT DATABASE BraTS_20 [DataBase/ *.aiSet]

Component name	Subject name	Subject id	Series descr.	Modify time	Last Use	User	Arch	Project	Group
Docu-1	-[AI LEARNING SET]	AI LEARNING SET		2021-03-31 21:3...	2021-03-31 2...	User1	BraTS_20		

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

2. Samples

Subject Name	Study Date	Time	Study Description	Series Description	Modification	Last Use	Mod	nt	nz	nx	ny	Organ
BraTS20_Training_0...	2020.07.13	20:51:57.8...	T1		2020-11-04 14:24...	2021-03-31 2...	MR	1	155	240	240	
BraTS20_Training_0...	2020.07.13	20:56:38.6...	T1		2020-11-04 14:25...	2021-03-31 2...	MR	1	155	240	240	
BraTS20_Training_0...	2020.07.13	21:06:49.1...	T1		2020-11-04 14:24...	2021-03-31 2...	MR	1	155	240	240	
BraTS20_Training_0...	2020.07.13	21:16:04.0...	T1		2020-11-04 14:24...	2021-03-31 2...	MR	1	155	240	240	
BraTS20_Training_0...	2020.07.13	21:02:35.2...	T1		2020-11-04 14:24...	2021-03-31 2...	MR	1	155	240	240	
BraTS20_Training_0...	2020.07.13	21:18:06.7...	T1		2020-11-04 14:25...	2021-03-31 2...	MR	1	155	240	240	
BraTS20_Training_0...	2020.07.13	21:05:42.1...	T1		2020-11-04 14:24...	2021-03-31 2...	MR	1	155	240	240	
BraTS20_Training_0...	2020.07.13	21:07:52.3...	T1		2020-11-04 14:25...	2021-03-31 2...	MR	1	155	240	240	
BraTS20_Training_0...	2020.07.13	21:15:23.5...	T1		2020-11-04 14:24...	2021-03-31 2...	MR	1	155	240	240	
BraTS20_Training_0...	2020.07.13	20:56:58.1...	T1		2020-11-04 14:25...	2021-03-31 2...	MR	1	155	240	240	
BraTS20_Training_0...	2020.07.13	21:15:04.4...	T1		2020-11-04 14:25...	2021-03-31 2...	MR	1	155	240	240	
BraTS20_Training_0...	2020.07.13	21:08:25.3...	T1		2020-11-04 14:25...	2021-03-31 2...	MR	1	155	240	240	
BraTS20_Training_0...	2020.07.13	21:04:45.9...	T1		2020-11-04 14:24...	2021-03-31 2...	MR	1	155	240	240	
BraTS20_Training_0...	2020.07.13	21:06:31.7...	T1		2020-11-04 14:24...	2021-03-31 2...	MR	1	155	240	240	
BraTS20_Training_0...	2020.07.13	21:09:50.8...	T1		2020-11-04 14:25...	2021-03-31 2...	MR	1	155	240	240	
BraTS20_Training_0...	2020.07.13	21:13:39.9...	T1		2020-11-04 14:24...	2021-03-31 2...	MR	1	155	240	240	
BraTS20_Training_0...	2020.07.13	21:04:56.0...	T1		2020-11-04 14:25...	2021-03-31 2...	MR	1	155	240	240	
BraTS20_Training_0...	2020.07.13	20:51:18.1...	T1		2020-11-04 14:24...	2021-03-31 2...	MR	1	155	240	240	
BraTS20_Training_0...	2020.07.13	20:53:18.7...	T1		2020-11-04 14:24...	2021-03-31 2...	MR	1	155	240	240	
BraTS20_Training_0...	2020.07.13	21:14:12.2...	T1		2020-11-04 14:24...	2021-03-31 2...	MR	1	155	240	240	

Add samples ✕ Remove ✓ ✕

3. Learning parameters | 4. Target settings | 5. Weights & Manifest

☑ Anonymize samples

□ Crop image: □ Crop to associated VOI Box size 50.0 50.0 50.0 [mm] Pixel size 2.0 2.0 2.0 [mm]

□ Gaussian Smoothing 1.0 6.0 6.0 [mm]

☑ Scale Values to: Z-Score of sample.

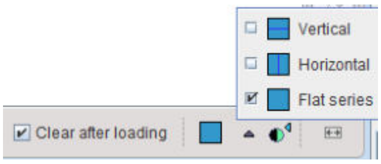
Architecture: Multichannel Segmentation ? □ Use GPU

Batch size: 5

Number of epochs: 5

Learning rate: 5.0E-4 Final LR in Manifest:

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



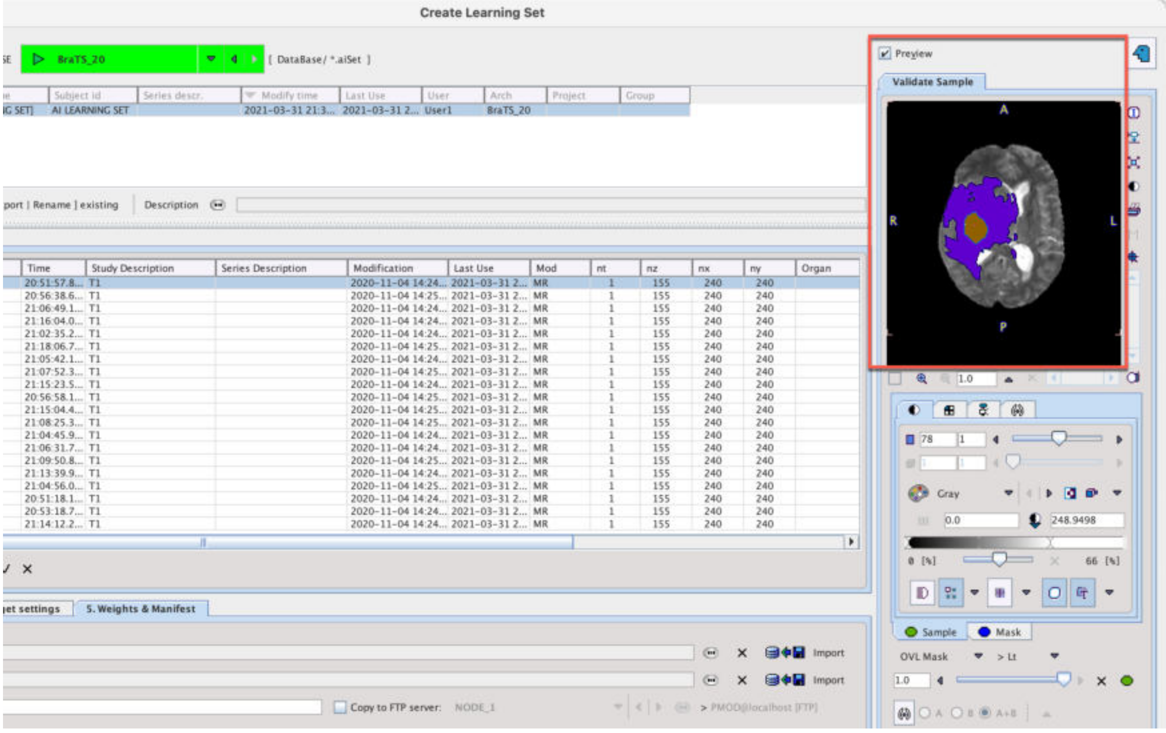
□ Vertical

□ Horizontal

☑ Flat series

☑ Clear after loading

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:



Create Learning Set

▷ BraTS_20 [DataBase/ *.aiSet]

Subject id	Series descr.	Modify time	Last Use	User	Arch	Project	Group
AI LEARNING SET		2021-03-31 21:3...	2021-03-31 2...	User1	BraTS_20		

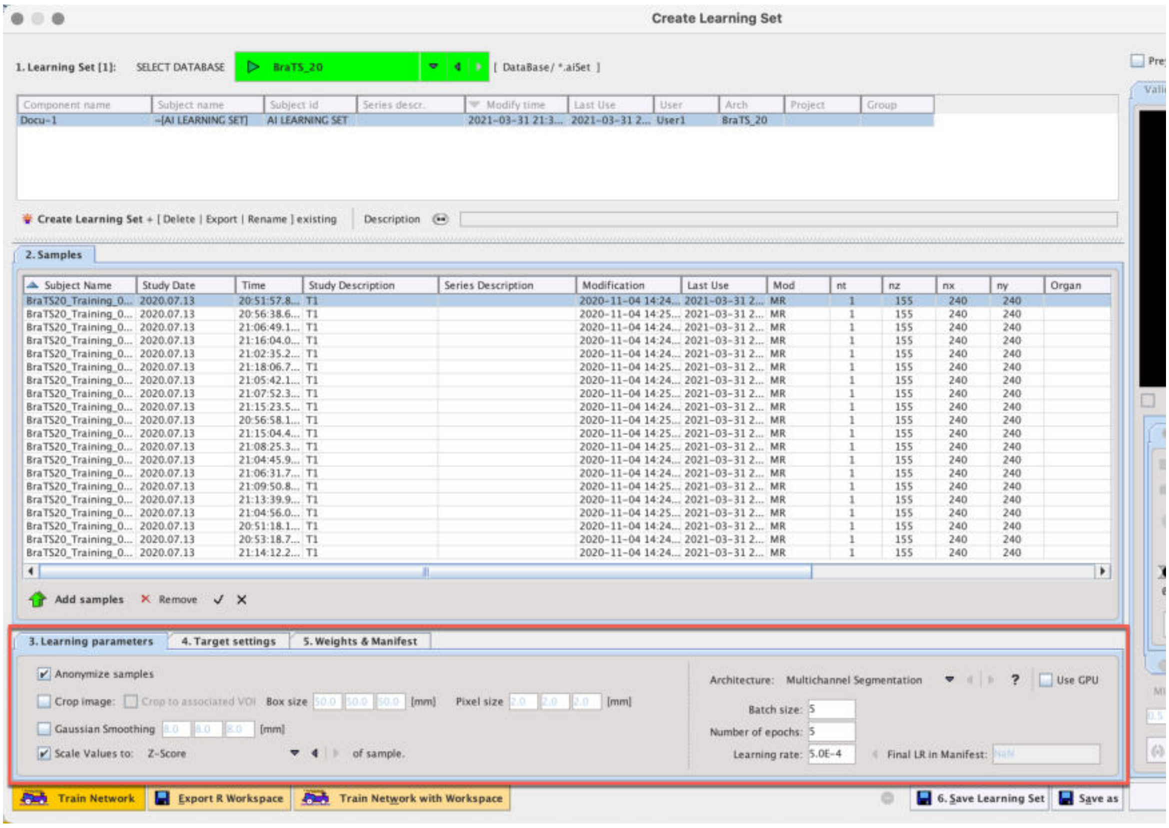
port | Rename] existing Description

Time	Study Description	Series Description	Modification	Last Use	Mod	nt	nz	nx	ny	Organ
20:51:57.8...	T1		2020-11-04 14:24...	2021-03-31 2...	MR	1	155	240	240	
20:56:38.6...	T1		2020-11-04 14:25...	2021-03-31 2...	MR	1	155	240	240	
21:06:49.1...	T1		2020-11-04 14:24...	2021-03-31 2...	MR	1	155	240	240	
21:16:04.0...	T1		2020-11-04 14:24...	2021-03-31 2...	MR	1	155	240	240	
21:02:35.2...	T1		2020-11-04 14:24...	2021-03-31 2...	MR	1	155	240	240	
21:18:06.7...	T1		2020-11-04 14:25...	2021-03-31 2...	MR	1	155	240	240	
21:05:42.1...	T1		2020-11-04 14:24...	2021-03-31 2...	MR	1	155	240	240	
21:07:52.3...	T1		2020-11-04 14:25...	2021-03-31 2...	MR	1	155	240	240	
21:15:23.5...	T1		2020-11-04 14:24...	2021-03-31 2...	MR	1	155	240	240	
20:56:58.1...	T1		2020-11-04 14:25...	2021-03-31 2...	MR	1	155	240	240	
21:15:04.4...	T1		2020-11-04 14:25...	2021-03-31 2...	MR	1	155	240	240	
21:08:25.3...	T1		2020-11-04 14:25...	2021-03-31 2...	MR	1	155	240	240	
21:04:45.9...	T1		2020-11-04 14:24...	2021-03-31 2...	MR	1	155	240	240	
21:06:31.7...	T1		2020-11-04 14:24...	2021-03-31 2...	MR	1	155	240	240	
21:09:50.8...	T1		2020-11-04 14:25...	2021-03-31 2...	MR	1	155	240	240	
21:13:39.9...	T1		2020-11-04 14:24...	2021-03-31 2...	MR	1	155	240	240	
21:04:56.0...	T1		2020-11-04 14:25...	2021-03-31 2...	MR	1	155	240	240	
20:51:18.1...	T1		2020-11-04 14:24...	2021-03-31 2...	MR	1	155	240	240	
20:53:18.7...	T1		2020-11-04 14:24...	2021-03-31 2...	MR	1	155	240	240	
21:14:12.2...	T1		2020-11-04 14:24...	2021-03-31 2...	MR	1	155	240	240	

☑ Preview

Data Preparation

In **3. Learning parameters**, the data preparation steps, architecture to be used and the training settings are configured.



Create Learning Set

1. Learning Set [1]: SELECT DATABASE ▷ BraTS_20 [DataBase/ *.aiSet]

Component name	Subject name	Subject id	Series descr.	Modify time	Last Use	User	Arch	Project	Group
Docu-1	-[AI LEARNING SET]	AI LEARNING SET		2021-03-31 21:3...	2021-03-31 2...	User1	BraTS_20		

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

2. Samples

Subject Name	Study Date	Time	Study Description	Series Description	Modification	Last Use	Mod	nt	nz	nx	ny	Organ
BraTS20_Training_0...	2020.07.13	20:51:57.8...	T1		2020-11-04 14:24...	2021-03-31 2...	MR	1	155	240	240	
BraTS20_Training_0...	2020.07.13	20:56:38.6...	T1		2020-11-04 14:25...	2021-03-31 2...	MR	1	155	240	240	
BraTS20_Training_0...	2020.07.13	21:06:49.1...	T1		2020-11-04 14:24...	2021-03-31 2...	MR	1	155	240	240	
BraTS20_Training_0...	2020.07.13	21:16:04.0...	T1		2020-11-04 14:24...	2021-03-31 2...	MR	1	155	240	240	
BraTS20_Training_0...	2020.07.13	21:02:35.2...	T1		2020-11-04 14:24...	2021-03-31 2...	MR	1	155	240	240	
BraTS20_Training_0...	2020.07.13	21:18:06.7...	T1		2020-11-04 14:25...	2021-03-31 2...	MR	1	155	240	240	
BraTS20_Training_0...	2020.07.13	21:05:42.1...	T1		2020-11-04 14:24...	2021-03-31 2...	MR	1	155	240	240	
BraTS20_Training_0...	2020.07.13	21:07:52.3...	T1		2020-11-04 14:25...	2021-03-31 2...	MR	1	155	240	240	
BraTS20_Training_0...	2020.07.13	21:15:23.5...	T1		2020-11-04 14:24...	2021-03-31 2...	MR	1	155	240	240	
BraTS20_Training_0...	2020.07.13	20:56:58.1...	T1		2020-11-04 14:25...	2021-03-31 2...	MR	1	155	240	240	
BraTS20_Training_0...	2020.07.13	21:15:04.4...	T1		2020-11-04 14:25...	2021-03-31 2...	MR	1	155	240	240	
BraTS20_Training_0...	2020.07.13	21:08:25.3...	T1		2020-11-04 14:25...	2021-03-31 2...	MR	1	155	240	240	
BraTS20_Training_0...	2020.07.13	21:04:45.9...	T1		2020-11-04 14:24...	2021-03-31 2...	MR	1	155	240	240	
BraTS20_Training_0...	2020.07.13	21:06:31.7...	T1		2020-11-04 14:24...	2021-03-31 2...	MR	1	155	240	240	
BraTS20_Training_0...	2020.07.13	21:09:50.8...	T1		2020-11-04 14:25...	2021-03-31 2...	MR	1	155	240	240	
BraTS20_Training_0...	2020.07.13	21:13:39.9...	T1		2020-11-04 14:24...	2021-03-31 2...	MR	1	155	240	240	
BraTS20_Training_0...	2020.07.13	21:04:56.0...	T1		2020-11-04 14:25...	2021-03-31 2...	MR	1	155	240	240	
BraTS20_Training_0...	2020.07.13	20:51:18.1...	T1		2020-11-04 14:24...	2021-03-31 2...	MR	1	155	240	240	
BraTS20_Training_0...	2020.07.13	20:53:18.7...	T1		2020-11-04 14:24...	2021-03-31 2...	MR	1	155	240	240	
BraTS20_Training_0...	2020.07.13	21:14:12.2...	T1		2020-11-04 14:24...	2021-03-31 2...	MR	1	155	240	240	

☑ Anonymize samples

□ Crop image: □ Crop to associated VOI Box size 50.0 50.0 50.0 [mm] Pixel size 2.0 2.0 2.0 [mm]

□ Gaussian Smoothing 8.0 8.0 8.0 [mm]

☑ Scale Values to: Z-Score of sample.

Architecture: Multichannel Segmentation ? □ Use GPU

Batch size: 5

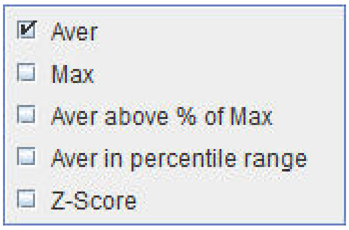
Number of epochs: 5

Learning rate: 5.0E-4 Final LR in Manifest: NaN

The available data preparation steps are:

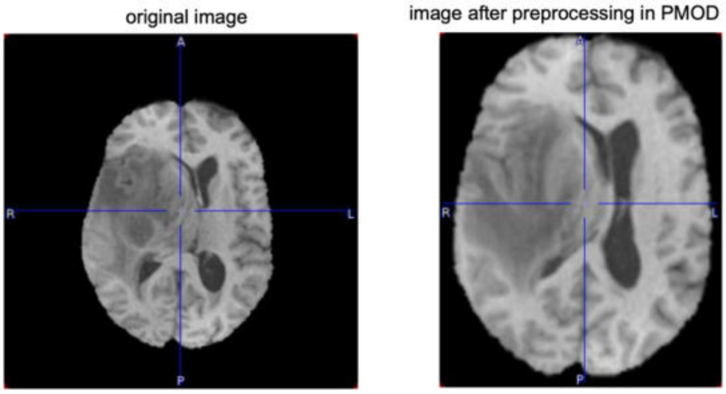
- **Anonymize Samples**: Anonymizes all images selected for training or workspace export (this is particularly relevant for training on cloud-computing infrastructure).

- **Crop Image**: Enables cropping to the associated VOI as described in Data Preparation 34 or to a fixed **Box size**.
- **Pixel Size**: Re-sampling of the images to a certain pixel size by either interpolation or down-sampling.
- **Gaussian Smoothing**: Input image smoothing to reduce noise.
- **Scale Values to**: Normalization of the pixel value range by scaling according to a method selected.


☑ Aver
□ Max
□ Aver above % of Max
□ Aver in percentile range
□ Z-Score

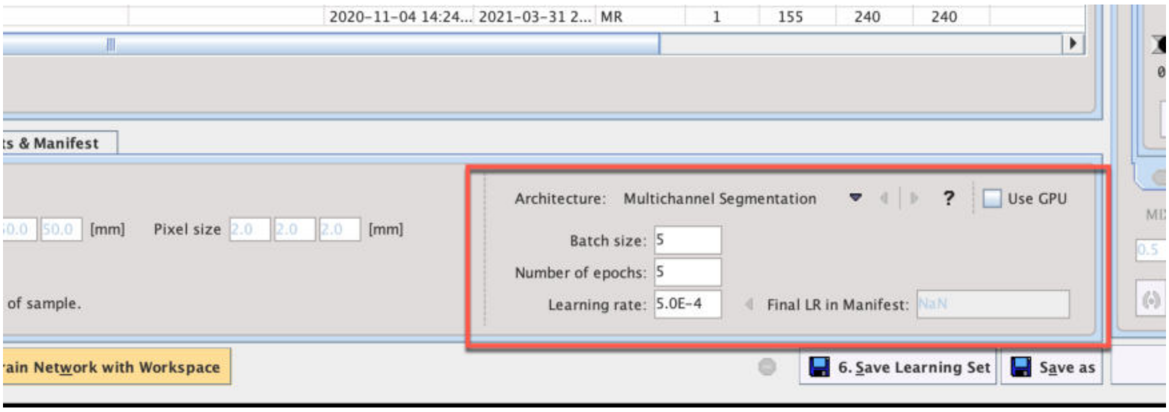
Note that as an alternative to such pre-processing steps, the input images could be (manually) pre-processed in other PMOD tools. In this case, however, the identical operations have to be applied to the input images before prediction.

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



Neural Network and Training Parameters

The neural network **Architecture** and the training parameters are selected in the area to the right.


2020-11-04 14:24... 2021-03-31 2... MR 1 155 240 240

ts & Manifest

50.0 50.0 [mm] Pixel size 2.0 2.0 2.0 [mm]

of sample.

Architecture: Multichannel Segmentation ? □ Use GPU

Batch size: 5

Number of epochs: 5

Learning rate: 5.0E-4

Final LR in Manifest: NaN

The neural network architecture is selected from the drop-down menu **Architecture**. The list corresponds to the the content of the Pmod4.2/resources/pai folder, where the neural network configurations are stored in sub-directories. Note that the **Multichannel Segmentation** architecture is designed to be retrainable. It 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.

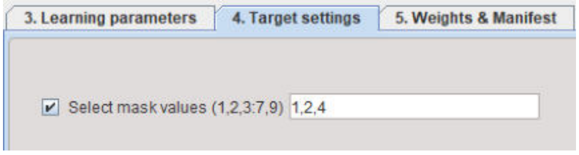
The checkbox **Use GPU** allows you to choose between training using the CPU or available/compatible GPU.

The training parameters 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.
- **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)

Definition of Target Segments

The reference segment image may contain more segments than actually required. The option **Select mask values** on the **4. Target Settings** panel allows the integer value of the required segments to be specified by entering the label values of the target segments, separated by a "comma".

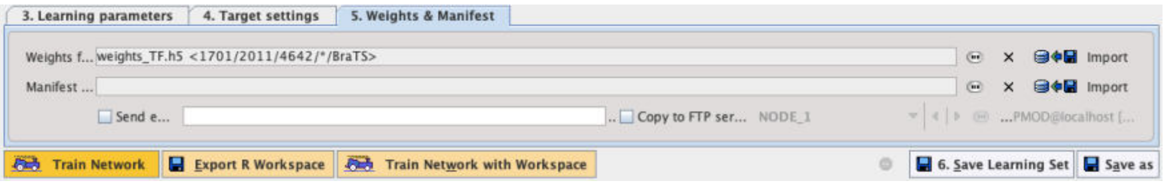


☑ Select mask values (1,2,3:7,9) 1,2,4

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.

The file locations are defined on the **5. Weights & Manifest** panel, and the files will be logically attached to the current learning set.



Weights f... weights_TF.h5 <1701/2011/4642/*/BraTS>

Manifest ...

□ Send e... .. □ Copy to FTP ser... NODE_1 ...PMOD@localhost [...

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 selected in the **3. Samples** list will be used for the training. As a result, the weights and manifest files will be updated.
2. When activating **Export R workspace** 51**,** 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 for loading a previously exported R workspace and starts the training locally.

Deployment

After completion of the training, the resulting **Weights** and **Manifest** files can be transferred, along with the definition of the model if necessary, to other PMOD installations for prediction.

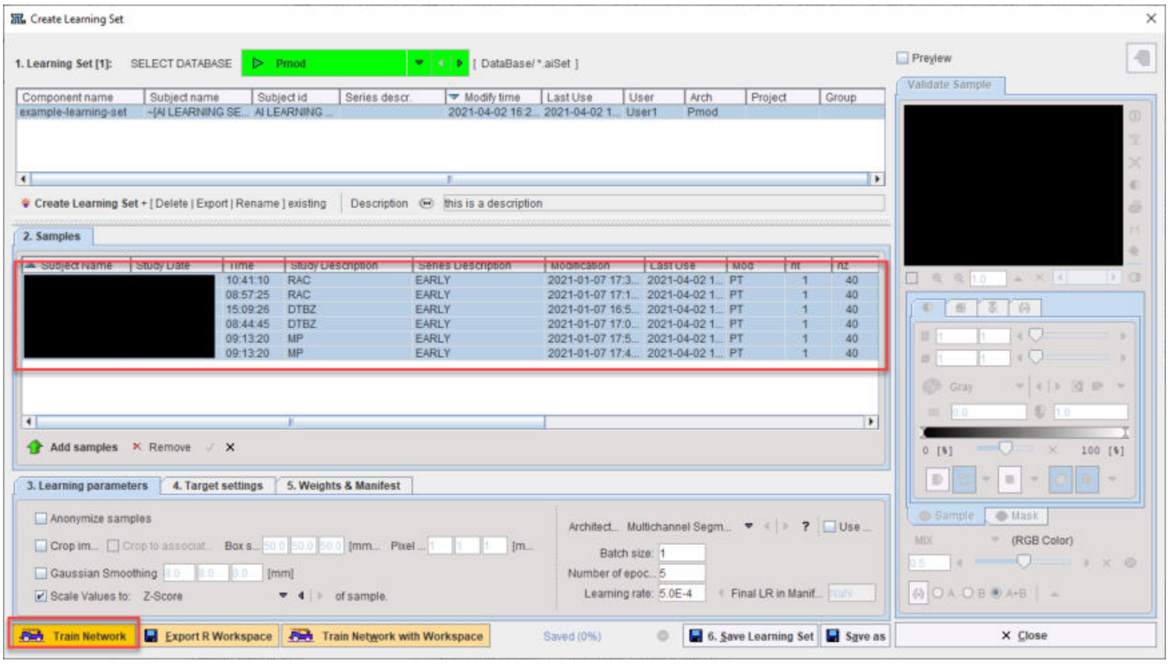
Recommendations

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. 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**:



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
example-learning-set	~[AI LEARNING SE...	AI LEARNING ...		2021-04-02 16:2...	2021-04-02 1...	User1	Pmod		

Create Learning Set + [Delete | Export | Rename] existing Description this is a description

2. Samples

Subject Name	Study Date	Time	Study Description	Series Description	Modification	Last Use	Mod	nt	nz
[redacted]	[redacted]	10:41:10	RAC	EARLY	2021-01-07 17:3...	2021-04-02 1...	PT	1	40
[redacted]	[redacted]	08:57:25	RAC	EARLY	2021-01-07 17:1...	2021-04-02 1...	PT	1	40
[redacted]	[redacted]	15:09:26	DTBZ	EARLY	2021-01-07 16:5...	2021-04-02 1...	PT	1	40
[redacted]	[redacted]	08:44:45	DTBZ	EARLY	2021-01-07 17:0...	2021-04-02 1...	PT	1	40
[redacted]	[redacted]	09:13:20	MP	EARLY	2021-01-07 17:5...	2021-04-02 1...	PT	1	40
[redacted]	[redacted]	09:13:20	MP	EARLY	2021-01-07 17:4...	2021-04-02 1...	PT	1	40

Add samples × Remove ✓ ×

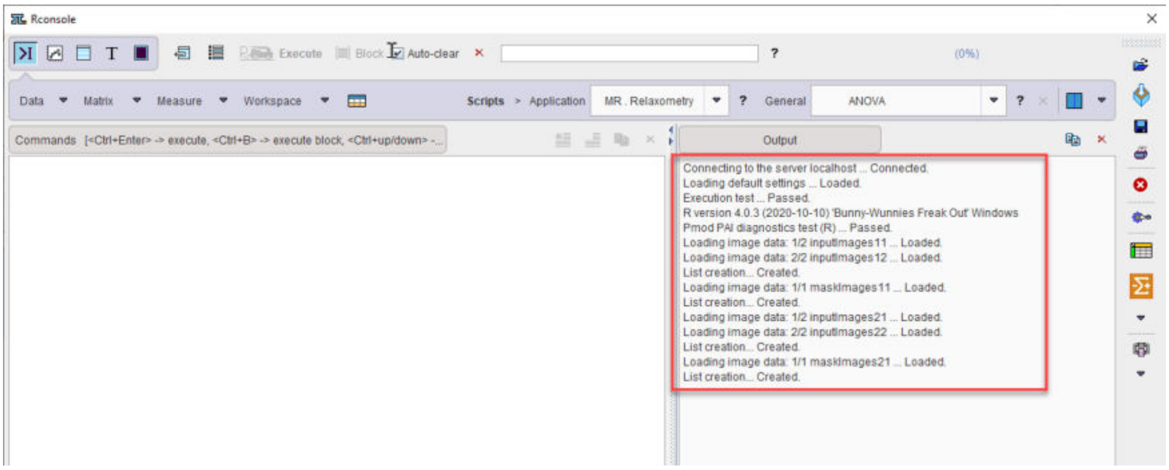
3. Learning parameters | 4. Target settings | 5. Weights & Manifest

□ Anonymize samples
□ Crop im... □ Crop to associat... Box s... 50.0 50.0 50.0 [mm... Pixel ... 1 1 1 [m...
□ Gaussian Smoothing 8.0 8.0 8.0 [mm]
☑ Scale Values to: Z-Score of sample.

Architect... Multichannel Segm... ? □ Use ...
Batch size: 1
Number of epoc... 5
Learning rate: 5.0E-4 Final LR in Manif... NaN

Train Network | Export R Workspace | Train Network with Workspace | Saved (0%) | 6. Save Learning Set | Save as | × Close

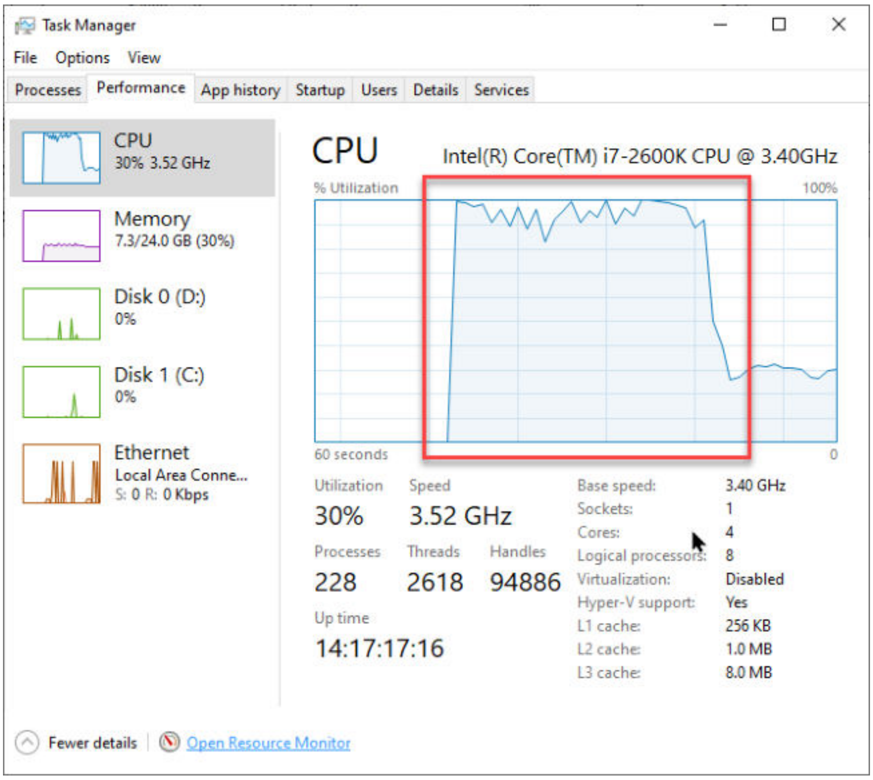
The **RConsole** opens and the Execution test and PAI diagnostics test are performed. If the tests are passed the selected samples (all input series and associated Segments) are loaded:



Rconsole

```
Connecting to the server localhost ... Connected.
Loading default settings ... Loaded.
Execution test ... Passed.
R version 4.0.3 (2020-10-10) 'Bunny-Wunnies Freak Out' Windows
Pmod PAI diagnostics test (R) ... Passed.
Loading image data: 1/2 inputImages11 ... Loaded.
Loading image data: 2/2 inputImages12 ... Loaded.
List creation... Created.
Loading image data: 1/1 maskImages11 ... Loaded.
List creation... Created.
Loading image data: 1/2 inputImages21 ... Loaded.
Loading image data: 2/2 inputImages22 ... Loaded.
List creation... Created.
Loading image data: 1/1 maskImages21 ... Loaded.
List creation... Created.
```

During training a high CPU/GPU load can be observed in the system monitor (Windows 10 Task Manager shown):



CPU 30% 3.52 GHz

Memory 7.3/24.0 GB (30%)

Disk 0 (D:) 0%

Disk 1 (C:) 0%

Ethernet Local Area Conne... S: 0 R: 0 Kbps

CPU Intel(R) Core(TM) i7-2600K CPU @ 3.40GHz

Utilization 30%

Speed 3.52 GHz

Processes 228

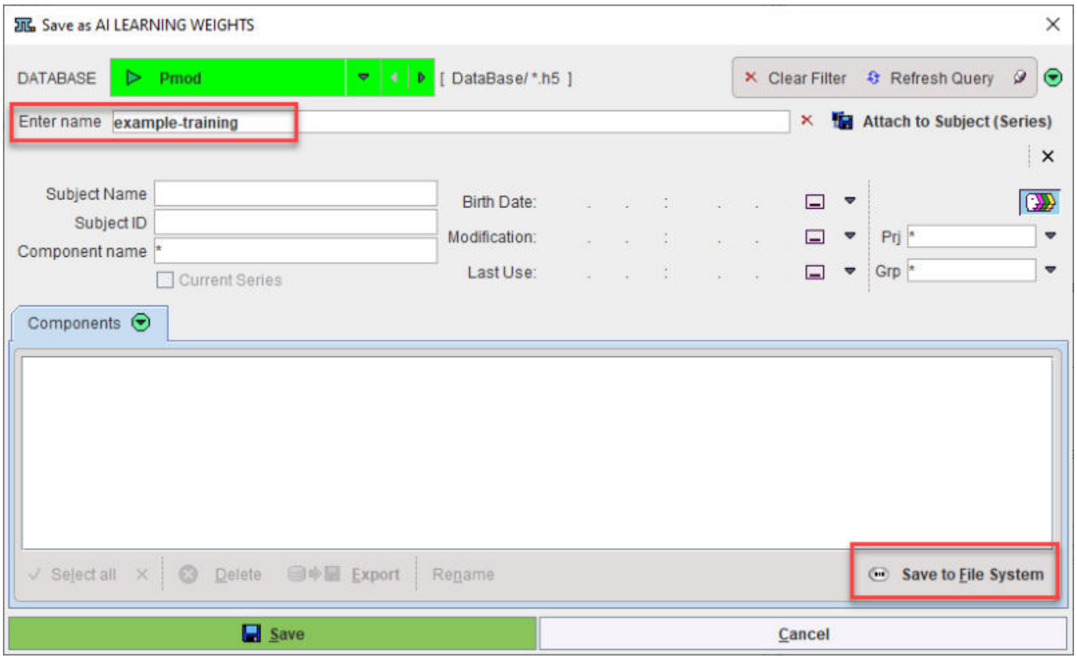
Threads 2618

Handles 94886

Up time 14:17:17:16

Base speed:	3.40 GHz
Sockets:	1
Cores:	4
Logical processors:	8
Virtualization:	Disabled
Hyper-V support:	Yes
L1 cache:	256 KB
L2 cache:	1.0 MB
L3 cache:	8.0 MB

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):



DATABASE Pmod [DataBase/ *.h5]

Enter name example-training

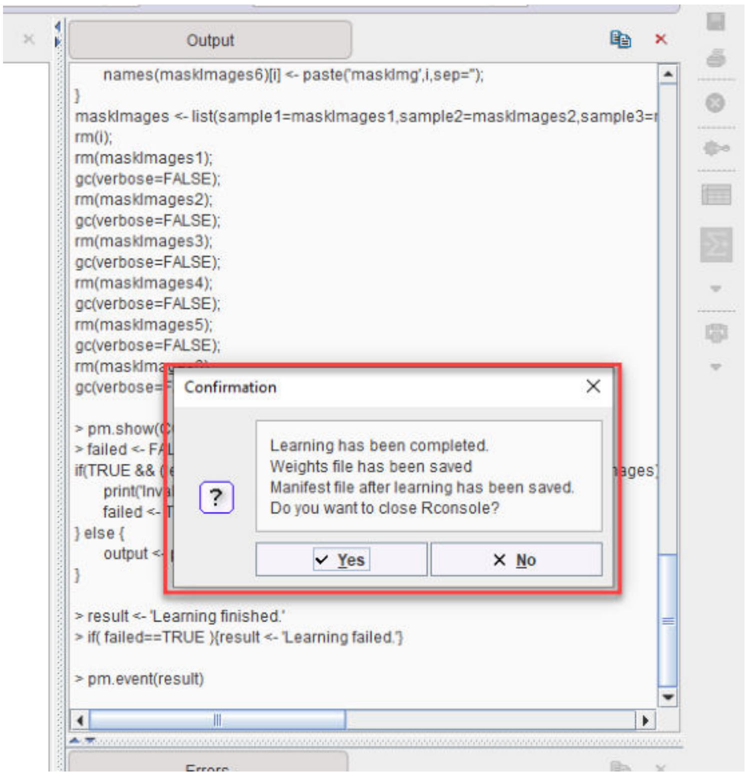
Component name *

☐ Current Series

Prj *

Grp *

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

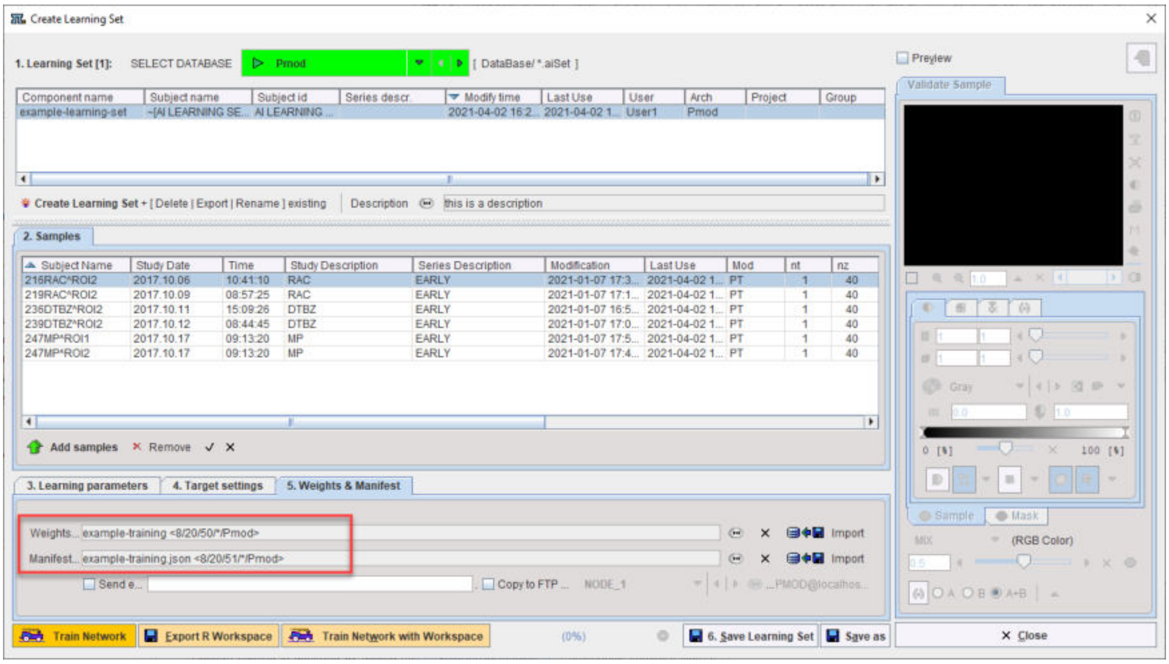


```
    names(maskImages6)[i] <- paste('maskImg',i,sep='');
}
maskImages <- list(sample1=maskImages1,sample2=maskImages2,sample3=
rm(i);
rm(maskImages1);
gc(verbose=FALSE);
rm(maskImages2);
gc(verbose=FALSE);
rm(maskImages3);
gc(verbose=FALSE);
rm(maskImages4);
gc(verbose=FALSE);
rm(maskImages5);
gc(verbose=FALSE);
rm(maskIma
gc(verbose=F

> pm.show(C
> failed <- FAL
if(TRUE && (                                                   ages)
    print('Inv
    failed <- T
} else {
    output <- (
}

> result <- 'Learning finished.'
> if( failed==TRUE ){result <- 'Learning failed.'}

> pm.event(result)
```

Confirmation

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

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



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
example-learning-set	~[AI LEARNING SE...	AI LEARNING ...		2021-04-02 16:2...	2021-04-02 1...	User1	Pmod		

Create Learning Set + [Delete | Export | Rename] existing   Description   this is a description

2. Samples

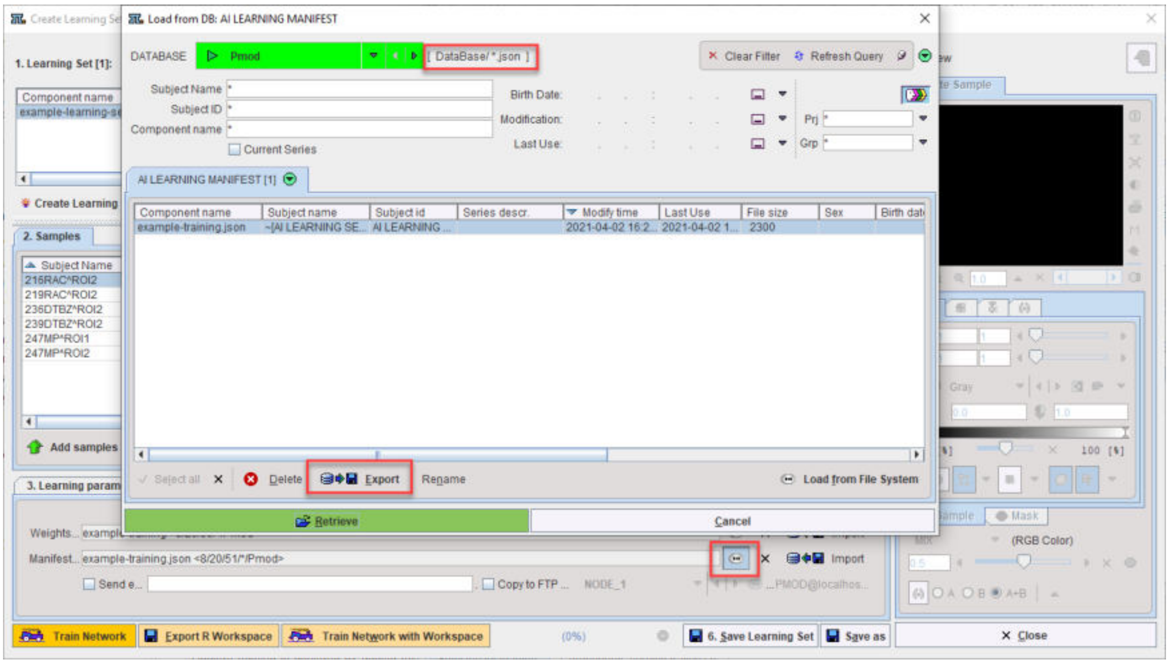
Subject Name	Study Date	Time	Study Description	Series Description	Modification	Last Use	Mod	nt	nz
216RAC^ROI2	2017.10.06	10:41:10	RAC	EARLY	2021-01-07 17:3...	2021-04-02 1...	PT	1	40
219RAC^ROI2	2017.10.09	08:57:25	RAC	EARLY	2021-01-07 17:1...	2021-04-02 1...	PT	1	40
236DTBZ^ROI2	2017.10.11	15:09:26	DTBZ	EARLY	2021-01-07 16:5...	2021-04-02 1...	PT	1	40
239DTBZ^ROI2	2017.10.12	08:44:45	DTBZ	EARLY	2021-01-07 17:0...	2021-04-02 1...	PT	1	40
247MP^ROI1	2017.10.17	09:13:20	MP	EARLY	2021-01-07 17:5...	2021-04-02 1...	PT	1	40
247MP^ROI2	2017.10.17	09:13:20	MP	EARLY	2021-01-07 17:4...	2021-04-02 1...	PT	1	40

Weights... example-training <8/20/50/*/Pmod>

Manifest... example-training.json <8/20/51/*/Pmod>

□ Send e...   □ Copy to FTP ...   NODE_1   ...PMOD@localhos...

Details of the training are recorded in the Manifest. It can be exported and read using a text editor:



Subject Name *

Subject ID *

Component name *

Prj *

Grp *

Component name	Subject name	Subject id	Series descr.	Modify time	Last Use	File size	Sex	Birth dat
example-training.json	~[AI LEARNING SE...	AI LEARNING ...		2021-04-02 16:2...	2021-04-02 1...	2300		

Component name
example-learning-se

Subject Name
216RAC^ROI2
219RAC^ROI2
236DTBZ^ROI2
239DTBZ^ROI2
247MP^ROI1
247MP^ROI2

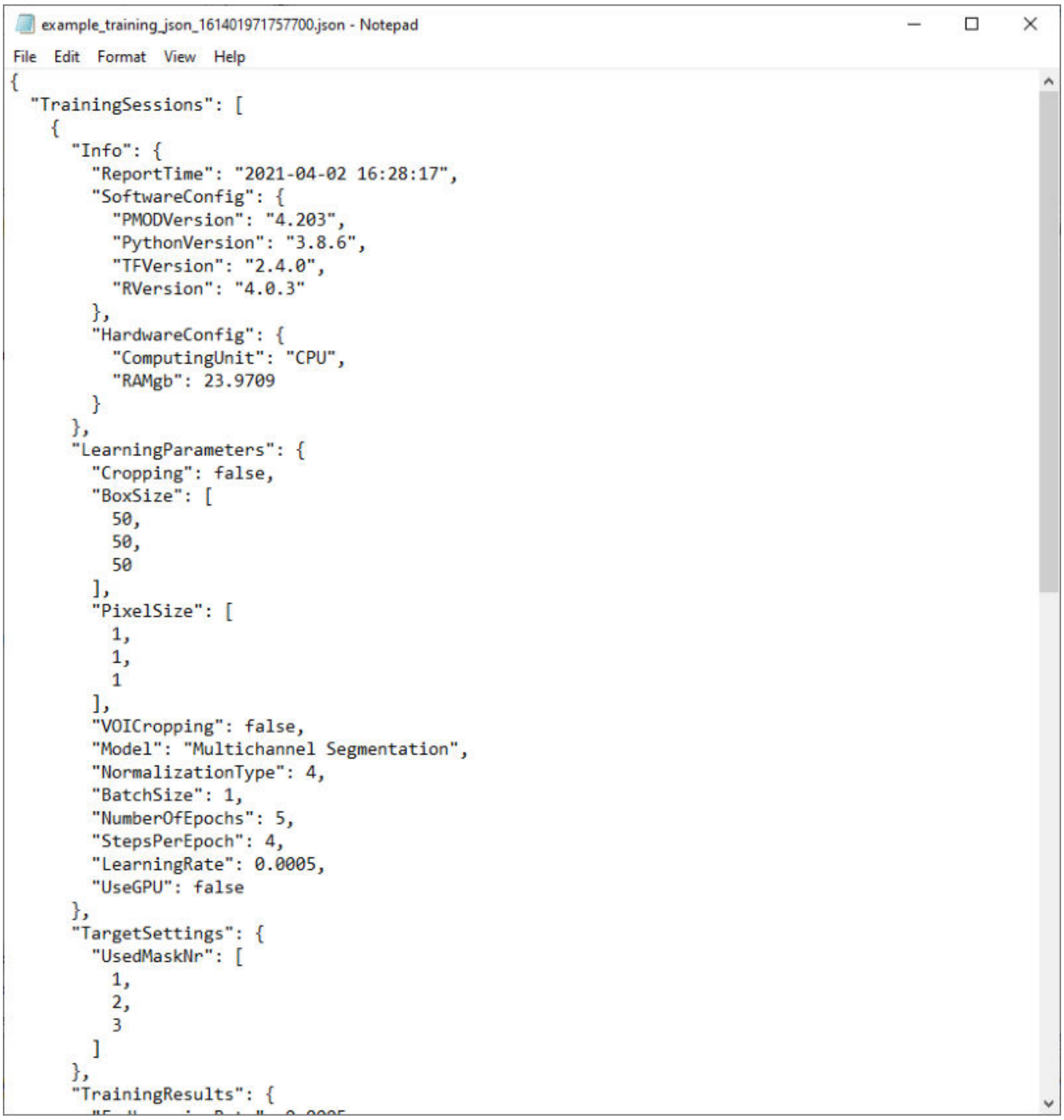
Weights... example

Manifest... example-training.json <8/20/51/*/Pmod>

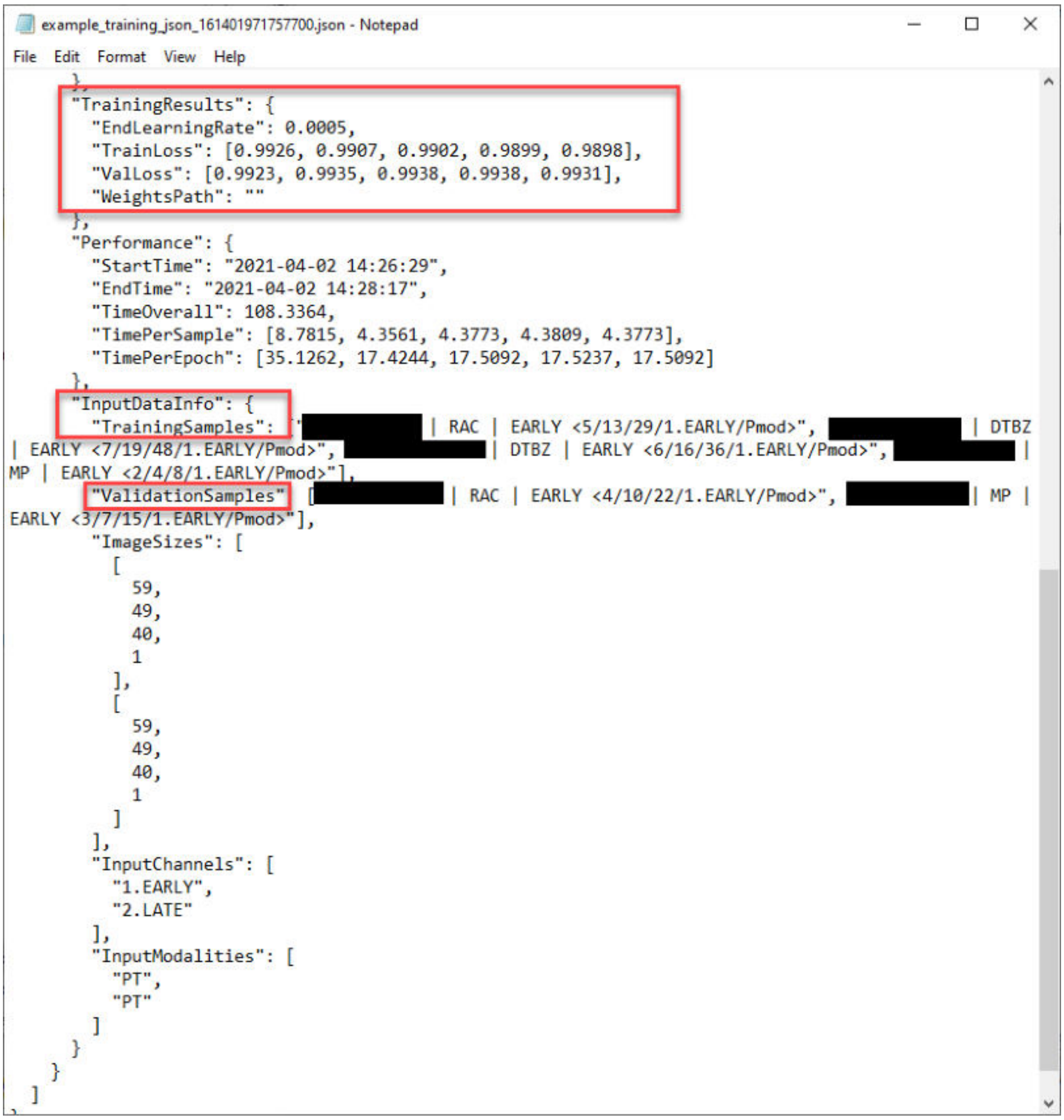
□ Send e...

□ Copy to FTP ... NODE_1

The content of the Manifest file is as follows. The evolution of the training loss value with each epoch can be observed, and those samples used for training and for validation can be identified:



```
{
  "TrainingSessions": [
    {
      "Info": {
        "ReportTime": "2021-04-02 16:28:17",
        "SoftwareConfig": {
          "PMODVersion": "4.203",
          "PythonVersion": "3.8.6",
          "TFVersion": "2.4.0",
          "RVersion": "4.0.3"
        },
        "HardwareConfig": {
          "ComputingUnit": "CPU",
          "RAMgb": 23.9709
        }
      },
      "LearningParameters": {
        "Cropping": false,
        "BoxSize": [
          50,
          50,
          50
        ],
        "PixelSize": [
          1,
          1,
          1
        ],
        "VOICropping": false,
        "Model": "Multichannel Segmentation",
        "NormalizationType": 4,
        "BatchSize": 1,
        "NumberOfEpochs": 5,
        "StepsPerEpoch": 4,
        "LearningRate": 0.0005,
        "UseGPU": false
      },
      "TargetSettings": {
        "UsedMaskNr": [
          1,
          2,
          3
        ]
      },
      "TrainingResults": {
```

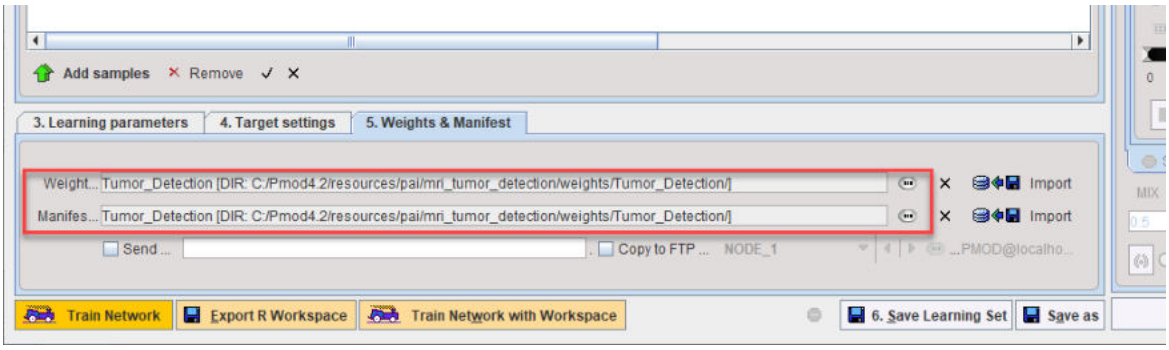


```
      },
      "TrainingResults": {
        "EndLearningRate": 0.0005,
        "TrainLoss": [0.9926, 0.9907, 0.9902, 0.9899, 0.9898],
        "ValLoss": [0.9923, 0.9935, 0.9938, 0.9938, 0.9931],
        "WeightsPath": ""
      },
      "Performance": {
        "StartTime": "2021-04-02 14:26:29",
        "EndTime": "2021-04-02 14:28:17",
        "TimeOverall": 108.3364,
        "TimePerSample": [8.7815, 4.3561, 4.3773, 4.3809, 4.3773],
        "TimePerEpoch": [35.1262, 17.4244, 17.5092, 17.5237, 17.5092]
      },
      "InputDataInfo": {
        "TrainingSamples": "[redacted] | RAC | EARLY <5/13/29/1.EARLY/Pmod>", [redacted] | DTBZ
| EARLY <7/19/48/1.EARLY/Pmod>", [redacted] | DTBZ | EARLY <6/16/36/1.EARLY/Pmod>", [redacted] |
MP | EARLY <2/4/8/1.EARLY/Pmod>"],
        "ValidationSamples" [[redacted] | RAC | EARLY <4/10/22/1.EARLY/Pmod>", [redacted]| MP |
EARLY <3/7/15/1.EARLY/Pmod>"],
        "ImageSizes": [
          [
            59,
            49,
            40,
            1
          ],
          [
            59,
            49,
            40,
            1
          ]
        ],
        "InputChannels": [
          "1.EARLY",
          "2.LATE"
        ],
        "InputModalities": [
          "PT",
          "PT"
        ]
      }
    }
  ]
```

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 Weight and Manifest (identified on 5. Weights & Manifest).



Weight... Tumor_Detection [DIR: C:/Pmod4.2/resources/pai/mri_tumor_detection/weights/Tumor_Detection/]

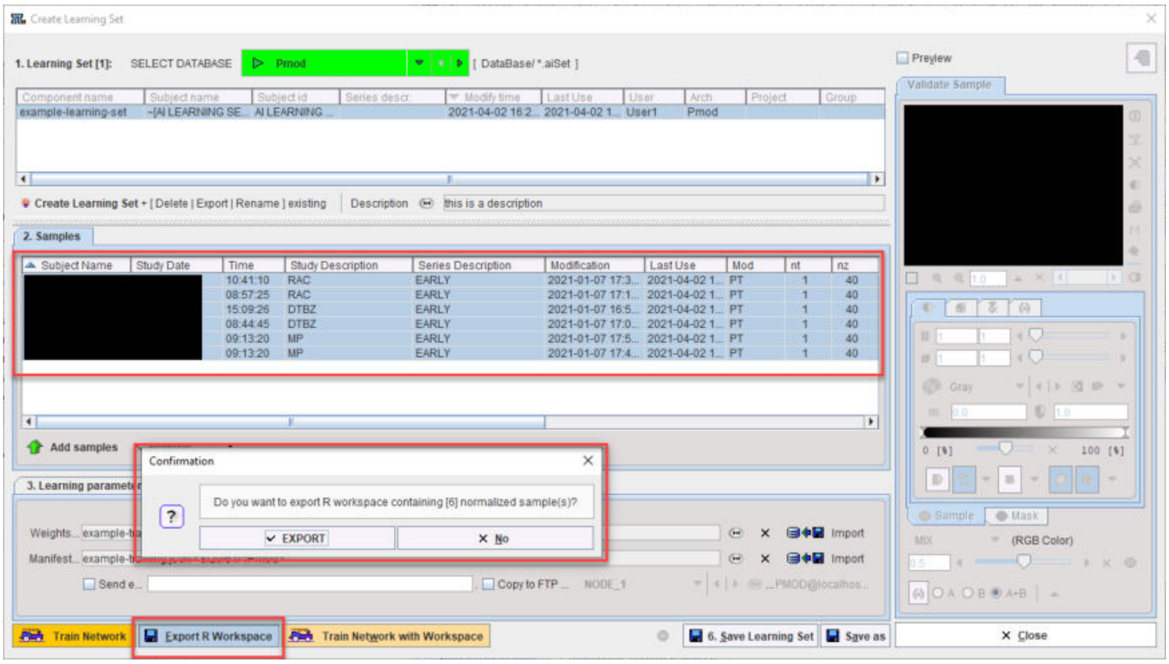
Manifes... Tumor_Detection [DIR: C:/Pmod4.2/resources/pai/mri_tumor_detection/weights/Tumor_Detection/]

□ Send ... □ Copy to FTP ... NODE_1 ...PMOD@localho...

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. Otherwise, a dialog window is shown listing the selected samples and providing an approximate RAM size needed for the training.



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
example-learning-set	~[AI LEARNING SE...	AI LEARNING ...		2021-04-02 16:2...	2021-04-02 1...	User1	Pmod		

Create Learning Set + [Delete | Export | Rename] existing Description: this is a description

2. Samples

Subject Name	Study Date	Time	Study Description	Series Description	Modification	Last Use	Mod	nt	nz
[redacted]	[redacted]	10:41:10	RAC	EARLY	2021-01-07 17:3...	2021-04-02 1...	PT	1	40
[redacted]	[redacted]	08:57:25	RAC	EARLY	2021-01-07 17:1...	2021-04-02 1...	PT	1	40
[redacted]	[redacted]	15:09:26	DTBZ	EARLY	2021-01-07 16:5...	2021-04-02 1...	PT	1	40
[redacted]	[redacted]	08:44:45	DTBZ	EARLY	2021-01-07 17:0...	2021-04-02 1...	PT	1	40
[redacted]	[redacted]	09:13:20	MP	EARLY	2021-01-07 17:5...	2021-04-02 1...	PT	1	40
[redacted]	[redacted]	09:13:20	MP	EARLY	2021-01-07 17:4...	2021-04-02 1...	PT	1	40

Confirmation

Do you want to export R workspace containing [6] normalized sample(s)?

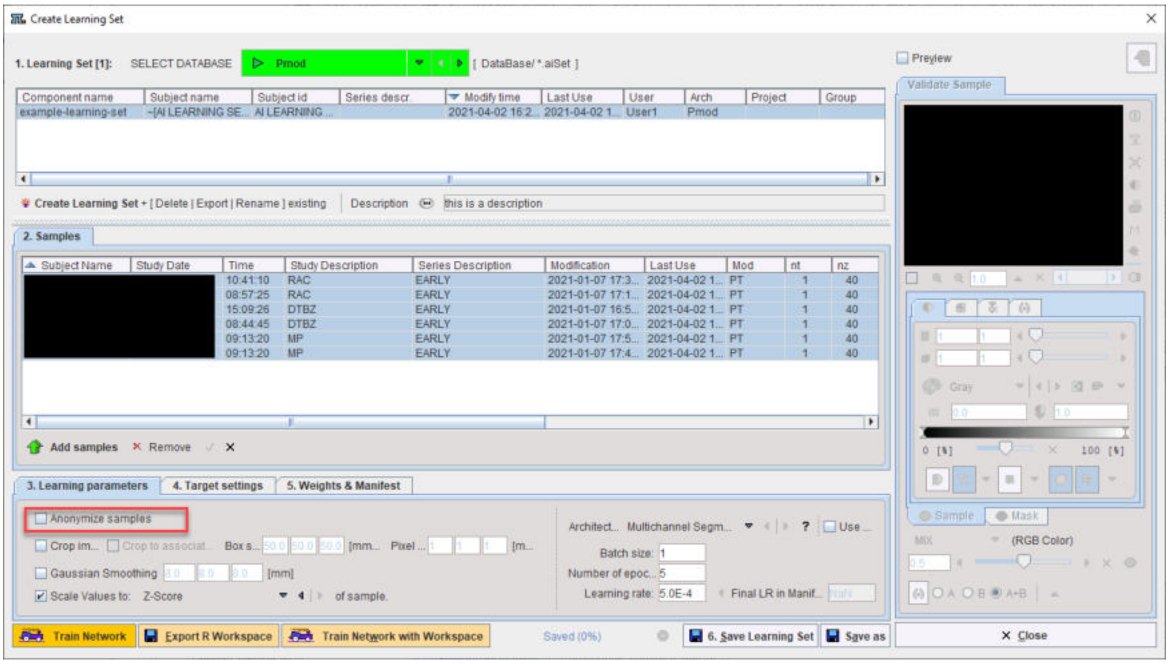
✓ EXPORT ✕ No

Weights... example-tr...

Manifest... example-t...

□ Send e... □ Copy to FTP ... NODE_1 ...PMOD@localhos...

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:



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
example-learning-set	~[AI LEARNING SE...	AI LEARNING ...		2021-04-02 16:2...	2021-04-02 1...	User1	Pmod		

Description: this is a description

2. Samples

Subject Name	Study Date	Time	Study Description	Series Description	Modification	Last Use	Mod	nt	nz
[redacted]	[redacted]	10:41:10	RAC	EARLY	2021-01-07 17:3...	2021-04-02 1...	PT	1	40
[redacted]	[redacted]	08:57:25	RAC	EARLY	2021-01-07 17:1...	2021-04-02 1...	PT	1	40
[redacted]	[redacted]	15:09:26	DTBZ	EARLY	2021-01-07 16:5...	2021-04-02 1...	PT	1	40
[redacted]	[redacted]	08:44:45	DTBZ	EARLY	2021-01-07 17:0...	2021-04-02 1...	PT	1	40
[redacted]	[redacted]	09:13:20	MP	EARLY	2021-01-07 17:5...	2021-04-02 1...	PT	1	40
[redacted]	[redacted]	09:13:20	MP	EARLY	2021-01-07 17:4...	2021-04-02 1...	PT	1	40

[redacted]

☐ Anonymize samples

☐ Crop im... ☐ Crop to associat... Box s... 50.0 50.0 50.0 [mm... Pixel ... 1 1 1 [m...

☐ Gaussian Smoothing 8.0 8.0 8.0 [mm]

☑ Scale Values to: Z-Score of sample.

Architect... Multichannel Segm... ☐ Use ...

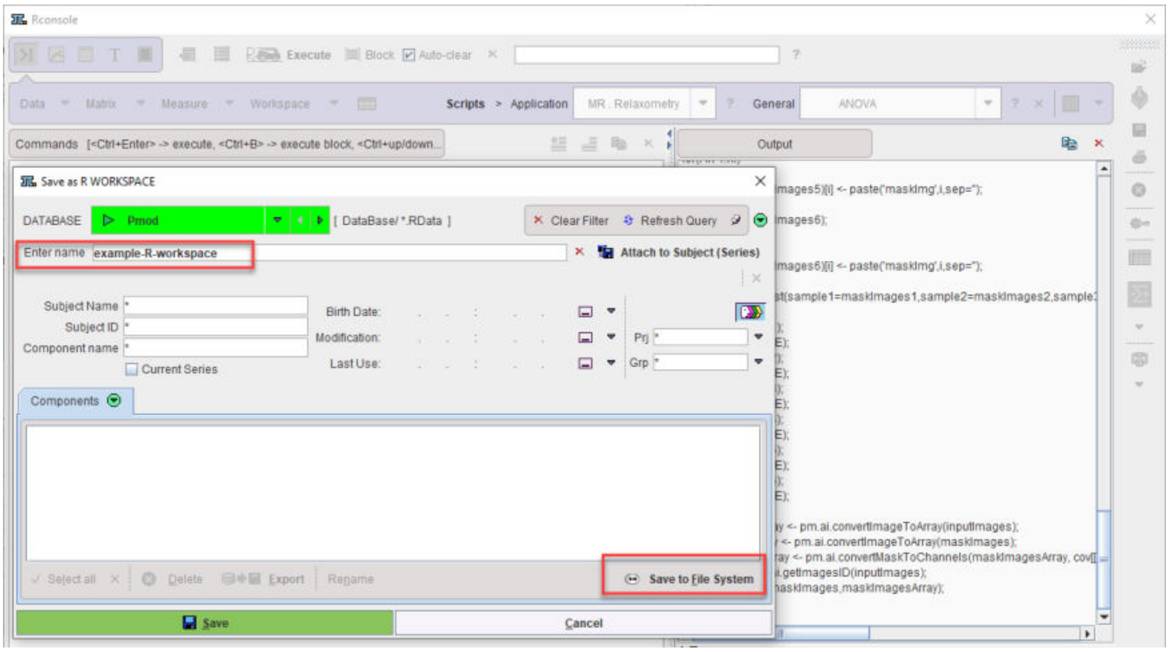
Batch size: 1

Number of epoc... 5

Learning rate: 5.0E-4 Final LR in Manif...

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:



Commands [<Ctrl+Enter> -> execute, <Ctrl+B> -> execute block, <Ctrl+up/down...

Save as R WORKSPACE

DATABASE Pmod [DataBase/ *.RData]

Enter name example-R-workspace

Subject Name *

Subject ID *

Component name *

☐ Current Series

Birth Date:

Modification:

Last Use:

Prj *

Grp *

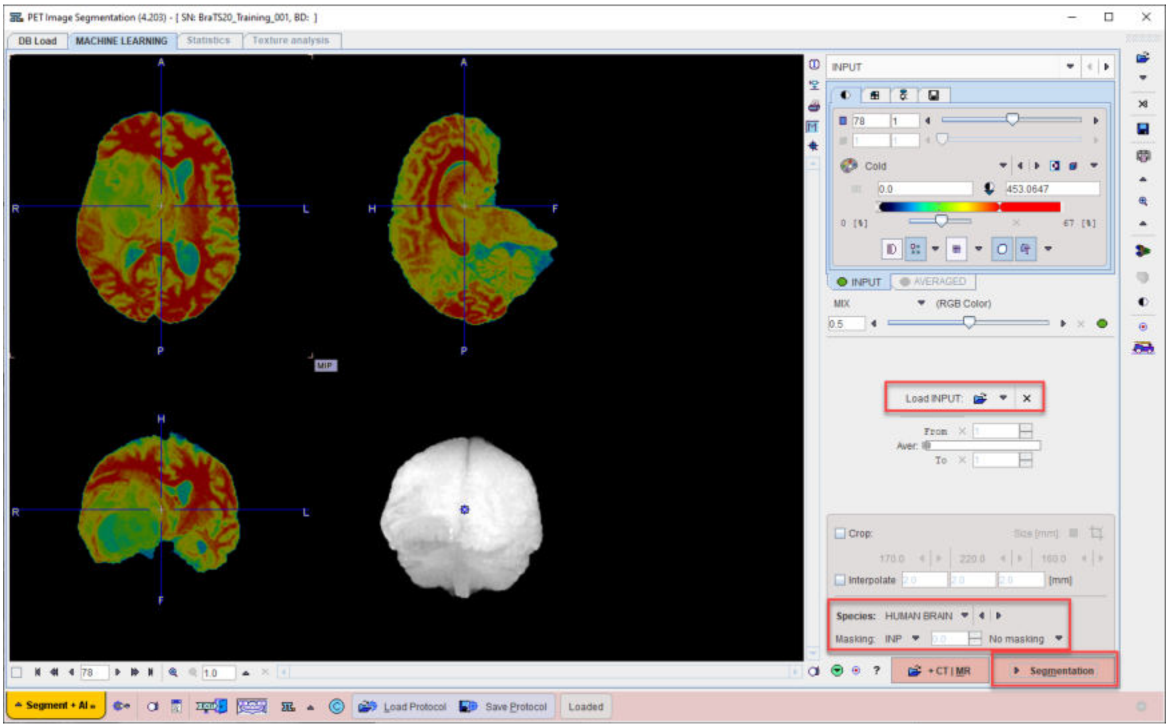
```
mages5)[i] <- paste('maskImg',i,sep='');
mages6);
mages6)[i] <- paste('maskImg',i,sep='');
st(sample1=maskImages1,sample2=maskImages2,sample
);
E);
);
E);
);
E);
);
E);
);
E);
);
E);
y <- pm.ai.convertImageToArray(inputImages);
<- pm.ai.convertImageToArray(maskImages);
ray <- pm.ai.convertMaskToChannels(maskImagesArray, cov
.getImagesID(inputImages);
naskImages,maskImagesArray);
```

5 Use of Trained Neural Network for Prediction

Trained networks can be used in PSEG for the segmentation of input images with the same characteristics as the training images. Please apply the same preparations 34 including associations and cropping VOI definition as for the training samples.

Image Loading

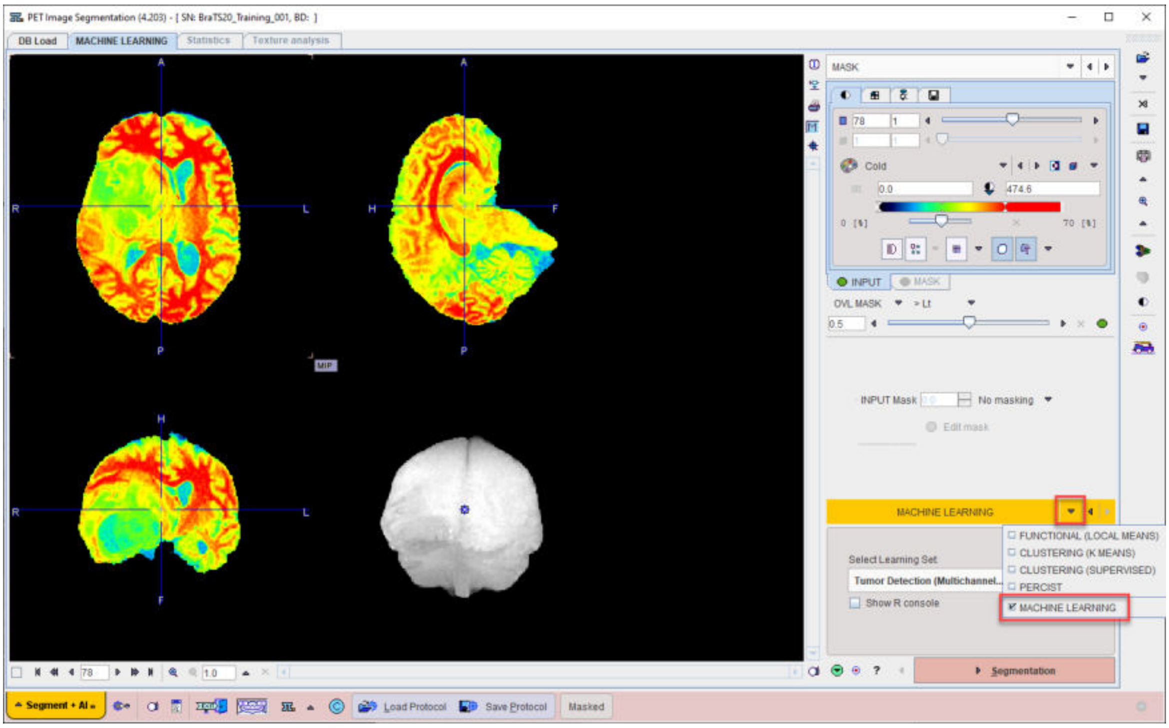
Start PSEG, and on the **INPUT** page use the **Load INPUT** button for loading the image to be segmented. In case of a segmentation requiring multiple input images, use the one which appears first in the association list.



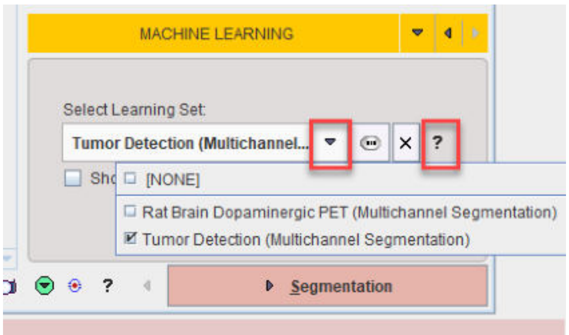
Do NOT apply cropping or interpolation in the lower right (this will be done by the model if needed), set **Masking** to **No Masking**, and continue using the **Segmentation** button.

Learning Set Configuration

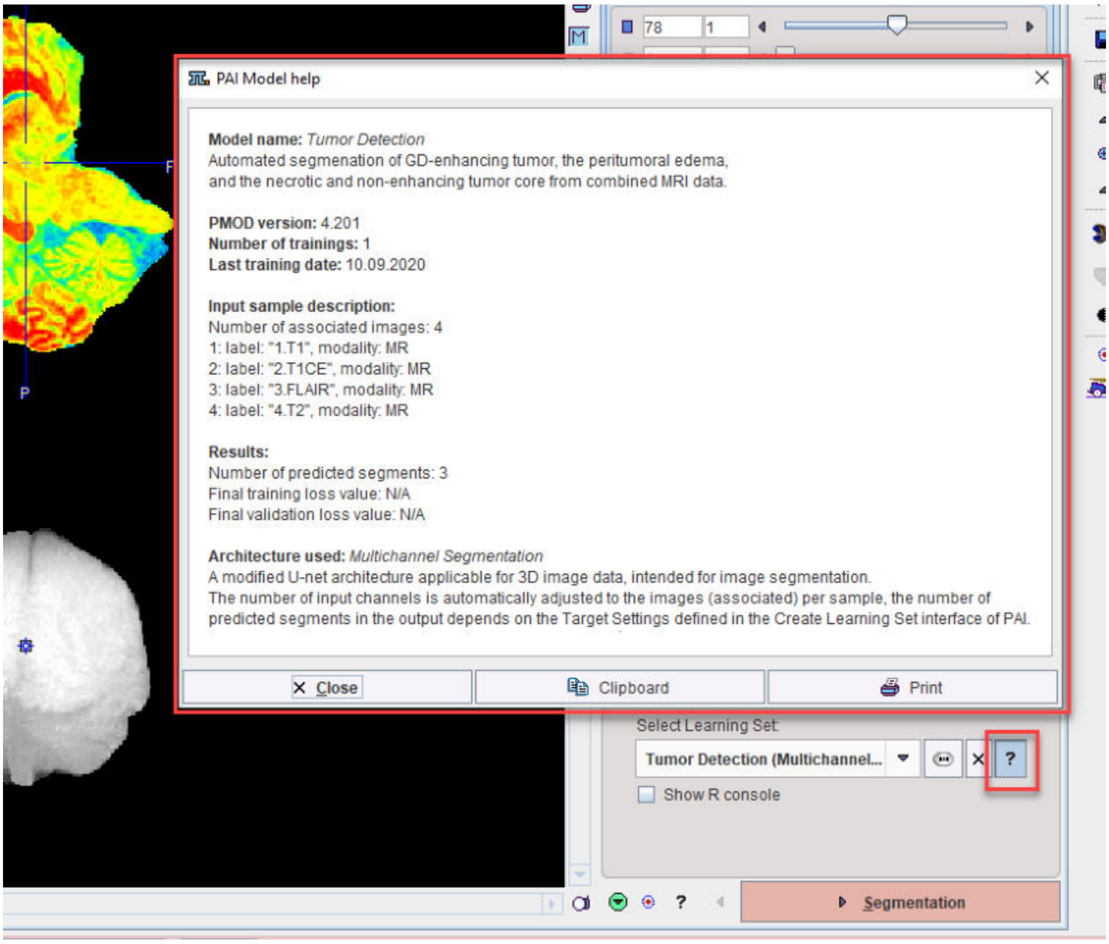
On the **MASK** page set the segmentation method to **MACHINE LEARNING**.



Then choose the appropriate model for the segmentation using the menu **Select Learning Set**.


MACHINE LEARNING
Select Learning Set:
Tumor Detection (Multichannel...
□ Sho
□ [NONE]
□ Rat Brain Dopaminergic PET (Multichannel Segmentation)
☑ Tumor Detection (Multichannel Segmentation)
?
Segmentation

Note that a description of the model can be retrieved using the **PAI Model Help** button:



PAI Model help

Model name: *Tumor Detection*
Automated segmentation of GD-enhancing tumor, the peritumoral edema, and the necrotic and non-enhancing tumor core from combined MRI data.

PMOD version: 4.201
Number of trainings: 1
Last training date: 10.09.2020

Input sample description:
Number of associated images: 4
1: label: "1.T1", modality: MR
2: label: "2.T1CE", modality: MR
3: label: "3.FLAIR", modality: MR
4: label: "4.T2", modality: MR

Results:
Number of predicted segments: 3
Final training loss value: N/A
Final validation loss value: N/A

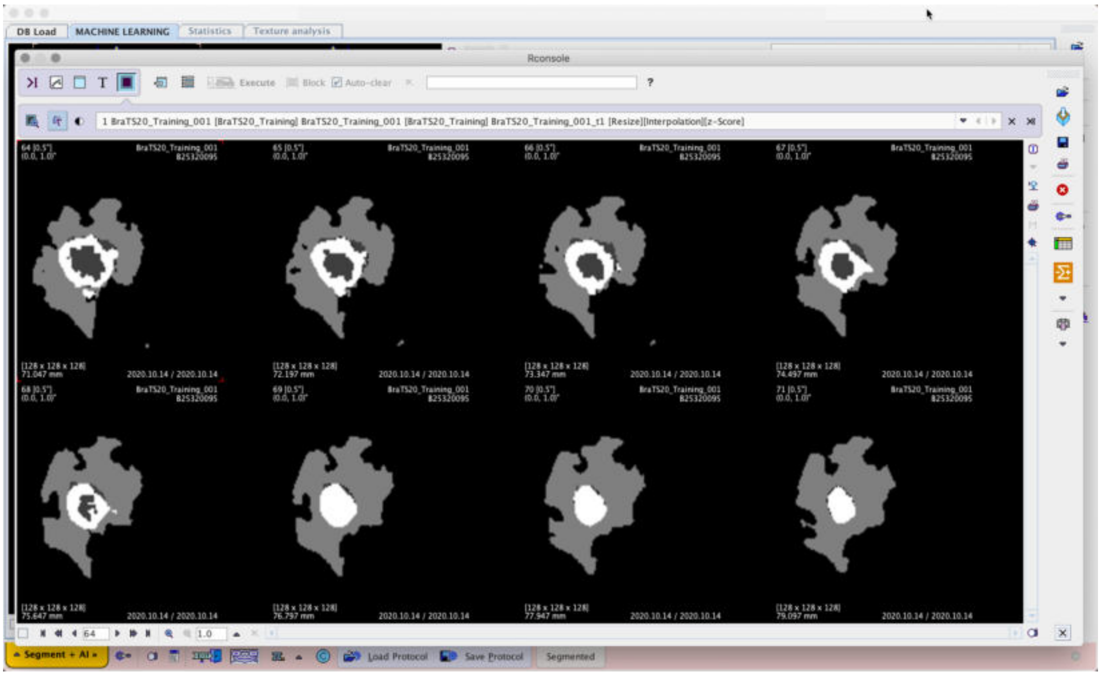
Architecture used: *Multichannel Segmentation*
A modified U-net architecture applicable for 3D image data, intended for image segmentation.
The number of input channels is automatically adjusted to the images (associated) per sample, the number of predicted segments in the output depends on the Target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

Select Learning Set: Tumor Detection (Multichannel...

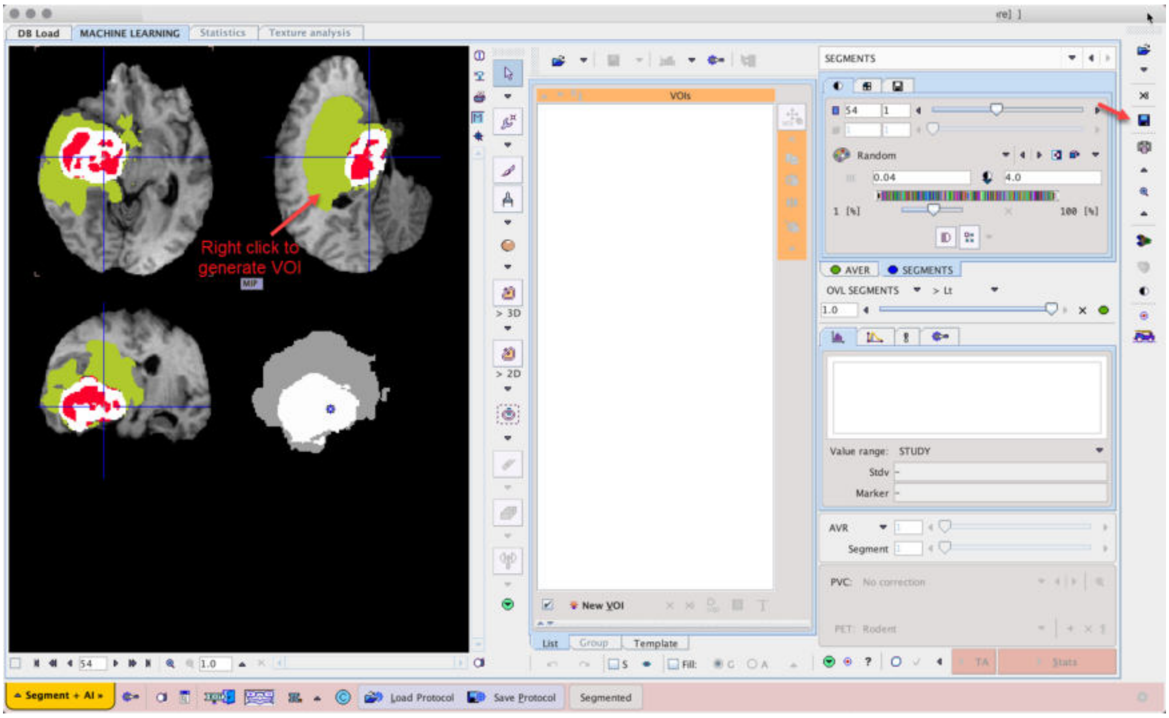
□ Show R console

Segmentation

Use the **Segmentation** workflow button to start prediction. The input data is transferred to the R Console and processed using the selected model. If **Show R console** was enabled, the resulting label maps are displayed on the image tab of the R Console:



The result of the segmentation is a label map which is overlaid on the input series on the **SEGMENTS** page (once the R Console has been closed).



The labels can be converted into a VOI 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, and then applies iso-contouring to generate the VOI. The label map itself can be saved using the save button in the taskbar to the right.

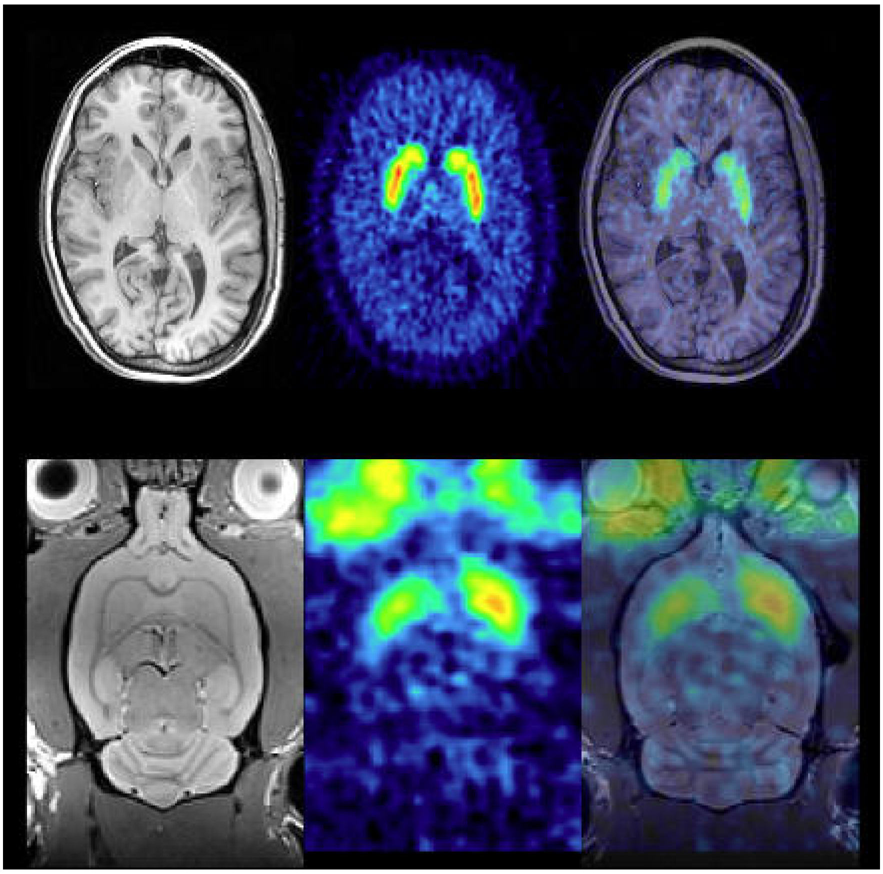
Please refer to the *PSEG User Guide* for details of the general segmentation functionality.

6 Case Study - Rat Brain Dopaminergic PET Model

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.

6.1 Sample Preparation

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



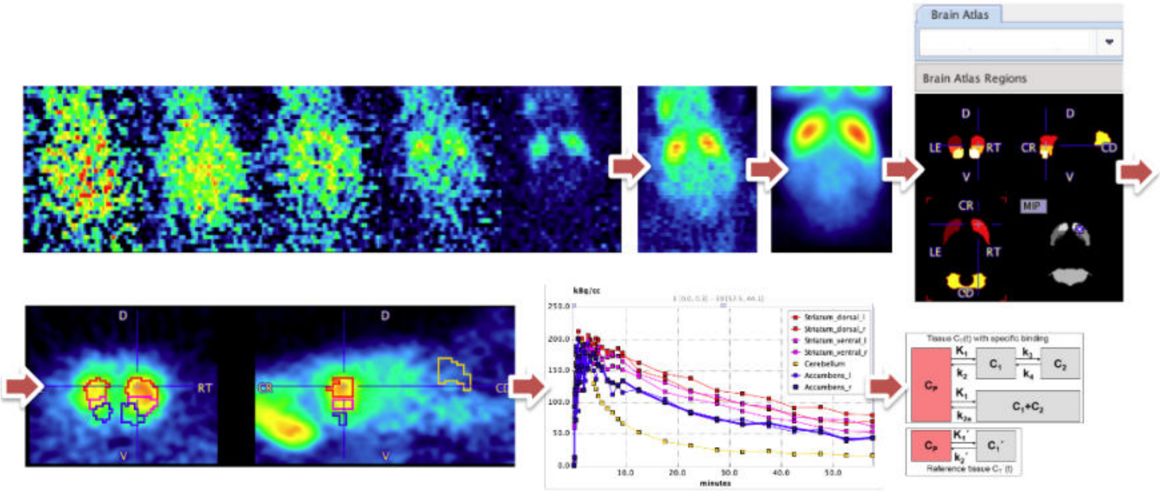
Fully quantitative analysis of PET with tracers for targets in the dopaminergic system typically uses time-activity curves from the striatum (in small animals; caudate and putamen in humans) and cerebellum (reference regions devoid of dopaminergic neurons/synapses). Dynamic PET studies typically cover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

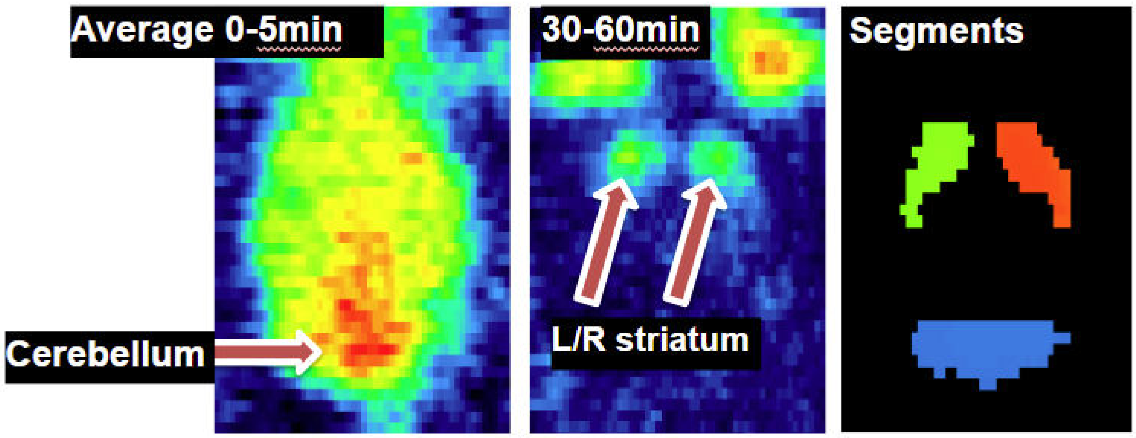
The traditional analysis workflow for this data is summarized below:



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**:



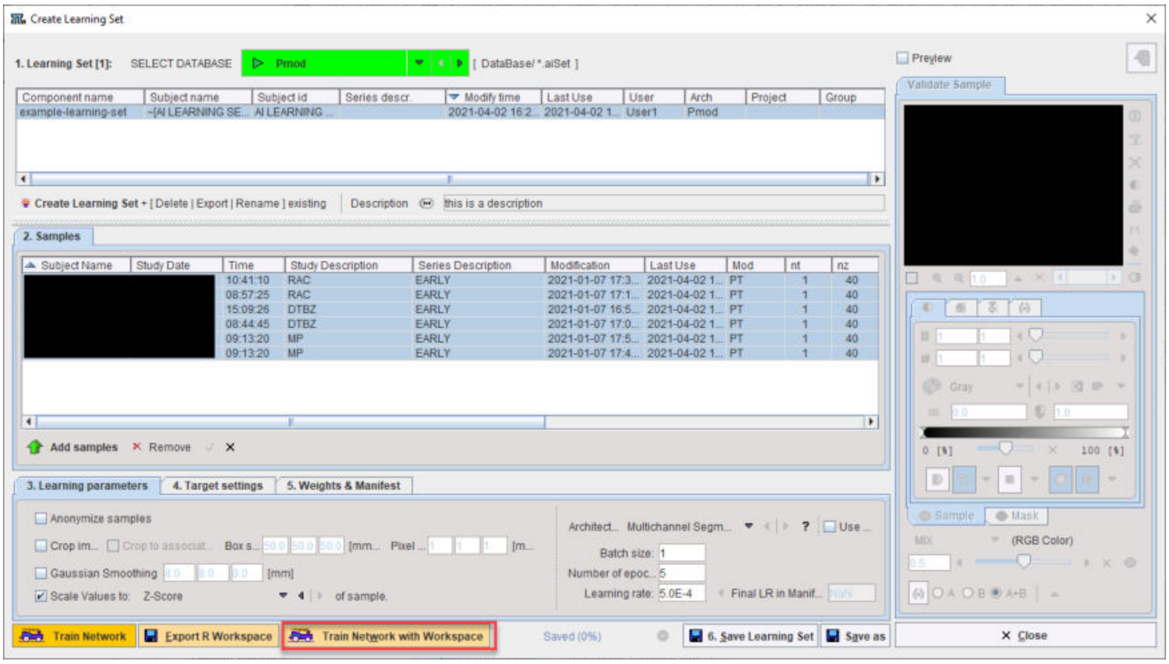
The VOI results from PNROD were converted into reference Segments using the **Mask By VOI Number** functionality available in the **View** tool and added to the **Database**.

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.

6.2 Training and Validation

The **Learning Set** was exported as an R workspace 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:



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
example-learning-set	~[AI LEARNING SE...	AI LEARNING ...		2021-04-02 16:2...	2021-04-02 1...	User1	Pmod		

Create Learning Set + [Delete | Export | Rename] existing Description: this is a description

2. Samples

Subject Name	Study Date	Time	Study Description	Series Description	Modification	Last Use	Mod	nt	nz
[redacted]	[redacted]	10:41:10	RAC	EARLY	2021-01-07 17:3...	2021-04-02 1...	PT	1	40
[redacted]	[redacted]	08:57:25	RAC	EARLY	2021-01-07 17:1...	2021-04-02 1...	PT	1	40
[redacted]	[redacted]	15:09:26	DTBZ	EARLY	2021-01-07 16:5...	2021-04-02 1...	PT	1	40
[redacted]	[redacted]	08:44:45	DTBZ	EARLY	2021-01-07 17:0...	2021-04-02 1...	PT	1	40
[redacted]	[redacted]	09:13:20	MP	EARLY	2021-01-07 17:5...	2021-04-02 1...	PT	1	40
[redacted]	[redacted]	09:13:20	MP	EARLY	2021-01-07 17:4...	2021-04-02 1...	PT	1	40

3. Learning parameters

- ☐ Anonymize samples
- ☐ Crop im... ☐ Crop to associat... Box s... 50.0 50.0 50.0 [mm... Pixel ... 1 1 1 [m...
- ☐ Gaussian Smoothing 8.0 8.0 8.0 [mm]
- ☑ Scale Values to: Z-Score of sample.

Architect... Multichannel Segm... ☐ Use ...

Batch size: 1

Number of epoc... 5

Learning rate: 5.0E-4

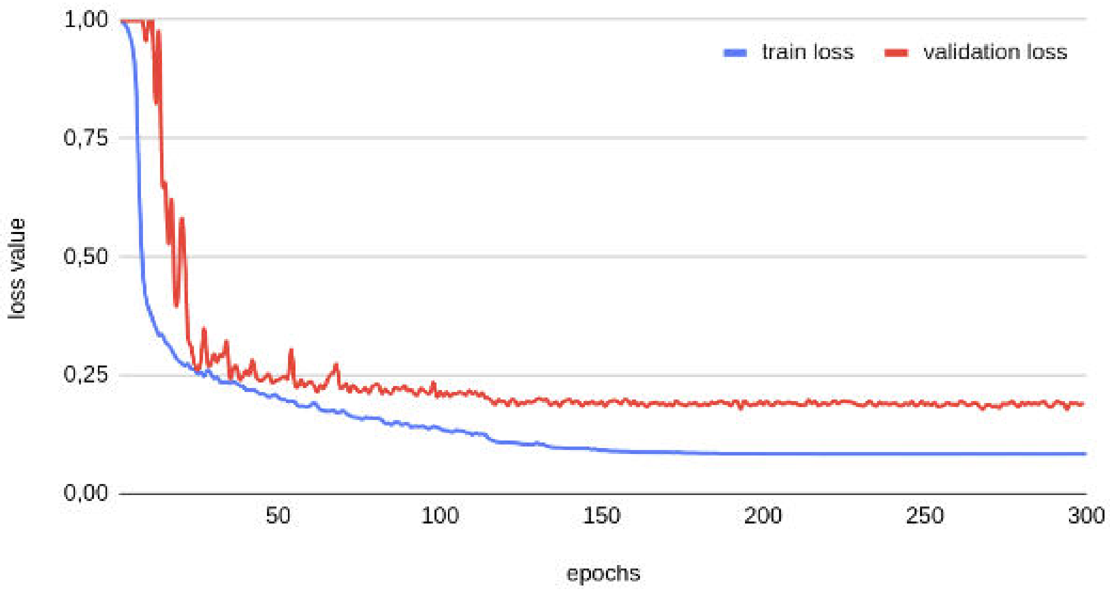
Final LR in Manif...

Saved (0%)

The parameters used were:

- 4 GPU V100, memory 488 GB
- 382 samples (305 training, 77 validation)
- pixel size 0.5 x 0.5 x 0.5 mm
- batch size 16
- Learning rate 0.05
- epochs 300

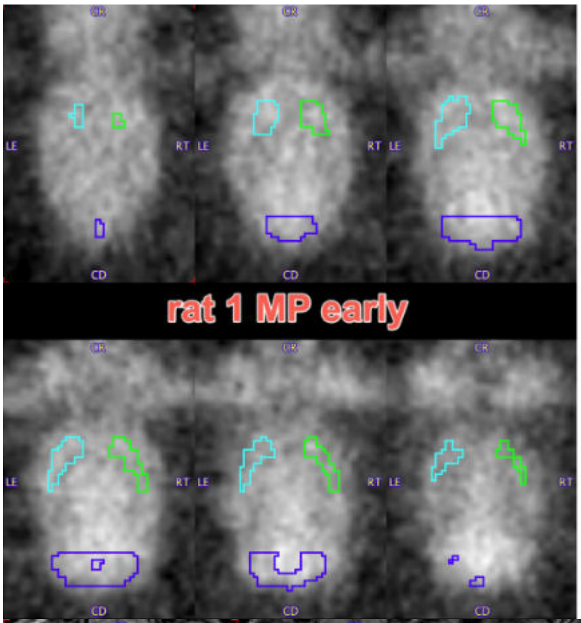
Training took 3 hours 8 minutes at 0.123 seconds/sample. The loss values for training and validation were extracted from the **Manifest** and plotted:

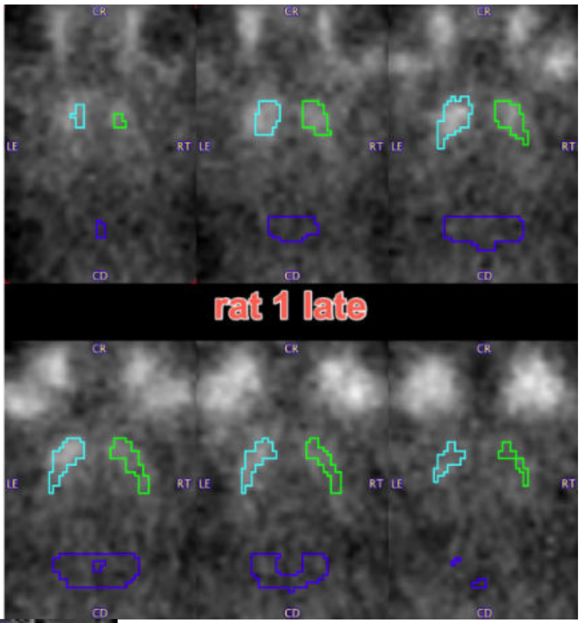


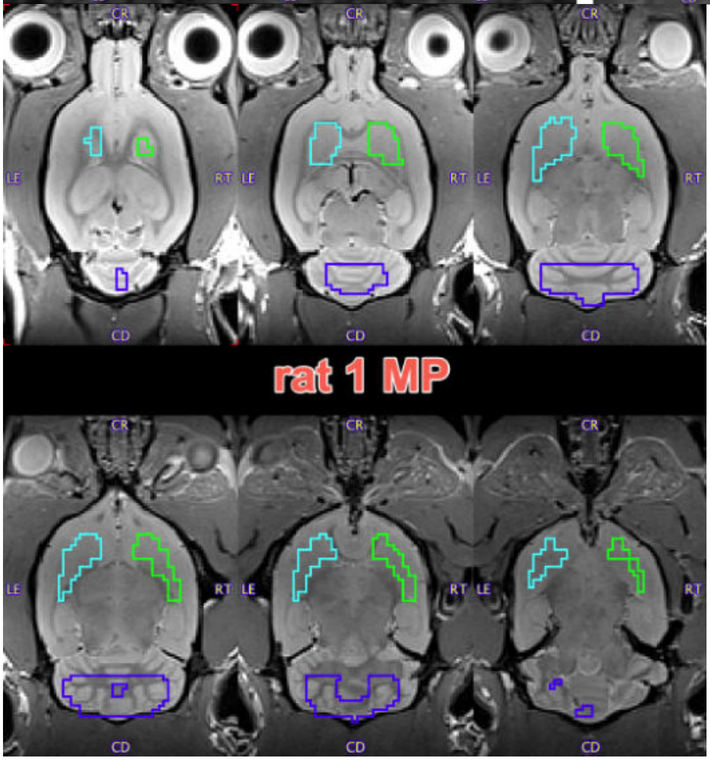
The plot illustrates how the loss value reduces with each epoch until a plateau is reached.

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 *Pmod4.2/resources/pai*. 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.

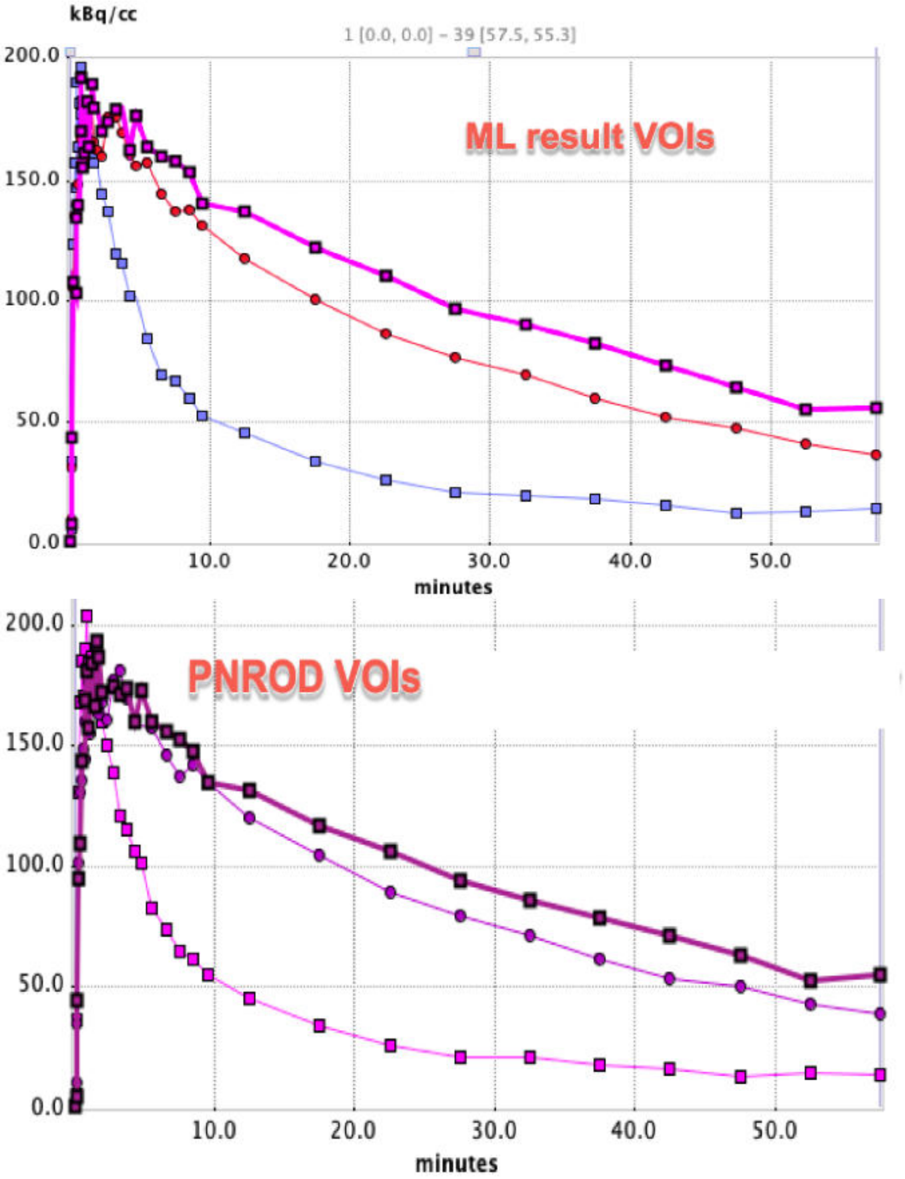
The resulting VOIs are shown below on the EARLY, LATE and anatomical reference MR images:







Time-activity curves were extracted from the dynamic PET series using both these VOIs and reference VOIs from PNROD. The two sets of TACs are shown below with matched axes:



The curve with faster washout in each case is the cerebellum.

7 PMOD Copyright Notice

PMOD Technologies LLC
Sumatrastrasse 25
8006 Zürich
Switzerland
+41 (44) 350 46 00
support@pmod.com
http://www.pmod.com