OXASL PCASL optimizer - Documentation

Release 0.0.1

Martin Craig

May 25, 2023

Contents:

OXASL_OPTPCASL is a package for generating optimal PLDs for PCASL experiments in order to maximise sensitivity to CBF, ATT or both.

CHAPTER 1

Installation

If using FSL, it is simplest to install using `fslpython` (however FSL is not required):

```
fslpython -m pip install oxasl_optpcasl --user
```

Otherwise can be installed into any Python 3 environment. Use of Conda or virtualenv is recommended:

```
python -m pip install oxasl_optpcasl
```

You may need to add `--user` if using system python.

CHAPTER 2

References

- Woods JG, Chappell MA, Okell TW. *A general framework for optimizing arterial spin labeling MRI experiments.* Magn Reson Med. 2019;81(4):2474-2488. doi:10.1002/mrm.27580
- Woods JG, Chappell MA, Okell TW. *Designing and comparing optimized pseudo-continuous Arterial Spin Labeling protocols for measurement of cerebral blood flow* NeuroImage 2020 223:1053-8119 doi:10.1016/j.neuroimage.2020.117246

2.1 Theory

This is a **brief** overvies of the theory behind multi-PLD PCASL optimization. For a more detailed account see Woods et al, 2018[1], 2020[2]

`OXASL_OPTPCASL` is based on minimising the Cramer-Rao lower bound on the variance of the parameters estimated from the data which for a multi-PLD PCASL experiment is typically the CBF and the ATT.

In this case the lower bound is obtained from the inverse of the Fisher information matrix which takes the following form:

$$F(t;\theta)_{jk} = \frac{A}{\sigma^2}\sum_{i=1}^{N}\frac{\partial\Delta M(t_i;\theta)}{\partial\theta_j}\frac{\partial\Delta M(t_i;\theta)}{\partial\theta_k}$$

Here N is the number of acquisitions (PLDs), A is the number of averages (repeats) for each PLD, σ^2 is the normally distributed noise variance and t are the timings (i.e. PLDs).

$\Delta M(t;\theta)$ is a model for the ASL signal (magnitude difference between labelled and control images) based on model parameters θ which in this case we will take as being CBF and ATT only.

[1] Woods JG, Chappell MA, Okell TW. A general framework for optimizing arterial spin labelling MRI experiments. Magn Reson Med. 2019;81:2474-2488. https://doi.org/10.1002/mrm.27580

[2] Woods JG, Chappell MA, Okell TW. Designing and comparing optimized pseudo-continuous Arterial Spin Labeling protocols for measurement of cerebral blood flow. NeuroImage 2020 223:1053-8119 doi:10.1016/j.neuroimage.2020.117246

`OPTPCASL` supports two approaches to minimising the Cramer-Rao lower bound. In each case the elements of the Fisher information matrix are evaluated by differentiating the basic PCASL kinetic model:

$$\begin{aligned}\Delta M(t) &= 0 && 0 < t < \Delta t\\ &= 2M_{0B}fT_1'\alpha\exp(\tfrac{-\Delta t}{T_{1b}})(1-\exp\tfrac{-(t-\Delta t)}{T_1'}) && \Delta t < t < \tau + \Delta t\\ &= 2M_{0B}fT_1'\alpha\exp(\tfrac{-\Delta t}{T_{1b}})\exp\tfrac{-t-\tau-\Delta t}{T_1'}(1-\exp\tfrac{-\tau}{T_1'}) && \tau + \Delta t < t\end{aligned}$$

The apparent T_1 relaxation time T_1' is calculated assuming a fixed CBF of 50 ml/100g/min - this limitation in practice leads to an insignificant error in the calculation of the kinetic model[1].

2.1.1 Optimising for both CBF and ATT variance

In this case we seek to minimise the cost function:

$$C = \frac{1}{det(F(t;\theta)})$$

i.e. minimising the overall magnitude of the covariance matrix

2.1.2 Optimising for either CBF or ATT variance only

In this case we simply minimise a particular component of F^{-1} which corresponds to the variance of the parameter of interest - e.g. if CBF is the first parameter we would minimise F_{00}^{-1} to optimize for CBF and minimise F_{11}^{-1} to optimize for ATT.

2.1.3 Priors

Since the elements of F depend on the parameters being estimated we adopt a pseudo-Bayesian approach where the cost function being minimised is averaged across a prior distribution for the parameters.

In the case of the ATT, a essentially uniform distribution is used with minimum and maximum ATT values. In addition, a *tapering* parameter is supported which reduces the probability density linearly within a fixed distance from the minimum and maximum times (default: 0.3s).

A prior distribution for CBF is not required since (with the use of a fixed CBF in the estimation of T_1'), the cost function is essentially independent of CBF.

2.1.4 Implementation of the cost minimisation algorithm

Minimisiation is accomplished by an iterative search method looping over the PLDs in turn. For each PLD the optimal value is obtained for that PLD allowing it to vary between the current values of the preceding and successive PLD. This optimization loop is repeated until the set of PLDs is unchanged.

References

2.2 Command line tool user guide

The command line tool is `oxasl_optpcasl`. The default arguments optimize for a 6 PLD PCASL acquisition:

```
$ oxasl_optpcasl
OXASL - PCASL Optimizer 0.0.1.post12
====================================
Optimizing PLDs for 300s 3D scan with readout time 0.500000s
PLD search limits: PLDs between 0.10s and 3.00s in steps of 0.02500s
Optimizing for 6 PLDs
BAT distribution: 1901 values between 0.20s and 2.100000s (weight taper=0.30s)
Optimization method: D-optimal

Finished optimization after 30 iters - PLDs unchanged
Optimal PLDs: [0.2, 0.7, 0.725, 1.55, 1.875, 2.075]
num_av = 8
Scan time = 296.400000
DONE
```

2.2.1 Varying the number of PLDs

Use the command line option `--scan-npld=<n>`:

```
$ oxasl_optpcasl --scan-npld=8
OXASL - PCASL Optimizer 0.0.1.post12
====================================
Optimizing PLDs for 300s 3D scan with readout time 0.500000s
PLD search limits: PLDs between 0.10s and 3.00s in steps of 0.02500s
Optimizing for 8 PLDs
BAT distribution: 1901 values between 0.20s and 2.100000s (weight taper=0.30s)
Optimization method: D-optimal

Finished optimization after 48 iters - PLDs unchanged
Optimal PLDs: [0.2, 0.7, 0.7, 0.725, 1.5, 1.775, 1.95, 2.1]
num_av = 6
Scan time = 298.200000
DONE
```

2.2.2 Optimizing for CBF or ATT individually

Use the command line options `--optimize=CBF` or `--optimize=ATT`:

```
$ oxasl_optpcasl --optimize=CBF
OXASL - PCASL Optimizer 0.0.1.post12
====================================
Optimizing PLDs for 300s 3D scan with readout time 0.500000s
PLD search limits: PLDs between 0.10s and 3.00s in steps of 0.02500s
Optimizing for 6 PLDs
BAT distribution: 1901 values between 0.20s and 2.100000s (weight taper=0.30s)
Optimization method: L-optimal

Finished optimization after 24 iters - PLDs unchanged
Optimal PLDs: [0.2, 1.175, 1.8, 2.025, 2.1, 2.1]
num_av = 7
Scan time = 291.200000
DONE
```

2.2.3 Varying the ATT prior distribution

The ATT prior distribution determines the range of arterial transit times that are considered relevant when optimizing the performance of the protocol. The default range is from 0.2s to 2.3s, with the weighting of the upper and lower 0.3s being linearly tapered to zero.

This is quite a wide range, and in some cases it may be more sensible to restric the ATT prior to a more limited range. For example the following will perform optimization based on ATTs between 1 and 2s, with the range 1s to 1.1s and the range 1.9s to 2s tapered in weight to zero:

```
$ oxasl_optpcasl --att-start=1.0 --att-end=2.0 --att-step=0.001 --att-taper=0.1
OXASL - PCASL Optimizer 0.0.1.post21
====================================
Optimizing PLDs for 300s 3D scan with readout time 0.500000s
PLD search limits: PLDs between 0.10s and 3.00s in steps of 0.02500s
Optimizing for 6 PLDs
BAT distribution: 1000 values between 1.00s and 2.000000s (weight taper=0.10s)
Optimization method: D-optimal

Finished optimization after 24 iters - PLDs unchanged
Optimal PLDs: [0.6, 0.6, 0.6, 1.575, 1.8, 2.025]
num_av = 8
Scan time = 297.600000
DONE
```

2.2.4 Varying the PLD search limits

This example uses finer search parameters to determine optimal PLDs. Using very fine search limits gives a more accurate result but the optimization will take longer. In this example we also increase the search range of PLDs up to 5s (although it turns out that such long PLDs are not optimal in this case and the largest value returned is 2.08s):

```
$ oxasl_optpcasl --pld-min=0.1 --pld-max=5 --pld-step=0.01
OXASL - PCASL Optimizer 0.0.1.post12
====================================
Optimizing PLDs for 300s 3D scan with readout time 0.500000s
PLD search limits: PLDs between 0.10s and 5.00s in steps of 0.01000s
Optimizing for 6 PLDs
BAT distribution: 1901 values between 0.20s and 2.100000s (weight taper=0.30s)
Optimization method: D-optimal


Finished optimization after 30 iters - PLDs unchanged
Optimal PLDs: [0.2, 0.7, 0.71, 1.54, 1.87, 2.08]
num_av = 8
Scan time = 296.000000
DONE
```

2.2.5 Varying other scan properties

Here we optimize for a longer scan and modify the readout time (in seconds). Note that the number of averages (repeats) has increased:

```
$ oxasl_optpcasl --scan-readout=0.75 --scan-duration=500
OXASL - PCASL Optimizer 0.0.1.post12
```

(continues on next page)

(continued from previous page)

```
====================================
Optimizing PLDs for 500s 3D scan with readout time 0.750000s
PLD search limits: PLDs between 0.10s and 3.00s in steps of 0.02500s
Optimizing for 6 PLDs
BAT distribution: 1901 values between 0.20s and 2.100000s (weight taper=0.30s)
Optimization method: D-optimal

Finished optimization after 36 iters - PLDs unchanged
Optimal PLDs: [0.2, 0.7, 0.7, 0.925, 1.8, 2.0]
num_av = 13
Scan time = 499.850000
DONE
```

2.3 GUI user guide

The `OXASL_OPTPCASL` GUI is started using the command `oxasl_optpcasl_gui`

2.3.1 GUI requirements

The wxpython GUI library is required for the GUI. This is not included as a requirements of `oxasl_optpcasl` since it is possible to use the optimizer solely through the command line. So, if you want to use the GUI you will need to install `wxpython`, for example using one of the following:

```
python -m pip install wxpython
conda install wxpython
```

2.3.2 GUI errors on Mac

A common issue on Mac when a Conda environment is being used is that the GUI will fail to start with a message about requiring a 'Framework Build'.

This is a well known issue with Conda which to date has not been fixed. To work around the problem you need to modify the wrapper script, for example using:

```
nano `which oxasl_optpcasl_gui`
```

If you are using FSL and installed *oxasl_optpcasl* into *fslpython* then change the first line to:

```
#!/usr/bin/env fslpythonw
```

Otherwise, change the first line to:

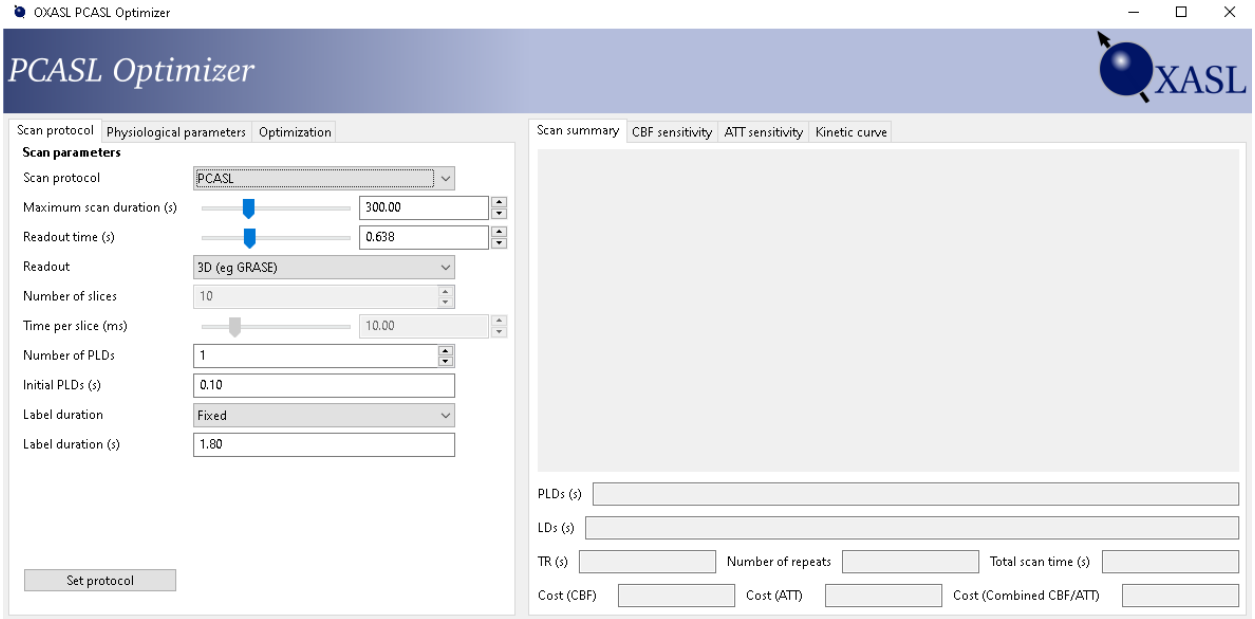
```
#!/usr/bin/env pythonw
```

This should enable to the script to run.

2.3.3 Overview of the GUI

The window is divided into two horizontally. The left hand pane is used to set properties of the scan protocol, assumed physiological parameters and options for the optimization process. The right hand pane displays characteristics of

the selected protocol including sensitivity to ATT and CBF and illustrations of the relationship between the effective PLDs of the protocol and the PCASL kinetic curve (using the Buxton model).

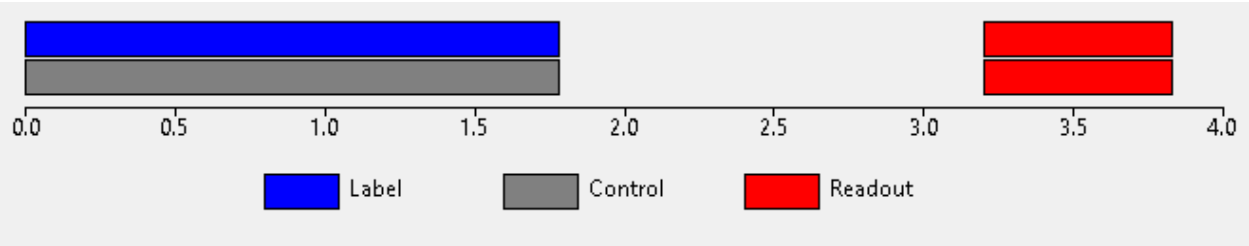


2.3.4 Setting the scan parameters

On the first tab we set the parameters for the scan we want to optimize. There are three basic scan protocols:

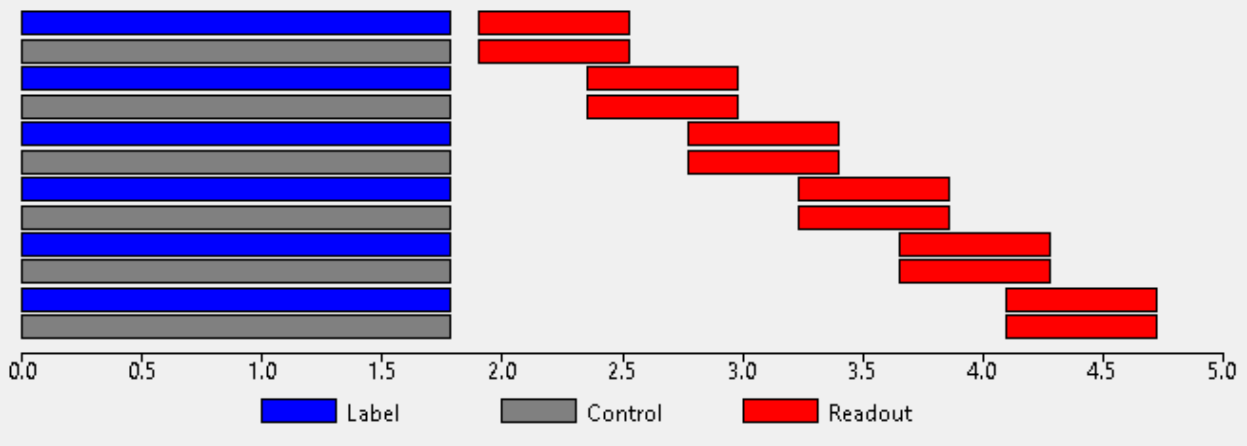
PCASL protocol

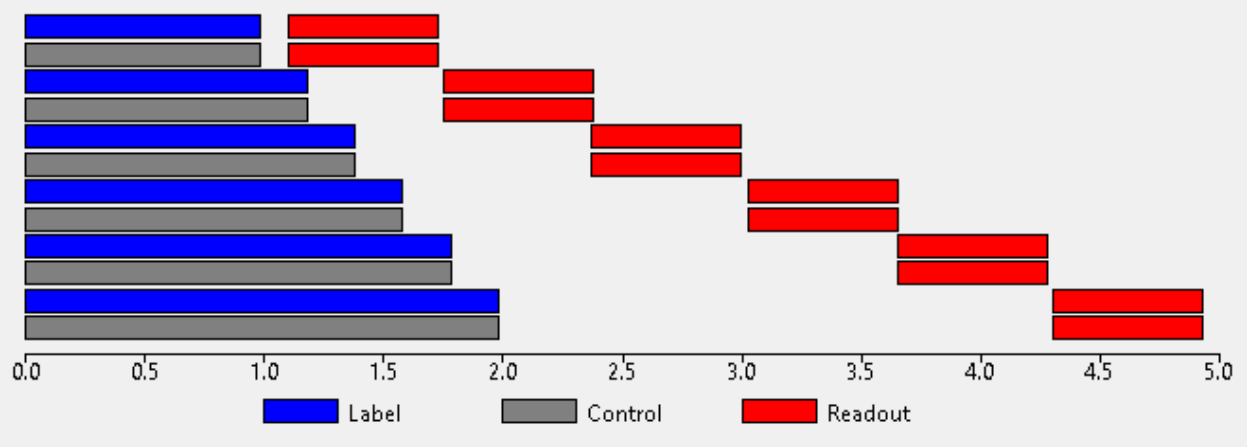
This protocol is a standard PCASL acquisition which consists of an inversion (labelling) pulse followed by a post-labelling delay (PLD) followed by readout. This cycle is then repeated but without the labelling pulse, substituting a matching control delay so the total time to readout is the same. For a single PLD this can be depicted graphically as follows:



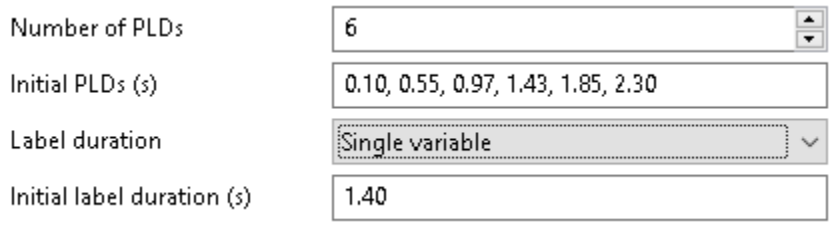
Subtraction of the labelled image from the control image is used to obtain the ASL signal.

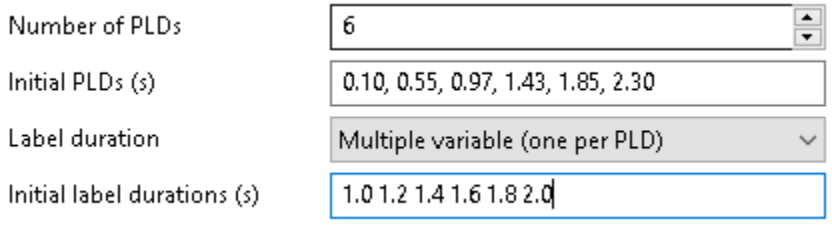
This label/control cycle can be repeated with different PLDs in order to sample the ASL kinetic curve at different points. The different PLDs may be paired with the same labelling duration (LD) or each PLD may have a different LD:





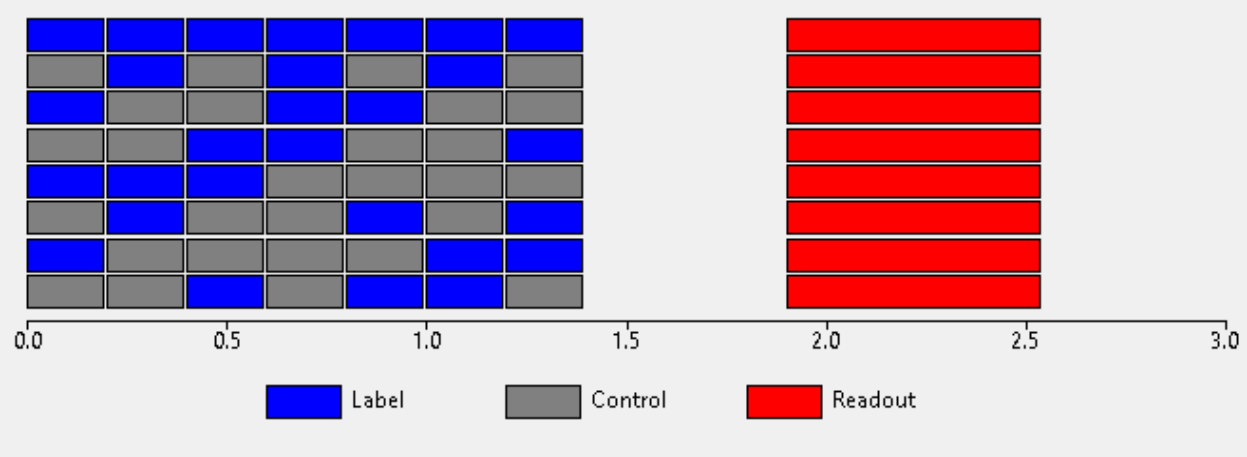
When using a PCASL protocol the initial PLDs must be specified as a space or comma separated list of values in seconds. The labelling duration may either be fixed (one value which is not optimized), single (one initial value which is optimized) or multiple (one initial value per PLD, all optimized).





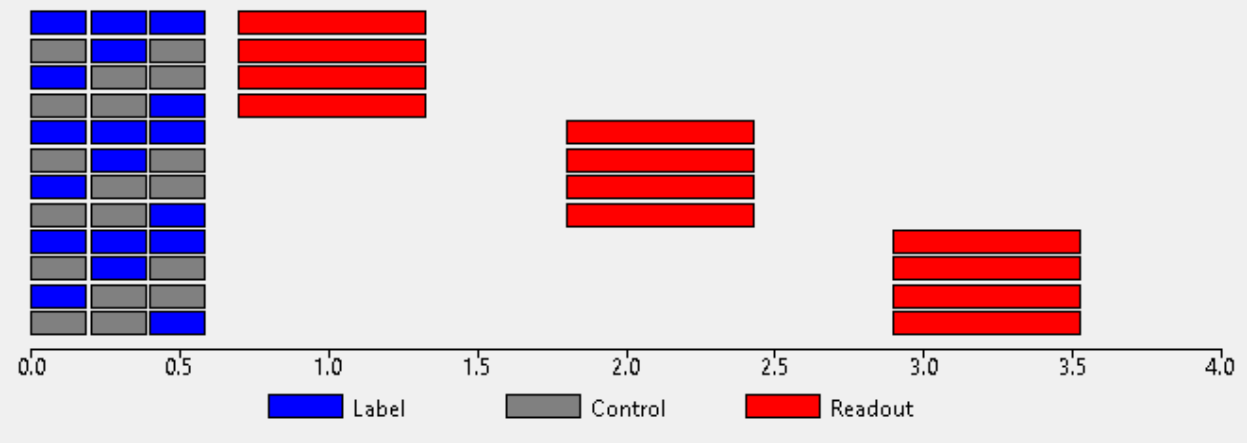
Hadamard time-encoded protocol

In this protocol, the labelling phase consists of a *sequence* of 'sub-boli' each of which may be either an inversion pulse or a control delay. After the full set of sub-boli there is again a post-labelling delay followed by readout. The cycle is repeated but with the label/control allocation of each sub-bolus changed for each repeat. The pattern of these label/control sub-boli across all the acquisition cycles forms a Hadamard matrix. Sizes of 4, 8 and 12 may be used (the number of sub-boli is one less than the matrix size). For example a Hadamard acquisition using a matrix of size 8 can be depicted as follows:

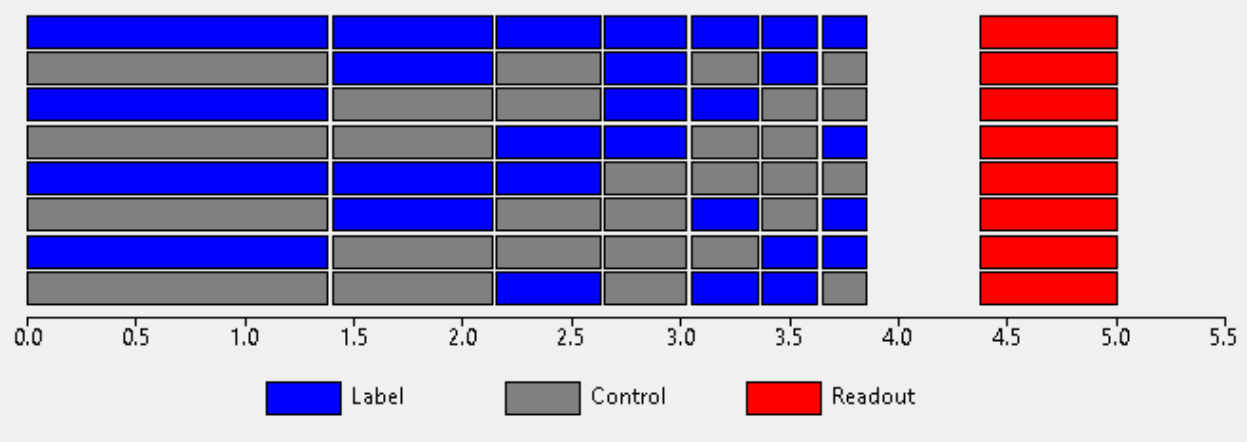


To obtain the ASL signal the images must be decoded by adding and subtracting each cycle. By doing this in different ways the contribution of each sub-bolus can be isolated without the need for a corresponding control cycle. Hence Hadamard encoding allows us to obtain 7 subtracted images from only 8 label/control cycles (conventional PCASL would require 14 label/control cycles to produce 7 subtracted images). We can therefore potentially extract more information in a given time.

Commonly Hadamard protocols only use a single PLD since each sub-bolus has a different effective PLD when decoded. However it is possible to repeat the entire Hadamard sequence at multiple PLDs, for example this acquisition (using a matrix size of 4 to reduce the time required):

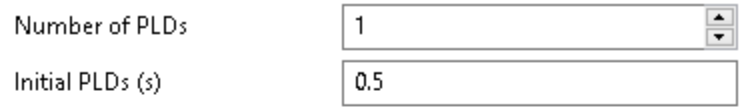


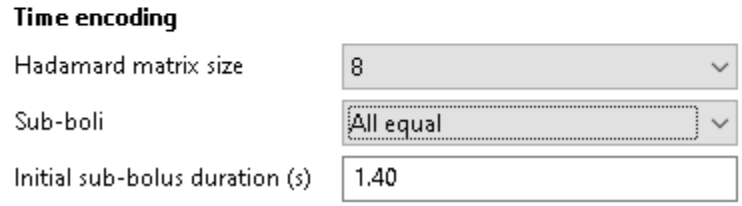
The lengths of the Hadamard sub-boli may all be equal, however do not have to be. Once approach is to choose the length of the first sub-bolus and then choose shorter lengths for the remainder to account for T1 decay in such a way that the total signal from each sub-bolus is the same:



Alternatively, all the sub-boli durations may be specified (and optimized) independently.

With the Hadamard protocol, the matrix size must be specified and the strategy for choosing sub-bolus durations. If these are all equal then a single initial value must be given. When using the T1 decay strategy the first sub-bolus duration is specified. When sub-bolus durations are unconstrained the initial values must be specified as space or comma separated values in seconds (one less than the matrix size):

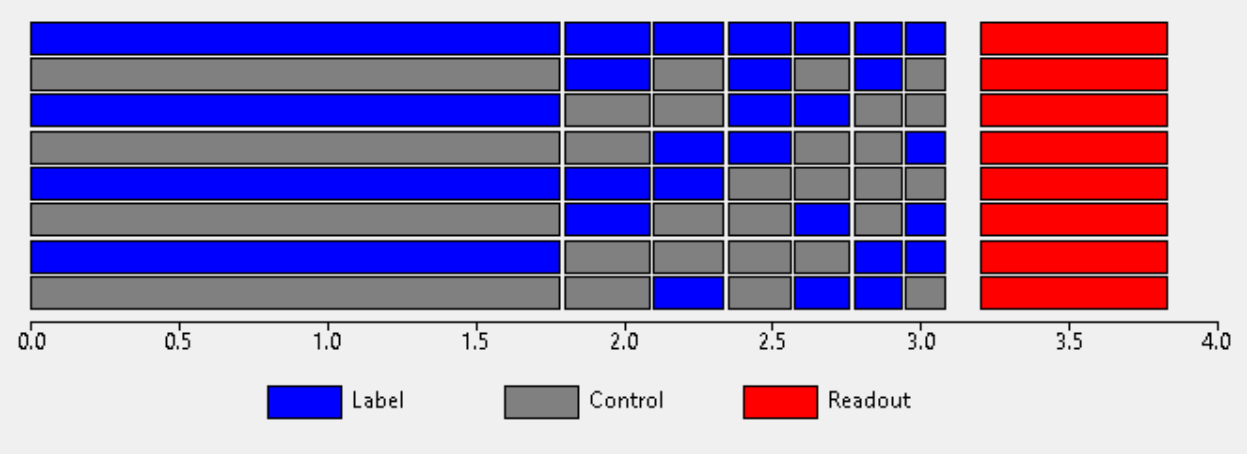




images/gui/had_opts_multiple.png

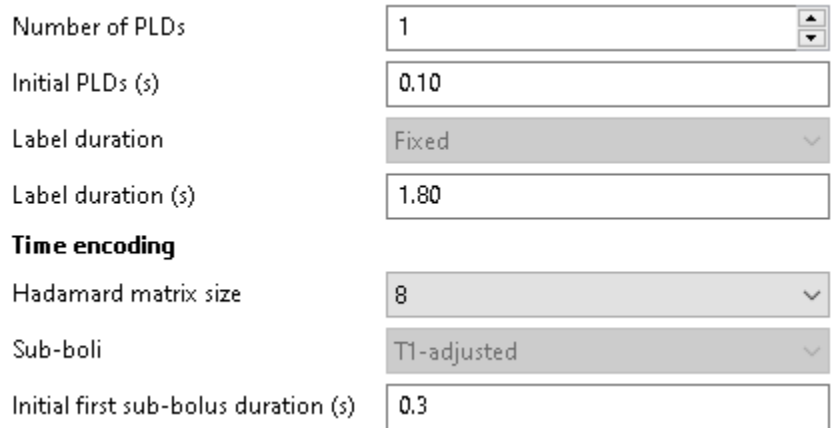
'Free lunch' protocol

This is essentially a Hadamard protocol, however the first sub-bolus is fixed at a 'long' value typically equal to that used in single-PLD PCASL experiments. The idea here is to do a 'normal' single delay PCASL experiment but fill in the long PLD with additional Hadamard encoded sub-boli. The resulting images can either be analysed as a collection of 4 single-PLD label/control pairs (the remaining sub-bolus contributions cancel out in this case), or undergo Hadamard decoding to gain additional information 'for free'. Typically we optimize the duration of the first of these sub-boli and assign the remainder using the T1 decay strategy.



Note the similarity of this to the single-PLD experiment shown above - they are essentially the same but with the 'spare' PLD space filled with time-encoded labelling pulses for additional information.

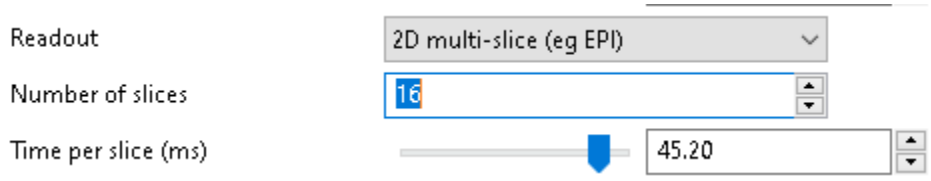
With the free-lunch protocol, a single fixed labelling duration must be given - this is not optimized. An initial value for the next sub-bolus duration must also be given, this is optimized and the remainder are chosen according to the T1 decay strategy. One or more initial PLDs is also required:



Currently it is not possible to change the strategy used to select sub-bolus durations, nor to optimize the initial 'long' LD. This may be added in the future.

2D/3D readout

If we are using a 2D multi-slice readout, additional options for the time per slice and number of slices are enabled:



2.3.5 Displaying information about the protocol

Clicking the `Set Protocol` button loads the initial protocol design into the output tabs on the right side of the viewing pane. Nothing has been optimized yet, but you can view information about the protocol's sensitivity to ATT

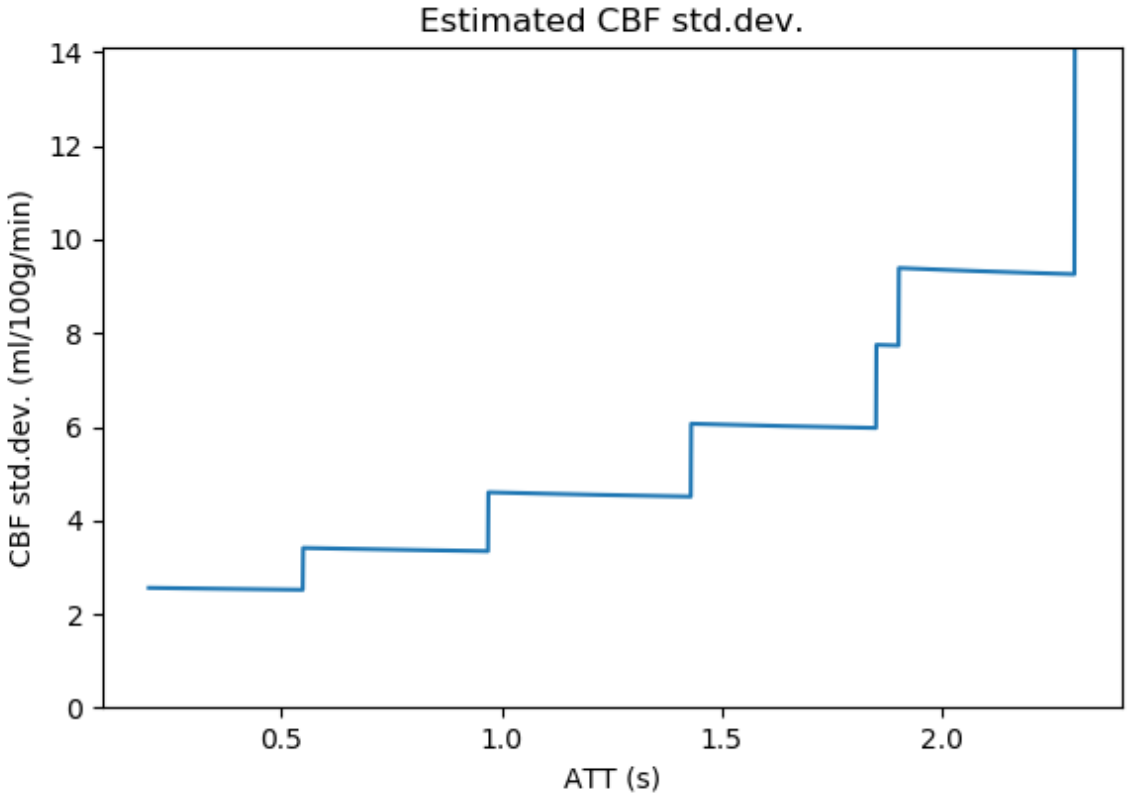
and CBF here.

The protocol summary

This tab shows a graphical representation of the protocol together with the calculated cost values for measuring CBF, ATT and combined CBF/ATT cost. The optimization process will seek to modify the PLDs/LDs of the protocol to minimise one of these values.

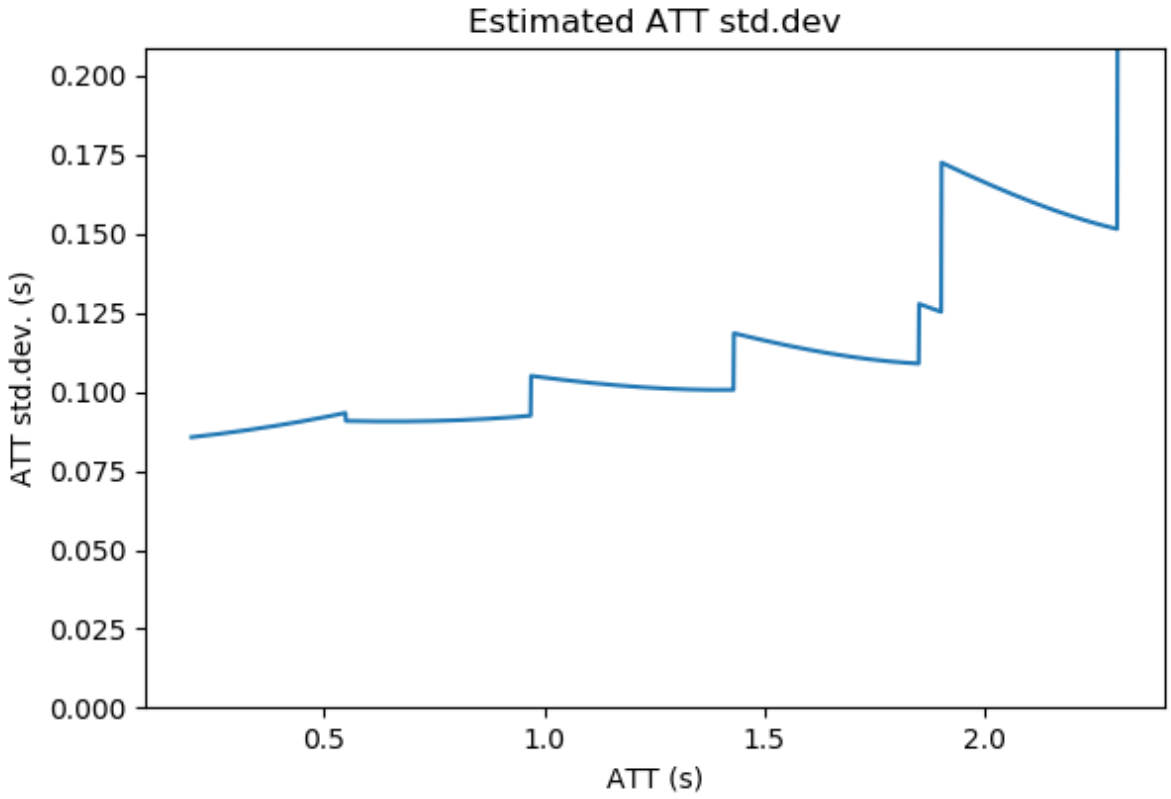
Sensitivity to CBF

This plot shows the sensitivity of the protocol to CBF at a range of arterial transit times. For example, in the plot below we can see that this protocol becomes less sensitive to CBF at longer ATTs. Sensitivity plots often have 'steps' in them which generally correspond to the PLDs.



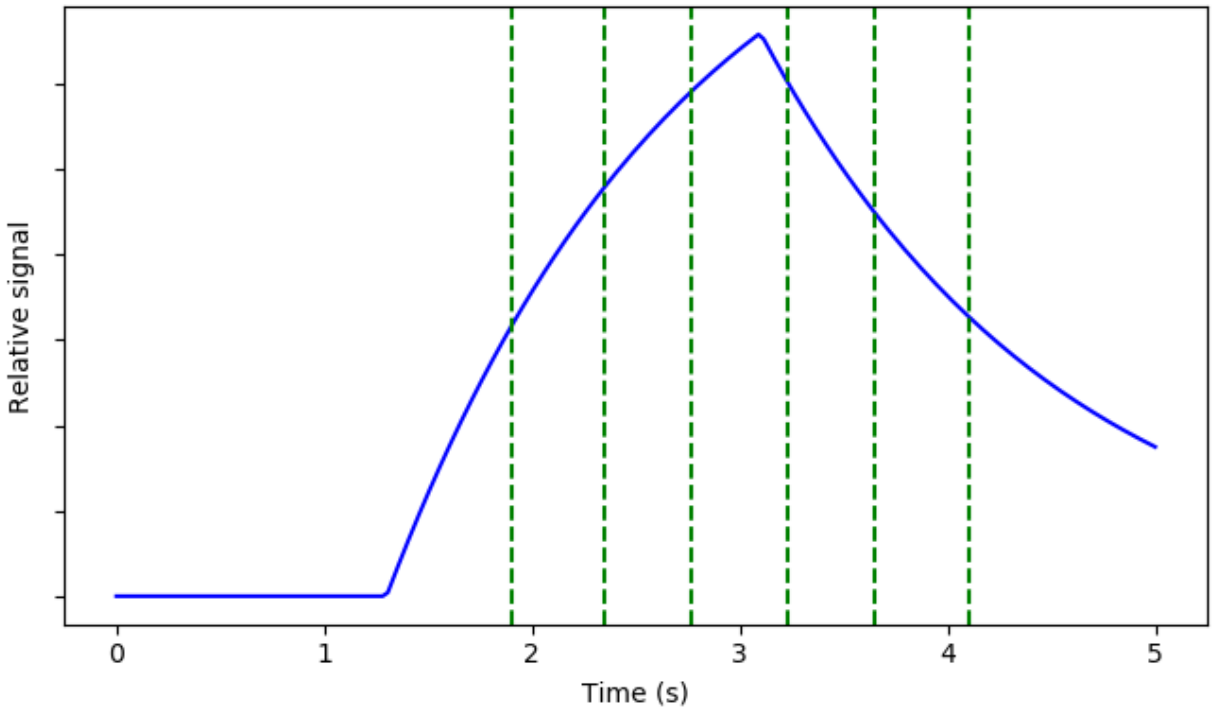
Sensitivity to ATT

This plot similarly shows sensitivity to measurements of ATT.

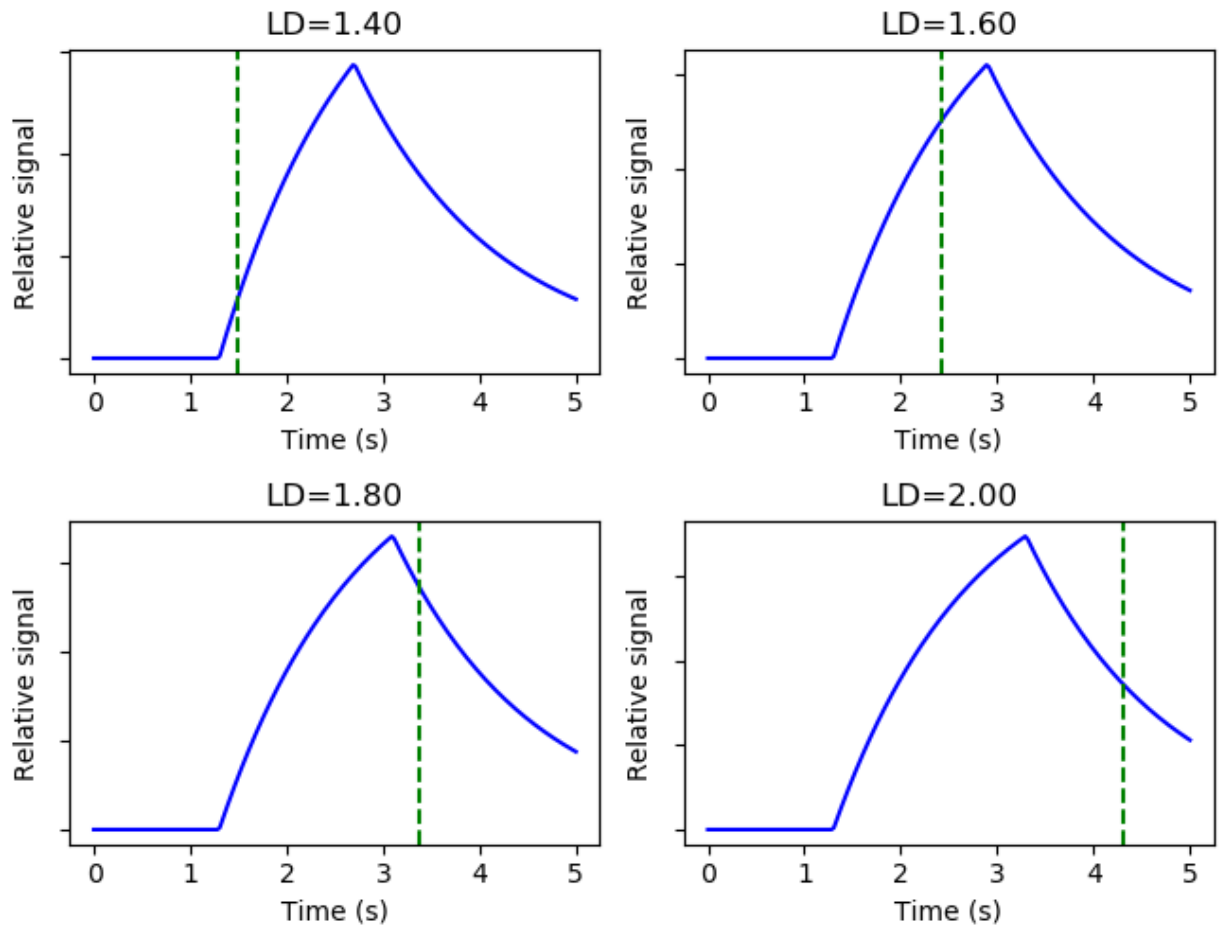


Kinetic curve

This tab shows an illustration of the simple Buxton model of the ASL signal with the protocol PLDs displayed as vertical dashed lines. Note that for time encoded (Hadamard) protocols, the PLDs shown are the effective PLDs for each sub-bolus after decoding. The slider at the bottom can be used to view the kinetic curve at different ATTs.

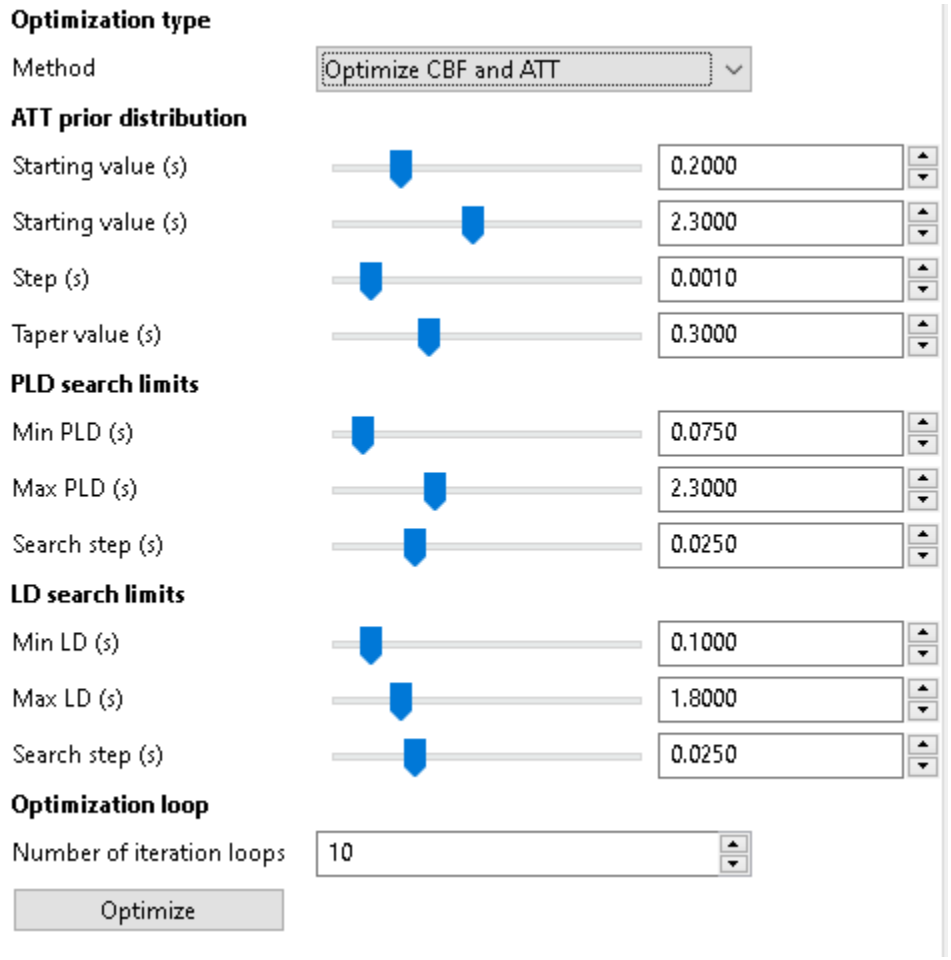


In protocols with multiple labelling durations a different kinetic curve is plotted for each.



2.3.6 Optimization configuration

On the 'Optimization' tab, we can control the optimization parameters. This includes the prior distribution of ATT and the search limits for PLDs. However the most likely setting to change here is whether we want to optimize for CBF, ATT or both.



The optimization works by modifying one parameter (PLD or LD) at a time and sometimes can reach different results depending on what order it chooses to optimize each parameter. To make the results more robust it is common to repeat the optimization cycle multiple times (the order is chosen randomly each time) and choose the outcome with the lowest cost. When optimizing a small number of parameters (e.g. a single LD and PLD) multiple runs may be unnecessary, however when multiple PLDs and LDs are to be optimized it may be helpful to run the optimization a large number of times - of course this may take a while!

2.3.7 Running the optimization

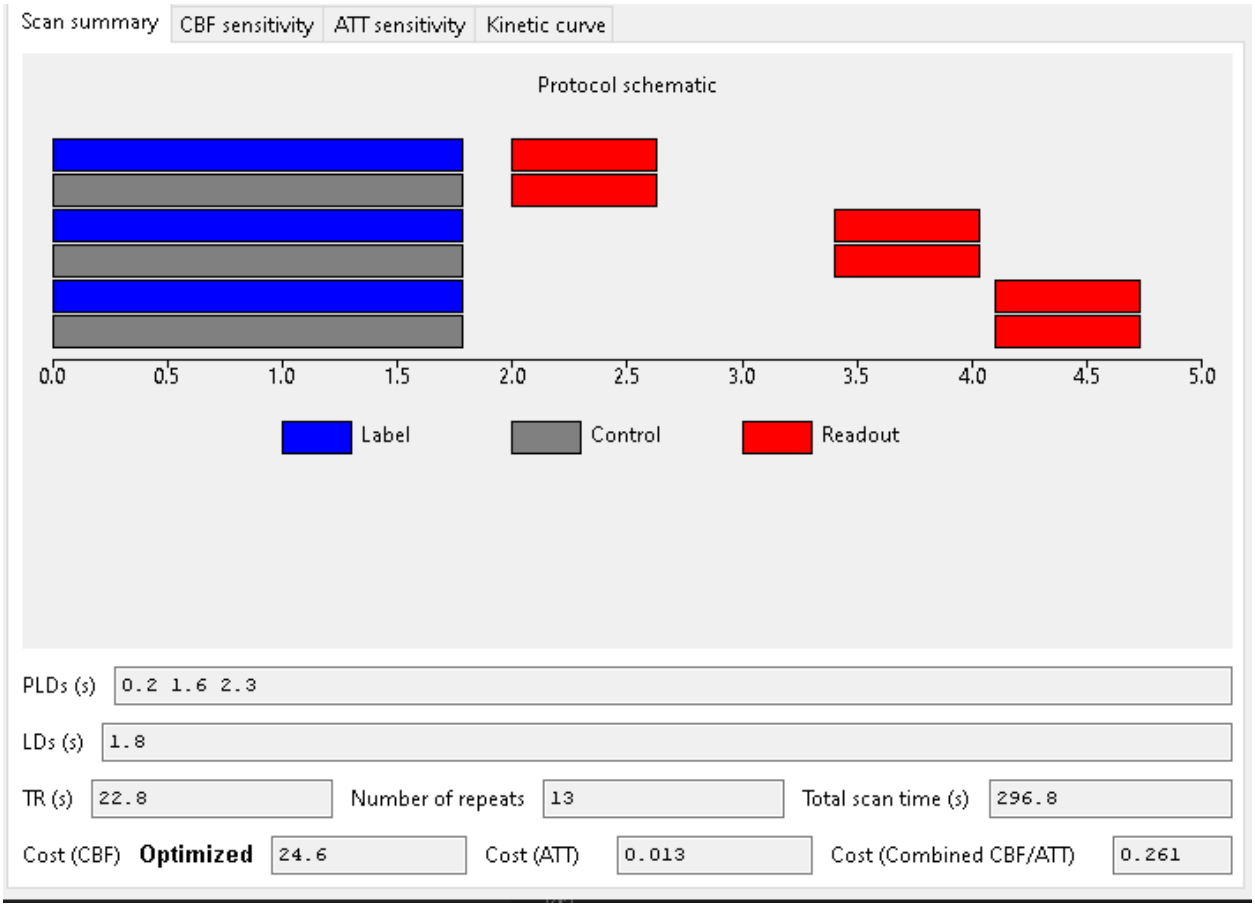
Click the `Run Optimization` button to start - it may take a few seconds for a simple optimization with a single repeat, or many minutes for cases with many parameters or where multiple optimization loops have been specified. Fine PLD/LD search spacings and large ATT prior distributions may also increase the time required for optimization.

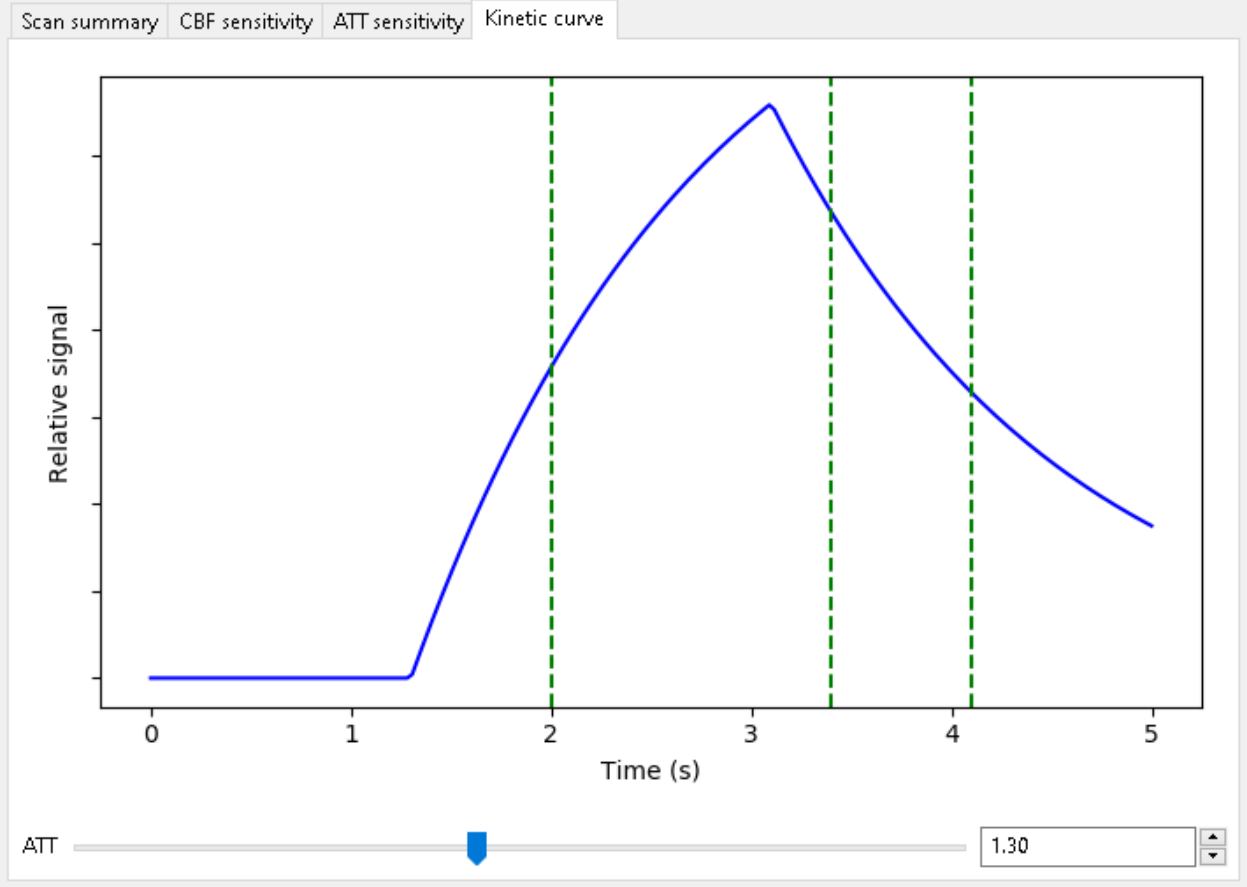
After optimization, the output pane will be updated to show the optimized protocol. The label `Optimized` will be shown beside the cost measure that has been optimized.

Sample output

This is the output of an optimized multi-PLD PCASL experiment with 3 PLDs and a single optimized labelling duration.

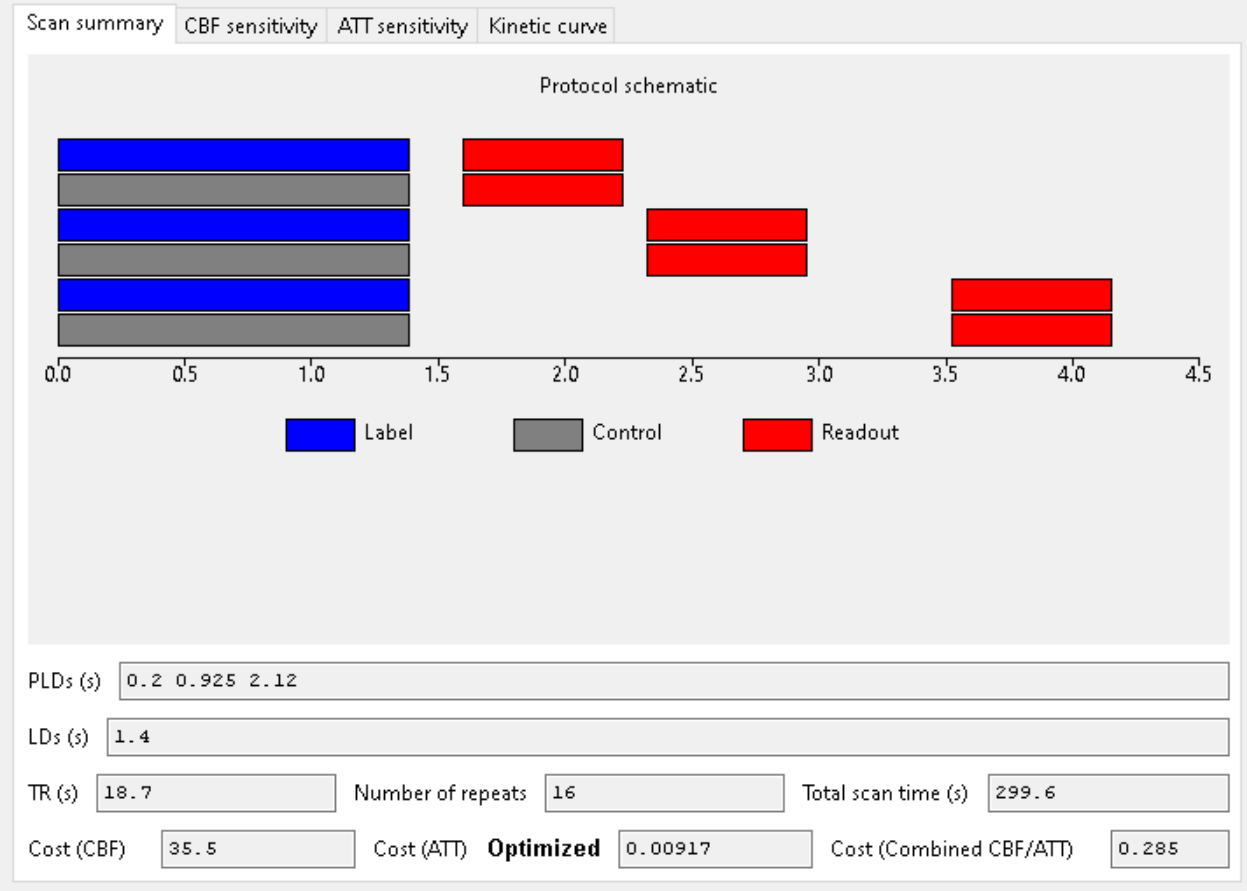
When optimizing for CBF alone, we get the following scan:

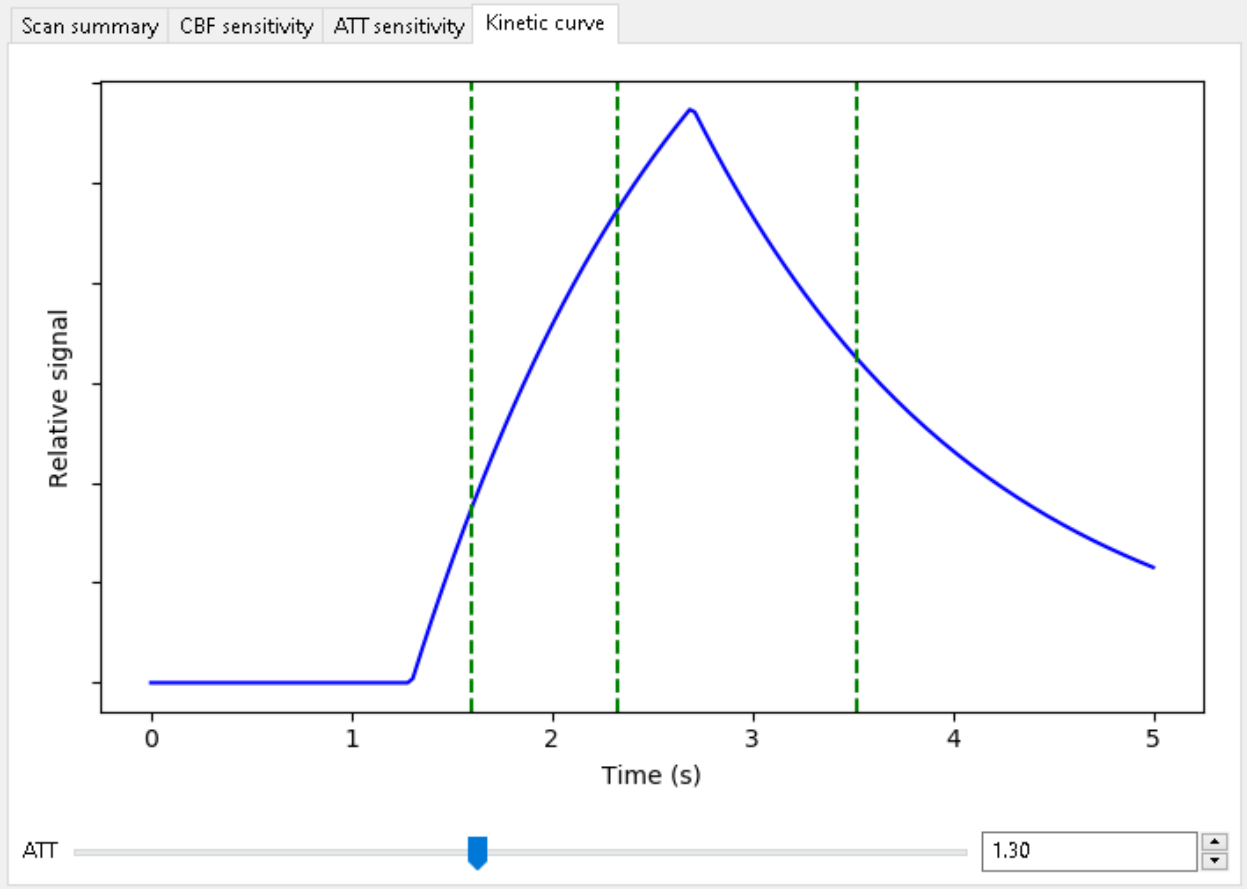

Scan summary
CBF sensitivity
ATT sensitivity
Kinetic curve
Protocol schematic
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
Label
Control
Readout
PLDs (s)
0.2 1.6 2.3
LDs (s)
1.8
TR (s)
22.8
Number of repeats
13
Total scan time (s)
296.8
Cost (CBF)
Optimized
24.6
Cost (ATT)
0.013
Cost (Combined CBF/ATT)
0.261



Note that two of the three PLDs are in the decay part of the curve for most ATTs (this plot was obtained for ATT=1.3). This is good for estimating CBF as the whole bolus has usually arrived by this point.

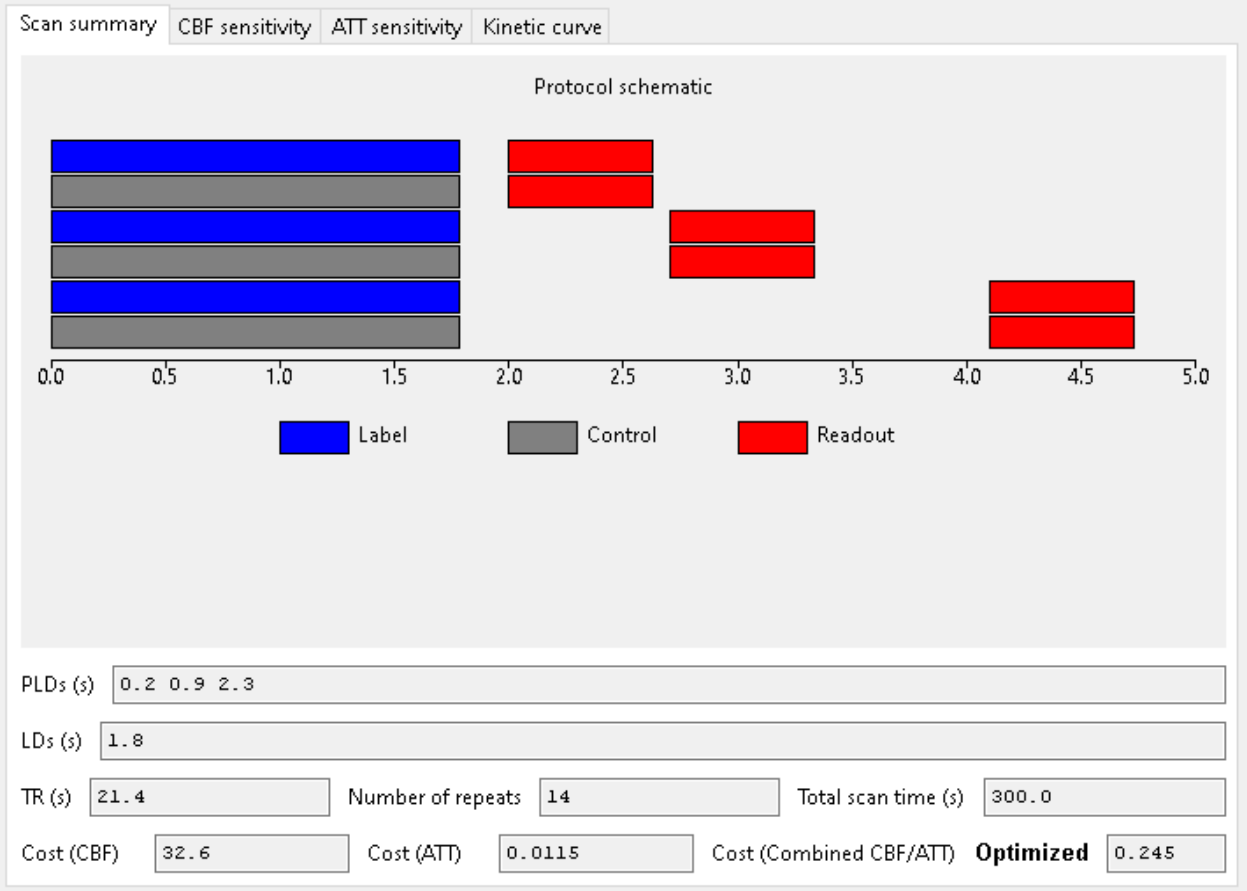
When optimizing for ATT alone, we get the following scan:

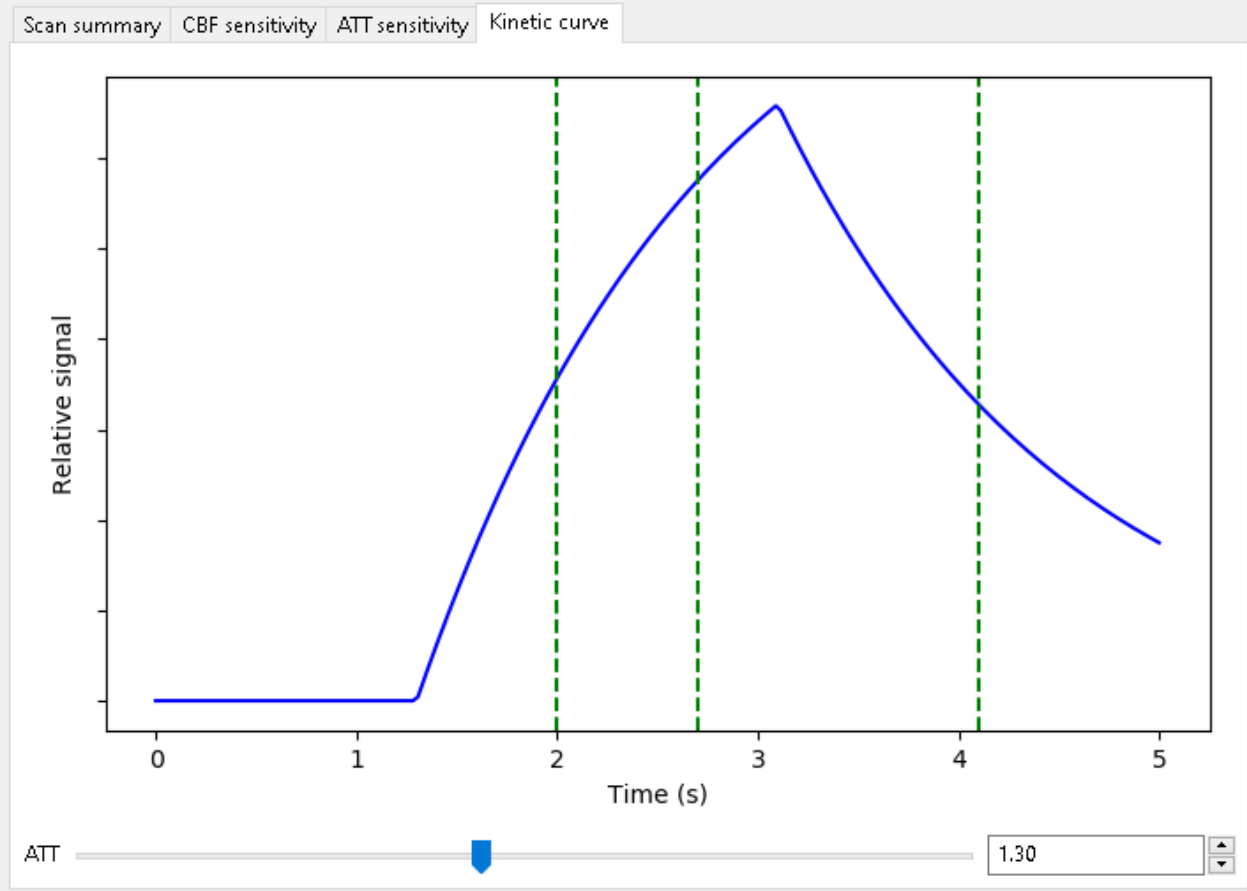

Scan summary
CBF sensitivity
ATT sensitivity
Kinetic curve
Protocol schematic
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
Label
Control
Readout
PLDs (s)
0.2 0.925 2.12
LDs (s)
1.4
TR (s)
18.7
Number of repeats
16
Total scan time (s)
299.6
Cost (CBF)
35.5
Cost (ATT)
Optimized
0.00917
Cost (Combined CBF/ATT)
0.285



As expected, when optimizing for ATT measurements, shorter PLDs are preferred to sample the inflow of the bolus.

When optimizing for CBF and ATT we obtain the following:

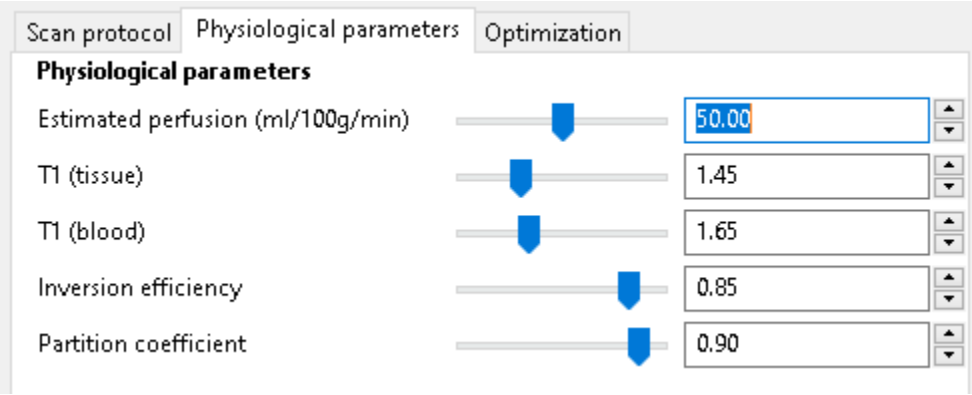

Scan summary
CBF sensitivity
ATT sensitivity
Kinetic curve
Protocol schematic
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
Label
Control
Readout
PLDs (s)
0.2 0.9 2.3
LDs (s)
1.8
TR (s)
21.4
Number of repeats
14
Total scan time (s)
300.0
Cost (CBF)
32.6
Cost (ATT)
0.0115
Cost (Combined CBF/ATT)
Optimized
0.245



The combined optimization is in between the CBF and ATT-only extremes with longer PLDs than the ATT-only optimization but shorter than the CBF-only case.

2.3.8 Physiological parameters

The Physiological parameters tab allows control of the assumed values of parameters used in the kinetic model (and hence affecting the cost function and optimization).



The estimated perfusion is a fixed value used to calculate the effective T1 of labelled blood. Modification of this value should not have a significant effect on the cost or optimization, however.