QUIT Documentation

Tobias Wood

Oct 04, 2018

Contents:

Welcome to the QUantitative Imaging Tools, a collection of C++ programs for analysing quantitative MR images. QUIT is divided into several modules, each of which contains numerous programs for processing images. The modules are:

CHAPTER 1

Categories

1.1 Relaxometry

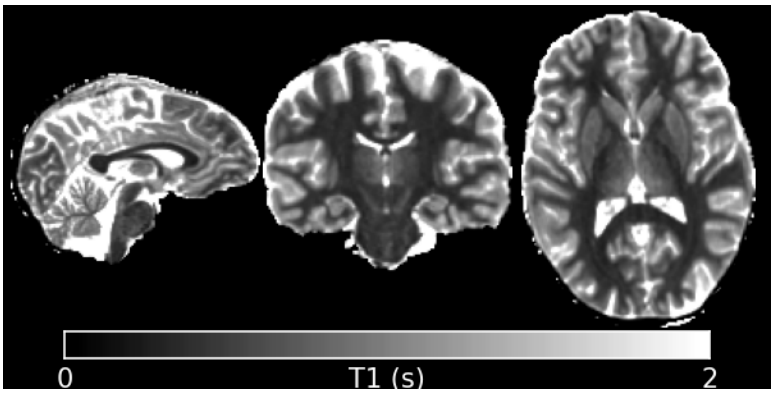
Relaxometry is the measurement of the longitudinal and transverse relaxation times T₁ and T₂. There are a multitude of different techniques for measuring both. The main focus of QUIT has been on the Driven-Equilibrium Single-Pulse Observation of T₁ (DESPOT1) family of techniques, but the classic multi-echo T₂ method and the more recent MP2RAGE T₁ measurement method are also implemented, along with the AFI and DREAM B1 mapping methods.

The following programs are available:

- *qidespot1*
- *qidespot1hifi*
- *qidespot2*
- *qidespot2fm*
- *qimcdespot*
- *qimultiecho*
- *qimp2rage*
- *qiafi*
- *qidream*

1.1.1 qidespot1

This program implements the classic Driven Equilibrium Single-Pulse Observation of T1 (DESPOT1) algorithm, also known as the Variable Flip-Angle (VFA) method in the literature. This is a fast way to measure longitudinal relaxation using a spoiled steady-state sequence, which is know by a different name by every scanner manufacturer just to be helpful. On GE, it's SPoiled Gradient Recalled echo (SPGR), on Siemens it's Fast Low-Angle SHot (FLASH) and on Phillips the sequence is Fast Field Echo (FFE).



Example Command Line

```
qidespot1 input_file.nii.gz --mask=mask_file.nii.gz --B1=b1_file.nii.gz < input.json
```

Example Command Line

```
{
    "SPGR": {
        "TR": 0.01,
        "FA": [3, 18]
    }
}
```

Outputs

- `D1_T1.nii.gz` - The T1 map. Units are the same as those used for TR in the input.
- `D1_PD.nii.gz` - The apparent Proton Density map. No units.

Important Options

- `--B1, -b`

  Specify an effective flip-angle or B1 map. This must be expressed as a fraction, e.g. a value of 1 in a voxel implies the nominal flip-angle was achieved.

- `--algo, -a`

  This specifies which precise algorithm to use. There are 3 choices, classic linear least-squares (l), weighted linear least-squares (w), and non-linear least-squares (n). If you only have 2 flip-angles then LLS is the only meaningful choice. The other 2 choices should produce better (less noisy, more accurate) T1 maps when you have more input flip-angles. WLLS is faster than NLLS for the same number of iterations. However, modern processors are sufficiently powerful that the difference is bearable. Hence NLLS is recommended for the highest possible quality.

References

- Christen et al, the original paper
- Chang et al, Weighted Least Squares Fitting
- Wood, Optimum Flip-Angle Formulas

1.1.2 qidespot1hifi

This is an extension of DESPOT1 to fit a map simultaneously using an MP-RAGE / IR-SPGR type sequence. Although DESPOT1-HIFI can produce a rough estimate of B1, it often fails to produce reasonable values in the ventricles, and

the fact that the MP-RAGE image is often acquired at lower resolution than the SPGR/FLASH data can also cause problems. Hence you should either smooth the B1 map produced as output, or fit it with a polynomial (*Utilities*), then recalculate T1 using the *qidespot1* program. Note that if your MP-RAGE image is not acquired at the same resolution as your SPGR data, it must be resampled to the same spacing before processing (and it should also be registered to your SPGR data).

Example Command Line

```
qidespot1hifi spgr_file.nii.gz irspgr_file.nii.gz --mask=mask_file.nii.gz < input.json
```

Example Command Line

```
{
    "SPGR": {
        "TR": 0.01,
        "FA": [3, 18]
    },
    "MPRAGE": {
        "FA": 5,
        "TR": 0.01,
        "TI": 0.45,
        "TD": 0,
        "eta": 1,
        "ETL": 64,
        "k0": 0
    }
}
```

For the MPRAGE sequence, the TR is the spacing between readouts/echoes, not the overall segment TR. TI is the Inversion Time, and TD is the Delay Time after the echo-train (often 0). Eta is the Inversion Efficiency, which should be set to 1. ETL is the Echo-Train Length - usually the number of phase encode steps in one segment. k0 defines the position in the echo-train that the center line of k-space is acquired. This is 0 for centric acquisition and ETL/2 for linear.

Outputs

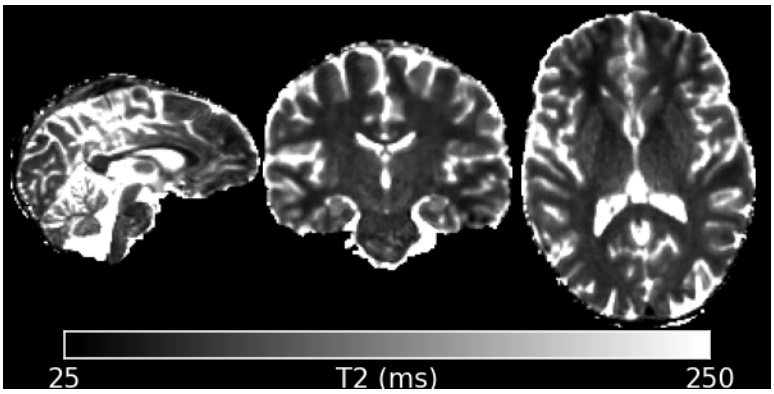
- `HIFI_T1.nii.gz` - The T1 map. Units are the same as those used for TR in the input.
- `HIFI_PD.nii.gz` - The apparent Proton Density map. No units.
- `HIFI_B1.nii.gz` - The relative flip-angle map.

References

- Original HIFI Paper

1.1.3 qidespot2

DESPOT2 uses SSFP data and a separate T1 map to calculate T2, using the same maths as DESPOT1. It does not account for the banding artefacts present in SSFP data at field-strengths of 3T and above. See *qidespot2fm* for a method that does account for them, or if you have at least 4 phase-increments and complex data then see *SSFP* for a way to remove them before using this program.



Example Command Line

```
qidespot2 t1_map.nii.gz input_file.nii.gz --mask=mask_file.nii.gz --B1=b1_file.nii.gz
↪< input.json
```

Example Command Line

```
{
    "SSFP": {
        "TR": 0.005,
        "PhaseInc": [180],
        "FA": [12, 60]
    }
}
```

Both `PhaseInc` and `FA` are measured in degrees. If the ellipse option is specified, then the sequence type must be `SSFPGS`, which does not require a `PhaseInc`. The units of `TR` must match the input T1 map.

Outputs

- `D2_T2.nii.gz` - The T2 map. Units are the same as those used for TR in the input.
- `D2_PD.nii.gz` - The apparent Proton Density map. No units. Will be corrected for T2 decay at the echo time.

Important Options

- `--B1, -b`

  Specify an effective flip-angle or B1 map. This must be expressed as a fraction, e.g. a value of 1 in a voxel implies the nominal flip-angle was achieved.

- `--algo, -a`

  This specifies which precise algorithm to use. There are 3 choices, classic linear least-squares (l), weighted linear least-squares (w), and non-linear least-squares (n). If you only have 2 flip-angles then LLS is the only meaningful choice. The other 2 choices should produce better (less noisy, more accurate) T1 maps when you have more input flip-angles. WLLS is faster than NLLS for the same number of iterations. However, modern processors are sufficiently powerful that the difference is bearable. Hence NLLS is recommended for the highest possible quality.

- `--ellipse, -e`

  This specifies that the input data is the SSFP Ellipse Geometric Solution, i.e. that multiple phase-increment data has already been combined to produce band free images.

References

- Original DESPOT2 Paper

1.1.4 qidespot2fm

DESPOT2-FM uses SSFP data with mulitple phase-increments (also called phase-cycles or phase-cycling patterns) to produce T2 maps without banding artefacts.

Example Command Line

```
qidespot2fm t1_map.nii.gz input_file.nii.gz --mask=mask_file.nii.gz --B1=b1_file.nii.
↪gz < input.json
```

The input file should contain all SSFP images concatenated together as a 4D file. The preferred ordering is flip-angle, then phase-increment (i.e. all flip-angles at one phase-increment, then all flip-angles at the next phase-increment).

Example Command Line

```
{
    "SSFP": {
        "TR": 0.005,
        "PhaseInc": [180, 180, 0, 0],
        "FA": [12, 60, 12, 60]
    }
}
```

Both `PhaseInc` and `FA` are measured in degrees. The length of `PhaseInc` and `FA` must match.

Outputs

- `FM_T2.nii.gz` - The T2 map. Units are the same as those used for TR in the input.
- `FM_PD.nii.gz` - The apparent Proton Density map. No units. Will be corrected for T2 decay at the echo time.

Important Options

- `--B1, -b`

  Specify an effective flip-angle or B1 map. This must be expressed as a fraction, e.g. a value of 1 in a voxel implies the nominal flip-angle was achieved.

- `--asym, -A`

  With the commonly used phase-increments of 180 and 0 degrees, due to symmetries in the SSFP magnitude profile, it is not possible to distinguish positive and negative off-resonance. Hence by default `qidespot2fm` only tries to fit for positive off-resonance frequences. If you acquire most phase-increments, e.g. 180, 0, 90 & 270, then add this switch to fit both negative and positive off-resonance frequencies.

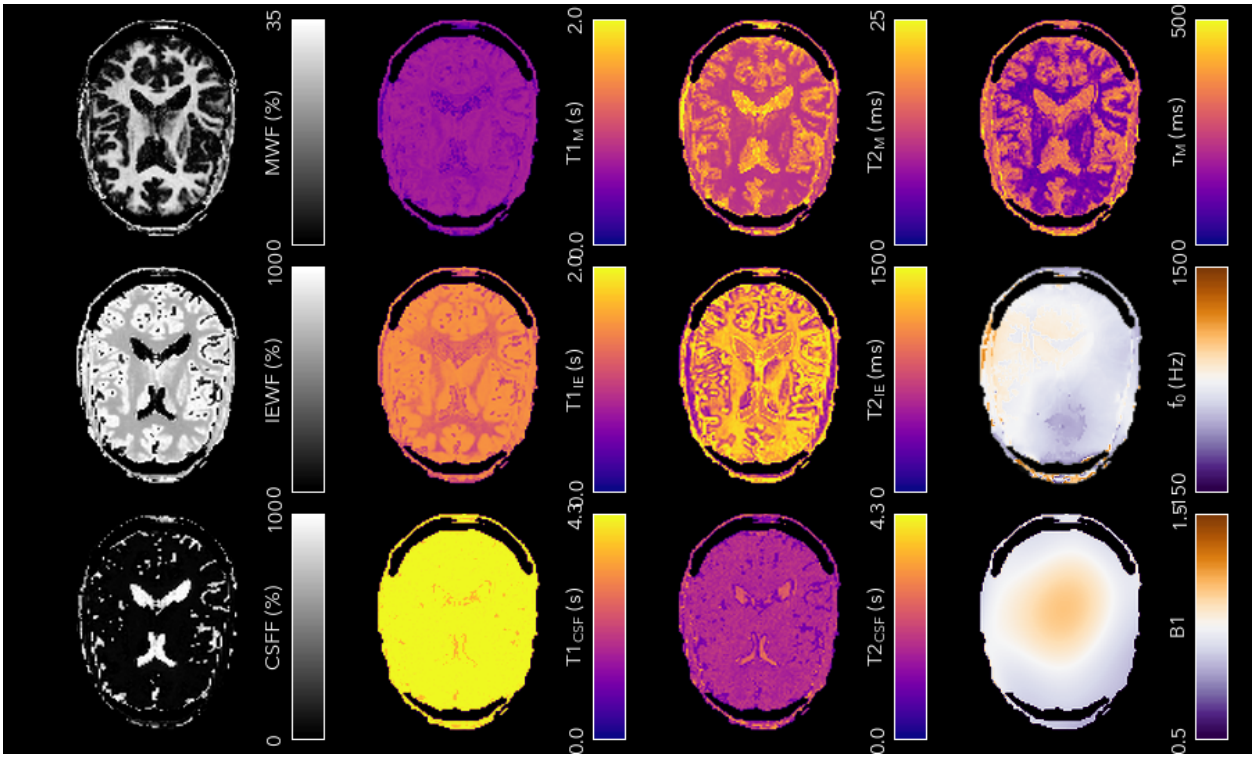
References

- Orignal FM Paper

1.1.5 qimcdespot

Multi-component DESPOT aims to separate SPGR and SSFP signals into multiple discrete pools with different T₁ and T₂. In the brain, the pool with shorter values is attributed to myelin water, while pools with longer values can be either intra/extra-cellular water or CSF.

It is recommended to have an off-resonance map to stabilise the fitting. This can be generated by using *qidespot1* and then *qidespot2fm* above. A B1 map is also essential for good results.



Example Command Line

```
qimcdespot spgr_file.nii.gz ssfp_file.nii.gz --mask=mask_file.nii.gz --B1=b1_file.nii.
↪gz --f0=f0_file.nii.gz --scale < input.json
```

The SSFP input file should contain all SSFP images concatenated together as a 4D file (see *qidespot2fm* above).

Example Command Line

```
{
    "Sequences": [
        {
            "SPGR": {
                "TR": 0.01,
                "FA": [3,4,5,7,9,12,15,18]
            }
        },
        {
            "SSFP": {
                "TR": 0.05,
                "FA": [12,16,20,24,30,40,50,60,12,16,20,24,30,40,50,60],
                "PhaseInc": [180,180,180,180,180,180,180,180,0,0,0,0,0,0,0,0]
            }
        }
    ]
}
```

The order that the sequences are listed must match the order the input files are specified on the command-line.

Outputs

Note - the output prefix will change depending on the model selected (see below). The outputs listed here are for the 3 component model.

- `3C_T1_m.nii.gz` - T1 of myelin water
- `3C_T2_m.nii.gz` - T2 of myelin water
- `3C_T1_ie.nii.gz` - T1 of intra/extra-cellular water
- `3C_T2_ie.nii.gz` - T2 of intra/extra-cellular water
- `3C_T1_csf.nii.gz` - T1 of CSF
- `3C_T2_csf.nii.gz` - T2 of CSF
- `3C_tau_m.nii.gz` - The residence time of myelin water (reciprocal of forward exchange rate)
- `3C_f_m.nii.gz` - The Myelin Water Fraction (MWF)
- `3C_f_csf.nii.gz` - The CSF Fraction
- `3C_f0.nii.gz` - The off-resonance frequency. If this was specified on the command line, it will be a copy of that file
- `3C_B1.nii.gz` - The relative flip-angle map. If this was specified on the command line, it will be a copy of that file

The intra/extra-cellular water fraction is not output, as it is not a free parameter (only 2 of the 3 pool fractions are required for the calculations). It is easy to calculate this post-hoc by subtracting the MWF and CSFF from 1.

Important Options

- `--algo, -a`
  - S - Stochastic Region Contraction
  - G - Gaussian Region Contraction

  Gaussian is recommended.
- `--tesla, -t`

  Specify the field-strength so sensible fitting ranges can be used. Currently only ranges for (3) and (7)T are defined. If you wish to specify your own ranges, set this option as (u) and then the ranges will be read from your input file.
- **`--model, -m`**
  - 1 - 1 component model (no fractions, just a single T1/T2)
  - 2 - 2 component model. Myelin and intra/extra-cellular water
  - 2nex - 2 component model without exchange
  - 3 - 3 component model. Myelin water, IE water & CSF
  - 3nex - 3 component model without exchange
  - 3f0 - 3 component model, allow an additional off-resonance offset between myelin and IE water pools

References

- Original mcDESPOT paper
- 3 component model
- Stochastic/Gaussian Region Contraction

1.1.6 qimp2rage

MP2RAGE adds a second inversion time to the standard T1w MPRAGE sequence. Combining the (complex) images with the expression $S_1 S_2^* / (|S_1^2 + S_2^2|)$ produces a real-valued image that is corrected for receive coil (B1-) inhomogeneity. In addition, if the two inversion times are carefully selected, a one-to-one mapping exists between the values in that image and T1, which is also robust to transmit (B1+) inhomogeneity. Finally, as the two images are implicitly registered, this method has several advantages over DESPOT1.

Example Command Line

```
qimp2rage input_file.nii.gz --mask=mask_file.nii.gz < input.json
```

The input file must be complex-valued.

Example Command Line

```
{
    "MP2RAGE" : {
        "TR" : 0.006,
        "SegTR" : 5,
        "TI" : [0.9, 2],
        "ETL": 128,
        "FA": [6, 8]
    }
}
```

`TR` is the readout or echo-train repetition time, while `SegTR` is the segment or overall TR. `ETL` is the echo-train length or number of readouts in one segment.

Outputs

- `{input}_contrast.nii.gz` - The MP2 contrast image. The range of this image is -0.5 to 0.5.
- `{input}_T1.nii.gz` - The T1 map. Units are the same as *TR* and *SegTR*.

Important Options

- `--beta, -b`

  Regularisation factor for robust contrast calculation (see references). It is recommended to experiment with this parameter to manually find an optimum value, which should then be kept constant for an entire dataset.

References

- Original MP2RAGE paper
- Robust contrast

1.1.7 qimultiecho

Classic monoexponential decay fitting. Can be used to fit either T2 or T2*.

Example Command Line

```
qimultiecho input_file.nii.gz --algo=a < input.json
```

Example Command Line

For regularly spaced echoes:

```
{
    "MultiEcho" : {
        "TR" : 2.5,
        "TE1" : 0.005,
        "ESP" : 0.005,
        "ETL" : 16
    }
}
```

`TE1` is the first echo-time, `ESP` is the subsequent echo-spacing, `ETL` is the echo-train length.

For irregularly spaced echoes:

```
{
    "MultiEchoFlex" : {
        "TR" : 2.5,
        "TE" : [0.005, 0.01, 0.03, 0.05]
    }
}
```

Note: The current implementation of the ARLO method will only work with regularly spaced echoes

Outputs

- `ME_T2.nii.gz` - The T2 map. Units are the same as *TE1* and *ESP*.
- `ME_PD.nii.gz` - The apparent proton-density map (intercept of the decay curve at TE=0)

Important Options

- `--algo, -a`
  - l - Standard log-linear fitting
  - a - ARLO (see reference below)
  - n - Non-linear fitting

References

- ARLO

1.1.8 qiafi

Calculates a relative flip-angle (B1) map using the Actual Flip-angle Imaging method.

Example Command Line

```
qiafi input_file.nii.gz
```

Does not read any input from `stdin`. The input file should contain two volumes, corresponding to TR1 and TR2.

Outputs

- `AFI_B1.nii.gz` - The relative flip-angle map.

Important Options

- `--flip, -f`

  The nominal flip-angle that should have been achieved, default 55 degrees.

- `--ratio, -r`

  The ratio of TR2 to TR1, default 5.

- `--save, -s`

  Output AFI_angle.nii.gz, the actual achieved angle in each voxel.

References

- Original AFI Paper
- Optimal parameters
- Steady-State Conditions

1.1.9 qidream

Calculates a relative flip-angle (B1) map using the DREAM method.

Example Command Line

```
qidream input_file.nii.gz
```

Does not read any input from *stdin*. The input file should contain two volumes, the FID and stimulated echo (STE).

Outputs

- `DREAM_B1.nii.gz` - The relative flip-angle map.
- `DREAM_angle.nii.gz` - The actual achieved angle in each voxel.

Important Options

- `--alpha, -a`

  The nominal flip-angle that should have been achieved, default 55 degrees.

- `--order, -O`
  - f - FID is the first volume, STE is second
  - s - STE is the first volume, FID is second
  - v - VST (Virtual Stimulated Echo) is the first volume, FID is second

References

- Original DREAM Paper
- Virtual Stimulated Echo

1.2 Perfusion

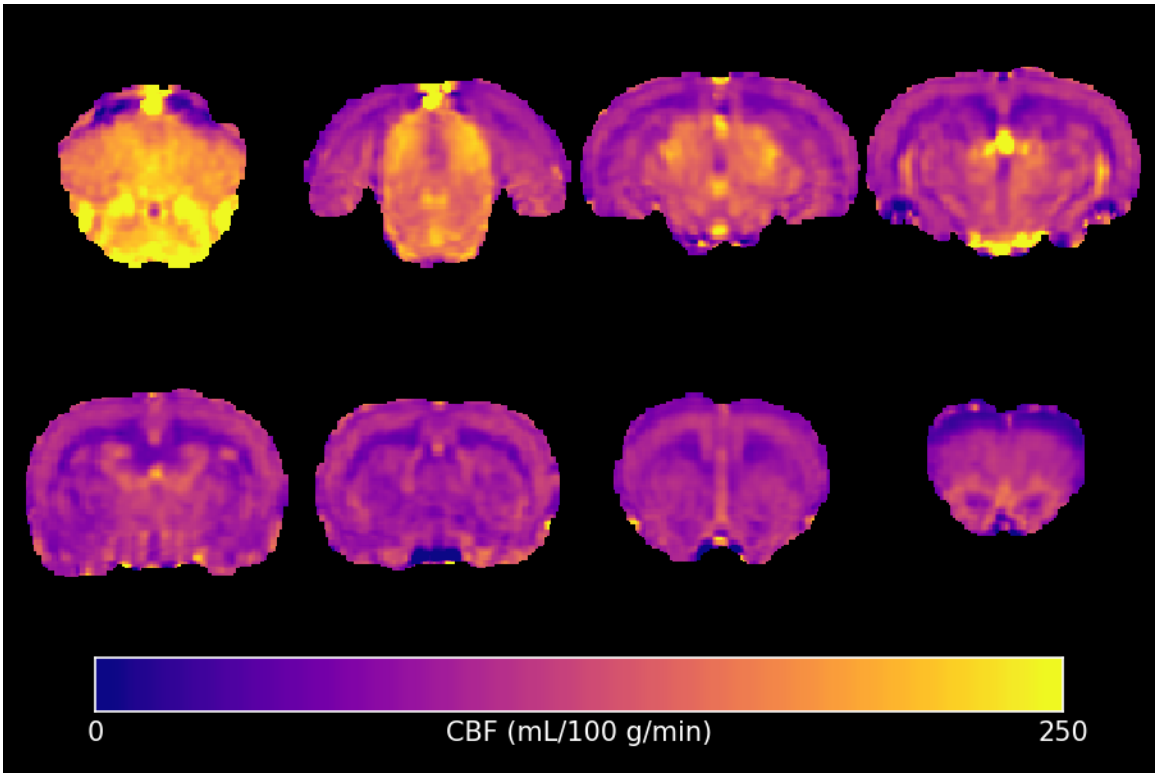
Perfusion is the study of blood flow within the brain. This module contains tools to calculate Cerebral Blood Flow (CBF) and the Oxygen Extraction Fraction (OEF).

The following programs are available:

- *qi_asl*
- *qi_ase_oef*

1.2.1 qi_asl

This program implements the standard equation to calculate CBF from either Continuous or pseudo-Continuous Arterial Spin Labelling data (CASL or pCASL). For the exact equation used, see the first reference below.



Example Command Line

```
qi_asl asl_file.nii.gz --blood=2.429 --alpha=0.9 --average --slicetime --pd=reference_
↪file.nii.gz <input.json
```

The input file must contain pairs of label & control volumes. Currently the order of these is hard-coded to label, then control. The file can contain multiple pairs if you are studying timeseries data. The arguments are discussed further below. It is highly recommended to provide either a separated Proton Density reference image or a tissue T1 map.

Example Command Line

```
{
    "CASL" : {
        "TR" : 4.0,
        "label_time" : 3.0,
        "post_label_delay" : [ 0.3, 0.6, 0.9 ]
    }
}
```

The units for all these values must be consistent, seconds are preferred. If single-slice or 3D data was acquired, then `post_label_delay` should contain a single value. For multi-slice data, specify the `--slicetime` option and then provide the effective post-labelling delay for each slice.

Outputs

- `input_CBF` - The CBF value, given in mL/(100 g)/min

Important Options

- `--blood, -b`

  The T1 value of blood at the field strength used. The default value is 1.65 seconds, corresponding to 3T. For 1.5T the value should be 1.35 seconds and at 9.4T it should 2.429 seconds. See reference 2.

- `--pd, -p`

  Provide a separate image to estimate of the Proton Density of tissue. If this is not provided, the label images are used instead.

- `--tissue, -t`

  Provide a T1 map to correct the Proton Density estimate. If a separate PD reference is not given, then an alternative is to correct the label images for incomplete T1 relaxation.

- `--alpha, -a`

  The labelling efficiency of the sequence.

- `--lambda, -l`

  The blood-brain partition co-efficient, default 0.9 mL/g.

References

- ISMRM Consortium Recommendations
- High-field blood T1 times

1.2.2 qi_ase_oef

Estimates the Oxygen Extraction Fraction (OEF) from Asymmetric Spin-Echo (ASE) data. If the signal evolution each side of a spin-echo in the presence of blood vessels is observed carefully, it does not display simple monoexponential T2* decay close to the echo, but is instead quadratically exponential. By measuring the T2* decay in the linear regime using an ASE sequence, it is possible to extrapolate back to the echo and obtain an estimate of what the signal would be if no blood was presence. The difference between this and the observed signal can be attributed to the Deoxygenated Blood Volume (DBV), and from there the OEF can be calculated.

Example Command Line

```
qi_ase_oef ase_file.nii.gz --B0=9.4 $DB --fmap=fieldmap.nii.gz <input.json
```

Example Command Line

```
{
    "MultiEcho" : {
        "TR" : 2.0,
        "TE1" : 0,
        "ESP" : 0.002,
        "ETL" : 10
    }
}
```

`TR` must be provided but is not used in the calculation. Echo-times below the critical time (Tc) will be excluded from the R2' calculation.

Outputs

- `input_R2prime.nii.gz` The R2' map. Units are the same as those used for `TR`, `TE1` and `ESP`.
- `input_DBV.nii.gz` The Deoxygenated Blood Volume, in percent.
- `input_OEF.nii.gz` The Oxygen Extraction Fraction, in percent.

- `input_dHb.nii.gz` The Deoxyhaemoglobin concentration.

Important Options

- `--B0, -b`

  Field-strength the data was acquired at. This is used to calculate Tc and appears elsewhere in several equations.

- `--fmap, -f`

  Provide a field-map (in Hertz). This will be used to provide first-order correction of Macroscopic Field Gradients (MFGs). If this option is specified, the derivative of the field-map in all 3 directions will also be saved.

- `--slice,-s`

  If the data was acquired with a slice-gap, use this option to specify the actual slice-thickness for the MFG calculation.

References

- Blockley

1.3 Magnetization Transfer

MR voxels often contain complex microstructure with multiple different components or pools, each with unique relaxation properties. It is possible for magnetization to be transferred between these pools via several mechanisms, such as exchange of individual protons or entire molecules, or simple dipolar coupling from molecules that are in close proximity. These mechanisms can be studied in the related fields of Magnetization Transfer (MT) and Chemical Exchange Saturation Transfer (CEST). QUIT currently contains some basic CEST analysis tools and one for calculating simple dipolar/inhomogeneous MT ratios.

In addition a tool is provided for calculating qMT parameters from SSFP data. This is in the *SSFP* module.

- *qi_lineshape*
- *qi_qmt*
- *qi_zspec_interp*
- *qi_lorentzian*
- *qi_dipolar_mtr.sh*

1.3.1 qi_lineshape

A utility to sample lineshapes and write them out to files, which can then be read by *qi_qmt* and interpolated values used instead of calculating the lineshape during fitting. This is used principally to speed up fitting the Super-Lorentzian lineshape, which takes a long time to calculate but is smoothly valued, so can be accurately approximated using interpolation. The lineshape definitions are the same as those in qMRLab.

Example Command Line

```
qi_lineshape --lineshape=SuperLorentzian --frq_start=500 --frq_space=500 --frq_
↪count=150 > superlorentz.json
```

No input file required. Output is to `stdout` in JSON format, so should be redirected into a file for further use.

Important Options

- `--lineshape, -l`

  Choose from a Gaussian, a Lorentzian or the Super-Lorentzian

- `--T2b, -t`

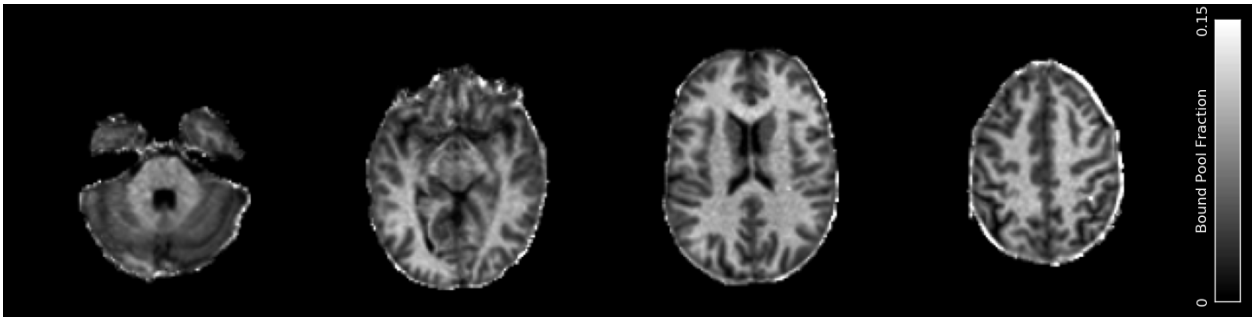
  Specify the nominal T2 of the lineshape. During fitting in *qi_qmt* scaling will be used to find the actual value. Should be specified in seconds.

- `--frq_count`, `--frq_start`, `--frq_space`

  These Control the position and number of samples to take on the lineshape. `frq_start` and `frq_space` should be in Hertz.

1.3.2 qi_qmt

Calculates Quantitative Magnetization Transfer parameters using the Ramani model from steady-state gradient-echo data acquired with multiple off-resonance saturation pulses. A massive thank you to Erika Raven, Mara Cercignani for helping get this working.



Example Command Line

```
qi_qmt --verbose MTSatData.nii.gz T1.nii.gz --mask brain_mask.nii.gz --
↪lineshape=superlorentz.json --B1=B1_map.nii.gz --f0=B0_map.nii.gz < input.json
```

Note that a T1 map is a required input to stabilise the fitting.

Example Command Line

```
{
    "MTSat" : {
        "TR": 0.055,
        "FA": 5,
        "sat_f0": [56360, 47180, 12060, 1000, 1000, 2750, 2770, 2790, 2890, 1000,␣
↪1000],
        "sat_angle": [332, 628, 628, 332, 333, 628, 628, 628, 628, 628, 628],
        "pulse": { "name": "Gauss", "Trf": 0.015, "p1": 0.416, "p2": 0.295 }
    }
}
```

`Trf` is the pulse-width (RF Time). `p1` and `p2` are the ratio of the integral of B_1 and B_1^2 (the integrals of the pulse amplitude and the square of the pulse amplitude) to the maximum amplitude of the pulse. Both `Trf` and `TR` should be in seconds. `sat_f0` is in Hertz.

Outputs

- `QMT_f_b.nii.gz` - The bound pool fraction
- `QMT_k_bf.nii.gz` - The forward exchange rate from bound to free pool

- `QMT_T1_f.nii.gz` - T1 of the free pool
- `QMT_T2_f.nii.gz` - T2 of the free pool
- `QMT_T2_b.nii.gz` - T2 of the bound pool
- `QMT_PD.nii.gz` - The apparent Proton Density / size of the free pool

Note that `T1_b`, the longitudinal relaxation rate of the bound pool, is fixed to 1 second in this model and so is not written out.

References

- Ramani et al

1.3.3 qi_zspec_interp

Interpolates a Z-spectrum to arbitrary precision. Can output asymmetry values instead of a Z-spectrum.

Example Command Line

```
qi_zspec_interp zspectrum.nii.gz --f0=LTZ_f0.nii.gz < input.json
```

The off-resonance map units must match the input frequencies (e.g. either PPM or Hertz)

Example Command Line

```
{
    "input_freqs" : [ -5, -2.5, 0, 2.5, 5],
    "output_freqs" : [ -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5]
}
```

`input_freqs` are the offset frequencies the Z-spectrum was acquired at. `output_freqs` are the frequencies you want the asymmetry calculated at.

Outputs

- `{input}_interp.nii.gz` The interpolated Z-spectrum.

Important Options

- `--f0, -f`

  Specify an off-resonance map. Units must be the same as the input & asymmetry frequencies.

- `-O, --order`

  The order of Spline interpolation used. Default is 3 (cubic).

- `-a, --asym`

  Output asymmetry (:math‘Z(+f) - Z(-f)‘) values.

1.3.4 qi_lorentzian

Fits a single Lorentzian to a Z-spectrum for B0 correction. Currently hard-coded to only fit the spectrum between +/-2ppm to avoid background MT contamination.

Example Command Line

```
qi_lorentzian zspectrum.nii.gz < input.json
```

The Z-spectrum must be a 4D file with each volume acquired at a different offset frequency.

Example Command Line

```
{
    "freq" : [ -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5]
}
```

These are the offset frequencies for each volume in the Z-spectrum input.

Outputs

- `LTZ_f0.nii.gz` - The center frequency of the fitted Lorentzian.
- `LTZ_w.nii.gz` - The width of the fitted Lorentzian.
- `LTZ_sat.nii.gz` - The saturation ratio of the fitted Lorentzian.
- `LTZ_PD.nii.gz` - The apparent Proton Density of the fitted Lorentzian.

1.3.5 qi_dipolar_mtr.sh

Calculates dipolar/inhomogeneous Magnetization Transfer Ratios (MTRs). Dipolar/inhomogeneous MT is a new (see note) contrast mechanism that is present in highly structured materials such as myelin and tendon. By applying off-resonance saturation at both positive and negative frequencies (instead of only one side as in classic MTR) it is possible to decouple the dipolar pool and hence produce an enhanced Magnetization Transfer (eMT) effect. The different between eMT and normal MT is the dipolar/inhomogeneous MT and is potentially highly specific to myelin within the brain.

Although the majority of the existing literature refers to this effect as inhomogeneous MT, this name was chosen before the physical phenomena underlying the effect was well understood. Current theory does not rely on inhomogeneous effects at all, so the name is a misnomer.

Example Command Line

```
qi_dipolar_mtr dipolar_mt_volumes.nii.gz
```

The input must consist of 5 volumes: Dipolar +/-, Dipolar -/+, Unsaturated, MT+, MT-. This scheme is not flexible and will be improved in a future version.

Outputs

- `DMT_mtr.nii.gz` - The classic MTR, expressed as a percentage
- `DMT_emtr.nii.gz` - The enhanced MTR, expressed as a percentage
- `DMT_dmtr.nii.gz` - The dipolar MTR, expressed as a percentage. This is the difference between eMTR and MTR.
- `DMT_mta.nii.gz` - The first-order MT-asymmetry (MT- subtracted from MT+, relative to unsaturated, in percent).

References

1. Original full paper
2. Dipolar versus inhomogeneous naming

1.4 SSFP

The Steady-State Free-Precession (SSFP), or more precisely balanced-SSFP (bSSFP), sequence is one of the oldest NMR sequences and can be used to give high SNR MR images with mixed T1/T2 contrast in very short scan time. However, it suffers from banding artefacts in areas of off-resonance which limit its clinical applicability. This module contains a tool for removing those banding artefacts, and then further tools for quantitative mapping using the ellipse signal model.

- *qi_ssfp_bands*
- *qi_ssfp_ellipse*
- *qi_ssfp_planet*
- *qi_ssfp_emt*

1.4.1 qi_ssfp_bands

There are several different methods for removing SSFP bands in the literature. Most of them rely on acquiring multiple SSFP images with different phase-increments (also called phase-cycling or phase-cycling patterns). Changing the phase-increments moves the bands to a different location, after which the images can be combined to reduce the banding. The different approaches are discussed further below, but the recommended method is the Geometric Solution which requires complex data.

Example Command Line

```
qissfpbands ssfp.nii.gz --method=G --2pass --magnitude
```

The SSFP file must be complex-valued to use the Geometric Solution or Complex Average methods. For the other methods magnitude data is sufficient. Phase-increments should be in opposing pairs, e.g. 180 & 0 degrees, 90 & 270 degrees. These should either be ordered in two blocks, e.g. 180, 90, 0, 270, or alternating, e.g. 180, 0, 90, 270.

Outputs

The output filename is the input filename with a suffix that will depend on the method selected (see below).

Important Options

- `--method`

  Choose the band removal method. Choices are:

  - `G` Geometric solution. Suffix will be `GSL` or `GSM`
  - *X*‘ Complex Average. Suffix will be `CS` (for Complex Solution)
  - `R` Root-mean-square. Suffix will be `RMS`
  - `M` Maximum of magnitudes. Suffix will be `Max`
  - `N` Mean of magnitudes. Suffix will be `MagMean`

- `--regularise`

  The Geometric Solution requires regularisation in noisy areas. Available methods are:

  - `M` Magnitude regularisation as in original paper
  - `L` Line regularisation (unpublished)
  - `N` None

  The default is `L`. If `L` or `M` are selected, then that character will be appended to the suffix.

- `--2pass, -2`

  Apply the second-pass energy-minimisation filter from the original paper. Can be likened to smoothing the phase data. If selected will append `2` to the suffix.

- `--alt-order`

  Phase-increments alternate, e.g. 180, 0, 90, 270. The default is the opposite (two blocks), e.g. 180, 90, 0, 270.

- `--ph-incs`

  Number of phase-increments. The default is 4. If you have multiple phase-increments and (for example) multiple flip-angles, `qissfpbands` can process them all in one pass.

- `--ph-order`

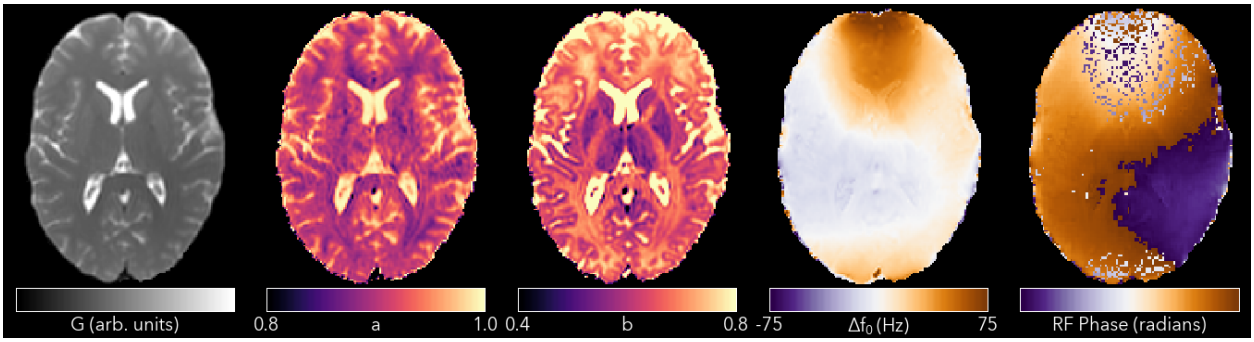
  The data order is phase-increment varying fastest, flip-angle slowest. The default is the opposite.

References

- Geometric Solution

1.4.2 qi_ssfp_ellipse

The most important result of Xiang & Hoff's Geometric Solution paper was that the SSFP signal equation can be expressed as an ellipse in the complex-plane. Shcherbakova built on this and showed it was possible to recover the ellipse parameters *G*, *a*, *b* from at least six phase-increments. They then proceeded to recover T1 & T2 from the ellipse parameters. This utility calculates the ellipse parameters, and `qi_ssfp_planet` then processes those parameters to calculate T1 & T2.



Example Command Line

```
qi_ssfp_ellipse ssfp_data.nii.gz < input.json
```

The SSFP file must be complex-valued. At least three pairs of opposing phase-increments are recommended (six images in total).

Outputs

- `ES_G` - The Geometric Solution point of the ellipse. Influences the overall size of the ellipse. This is called (M) in the Hoff and Shcherbakova papers, but it is not a measurable magnetization and hence to distinguish it a different letter is used.
- `ES_a` - The ellipse parameter that along with (G) controls the ellipse size.
- `ES_b` - The ellipse parameter that determines how flat or circular the ellipse is.
- `ES_theta_0` - The accrued phase due to off-resonance, divide by $2\pi TE$ (or πTR) to find the off-resonance frequency.

- `ES_phi_rf` - The effective phase of the RF pulse.

Important Options

- `--algo, -a`

  There are two available methods for calculating the ellipse parameters

  - `h` Hyper-Ellipse method, similar to that used in the Shcherbakova paper. Can fail when (alpha) falls below the Ernst angle, where there is an inversion of the ellipse properties.
  - `d` Direct non-linear fitting of the data, which does not suffer the above properties. The default.

References

- PLANET
- Hyper-Ellipse

1.4.3 qi_ssfp_planet

Converts the SSFP Ellipse parameters into relaxation times.

Example Command Line

```
qi_ssfp_planet ES_G.nii.gz ES_a.nii.gz ES_b.nii.gz
```

Outputs

- `PLANET_T1.nii.gz` - Longitudinal relaxation time
- `PLANET_T2.nii.gz` - Transverse relaxation time
- `PLANET_PD.nii.gz` - Apparent Proton Density

References

- PLANET

1.4.4 qi_ssfp_emt

Due to the short TR commonly used with SSFP, at high flip-angles the sequence becomes MT weighted. It is hence possible to extract qMT parameters from SSFP data. More details will be in a forthcoming paper.

Example Command Line

```
qi_ssfp_emt ES_G.nii.gz ES_a.nii.gz ES_b.nii.gz
```

Outputs

- `EMT_T1f.nii.gz` - Longitudinal relaxation time of the free water bool
- `EMT_T2f.nii.gz` - Transverse relaxation time of the free water pool
- `EMT_M0.nii.gz` - Apparent Proton Density
- `EMT_F.nii.gz` - Bound pool fraction
- `EMT_kf.nii.gz` - Forward exchange rate

References

- Bieri et al

- Gloor et al

1.5 Susceptibility

Susceptibility is a fundamental magnetic property of a material, and determines whether materials are paramagnetic (positive susceptibility) or diamagnetic (negative susceptibility). Quantitative Susceptibility Mapping (QSM) is a branch of MRI that aims to measure the susceptiblity of objects from the phase of the MR data. QUIT currently does not contain a full QSM processing pipeline, but does contain some phase unwrapping tools.

- *qi_unwrap_path*
- *qi_unwrap_laplace*

1.5.1 qi_unwrap_path

An implementation of the quality-guided path-based unwrapping of Abdul-Rahman et al. This is the recommended method to use (preferable over Laplacian).

Example Command Line

```
qi_unwrap_path phase_file.nii.gz
```

The phase file must be specified in radians (i.e. between $-\pi$ and $+\pi$). Does not read input from `stdin`, and currently there are no arguments to control the algorithms behaviour.

Outputs

- `input_unwrapped.nii.gz` - The unwrapped phase value, in radians.

References

- Abdul-Rahman et al 1
- Abdul-Rahman et al 2

1.5.2 qi_unwrap_laplace

Implements Laplacian-based phase-unwrapping. Along with phase-unwrapping, the Laplacian method implicitly removes background fields. This means it can alter phase values in undesirable ways and hence is not the preferred method.

Example Command Line

```
qi_unwrap_laplace phase_file.nii.gz
```

The phase file must be specified in radians (i.e. between -pi and +pi). Does not read input from *stdin*.

Outputs

- `input_unwrapped.nii.gz` The unwrapped phase, in radians.

Important Options

- `--mask, -m`

  Specify a mask, the phase will only be unwrapped inside this.

- `--erode, -e`

Radius to erode the input mask by (default 1 mm).

References

- Bakker et al

1.6 Statistics / GLM Tools

QUIT contains a few tools to help prepare your data for statistical analysis with outside tools, for instance non-parametric tests with Randomise or an ROI analysis using Pandas. These tools are:

- *qi_glmsetup*
- *qi_glmcontrasts*
- *qi_rois*

1.6.1 qi_glmsetup

FSL randomise takes a single 4D file with one volume per subject/timepoint as input, along with some simple text files that represent the GLM. Creating these files can be tedious, particularly with the FSL GUI. This tool makes it quick to create the relevant files.

Example Command Line

```
qi_glmsetup --groups=groups.txt --covars="brain_volume.txt,brain_volume.txt" --
↪design=glm.txt --out=merged.nii --sort subject_dirs*/D1_T1.nii
```

This command line will merge all the T1 maps in the directories matching the pattern. The file `groups.txt` should contain a single number per line, one for each T1 map. The number represents the group or cell that image belongs to. A group of 0 means exclude this file (so you don't have to work out a pattern that won't match that file). For example, with 8 scans belonding to 3 groups with 1 excluded scan, the `groups.txt` might look like:

```
1
3
3
2
0
1
2
1
```

The design matrix corresponding to the specified groups will be saved to the `glm.txt` file (Note - this will still need to be processed with `Text2Vest` to make it compatible with `randomise`). If `--sort` is specified, then the images and design matrix will be sorted into ascending order.

1.6.2 qi_glmcontrasts

Randomise does not save any `contrast` files, i.e. group difference maps, it only saves the statistical maps. For quantitative imaging, the contrasts can be informative to look at, as if scaled correctly, they can be interpreted as effect size maps. A group difference in human white matter T1 of only tens of milliseconds, even if it has a high p-value, is perhaps not terribly interesting as it corresponds to a change of about 1%. These contrast maps are particularly useful if used with the dual-coding visualisation technique.

Example Command Line

```
qi_glmcontrasts merged_images.nii design.txt contrasts.txt --out=contrast_prefix
```

The design and contrasts files should be raw text (not passed through `Text2Vest`). One contrast image will be generated for each row of the contrast matrix.

1.6.3 qi_rois

An alternative to voxel-wise statistics is to average the values over a pre-defined, anatomically meaningful region-of-interest in each quantitative image, and the perform statistics on those ROI values. This approach has several advantages, as more traditional and robust statistical methods can be used than the simple parametric T-tests that voxel-wise analysis tools use.

To avoid resampling issues, it is preferable to warp the ROI definitions (atlas files) to the subject space and sample the quantitative maps at their native resolution. This can make extracting all the ROI values tedious. This tool can extract ROI values from multiple files at once and produce a Comma-Separated Value (.csv) file as output for use with a stats tool such as Pandas. It can also calculate the volumes of the warped ROIs, i.e. for a Tensor/Deformation Based Morphometry analysis. The registrations required for this should be carried out with external tools, e.g. ANTs or FSL.

Example Command Line

For quantitative ROIs:

```
qi_rois labels_subject1.nii labels_subject2.nii ... labels_subjectN.nii data_subject1.
↪nii data_subject2.nii ... data_subjectN.nii --ignore_zero --header=subject_ids.txt
```

For ROI volumes:

```
qi_rois --volumes labels_subject1.nii labels_subject2.nii ... labels_subjectN.nii ---
↪header=subject_ids.txt
```

Any header files should contain one line per subject, corresponding to the input image files. The output of `qi_rois` is fairly flexible, and can be controlled with the `--transpose`, `--delim`, `--precision`, and `--sigma` options.

1.7 Utilities

QUIT contains a number of utilities. Note that these are actually compiled in two separate modules - `CoreProgs` contains the bare minimum of programs for the QUIT tests to run, while the actual `Utils` modules contains a larger number of useful tools for neuro-imaging pipelines. Their documentation is combined here.

- *qi_coil_combine*
- *qi_rfprofile*
- *qiaffine*
- *qicomplex*
- *qihdr*
- *qikfilter*
- *qimask*
- *qipolyfit/qipolyimg*
- *qisplitsubjects*
- *qidiff*

- *qinewimage*

1.7.1 qi_coil_combine

The program implements both the COMPOSER and Hammond methods for coil combination. For COMPOSER, a wrapper script that includes registration and resampling of low resolution reference data to the image data can be found in `qi_composer.sh`.

Example Command Line

```
qi_coil_combine multicoil_data.nii.gz --composer=composer_reference.nii.gz
```

Both the input multi-coil file and the reference file must be complex valued. Does not read input from `stdin`. If a COMPOSER reference file is not specifed, then the Hammond coil combination method is used.

Outputs

- `input_combined.nii.gz` - The combined complex-valued image.

Important Options

- `--composer, -c`

  Use the COMPOSER method. The reference file should be from a short-echo time reference scan, e.g. UTE or ZTE. If

- `--coils, -C`

  If your input data is a timeseries consisting of multiple volumes, then use this option to specify the number of coils used in the acquisition. Must match the number of volumes in the reference image. Does not currently work with the Hammond method.

- `--region, -r`

  The reference region for the Hammond method. Default is an 8x8x8 cube in the center of the acquisition volume.

References

- COMPOSER
- Hammond Method

1.7.2 qi_rfprofile

This utility takes a B1+ (transmit field inhomogeneity) map, and reads an excitation slab profile from `stdin`. The two are multiplied together along the slab direction (assumed to be Z), to produce a relative flip-angle or B1 map.

Example Command Line

```
qi_rfprofile b1plus_map.nii.gz output_b1_map.nii.gz < input.json
```

Example Command Line

```
{
    "rf_pos" : [ -5, 0, 5],
    "rf_vals" : [[0, 1, 0],
                 [0, 2, 0]]
}
```

`rf_pos` specifies the positions that values of the RF slab have been calculated at, which are specified in `rf_vals`. Note that `rf_vals` is an array of arrays - this allows `qi_rfprofile` to calculate profiles for multiple flip-angles in a single pass. The units for `rf_pos` are the same as image spacing in the header (usually mm). `rf_vals` is a unitless fraction, relative to the nominal flip-angle.

These values should be generated with a Bloch simulation. Internally, they are used to create a spline to represent the slab profile. This is then interpolated to each voxel's Z position, and the value multiplied by the input B1+ value at that voxel to produce the output.

Outputs

- `output_b1map.nii.gz` - The relative flip-angle/B1 map

1.7.3 qiaffine

This tool applies simple affine transformations to the header data of an image, i.e. rotations or scalings. It was written because of the inconsistent definitions of co-ordinate systems in pre-clinical imaging. Non-primate mammals are usually scanned prone instead of supine, and are quadrupeds instead of bipeds. This means the definitions of superior/inferior and anterior/posterior are different than in clinical scanning. However, several pre-clinical atlases, e.g. Dorr et al, rotate their data so that the clinical conventions apply. It is hence useful as a pre-processing step to adopt the same co-ordinate system. In addition, packages such as SPM or ANTs have several hard-coded assumptions about their input images that are only appropriate for human brains. It can hence be useful to scale up rodent brains by a factor of 10 so that they have roughly human dimensions.

Example Command Line

```
qiaffine input_image.nii.gz --scale=10.0 --rotX=90
```

If no output image is specified, the output will be written back to the input filename.

Common Options

- `--scale, -s`

  Multiply the voxel spacing by a constant factor.

- `--rotX, --rotY, --rotZ`

  Rotate about the specified axis by the specified number of degrees. Note that currently, each rotation can only be specified once and the order will always be X, Y, then Z.

- `--offX, --offY, --offZ`

  Add the specified offset to the origin.

- `--center, -c`

  Set the image origin to be the Center of Gravity of the image.

1.7.4 qicomplex

Manipulate complex/real/imaginary/magnitude/phase data. Created because I was fed up with how `fslcomplex` works.

Example Command Line

```
qicomplex -m input_magnitude.nii.gz -p input_phase.nii.gz -R output_real.nii.gz -I␣
↪output_imaginary.nii.gz
```

Lower case arguments `--mag, -m, --pha, -p, --real, -r, --imag, -i, --complex, -x` are inputs (of which it is only valid to specify certain combinations, complex OR magnitude/phase OR real/imaginary).

Upper case arguments `--MAG, -M, --PHA, -P, --REAL, -R, --IMAG, -I, --COMPLEX, -X` are outputs, any or all of which can be specified.

An additional input argument, `--realimag` is for Bruker "complex" data, which consists of all real volumes followed by all imaginary volumes, instead of a true complex datatype.

The `--fixge` argument fixes the lack of an FFT shift in the slab direction on GE data by multiplying alternate slices by -1. `--negate` multiplies the entire volume by -1. `--double` reads and writes double precision data instead of floats.

1.7.5 qihdr

Prints the header of input files as seen by ITK to `stdout`. Can extract single header fields or print the entirety.

Example Command Line

```
qihdr input_file1.nii.gz input_file2.nii.gz --verbose
```

Multiple files can be queried at the same time. The `--verbose` flag will make sure you can tell which is which.

Important Options

If any of the following options are specified, then only those fields will be printed instead of the full header. This is useful if you want to use a header field in a script: * `--origin, -o` * `--spacing, -S` - The voxel spacing * `--size, -s` - The matrix size * `--voxvol, -v` - The volume of one voxel

Another useful option is `--meta, -m`. This will let you query specific image meta-data from the header. You must know the exact name of the meta-data field you wish to obtain.

1.7.6 qikfilter

MR images often required smoothing or filtering. While this is best done during reconstruction, sometimes it is required as a post-processing step. Instead of filtering by performing a convolution in image space, this tool takes the Fourier Transfrom of input volumes, multiplies k-Space by the specified filter, and transforms back.

Example Command Line

```
qikfilter input_file.nii.gz --filter=Gauss,0.5
```

Outputs

- `input_file_filtered.nii.gz`

Important Options

- `--filter,-f`

  Specify the filter to use. For all filters below the value (r) is the fractional distance from k-Space center, i.e. $r = \sqrt{(((k_x/s_x)^2 + (k_y/s_y)^2 + (k_z/s_z)^2)/3)}$. Valid filters are:

  - `Tukey,a,q`

    A Tukey filter with parameters a and q. Filter value is 1 for $r < (1 - a)$ else the value is $\frac{(1+q)+(1-q)\cos(\pi\frac{r-(1-a)}{a})}{2}$

  - `Hamming,a,b`

A Hamming filter, parameters *a* and *b*, value is $a - b\cos(\pi(1 + r))$

- `Gauss,w` or `Gauss,x,y,z`

  A Gaussian filter with FWHM specified either isotropically or for each direction independantly.

- `Blackman` or `Blackman,a`

  A Blackman filter, either with the default parameter of $\alpha = 0.16$ or the specified α. Refer to Wikipedia for the relevant equation.

- `Rectangle,Dim,Width,Inside,Outside`

  A rectangular or top-hat filter along the specified dimension (must be 0, 1 or 2).

If multiple filters are specified, they are concatenated, *unless* the `--filter_per_volume` option is specified.

- `--filter_per_volume`

  For multiple flip-angle data, the difference in contrast between flip-angles can lead to different amounts of ringing. Hence you may wish to filter volumes with more ringing more heavily. If this option is specified, the number of filters on the command line must match the number of volumes in the input file, and they will be processed in order.

- `--complex_in` and `--complex_out`

  Read / write complex data.

1.7.7 qimask

Implements several different masking strategies. For human data, BET, antsBrainExtraction of 3dSkullStrip are likely better ideas. For pre-clinical data, the strategies below can provide a reasonable mask with some tweaking. There are potentially three stages to generating the mask:

1 - Binary thresholding. If lower or upper thresholds are specified, these are used to separate the image into foreground and background. If neither are specified, then Otsu's method is used to automatically estimate a reasonable threshold value. 2 - (Optional) Run the RATs algorithm 3 - (Optional) Hole-filling

Example Command Line

```
qimask input_image.nii.gz --lower=10 --rats=1200 --fillh=1
```

In this case an intensity value of 10 will be used as the threshold, RATs will be run with a target volume of 1200 mm^3, and then holes with a radius of 1 voxel will be filled.

Outputs

- `input_image_mask.nii.gz`

Important Options

- `--lower,-l`/`--upper,-u`

  Specify lower and/or upper intensity thresholds. Values below/above these values are set to 0, those inside are set 1. If this option is not specified, Otsu's method will be used to generate a threshold value. If no thresholding is desired, specify `--lower=0`.

- `--rats, -r`

Use the RATs algorithm to remove non-brain tissue. The RATs algorithm uses erode & dilate filters of progressively increasing size until the largest connected component falls below a target size. For rats, target values of around 1000 mm^3 are reasonable.

- `--fillh, -F`

  Fill holes in the mask up to radius N voxels.

References

- RATs algorithm

1.7.8 qipolyfit/qipolyimg

These tools work together to fit Nth order polynomials to images. This is typically used for smoothing a B1 field.

`qipolyfit` will output the polynomial co-efficients and origin to `stdout`. `qipolyimg` can then read these to generate the polyimage image, using a different image as the reference space. In this way the polynomial image can be created without having to use upsampling.

Example Command Line

```
qipolyfit noisy_b1_map.nii.gz --mask=brain_mask.nii.gz --order=8 | qipolyimg hires_t1_
↪image.nii.gz hires_smooth_b1_map.nii.gz --order=8
```

With the above command-line the output of `qipolyfit` is piped directly to the output of `qipolyimg`. You can instead redirect it to a file with > and read it in separately. The `--order` argument must match between the two commands.

Important Options

- `--order, -o`

  The order of the fitted polynomial. Default is 2 (quadratic)

- `--mask, -m`

  Only fit the data within a mask. This is usually the brain or only white-matter.

- `--robust` (`qipolyimg` only)

  Use Robust Polynomial Fitting with Huber weights. There is a good discussion of this topic in the Matlab help files.

1.7.9 qisplitsubjects

This program is deprecated. It was used to separate ex-vivo images containing multiple subjects into distinct images.

1.7.10 qidiff

Calculates the mean square difference between two images and checks if it is below a tolerance value. Used in the QUIT tests to ensure that calculated parameter maps are close to their baseline values.

Example Command Line

```
qidiff --baseline=original.nii --input=calculated.nii --noise=0.01 --tolerance=30
```

The program simply returns `FAILURE` or `SUCCESS`, which is detected by BATS. Note, to make useage clearer, unlike most other QUIT programs all input is specified as arguments.

Important Options

- `--baseline`

  The baseline image. Required.

- `--image`

  The image to compare to the baseline. Required.

- `--noise`

  The added noise level.

- `--tolerance`

  The tolerance is relative to the added noise level (i.e. it is a noise amplification factor).

- `--abs, -a`

  Use absolute difference instead of fractional difference (i.e. do not divide by the baseline image). Useful when images contain genuine zeros (e.g. off resonance maps).

1.7.11 qinewimage

Creates new images filled with specified patterns. Used for generating test data.

Example Command Line

```
qinewimage --size 32,32,32 --grad "0 0.5 1.5" output_image.nii.gz
```

The file specified on the command line is the *output* file.

Important Options

- `--dims, -d`

  The output dimension. Valid values are 3 and 4.

- `--size, -s`

  Matrix size of the output image.

- `--fill, -f`

  Set all voxels in the image to the specified value.

- `--grad, -g "DIM,LOW,HIGH"`

  Fill voxels with a gradient along the specified dimension, starting at the low value at one edge and finishing at the high value on the other. It is recommended to encase `DIM,LOW,HIGH` with quotation marks as they must be passed as a single string to be interpreted properly.

- `--step, -t "DIM,LOW,HIGH,STEPS"`

  Similar to `--grad`, but instead of a smooth gradient will with a number of discrete steps.

- `--wrap, -w`

  Wrap output voxels at the specified value. Useful for simulating phase data.

1.8 Developer

Below are a few introductory notes for anyone who wants to compile QUIT from source or might be interested in contributing to QUIT. Although I hope that most QUIT code is fairly clear, the following is provided to give a high-level overview of how most QUIT programs are structured.

1.8.1 Basic Requirements

1. A C++17 compliant compiler (GCC 7.0.0+, Clang 3.9+)
2. CMake version 3.10.2 or higher (http://www.cmake.org)

WARNING - You will require a recent compiler for QUIT as it uses C++17 features. On Mac simply having the most recent software updates is enough, on Linux you will need GCC 7.0.0. No one has been brave enough to compile QUIT on Windows to date.

1.8.2 Installing GCC 7

Most Linux systems currently ship with GCC 4.8 or lower as their system compiler. As of QUIT 2.1 you will require GCC 7 as QUIT uses C++17 features. Joost Kuijer has kindly provided the following recipe for installing newer GCC versions in a user directory and using it to compile QUIT.

1. Follow the GCC guide here: https://gcc.gnu.org/wiki/InstallingGCC
2. **Before running the `build.sh` script let cmake know about the new compiler**

```
export PATH=$HOME/GCC-7.0.0/bin:$PATH
export LD_LIBRARY_PATH=$HOME/GCC-7.0.0/lib:$HOME/GCC-7.0.0/lib64:$LD_LIBRARY_
↪PATH
export CC=$HOME/GCC-7.0.0/bin/gcc
export CXX=$HOME/GCC-7.0.0/bin/g++
```

3. **Before executing the compiled code also do (you can add this line to your *.bashrc* file:**

```
export LD_LIBRARY_PATH=$HOME/GCC-7.0.0/lib:$HOME/GCC-7.0.0/lib64:$LD_LIBRARY_
↪PATH
```

1.8.3 External Libraries

QUIT is built using several C++ libraries. These are currently included in the project as git submodules. The easiest way to initialise these is with the `easy_build.sh` script. However, if you are already have some of them available you may wish to not run the script and instead configure them yourself. This is discussed further below. The libraries are:

- Eigen

  A high performace linear algebra library. Used for all signal equations in QUIT.

- Ceres

  A high performace, high quality collection of non-linear least squares solvers.

- ITK

The Insight Tool-Kit. This is intended as a registration library for medical images. Although the registration features are not used in QUIT, the overall framework is extremely useful. In particular, ITK reads a large variety of common file-formats, and all ITK filters automatically check that their inputs share a common space / co-ordinate definition. This single feature alone prevents the vast majority of easy user mistakes - e.g. trying to use a B1 map that has a different voxel spacing to the input data.

1.8.4 Compilation

If you are unfamiliar with C++/CMake/git etc. a script is provided that should be able to build the tools provided the right software is available on your system. To use it, in a terminal window, change directory to where you unpacked QUIT. Then type `./easy_build.sh`. This should correctly checkout the git repositories for each external library, build Ceres and ITK, and then configure and build QUIT.

If you are familiar with C++/CMake/git etc. then compiling QUIT should be straightforward. The rough steps are:

1. Ensure you have all the libraries above available. If you do not have system versions, then run `git submodule init; git submodule update` in the QUIT directory.
2. Compile the Ceres library. Make sure it uses the same version of Eigen that you will use for QUIT.
3. Compile ITK following their instructions.
4. Create a build directory for QUIT and use cmake/ccmake to configure the project. Specify the paths to the library when prompted. It is likely that CMake will report an error on the first 'Configure' attempt if it cannot locate Eigen, specify the correct directory and run the configure step again.
5. Compile QUIT.

1.8.5 File Formats

The available file formats are controlled by the main `CMakeLists.txt` in the root QUIT directory, by listing them as `COMPONENTS` in the ITK `find_package()` step. Add any additional file formats you wish to use here.

1.8.6 Tests

QUIT programs are tested using the Bash Automated Test System, which is included as a git submodule. To run the tests, point BATS at the Tests directory, e.g. `path/to/bats QUIT/Test`, which will run all of the tests. It is possible to run the test files individually as well. It was decided to use BATS instead of a unit-test based framework because this allows the QUIT programs to be tested as a whole, including command-line arguments.

Most QUIT programs are tested by generating ground-truth parameter files with `qinewimage`, feeding these into `qisignal` to generate simulated MR images with added noise, and then running the particular QUIT program to calculate some parameter maps, then comparing these to the ground-truth with `qidiff`. `qidiff` calculates figure-of-merit based on noise factors, i.e. they are a measure of how much the signal noise is amplified in the final maps. In this way the tests also serve to illustrate the quality of the methods as well as whether the programs run correctly. For programs where a ground-truth image cannot be generated easily, the tests at least ensure that the program runs and does not crash.

1.8.7 The ModelFitFilter

The core part of QUIT is the `ModelFitFilter` and its dependent type `FitFunction`, found in `Source/Core/`. This is a sub-class of the ITK `ImageToImageFilter`. The vast majority of QUIT programs declare an *Model* and *FitFunction* sub-class and use these to process the data. `ModelFitFilter` abstracts out most of the heavy lifting of

extracting voxel-wise data from multiple inputs and writing it out to multiple outputs, leaving the `FitFunction` to process a single-voxel. A `Model` defines the number of expected inputs and their size, the number of fixed & varying parameters, and the number of outputs.

1.8.8 Example: `qidespot1`

The structure of `qidespot1` is similar to most QUIT programs, and is a good example of most features. At the start are the includes (obviously). After that several `FitFunction` subclasses are defined, as well as a Ceres cost-function. The Ceres documentation is excellent, so refer to that for more information. After all the `FitFunction` classes are defined, the main program body begins. At the start of the program, all the command-line options are defined and then parsed. Then the various inputs are read and passed to the `ModelFitFilter`, which is then updated. Finally, the outputs are written back to disk.

CHAPTER 2

Citing

QUIT has been published in the Journal of Open Source Software. If you use it, please cite:

Note: Wood, (2018). QUIT: QUantitative Imaging Tools. Journal of Open Source Software, 3(26), 656, https://doi.org/10.21105/joss.00656

Thank you.

CHAPTER 3

Installation

Pre-compiled binaries are provided for Linux and Mac OS X in a .tar.gz archive from the releases page.

Download the correct archive for your platform, untar it and then ensure that the binaries can be found via your *PATH* environment variable. Either edit your shell profile (e.g. `.bashrc`) and add the QUIT directory your `$PATH` variable there, or copy the files to somewhere that will be on your path, e.g. `/usr/local/bin` (this will likely require `sudo` permissions).

The Linux binaries are built with Ubuntu 14.04 with GCC 7. If you need to run on an older version of Linux with a previous version of `glibc` then you may need to compile from source.

CHAPTER 4

Compile From Source

See the *Developer* documentation.

CHAPTER 5

General Usage

QUIT programs take their input as a combination of command-line arguments and text files passed to `stdin`. If you run a QUIT program with no arguments then will see a message like this:

```
~: qidespot1
SPGR FILE was not specified. Use --help to see usage.
>
```

If you then run it with `--help`, you will see some usage instructions:

```
~: qidespot1 --help
qidespot1 {OPTIONS} [SPGR FILE]

    Calculates T1 maps from SPGR data
    http://github.com/spinicist/QUIT

OPTIONS:

    SPGR FILE                          Path to SPGR data
    -h, --help                         Show this help message
    -v, --verbose                      Print more information
    -T[THREADS], --threads=[THREADS]   Use N threads (default=4, 0=hardware
                                         limit)
    -o[OUTPREFIX], --out=[OUTPREFIX]   Add a prefix to output filenames
    -b[B1], --B1=[B1]                  B1 map (ratio) file
    -m[MASK], --mask=[MASK]            Only process voxels within the mask
    -s[SUBREGION],
    --subregion=[SUBREGION]            Process subregion starting at voxel
                                         I,J,K with size SI,SJ,SK
    -r, --resids                       Write out residuals for each data-point
    -a[ALGO], --algo=[ALGO]            Choose algorithm (l/w/n)
    -i[ITERS], --its=[ITERS]           Max iterations for WLLS/NLLS (default
                                         15)
    -p[CLAMP PD], --clampPD=[CLAMP
    PD]                                Clamp PD between 0 and value
```

(continues on next page)

(continued from previous page)

```
    -t[CLAMP T1], --clampT2=[CLAMP
    T1]                                 Clamp T1 between 0 and value
    "--" can be used to terminate flag options and force all following
    arguments to be treated as positional options
```

The first line shows that DESPOT1 expects a single input image, in this case SPGR/FLASH/FFE, specified on the command line. However, if you naively run:

```
~: qidespot1 some_spgr_data.nii.gz
```

Nothing will happen. This is because most imaging formats do not store parameters that are required for quantitative imaging, e.g. the repetition time and flip-angles in their headers. QUIT programs hence expect to read this information in a text file passed to `stdin`. If you create a small text file `spgr.json` containing the following:

```
{
    "SPGR": {
        "TR": 0.01,
        "FA": [3, 18]
    }
}
```

and run the following:

```
~: qidespot1 some_spgr_data.nii.gz < spgr.json
```

then - provided your input data does contain two volumes corresponding to flip-angles 3 and 18 degrees - DESPOT1 will run, and you should see two files created (`D1_T1.nii.gz` and `D1_PD.nii.gz`). If you want to see what the programs are doing while running, specify the `--verbose` or `-v` options.

CHAPTER 6

Common Options

The following options are supported by most, but not necessarily all, QUIT programs.

- `--out, -o`

  Add a prefix to the output parameter files. By default, most QUIT programs write their output files using filenames in a pattern *PROGRAM_PARAMETER.nii.gz*. They will overwrite any existing files with the same names. If you need to save the output from multiple runs of the same program, or want to save output to a particular directory, use this option to add an addtional prefix to the output names.

- `--mask, -m`

  Specify a mask file, where non-zero values indicate that voxels should be processed, and zero values indicate that voxels should not be processed. The background voxels will be set to zero in the final image. This is useful for two reasons, first is to simply speed up processing for long-running programs (e.g. *qimcdespot*), second is that outside the head fitting data is non-sensical, as there is no signal. Hence these regions appear very noisy on output maps, which can make visualization difficult.

- `--threads, -t`

  Control the maximum number of threads used. The majority of QUIT programs are multi-threaded across voxels to improve processing times. In some parallel computing environments (e.g. Sun Grid Engine), it is possible to set the maximum number of cores available to a program, and it is hence good for CPU utilisation to match the number of threads to the number of cores. The default is 4. Note that HyperThreading may make the number of logical cores appear to be double the number of physical cores - QUIT programs are CPU bound, not IO bound, and hence gain no benefit from HyperThreading. You are better to specify the number of physical cores available rather than the number of logical cores.

- `--subregion, -s`

  Similar to *–mask*, this command will only process a sub-region of the input images. The argument needs to be in the format *"start_i,start_j,start_k,size_i,size_j,size_k"* where *i,j,k* are voxel indices (not physical co-ordinates). This is useful to speed up processing for trial-runs of pipelines.

- `--resids, -r`

Most QUIT programs will write out a single root-sum-squared residual image along with their parameter maps. Use this option to also output residuals for each data-point to look for systematic offsets. Note that if multiple inputs are specified (e.g. *qimcdespot*), then this option will write out a single cocatenated file for all input data-points in order.

- `--B1, -b` & `--f0, -f`

  Several of the QUIT programs take B1 (relative flip-angle) and f0 (off-resonance in Hz) maps as correction factors.

CHAPTER 7

File Formats

By default, QUIT is compiled with support for NIFTI and NRRD formats. The preferred file-format is NIFTI for compatibility with FSL and SPM. By default QUIT will output `.nii.gz` files. This can be controlled by the *QUIT_EXT* environment variable. Valid values for this are any file extension supported by ITK that QUIT has been compiled to support, e.g. `.nii` or `.nrrd`, or the FSL values `NIFTI`, `NIFTI_PAIR`, `NIFTI_GZ`, `NIFTI_PAIR_GZ`.

The ITK library supports a much wider variety of file formats, but adding support for all of these almost triples the size of the compiled binaries. Hence by default they are excluded. You can add support for more file formats by compiling QUIT yourself, see the *Developer* documentation. Note that ITK cannot write every format it can read (e.g. it can read Bruker 2dseq datasets, but it cannot write them).

CHAPTER 8

Scripting

The QUIT tools are designed to be used inside shell scripts or SGE jobs. 3 scripts are provided along with the binaries. `qi_composer.sh` uses the `qi_coil_combine` program to correctly combine complex data from multiple element RF coils. `qi_example_mcd_b1.sh` and `qi_example_mcd_hifi.sh` give example mcDESPOT processing pipelines. The first assumes that you have a good B1 map and have already registered all the input volumes to a common reference. The second shows one of doing this using FSL. Registration is beyond the scope of QUIT, so you must have additional tools installed for this.