

# Lossy Compression Should Also Be Used in Functional MRI Research

Zalán Rajna<sup>1,2</sup>(✉), Pekka Nyrönen<sup>1</sup>, Vesa Kiviniemi<sup>2</sup>, and Tapio Seppänen<sup>1</sup>

<sup>1</sup> Center for Machine Vision and Signal Analysis, University of Oulu, Oulu, Finland  
`zalan.rajna@oulu.fi`

<sup>2</sup> Oulu Functional Neuroimaging Group, Oulu University Hospital, Oulu, Finland

**Abstract.** The amount of functional MRI (fMRI) data processed in research is growing, yet no practice or protocol to store them in a lossy format exists. Many researchers are struggling with limited storage space, and speed of common processing tools are often bound by storage speed. In this work, we present a lossy compression framework for fMRI data with user adjustable trade-off between compression ratio and root mean squared error (RMSE). Our goal is to demonstrate the usability of on-the-fly lossy compression for fMRI data. On one hand, the storage footprint and processing speeds both benefit from higher data compression rates achieved with lossy compression. On the other hand, data quality for functional analysis remains effectively the same. With this short demonstration we encourage the research community to develop a lossy data standard for fMRI data.

**Keywords:** fMRI, lossy compression, neuroimaging

## 1 Introduction

The most commonly used and standardized way of handling functional MRI (fMRI) data is the Neuroimaging Informatics Technology Initiative (NIfTI) format[1]. The data format claims to deliver a unified representation for medical professionals, scientists and engineers. However, fMRI nowadays produces immense amounts of raw data, and the standard has not been updated to provide lossy compression where applicable. While work has been done previously on lossy and lossless fMRI data compression in [2–7], none of them is actively used. Meanwhile, the size of recordings is growing with increased frame rates, e.g. magnetic resonance encephalography (MREG)[8].

In direct medical, i.e. diagnostic use of MRI recordings, introducing lossy compression is usually unacceptable, since the cost of buying and operating recording devices are particularly high, and there is no room for technical degradation of accuracy in medical diagnosis. But the situation is different in research, where prototyping and feasibility analysis of methods do not necessarily require lossless compression. For example, the commonly used process of registering fMRI recordings of multiple subjects introduces data loss through interpolation.

Some data loss may be even beneficial, in case the smaller storage space requirements and analysis speed improvements compensate for the drawbacks of lossy data compression.

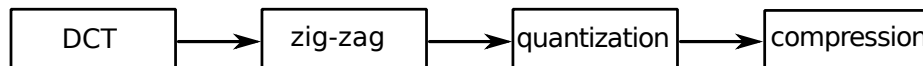
In this work, a novel lossy compression encoding and decoding framework (codec) for fMRI data and its effects on analysis are presented. We show, that our codec significantly reduces fMRI file sizes, while maintaining acceptable quality for analysis.

## 2 Methods

In the NIfTI data format[1], 4D fMRI voxels are stored sequentially in a one-dimensional array in row-major order with a leading 352-bit header. Optionally, the file is compressed losslessly in the `gzip` format. This introduced serious seek speed limitations earlier, which have been solved with `indexed_gzip`[9] as proposed in [10].

In this work, we are focusing on IEEE 754 32-bit floating point data only, since that is the most common data representation format in fMRI research in the processing pipelines.

To simplify file format definitions, we used the widely adapted HDF5[11] file format for our result, which can also conveniently store the header information as a separate entity within the same file. Figure 1 presents the overview of the encoding pipeline used for our demonstration. All elements have been widely used for many years in encoding various data types. Our algorithm is similar to established image and video compression algorithms.



**Fig. 1.** The main steps of the encoding pipeline: discrete cosine transform, zig-zag reordering, quantization, generic compression.

### 2.1 Discrete Cosine Transform

For simplicity, the data is processed in  $8 \times 8 \times 8$  blocks (512 voxels) for each time step, and the coding steps are executed block-wise. Zero padding was used where necessary. Discrete cosine transform (DCT)[12] is applied using FFTW library[13] on the data for its attractive computational attributes: operating in real-to-real domain, reducing required operations, and saving time compared to discrete Fourier transform.

The one-dimensional DCT-II is formulated as

$$X_k = \sum_{n=0}^{N-1} x_n \cos \left[ \frac{\pi}{N} \left( n + \frac{1}{2} \right) k \right] \quad (1)$$

where  $k = 0, \dots, N - 1$ ,  $N$  is the number of samples in the sequence  $x = [x_1, x_2, \dots, x_n]$  and  $k$  is the index of the currently computed element in the transformed vector  $X$ . The inverse DCT-II (IDCT-II or DCT-III) is

$$X_k = \frac{1}{2}x_0 + \sum_{n=0}^{N-1} x_n \cos \left[ \frac{\pi}{N} n \left( k + \frac{1}{2} \right) \right] \quad (2)$$

where denotations are as previously. In addition, as the transformed objects are three-dimensional, and multi-dimensional DCT transforms are simply the product of the transforms for each dimension with FFTW, data is multiplied by scaling factor  $(1/2)^3$ .

## 2.2 Reordering (Zig-zag)

The elements of a DCT-transformed block are reordered by frequency in ascending order using L1 distance. The first element is thus the DC coefficient.

This sort of arranging is known as zig-zag reordering in digital image processing, which we are extending here to three spatial dimensions. It is used e.g. in JPEG image compression as well[14]. The motivation is to sort the DCT domain elements by frequency: as the low-frequency coefficients usually contain most of the energy of the block, they are to be quantized as little as possible[14].

## 2.3 Quantization

Part of the data is decimated by gradually zeroing least significant digits of mantissas towards the end of the reordered block. The DC element is always left untouched.

We decided on the decimating method with the help of a short experiment. Three different sample blocks were selected ( $S_1$ : inside brain,  $S_2$ : brain edge,  $S_3$ : mostly zero) to evaluate achievable compression and normalized root mean square error (NRMSE) defined as  $NRMSE = \frac{RMSE}{\bar{y}}$ , where  $\bar{y}$  is the average value of the data for which root mean squared error (RMSE) was calculated.

The three experiments were following a linear, a quadratic, and a logarithmic function between the distance of the DC coefficient and the number of least significant bits zeroed from the mantissa. The mathematical formulation for the linear function is

$$f_q^{lin}(x) = \text{round} \left[ \frac{(B_{max} - B_{min})(x - B_{min})}{N - B_{min}} \right] \quad (3)$$

where  $0 \leq B_{min} < N \leq B_{max}$  and  $x$  is the element index counted from the DC coefficient,  $B_{max}$  is the index of the last bit that is zeroed,  $B_{min}$  is the index for the first zeroed bit, and  $N$  is the index of the element at which  $B_{max}$  is reached, hence referred to as the cut-off index. Only the mantissa is decimated, so  $B_{max}$  cannot be more than 22 for single precision float. For the rest of the paper,  $B_{max}$

is referred to as the quantization level. The lower limit for  $B_{min}$  is 1. In a similar manner, the formula for the quadratic function is

$$f_q^{quad}(x) = \text{round} \left[ -\frac{B_{max}}{N^2}x^2 - 2x \left( -\frac{B_{max}}{N} \right) \right] \quad (4)$$

and for the logarithmic function

$$f_q^{log}(x) = \min [B_{max}, \max [0, \text{round}(c \cdot \log(x))]] \quad (5)$$

where  $c$  is the scaling coefficient that can be adjusted to obtain the desired curve.

Various cut-off indices for the linear and quadratic functions, and different coefficients for the logarithmic function were tested on  $S_{1-3}$  blocks, and their NRMSE on  $10^{-3}$  scale is presented in Table 1.

**Table 1.** NRMSE and compression ratio improvement (CR+) with different quantization functions and parameters for three different data block types

Block	$S_1$		$S_2$		$S_3$	
	NRMSE ( $10^{-3}$ )	CR+	NRMSE ( $10^{-3}$ )	CR+	NRMSE ( $10^{-3}$ )	CR+
<i>Cut-off</i>	<b>Linear</b>					
$N = 512$	0.0864	28.1%	0.2261	32.1%	4.288	33.8%
$N = 384$	1.083	42.5%	3.542	47.3%	65.11	48.8%
$N = 256$	2.893	60.3%	10.10	63.8%	154.1	64.8%
$N = 128$	7.635	58.6%	33x.56	62.1%	418.8	63.5%
<i>Cut-off</i>	<b>Quadratic</b>					
$N = 512$	0.2553	40.7%	0.8633	47.1%	14.01	49.6%
$N = 384$	0.5121	47.5%	1.892	55.6%	26.95	58.2%
$N = 256$	0.7868	39.8%	3.416	46.3%	49.37	48.7%
$N = 128$	3.883	57.6%	15.45	62.5%	189.77	64.6%
<i>Coeff.</i>	<b>Logarithmic</b>					
$c = 3.6$	0.4549	57.7%	1.699	66.3%	23.81	70.0%
$c = 4.5$	2.291	69.0%	8.194	79.7%	93.30	83.1%
$c = 5.4$	5.030	72.4%	16.55	84.0%	125.8	87.2%
$c = 6.3$	8.123	74.0%	23.97	85.7%	179.9	88.8%

## 2.4 Compression

Table 1 also presents the compression ratio improvement (CR+) compared to the original (losslessly compressed) block. With these results it was concluded, that the logarithmic function has the most desirable properties for quantization: while NRMSE is generally low with every function, the logarithmic function produces higher compression ratios. The later experiments were conducted with logarithmic quantization, with coefficient  $c = 4.5$ .

In general, all the data were compressed with `gzip` level 6 parameter. The encoded data was stored in the HDF5 format[11] with data chunks of 327,680 blocks and `gzip` compression. To improve performance on blocks with many zeros, each lossily compressed block was compared to its losslessly compressed counterpart (no DCT, no reordering, no quantization). In case the lossless compression outperformed the lossy block, the original block was used, and the binary mask of losslessly compressed blocks was saved within the HDF5 file.

All steps described here were implemented in C++17 with GNU GCC compiler into a single threaded application. Tests were run on an Intel i7-8700 desktop PC with Ubuntu 18.04.

### 3 Results

To evaluate the performance of our lossy compression method, we compressed 72 MREG brain recordings of 5 minutes (2961 frames at sampling rate of 10 Hz) at spatial resolution of  $61 \times 73 \times 61$  voxels per frame. The sample data was registered frame-wise and subject-wise to  $3mm \times 3mm \times 3mm$  standard MNI space. This is an excellent use case for lossy compression, since data had to be interpolated already, lossy transformations were applied already earlier during data preprocessing (e.g. registration).

#### 3.1 Coding Speed with Different Quantization Levels

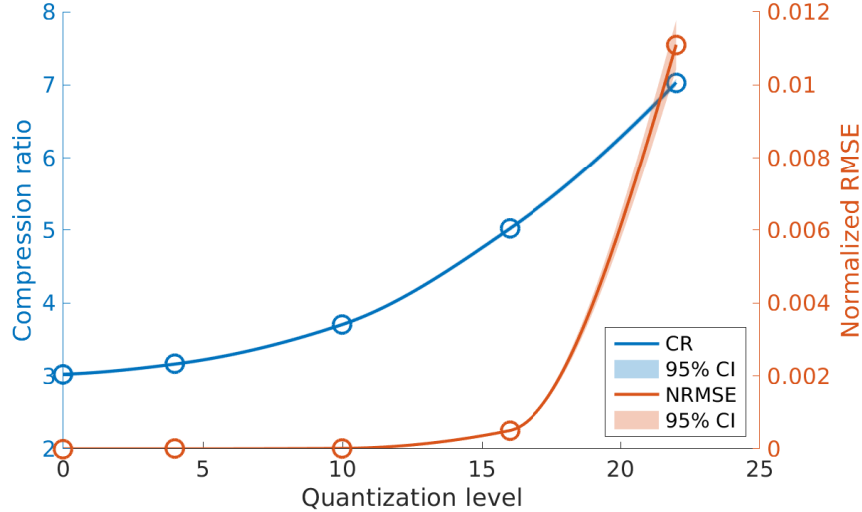
We have measured encoding and decoding speeds with different quantization levels, which are presented in Table 2. This includes storage read and write times, together with the HDF5 overhead. The quantization level is the parameter  $B_{max}$  of equation 5.

**Table 2.** Encoding and decoding speed with different quantization levels

	q=4	q=10	q=16	q=22
<i>Encode (<math>10^6</math> voxel/s)</i>	3.01	2.68	29.19	2.45
<i>95% conf. range</i>	$\pm 0.04$	$\pm 0.06$	$\pm 3.35$	$\pm 0.02$
<i>Decode (<math>10^6</math> voxel/s)</i>	35.72	29.19	36.06	37.78
<i>95% conf. range</i>	$\pm 3.77$	$\pm 3.35$	$\pm 3.75$	$\pm 3.24$

#### 3.2 Compression Ratio and NRMSE

Figure 2 presents the NRMSE values against achieved compression ratios for the selected quantization levels 4, 10, 16, and 22. Quantization level 0 denotes lossless compression. 95% confidence intervals are also plotted in the background.



**Fig. 2.** Compression ratios and NRMSE values with lossless compression ( $q = 0$ ) and quantization levels 4, 10, 16, 22.

### 3.3 Coding and Storage Speed

Hard disks were used as storage in the previous experiments. We realized, that I/O operation (disk read and write) were the slowest components of encoding and decoding. Therefore, using the samples from section 2.3, we evaluated I/O operation speeds of encoding and decoding with fast non-volatile memory express (NVMe) solid state (SSD) storage as well. The speed of reading (from disk), encoding (in memory), decoding (in memory), and writing (to disk) operations are presented in Table 3 in million ( $10^6$ ) voxel/second.

**Table 3.** Speed of coding parts using fast NVMe SSD storage

	Read	Encode	Decode	Write
<i>Average speed (<math>10^6</math> voxel/s)</i>	56.68	73.35	169.03	118.21
<i>95% confidence range</i>	$\pm 5.04$	$\pm 6.13$	$\pm 15.82$	$\pm 25.46$

The table clearly shows, that the read and write I/O operations are slower than encoding and decoding. Furthermore, encoding and decoding can be conveniently parallelized, since compression chunks (327,680 block units) are completely independent computationally. This means, that it is feasible for our algorithm to be used on-the-fly, keeping the storage operations as speed limiting factors. Also note, that times spent on storage operations are reduced linearly with higher compression rates: there is simply less data to read or write.

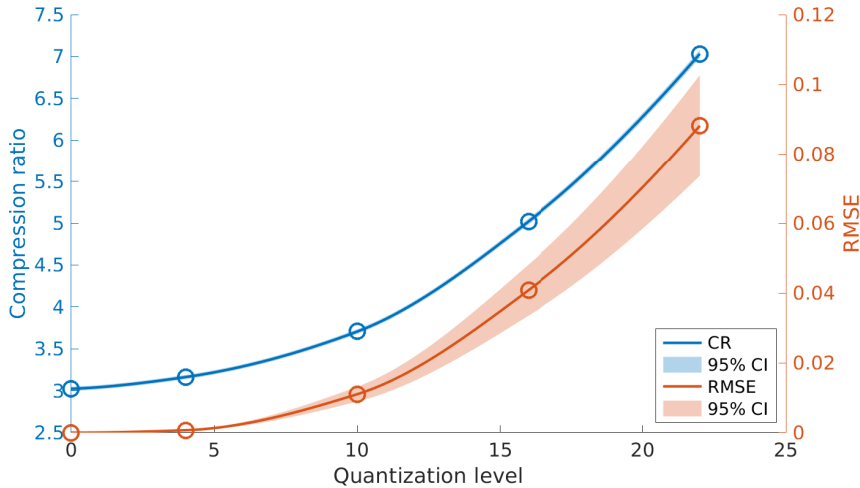
### 3.4 Effect on Functional Analysis

We chose a simple example to test feasibility in functional analysis. Seed-based correlation analysis shows the areas of the brain correlating with a seed region of interest (ROI). We chose the posterior cingulate cortex (PCC) ROI, according to the Harvard-Oxford cortical atlas[15] with a 50% probability threshold in the standard MNI brain. The analysis was performed with FSL[16] dual regression[17]. Results were compared to original data with different quantization parameters (c.f. Table 4).

**Table 4.** RMSE of seed-based correlation analysis results

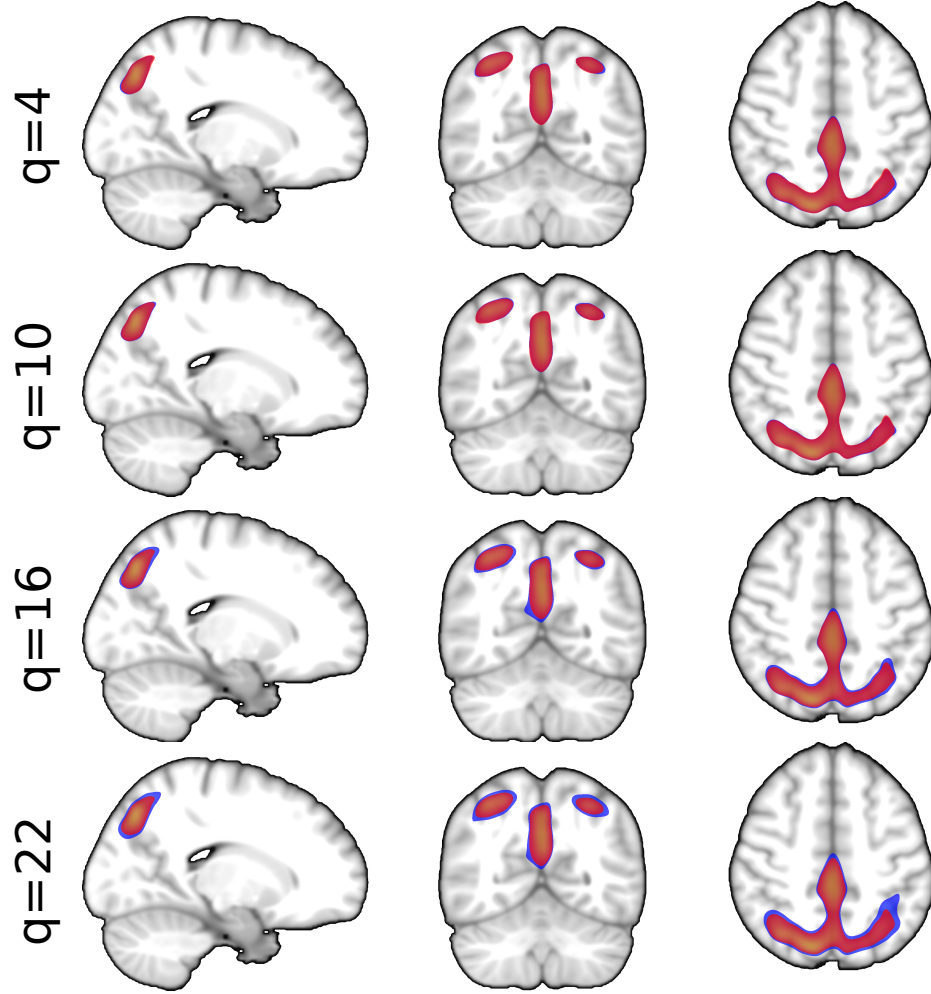
	<b>q=4</b>	<b>q=10</b>	<b>q=16</b>	<b>q=22</b>
<i>RMSE (<math>10^{-3}</math>)</i>	0.6916	11.01	40.84	88.15
<i>95% conf. range (<math>10^{-3}</math>)</i>	$\pm 0.16$	$\pm 2.122$	$\pm 7.317$	$\pm 14.54$

Since the results of this analysis are zero-centered, we used RMSE without normalization. The RMSE values together with compression ratios are presented in Figure 3 for comparison. Figure 4 presents the group averages of the analysis result, the areas exceeding threshold of 1.0.



**Fig. 3.** Compression ratios against RMSE of seed-based correlation analysis results with lossless compression ( $q = 0$ ), and quantization levels 4, 10, 16, 22.

Figure 4 demonstrates, that there are some differences introduced with lossy compression, but it minimally affects the shape and extent of the results. Data



**Fig. 4.** Group average of seed-based correlation analysis result maps of orthogonal planes intersecting MNI point (18, -60, 48). Lossless results are shown in red color with threshold 1.0, and results from lossily compressed data with quantization levels 4, 10, 16, 22 similarly in blue color.



are overlapping nearly exactly, and there is no structural difference. If higher accuracy is necessary, lower quantization levels can be applied by the user.

## 4 Discussion

We have shown that compression ratio of fMRI data can be improved considerably with lossy compression, while it does not affect functional analysis substantially. Higher compression ratios enable lower storage space and higher processing speeds. Therefore, lossy compression is a valuable option, especially in research with large number of recordings. Furthermore, we have demonstrated, that encoding and decoding can be performed on-the-fly, since they are always shorter tasks than reading and writing data to disk, even on high performance NVMe SSD storage.

The trade-off between accuracy and compression ratio can be balanced with the quantization parameter to suite user needs. Additionally, `gzip` compression levels can be adjusted to trade-off between storage size and processing speeds. The HDF5 file format is also extendable with other compression protocols, but `gzip` turned out to be superior for floating point data against `szip`. HDF5 also provides compatibility with many existing software tools, and random access (seek) times can be adjusted with number of blocks included in a data chunk.

This work did not present data on a large variety of fMRI sequences, and many parameters can be fine-tuned with further measurements in the future. We did not intend to make an exhaustive presentation of compression options or choosing the best lossy compression method. Instead, we would like to encourage the research community to develop and standardize an fMRI data compression protocol, because it would have many benefits in fMRI research and the implementation can be as simple as in our example. We only wrote 3600 lines of code which included multiple tests and prototypes.

## Statements

**Conflict of Interest** The authors declare that they have no conflict of interest.

**Human Subjects** The study protocol was approved by the ethics committee of the Northern Ostrobothnia Hospital District. Written informed consent was obtained from each subject individually prior to scanning, in accordance with the Helsinki declaration.

## References

1. Neuroimaging Informatics Technology Initiative (2015), NIFTI-1 data format. Retrieved January 24, 2020, from <https://nifti.nimh.nih.gov/nifti-1>

2. Kassim, A., Yan, P., Lee, W. S.: Motion compensated lossy-to-lossless compression of 4-D medical images using integer wavelet transforms. *IEEE Trans. on Information Technology in Biomedicine*, vol. 9, no. 1, pp. 132-138 (2005). doi:10.1109/TITB.2004.838376
3. Ying, L., Pearlman, W. A.: Four-dimensional wavelet compression of 4-D medical images using scalable 4D-SBHP. *Data Compression Conference*, pp. 233-242 (2007). doi:10.1109/DCC.2007.39
4. Rajeswari, R., Rajesh, R.: On the Efficient Compression of fMRI Data Series of Brain. *The Neuroradiology Journal*, vol. 21, no. 6, pp. 737-743 (2008). doi:10.1177/197140090802100601
5. Sanchez, V., Nasiopoulos, P., Abugharbieh, R.: Efficient 4D Motion Compensated Lossless Compression of Dynamic Volumetric Medical Image Data. *International Conference of Acoustics, Speech and Signal Processing (ICASSP)*, pp. 549-552 (2008). doi:10.1109/ICASSP.2008.4517668
6. Sanchez, V., Abugharbieh, R., Nasiopoulos, P.: Symmetry-Based Scalable Lossless Compression of 3D Medical Image Data. *IEEE Trans. on Medical Imaging*, vol. 28, no. 7, pp. 1062-1072 (2009). doi:10.1109/TMI.2009.2012899
7. Bruylants, T., Munteanu, A., Schelkens P.: Wavelet based volumetric medical image compression. *Image Commun.* vol. 31, no. C, pp. 112-133 (2015). doi: 10.1016/j.image.2014.12.007
8. Assländer, J., Zahneisen, B., Hugger, T., Reisert, M., Lee, H. L., LeVan, P., and Hennig, J.: Single shot whole brain imaging using spherical stack of spirals trajectories. *Neuroimage*, vol. 73, pp. 59-70 (2013). 10.1016/j.neuroimage.2013.01.065
9. McCarthy, P.: Indexed gzip. Retrieved January 24, 2020, from [https://github.com/pauldmccarthy/indexed\\_gzip](https://github.com/pauldmccarthy/indexed_gzip)
10. Rajna, Z., Keskinarkaus, A., Kiviniemi, V., Seppänen, T.: Speeding up the file access of large compressed NIFTI neuroimaging data. *37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 654-657(2015). doi:10.1109/EMBC.2015.7318447
11. Folk, M., Heber, G., Koziol, Q., Pourmal, E., Robinson, D.: An overview of the HDF5 technology suite and its applications. In *Proceedings of the EDBT/ICDT 2011 Workshop on Array Databases*, pp. 36-47 (2011). doi:10.1145/1966895.1966900
12. Ahmed, N., Natarajan, T., Rao, K. R.: Discrete cosine transform. *IEEE transactions on Computers*, vol. 100, no. 1, pp. 90-93 (1974).
13. Frigo, M., Johnson, S. G.: FFTW: An adaptive software architecture for the FFT. *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 3, pp. 1381-1384 (1998). doi:10.1109/ICASSP.1998.681704
14. Wallace, G. K.: The JPEG still picture compression standard. *IEEE transactions on consumer electronics*, vol. 38, no. 1, pp. xviii-xxxiv (1992). doi:10.1109/30.125072
15. Desikan, R. S., Ségonne, F., Fischl, B., Quinn, B. T., Dickerson, B. C., Blacker, D., Buckner, R. L., Dale, A. M., Maguire, R. P., Hyman, B. T., Albert, M. S., Killiany, R. J.: An automated labeling system for subdividing the human cerebral cortex on MRI scans into gyral based regions of interest. *NeuroImage*, vol. 31, no. 3, pp. 968-980 (2006). doi:10.1016/j.neuroimage.2006.01.021
16. Jenkinson, M., Beckmann, C. F., Behrens, T. E., Woolrich, M. W., Smith, S. M.: Fsl. *Neuroimage*, vol. 62, no. 2, pp. 782-790 (2012). doi:10.1016/j.neuroimage.2011.09.015

17. Beckmann, C. F., Mackay, C. E., Filippini, N., Smith, S. M.: Group comparison of resting-state FMRI data using multi-subject ICA and dual regression. *Neuroimage*, vol. 47, no. Suppl. 1, p. S148 (2009). doi:10.1016/S1053-8119(09)71511-3