Introduction to the BART Toolbox

Martin Uecker

Graz University of Technology

BART Toolbox for Computational MRI

- **For research use**
- Available for Linux, MacOS X, (Windows), ...
- BSD license (free for commercial use)
- `https://mrirecon.github.io/bart/`

Contributors:
Jon Tamir, Christian Holme, Moritz Blumenthal, Nick Scholand, Philip Schaten, Daniel Mackner, ... many more (see ACKNOWLEDGMENTS)

Research Support
American Heart Association Grant 12BGIA9660006, NIH Grant R41RR09784 and Grant R01EB009690, UC Discovery Grant 193037, Sloan Research Fellowship, GE Healthcare, DZHK (German Centre for Cardiovascular Research), and a personal donation from David Donoho's Shaw Prize, NIH Grant U24EB029240-01

TU Graz
Graz University of Technology

UNIVERSITÄTSMEDIZIN GÖTTINGEN UMG

DZHK
DEUTSCHES ZENTRUM FÜR
HERZ-KREISLAUF-FORSCHUNG E.V.

Motivation

- Rapid prototyping
  (similar to Matlab, octave, ...)
- Reproducible research
  (i.e. scripts to reproduce experiments)
- Robustness and clinically feasible runtime
  (C/C++, OpenMP, GPU programming)
- Educational tool
- Platform for collaboration

Motivation

- Rapid prototyping
  (similar to Matlab, octave, ...)
- Reproducible research
  (i.e. scripts to reproduce experiments)
- Robustness and clinically feasible runtime
  (C/C++, OpenMP, GPU programming)
- Educational tool
- Platform for collaboration

Programming library

- Consistent API based on multi-dimensional arrays
- Fast transforms: FFT, nuFFT, wavelet
- Generic iterative (matrix-free) algorithms
  (conjugate gradients, FISTA, IRGNM, ADMM, iPALM, . . .)
- Composable operators / algorithms
- Transparent GPU acceleration of most functions
- Cluster support (MPI)
- Neural networks
- MRI sequences

Command-line tools

- Simple file format
- Looping, parallelization, streaming
- Interoperability with Matlab, Python
- Basic operations: fft, resize, slice, . . .
- Sensitivity, motion, etc. calibration and image reconstruction

Programming library

- Consistent API based on multi-dimensional arrays
- Fast transforms: FFT, nuFFT, wavelet
- Generic iterative (matrix-free) algorithms (conjugate gradients, FISTA, IRGNM, ADMM, iPALM, . . .)
- Composable operators / algorithms
- Transparent GPU acceleration of most functions
- Cluster support (MPI)
- Neural networks
- MRI sequences

Command-line tools

- Simple file format
- Looping, parallelization, streaming
- Interoperability with Matlab, Python
- Basic operations: fft, resize, slice, . . .
- Sensitivity, motion, etc. calibration and image reconstruction

Generic Tools

fft	FFT
slice	extract slice
extract	extract block
resize	crop / zeropad
flip	reverse dimensions
join	merge files
zeros/ones	create files
circshift	circular shift
conv	convolution
scale	scale
conj	complex ocnjugate
fmac	fused multiply-add
saxpy	scalar mult. and add
sdot	dot product
rss	root-sum-of-squares
repmat	repmat
svd	singular-value d.

MRI Tools

cc	channel compression
walsh	Walsh method
whiten	whiten k-space
caldir	Direct calibration
ecalib	ESPIRiT calibration
ncalib	non-linear calibration
pics	ℓ_1-SENSE/ESPIRiT
moba	model-based recon
reconet	deep-learning recon
nlinv	non-linear inversion
nufft	non-uniform FFT
pocsense	POCSENSE
fovshift	Retro-shift FOV
pattern	extract pattern
poisson	Poisson-disc pattern
signal	create signal curves

pics: Parallel Imaging Compressed Sensing

$>$ bart pics -R*A*:*B*:*C*:*D* -R ... [-t trj] [-B basis] kspace sens image

- parallel imaging and compressed sensing
- non-Cartesian k-space trajectories
- subspace reconstruction
- multiple regularization terms
- A: different types of regularization:
  ℓ_2, ℓ_1, total (generalized) variation, infimal convolution,
  ℓ_1-wavelet, (multi-scale) low-rank, deep priors
- B: transforms along arbitrary dimensions (space, time, etc.)
- C: joint-thresholding along arbitrary dimensions
- D: regularization parameter

moba: Model-Based Reconstruction

```
> bart moba - model -r ... [-t trj] kspace enc output [sens]
```

- signal model (T_1, T_2, $T_2^\star$, $T_2^\star + \delta B_0$, Bloch)
- parallel imaging and compressed sensing
- non-Cartesian k-space trajectories
- A, B, C, D: regularization

reconet: Deep-Learning Reconstruction

```
> bart reconet –network ... [–trajectory trj]
            kspace sens weights out/ref
```

- network (VarNet, MoDL)
- parallel imaging
- non-Cartesian k-space trajectories

Resources

1. README, doc/
2. Website: `https://mrirecon.github.io/bart`
3. Source code: `https://github.com/mrirecon/bart`
4. Mailing list with public archive (see website)
5. Workshops:
   `https://github.com/mrirecon/bart-workshop`
6. Webinars:
   `https://github.com/mrirecon/bart-webinars`
7. Help: Each command has '-h' option for a very brief help.

Resources

1. README, doc/
2. Website: `https://mrirecon.github.io/bart`
3. Source code: `https://github.com/mrirecon/bart`
4. Mailing list with public archive (see website)
5. Workshops: `https://github.com/mrirecon/bart-workshop`
6. Webinars: `https://github.com/mrirecon/bart-webinars`
7. Help: Each command has '-h' option for a very brief help.

DACH+NORDIC ISMRM: OpenMR Days

https://github.com/mritogether/bart-workshop/dach_ismrm2025/

Schedule

- Introduction to the BART Toolbox (Martin Uecker)
- Break
- Sequence Framework (New!) (Daniel Mackner)
- Scanning
- Subspace Tutorial (Vitali Telezki)