Neuroconductor: An R Platform for Medical Imaging Analysis

John Muschelli[*,a], Adrian Gherman[a], Jean-Philippe Fortin[b], Brian Avants[b], Brandon Whitcher[c,d], Jonathan D. Clayden[e], Brian S. Caffo[a], Ciprian M. Crainiceanu[a]

[a] *Johns Hopkins Bloomberg School of Public Health, Department of Biostatistics, 615 N Wolfe St, Baltimore, MD, 21205*

[b] *Perelman School of Medicine, University of Pennsylvania, Department of Biostatistics and Epidemiology, 423 Guardian Drive, Philadelphia, PA 19104*

[c] *Klarismo Ltd, London, UK*

[d] *Department of Mathematics, Imperial College London, London, UK*

[e] *Developmental Imaging and Biophysics Section, UCL Great Ormond Street Institute of Child Health, University College London, 30 Guilford Street, London WC1N 1EH, United Kingdom.*

Abstract

Neuroconductor (https://neuroconductor.org) is an open-source platform for rapid testing and dissemination of reproducible computational imaging software. The goals of the project are to: 1) provide a centralized repository of R software dedicated to image analysis, 2) disseminate software updates quickly, 3) train a large, diverse community of scientists using detailed tutorials and short courses, 4) increase software quality via automatic and manual quality controls, and 5) promote reproducibility of image data analysis. Based on the programming language R (https://www.r-project.org/), Neuroconductor starts with 51 inter-operable packages that cover multiple areas of imaging including visualization, data processing and storage, and statistical inference. Neuroconductor accepts new R package submissions, which are subject to a formal review and continuous automated testing. We provide a description of the purpose of Neuroconductor and the user and developer experience.

[*] Corresponding Author

Email addresses: `jmusche1@jhu.edu` (John Muschelli), `adig@jhu.edu` (Adrian Gherman), `fortin946@gmail.com` (Jean-Philippe Fortin), `stnava@gmail.com` (Brian Avants), `bwhitcher@gmail.com` (Brandon Whitcher), `j.clayden@ucl.ac.uk` (Jonathan D. Clayden), `bcaffo@jhsph.edu` (Brian S. Caffo), `ccraini1@jhu.edu` (Ciprian M. Crainiceanu)

Introduction

Medical imaging analysis software is heterogeneous, complex, and difficult to use in fully reproducible analysis pipelines. These problems have been accentuated by the diversity of new imaging datasets and associated scientific problems. Indeed, many studies now collect data on thousands of subjects, at multiple visits, and using different modalities. Storing, understanding, and analyzing such data is daunting. Neuroconductor provides the infrastructure for using, improving and designing open-source, scripted software that depends on a minimum number of software platforms and is dedicated to improving the correctness, reproducibility, and speed of medical image data analysis. To achieve this, Neuroconductor interweaves pre- and post-processing image analysis and provides integrated data-analytic approaches.

Neuroconductor provides data, methods, and software packages designed to support the analysis of populations of images in R [55], a programming language with state-of-the-art statistical analysis tools and a vectorized data paradigm that is well suited for neuroimaging [65]. Neuroconductor supports many types of imaging data including magnetic resonance imaging (MRI: structural, functional, and dynamic), computed tomography (CT), single-photon and positron emission computed tomography, (SPECT and PET), electroencephalography (EEG), and magnetoencephalography (MEG). It is currently focused on human imaging, especially of the brain, but it also supports other biological imaging, such as the lungs and MR spectroscopy. Neuroconductor is able to interface with mature R distribution platforms, such as Bioconductor [27, 32] and the comprehensive R archive network (CRAN) [31, 55], as well as other R-based imaging projects such as TractoR [15]. Although these other distribution platforms are well-established, we believe that Neuroconductor enables better support and testing for imaging-specific packages, and provides a focal point for imaging-specific training materials, similar to how Bioconductor supports bioinformatics packages.

NiPype (http://nipy.org/packages/nipype/index.html) is a Python project with a goal of standardizing neuroimaging software syntax, similar to Neuroconductor. NiPype is incorporated in the larger NiPy project (http://nipy.org/), which provides tools for neuroimaging analysis in Python on multiple neuroimaging modalities. We believe Neuroconductor is in many ways the R analog to NiPy, with the hopes of including additional imaging problems and platforms. Although the utility of each of these projects may be similar, we do not believe Neuroconductor competes with NiPy, but allows R users similar capabilities as Python users. Moreover, with R-Python interfaces such as *rPython* [12], *PythonInR* [58], and *reticulate* [3] in R, and the *rpy2* (https://pypi.python.org/pypi/rpy2) and *pyRserve* (https://pypi.python.org/pypi/pyRserve/) modules in Python, it is possible for users can combine these two efforts.

Neuroconductor may be used seamlessly for data fusion analyses where imaging, genomics, and other types of high-throughput data types are analyzed together with traditional measurements and health outcomes. In R, these analyses are integrated with well-designed and thoroughly tested analytic packages for visualization, statistical inference, longitudinal and survival analysis, regression, network analysis, and machine learning.

The user perspective

Neuroconductor users interact with the platform via the Neuroconductor website (https://neuroconductor.org). Users may install an R package, explore packages, or identify a workflow designed for their specific problem. If the workflow does not exist in Neuroconductor, then users are able to create and submit their own. To download a package a user needs to know the package name, which is obtained from the list of packages on the Neuroconductor webpage (https://neuroconductor.org/list-packages/all). For example, to download the *fslr* and *neurohcp* packages in R, the command line instructions are:

```r
source("https://neuroconductor.org/neurocLite.R")
neuro_install(c("fslr","neurohcp"))
```

Many Neuroconductor packages contain help manuals and documentation as vignettes, which developers are strongly encouraged to provide. Large areas of research covering multiple package combinations are described under Help. Some of these areas contain tutorials and Massive Open Online Courses (MOOCs) that provide a rapid introduction to more complex concepts and workflows. Examples of such courses are Principles of fMRI (https://www.coursera.org/

learn/functional-mri) and Introduction to Neurohacking in R (https://www.coursera.org/learn/neurohacking). If the user does not find what they are looking for, they can request a tutorial on a specific topic and developers can also submit their own tutorial or course to Neuroconductor. Furthermore, the user can turn to the Support forum (https://neuroconductor.org/forum), can attend short courses, take free online MOOCs, or read tutorials (https://neuroconductor.org/neuroc-help). All package developers are users and some users become package developers.

The user may have their own data, use a data package from Neuroconductor, or use a Neuroconductor package to download data from an internet repository. After obtaining data, a first step is to manipulate, visualize, and ensure data quality. Most quality control procedures can be performed at the image or subject level, though the distribution of quality metrics in the population can be used as well; see Mejia et al. [42] for an example. Once data quality has been assessed, an analytic database of subjects for subsequent analyses may be created.

A major advantage of the R environment is that analyses can take advantage of a large collection of well-tested, state-of-the-art statistical packages. For example, incorporating demographic information, regression, mixed-effects models, or survival analysis is straightforward in R. Moreover, the *voxel* package implements most of these modeling methods at the voxel level directly on images [26]. Results containing spatial information are mapped back to an image, while other results may be displayed using plotting packages such as *ggplot2* [76]. All of these steps can be wrapped in a series of reproducible reports using *knitr* and *rmarkdown* [2, 78]. Another advantage of R is that complex processing steps and complete analyses can also be wrapped into R packages. For example, the *ichseg* [44, 49] R package is designed for intra-cranial hemorrhage segmentation and uses multiple functions from other Neuroconductor packages. It uses a single CT scan as input, processes the data using FSL [61], called through *fslr* in R [48], and creates a segmentation of the hemorrhagic stroke based on a pre-trained supervised machine learning algorithm. Last, data products resulting from analysis and package development may be deployed as Shiny applications [13], which are web applications built in R and accessed via a browser. Shiny applications are hosted on a user's custom launched or commercial Shiny server (such as https://www.shinyapps.io). As an example, the segmentation of the hemorrhagic stroke method described above has been implemented in a Shiny application (http://johnmuschelli.com/ich_segment.html). Below we provide several case studies to build up the intuition for interfacing with Neuroconductor.

Case study 1: Processing of structural magnetic resonance imaging

Figure 1 provides an example of a typical processing pipeline for multi-sequence structural MRI data. Analysis of such data often begins with a conversion of Digital Imaging and Communications in Medicine (DICOM) files, which are essentially binary formatted pictures with header information. DICOM is one of the most common formats for imaging data that come directly from the imaging device. DICOM files are commonly converted to the much simpler Neuroimaging Informatics Technology Initiative (NIfTI) format, which represents imaging data as a multi-dimensional array instead of a collection of single two-dimensional (2D) slices [41]. This process may be performed in `R` using the *dcm2niir* package [43], which calls `dcm2niix` (https://github.com/rordenlab/dcm2niix) from the command line, the *divest* package [14], which incorporates `dcm2niix` and provides an in-memory bridge to it, or the *oro.dicom* [75] package, which uses completely native `R` code. The resulting NIfTI files may be converted to array-based S4 objects using the *oro.nifti* [75] package or objects based on C++ pointers using the *ANTsR* [5] or *RNifti* [16] packages. These packages also provide the ability to read other medical imaging formats, such as ANALYZE, AFNI, NRRD or any other format supported by the insight toolkit (ITK) [79].

After conversion, image intensity inhomogeneity correction is applied to ensure that each tissue has similar intensity distributions across locations in the brain (e.g. top versus bottom of the brain). Image inhomogeneities are typically caused by magnetic field inhomogeneities and may be handled by multiple methods. Neuroconductor currently contains four such methods implemented in FSL via *fslr* [80], *freesurfer* [60], and *ANTsR* (N3 and N4 correction) [60, 69]. The next step is co-registration, which spatially aligns all images in the sequence to one of the images in the sequence, usually a T1-weighted (T1w) image. Co-registration may be performed using `flirt` (*fslr*), `antsRegistration` (*ANTsR*), `niftyreg` (*RNiftyReg*) [17], or `dramms` (*drammsr*) [51]. Co-registration is followed by brain extraction, also known as skull stripping. This procedure is commonly performed on the T1w image, and may be implemented using *spm12r* [53, 47], *fslr*, *freesurfer*, or *ANTsR*. At this point, the image sequences for the subject are in the same space and extracranial tissues have been removed.

Brain extraction is followed by intensity normalization, which transforms the arbitrary MRI units into interpretable units across subjects. This is not always thought of as a standard pre-processing step, but it is important in many applications. For example, one could be interested in subtracting two images that were collected longitudinally in order to identify changes or one may want to investigate changes in voxel or region-of-interest (ROI) intensities over time. This is typically achieved using z-scoring with respect to a particular tissue class as implemented in the WhiteStripe method (*WhiteStripe*) [59], standard and robust z-scoring relative to whole brain (*neurobase*) [45], histogram matching (implemented in *RAVEL*), or removal of unwanted variation (*RAVEL*) [24]. Further subject-specific and population analyses may require registration to a population-level template and/or tissue class segmentation. Both of these approaches are possible using Neuroconductor packages including *spm12r*, *fslr*, and *ANTsR*.

While the flowchart in Figure 1 provides a conceptual pre-processing pipeline, deploying an explicit and reproducible pipeline requires specific choices at every step that may depend on multiple tuning parameters. In R one can be explicit about these choices, provide a software suite of packages and tuning parameters, and quickly compare results based on different combinations of software and platforms.

Case study 2: A cross-package workflow for diffusion tensor imaging

In this section, we discuss an example of complete analysis (preprocessing and statistical analysis) performed entirely within Neuroconductor. We start with a simple question: in a population of healthy subjects, is there any difference in the white matter (WM) microstructure between males and females? Diffusion tensor imaging (DTI) has been used extensively to study WM fiber structure by taking advantage of the differential water diffusivity in the WM tracts relative to other brain structures [10, 37]. Fractional anisotropy (FA) and mean diffusivity (MD) are two scalar maps commonly derived from DTI images to study the diffusivity properties of the brain. FA measures the degree of directional diffusivity in a voxel and MD measures the total diffusion within a voxel [35].

To investigate potential gender differences in WM fiber tracts, we use DTI data from healthy young adults in the Human Connectome Project (HCP), available at http://db.humanconnectome.org. HCP includes a large cohort of individuals ($N > 1200$) with a vast amount of neuroimaging data, including structural magnetic resonance imaging (sMRI), task and resting-state functional MRI (fMRI) and diffusion tensor imaging (DTI).

Step 1: Downloading data using the neurohcp package

The first step is to download the minimally preprocessed DTI data [28] and structural T1w MRI images, available for 781 subjects (436 females and 345 males), using the *neurohcp* [70] package. The *neurohcp* package is an `R` interface for downloading data from the HCP database, which is publicly available from the Amazon Web Services (AWS). A description of how to connect to the HCP database via AWS is available at the Human Connectome Wiki. After accepting the data use terms, one needs to obtain AWS credentials, which include an access key identifier and a secret key. The *neurohcp* package in `R` accesses the data from an Amazon Simple Storage Solution (S3) bucket using these two AWS access keys. We can set these keys using `set_aws_api_key`:

```r
library(neurohcp)
set_aws_api_key(access_key = "ACCESS_KEY", secret_key = "SECRET_KEY")
```

After the access keys are set, *neurohcp* can download data from the S3 bucket. For instance, the complete diffusion data directory for subject 100307 may be downloaded using the `download_hcp_dir` function in the *neurohcp* package:

```r
result <- download_hcp_dir("HCP/100307/T1w/Diffusion")
```

The `result` is an `R` list containing the file names of all downloaded files, the directory where the files were downloaded, and the `http` request that was sent to the Amazon S3 bucket. Data from other subjects are downloaded similarly. The demographic data, which includes age and gender, is not located on the S3 bucket and must be downloaded directly from the website.

Step 2: Processing DTI data with rcamino

After the minimally processed DTI data are downloaded, processing continues using the Neuroconductor package *rcamino* [46], an `R` interface for the open-source DTI software `Camino` [18]. The package creates and fits the diffusion tensor models, and generates the FA and MD maps. We illustrate below the associated `R` code.

```r
# Extract the DTI files for the subject and name them using the NIfTI format
library(neurobase)
library(rcamino)
outfiles <- result$output_files
names(outfiles) <- neurobase::nii.stub(basename(outfiles))

# Specify the b-values and b-vectors used in the
# HCP database for further processing:
camino_fsl2scheme(bvecs = outfiles[["bvecs"]], bvals = outfiles[["bvals"]],
    outfile = "hcp.scheme")

# Convert the diffusion data from NIfTI to Camino format:
camino_image2voxel(infile = outfiles[["data"]], outfile = "dwi.Bfloat")

# Fit the diffusion tensor imaging model:
camino_modelfit(infile = "dwi.Bfloat", outfile = "dt.Bdouble",
    scheme = "hcp.scheme", gradadj = outfiles[["grad_dev"]],
    model = "ldt", mask = outfiles[["nodif_brain_mask"]])

# Produce the FA and MD maps from the fitted tensor data:
fa <- camino_fa_img(infile = "dt.Bdouble", inputmodel = "dt",
    header = outfiles[["data"]])
md <- camino_md_img(infile = "dt.Bdouble", inputmodel = "dt",
    header = outfiles[["data"]])
```

Note that Camino requires the b-values and b-vectors of the DTI to conduct the DTI model fit. The b-values are the amount of diffusion weighting used for each volume. The b-vectors are the gradients of the magnetic field that imply direction of flow; for more details, see O'Donnell and Westin [52]. The process is then repeated for all subjects with available DTI data, and may be sped up via parallel computing. Now, we have an FA and MD image for each subject.

Step 3: Nonlinear registration to template

The next step is to prepare the data for voxel-wise analysis, which requires the FA and MD maps to be spatially registered to a common template. For each subject, we use the download_hcp_file function from the *neurohcp* package to download the T1w image with extra-cranial voxels removed. We then use the symmetric diffeomorphic non-linear registration implemented in *ANTsR*, wrapped in *extrantsr* [4], to register FA and MD maps to the 1mm isotropic Eve template T1w image [50]. The Eve template is a single-subject template created by the Laboratory of Brain Anatomical MRI led by Professor Susumu Mori at Johns Hopkins University [50]. The Eve template is made available in the Neuroconductor data package *EveTemplate* [22]. Alternatively, one could register images to the MNI template [29], which is available in the Neuroconductor data package *MNITemplate* [23]. Each registered DTI map is saved as a standard NIfTI file for further analyses. The R code is presented below.

```r
# Specify the path of the Eve template file:
library(EveTemplate)
eve_template = getEvePath()

# Perform non-linear registration using SyN in ANTsR:
t1_file <- download_hcp_file("HCP/100307/T1w/T1w_acpc_dc_restore_brain.nii.gz")
reg <- extrantsr::registration(filename = t1_file,  template.file = eve_template,
    typeofTransform = "SyN", interpolator = "Linear", remove.warp = FALSE)

# Create the registered FA and MD maps to the Eve atlas:
fa_eve <- extrantsr::ants_apply_transforms(fixed = eve_template, moving = fa,
    transformlist = reg$fwdtransforms)
md_eve <- extrantsr::ants_apply_transforms(fixed = eve_template, moving = md,
    transformlist = reg$fwdtransforms)
```

The results of these processing steps are images containing the FA and MD maps registered to the Eve atlas for every subject. While at this point we do not conduct ROI analyses, the template may be used to extract subject-specific ROIs. An ROI is defined as an anatomical region, regions obtained from other analyses, or regions that are manually delineated. Here, we focus on voxel-wise analysis across subjects, which can be performed because images are registered to a common template space. Although we do not present any measures of quality control or registration accuracy, users should inspect registration and image quality using either automated methods or visual inspection.

Step 4: Statistical analysis

The next step is the analysis of the population of images using statistical inference. Neuroimaging convention refers to this step as "statistical analysis", a convention we adopt here, despite the fact that earlier steps involve a substantial amount of statistical operations.

We first read the registered NIfTI files into `R` and create a matrix of voxels intensities with voxels as rows, and subjects as columns. For the analysis, we only consider voxels in the WM and GM. We create one matrix for the FA maps, and one for the MD maps, using the function `images2matrix` from the package *neurobase*. The registered-to-Eve images for all subjects are located in the lists `files.fa` and `files.md` for FA and MD maps, respectively. For brevity, we present only the analysis for the MD maps. In this example, the matrix (`Y.md`) has $1,372,619$ rows (number of GM and WM voxels in the Eve template) and 781 columns (number of subjects in the dataset):

```
# Create the GM and WM mask in Eve template space:
mask <- readEveSeg()


# Discard voxels in the CSF
```

```
mask[mask ==1] <- 0

# All voxels in GM (label = 2) and WM (label =3) are kept
mask[mask %in% c(2,3)] <- 1

# Create the matrix for MD values at every voxel in the mask
Y.md <- images2matrix(files.md, mask=mask)
```

There are many options in R to quantify the association of the FA and MD intensities with gender. The simplest approach is to calculate mass-univariate two-sample t-statistics (t-statistics computed at each voxel separately), which can be quickly computed using *limma* [62, 63, 56], a popular R package from the Bioconductor project (https://www.bioconductor.org/). The package was originally developed for the analysis of high-throughput genomics data, but much of its functionality may be used in neuroimaging applications without additional effort. Among other methods, the *limma* package implements Empirical-Bayes (EB) methods to estimate t-statistics based on variance shrinkage (called moderated t-statistics). Below, we present the code to compute the moderated t-statistics for the association between gender and MD, adjusted for age.

```
# Fit the EB linear model with limma:
library(limma)
fit.md <- eBayes(lmFit(Y.md, model.matrix(~gender+age)))

# Get moderated t-statistics for gender:
t.md <- fit.md$t[,2]

# Store the moderated t-statistics into template space:
img.t.md <- remake_img(t.md, img=mask, mask=mask)
```

The code produces t-statistics comparing males versus females in the Eve-template space. The gender value is coded so that negative values of the t-statistic correspond to higher values of

MD in females. Below, we investigate where the largest differences are located.

Step 5: Visualization and localization of the results

A considerable advantage of the close integration between pre- and post-processing tools is that results can be easily mapped back into the native or template space. This helps localize significant associations using template labels. For example, the voxels that exhibit the largest differences between males and females are identified using the Eve white matter parcellation map (WMPM) [50], included in the package *EveTemplate*. Below, we provide the `R` code for localization of these results.

```r
# Get the labels from the WMPM
library(EveTemplate)
map <- readEveMap()
labels <- getEveMapLabels()
map_labels <- labels$text_label[match(map, as.numeric(labels$integer_label))]

# Get x, y and z coordinates of the voxels
coordinates <- getXYZ(map)

# Create a data frame containing the results and labels
results <- data.frame(t_gender=as.vector(img.t.md),
        label = as.vector(map_labels))
results <- cbind(results, coordinates)

# Remove unlabeled voxels and sort
results <- results[results$label!="background",]
results <- results[order(-abs(results$t_gender)),]

# Display the first few rows of results
head(results)
```

```
         t_gender             label   x   y  z
2298041 -15.88617 hippocampus_right  65 111 59
1984593 -14.98982  hippocampus_left 109 115 51
1984413 -14.96482  hippocampus_left 110 114 51
2298042 -14.77591 hippocampus_right  66 111 59
3043064 -14.77390     thalamus_left  92 104 78
1984774 -14.76050  hippocampus_left 109 116 51
```

A quick inspection of the top six voxels reveals that the most pronounced differences are located in the hippocampus and thalamus, where females have higher mean diffusivity compared to males. However, it may be useful to locate these findings on the template and visually study the degree of spatial clustering. The following commands produce the three panels of Figure 2, using the function `ortho2` from the *neurobase* package:

```r
# Read the Eve Template, brain only
template <- readEve("Brain")

# Set the plotting parameters
bound <- max(abs(t.md))
xyz <- c(77,114,84)
mfrow <- c(1,3)
myColors <- rev(colorRampPalette(c("blue", "grey95", "red"))(255))
ybreaks  <- seq(-bound, bound, length.out=length(myColors)+1)
colors <- c("orange", "yellow", "red", "firebrick", "blue", "deepskyblue3")

# Panel a: plot the Eve template space
ortho2(template, mfrow = mfrow, xyz = xyz)

# Panel b: plot the t-statistics for MD maps
ortho2(template, y=img.t.md, mfrow=mfrow, xyz=xyz,
    col.y = myColors, ybreaks=ybreaks, ycolorbar=TRUE)
```

```
# Focus on 6 anatomical WM regions
map_roi <- c(map)
map_roi[!map_roi %in% c(149,61, 147, 59, 115, 27)] <- 0
map_roi = plyr::mapvalues(x = map_roi,
        from = c(149, 61, 147, 59, 115, 27),
        to = 1:6)
map_roi = niftiarr(map, map_roi)

#Panel c: plot the 6 selected regions
ortho2(map, y=map_roi, mfrow=mfrow, xyz=xyz, col.y=colors)
```

Figure 2 displays the T1w Eve template in panel a), the map of t-statistics for gender differences in MD values in this population in panel b), and the annotated neuroanatomical structures of the hippocampus, thalamus, and caudate nucleus in panel c). For panel b), blue represents higher values of MD in males and red represents higher values of MD in females. Much more detailed information can be obtained and quantified from these results including percent voxels in the thalamus with a t-statistic passing a particular threshold (e.g. the Bonferroni correction) or the difference in the number of voxels with higher MD for females versus males in the right hippocampus, adjusting for age.

Advanced data analysis and visualization

An advantage of the `R` environment is that more sophisticated voxel-level analyses may be easily implemented. For example, one may want to investigate whether there are additional confounders for the association between FA and gender, or whether image intensities predict health outcomes. The `lm` and `glm` functions in `R` are designed specifically to address such questions. If images are observed longitudinally or they are used as baseline predictors in longitudinal studies, one could use the mixed effects *gee* [68] and *lme4* [11] packages. If one is interested in modeling survival time based on baseline images, then the *survival* package [67, 66] may be used. More advanced visualization of data and statistical results can be done using the packages *papayar*, which is an R wrapper around Papaya (https://www.nitrc.org/projects/papaya/), a JavaScript medical research image viewer. The *brainR* package in Neuroconductor is an R package for 3D and 4D visualization based on *rgl* [1].

Harmonization of multi-site neuroimaging data

An increasingly common strategy in neuroimaging is to combine multi-site imaging data across scanners and protocols. This approach pools results from different sources and may improve statistical power to detect small effects that would be undetectable in separate studies. However, the combination of multi-site data can introduce substantial unwanted variation in the data, due to differences in the scanner characteristics, acquisition parameters, and preprocessing pipelines. This is highly problematic when the level of technical variability is larger than the biological variation of the phenotype of interest. We refer to the process of removing the technical variation in multi-site studies as "harmonization".

The harmonization problem also exists in genomics, where data across batches often exhibit large technical variability. In genomics, this technical variability is referred to as batch effects. When there is confounding between batch and phenotype, failing to account for batch effects can lead to spurious associations [40]. Batch effects have been under intense methodological development, which produced widely-used R software packages for the removal of batch effects. Some popular examples of such packages deployed in Bioconductor are *ComBat* [33], *SVA* [38, 39], *RUV* [25], and *limma* [62, 63].

The Neuroconductor platform allows users to integrate, adapt and extend these methods to imaging studies. For example, the RUV approach was adapted to structural MRI images and was shown to successfully remove technical variation in multi-site data from the Alzheimer's Disease Neuroimaging Initiative (ADNI). The method is implemented in the Neuroconductor package *RAVEL* [24].

ANTsR: a Neuroconductor package highlight

Neuroconductor packages provide scripting access to large toolkits constructed primarily in other languages. For example, *ANTsR* [5] is a general purpose biomedical image analysis package built on top of the C++ based Insight ToolKit (ITK) [79], wrapped in the *ITKR* package, and Advanced Normalization Tools (ANTs) [6]. *ANTsR* leverages *Rcpp* [20] to wrap well-validated ANTs methods such as joint label fusion for image labeling based on a library of anatomical templates [72], Atropos for multi-channel segmentation [7], as well as core methods of computational anatomy [9]. *ANTsR* also contributes `R`-level access to ITK image iterators, multi-channel image classes and transformation objects as well as specialized methods for arterial spin labeling and BOLD-based network analysis via *igraph* interfaces [73].

ANTsR is an example of how Neuroconductor can support innovative applications of machine learning to medical imaging. *ANTsR* provides the first freely available implementation of RIP-MMARC (Rotation Invariant Patch-based Multi-Modality Analysis aRChitecture) [34], which uses dictionary learning to extract modality-specific information from a multi-modal data set. ANTsR also provides prior-constrained, sparse dimensionality reduction methods, including implementations for both eigenanatomy [19] and sparse canonical correlation analysis for neuroimaging [8]. Furthermore, *ANTsR* serves as a platform for disseminating potentially clinically useful tools such as LINDA (Lesion Identification with Neighborhood Data Analysis, https://github.com/dorianps/LINDA) [54] which implements a “convolutional” random forest for T1-based lesion segmentation. We anticipate that such downstream benefits of Neuroconductor will become more common in the future.

Because *ANTsR* leverages most of the functionality of the ITK library, this package is large on disk. It also requires additional build tools, such as CMake (https://cmake.org/), which is not allowed in other repositories, such as CRAN. However, this is a very important package and provides an example of how Neuroconductor can handle large and complex packages.

External Software Dependencies

Many pieces of software exist for medical imaging analyses that are standalone. For example, *fslr* requires a working installation of FSL [61]. The commands from FSL are called from the command line through `R` after checks on the data inputs. The thinking of this methodology is that FSL is an updated, well-maintained, and well-documented piece of software that will change with time. Without working extensively with the authors of FSL (or any other external software), porting all of the functionality would be redundant work and would need to be updated with a new release of the software. Therefore, we believe that requiring an additional system requirement (in the `SystemRequirements` field of the `DESCRIPTION` file) for user in some instances is necessary. Other packages, such as *ANTsR*, *dcm2niir*, and *rcamino*, external libraries are either bundled in the package or downloaded and installed at package building. If possible, installation at run-time or build-time is desired to reduce the necessary installation steps for the user.

Data Packages

Neuroconductor hosts data packages that allow users to test software or contain highly relevant data, such as templates. To be posted, data need to be de-identified, the author/maintainer needs to have approval to make the data public, and all user agreements must be respected. Neuroconductor also hosts packages that can access data from public repositories, while respecting the data user agreements.

Neuroconductor starts with a series of packages based on a multi-parametric neuroimaging study, which we refer to as Kirby21. Kirby21 contains data on 21 subjects scanned a day apart [36] and includes the following modalities: T1w, T2, FLAIR, proton density (PD), DTI, and fMRI. Neuroconductor also contains templates for T1w images: *EveTemplate* and *MNITemplate*; for a complete list of data packages see Table 1. Although Neuroconductor interacts with many neuroimaging data platforms, including the Neuroimaging Informatics Tools and Resources Clearinghouse (NITRC), many studies have restrictions on redistribution of data.

Medical Imaging and Neuroimaging Analyses

The analysis of medical imaging data is not restricted to a specific region or organ in the body. The acquisition of medical imaging data may occur anywhere from head to toe and the choice of anatomical coverage is driven by the clinical need. While some packages in Neuroconductor have been designed specifically to analyze neuroimaging data, all packages that manipulate file formats (DICOM, NIfTI, etc.) and provide visualization are not specific to the brain. In addition, some of the analysis packages have already been applied below the neck. For example, the *dcemriS4* package provides estimates for the key parameters in T1w DCE-MRI (dynamic contrast-enhanced MRI) experiments for perfusion imaging commonly used to assess tissue in cancer or from inflammatory processes; such as the breast or neck [57, 74]. The *oro.pet* package provides estimates of standard uptake values (SUVs) in PET experiments, where anatomical coverage may vary from the whole body (to assess the primary cancer and/or metastatic disease) to specific anatomical regions, such as the breast, liver or prostate [71].

The developer perspective

The comprehensive `R` archive network (CRAN) is the most standardized, popular, and common way to distribute `R` packages. Neuroconductor is a complementary platform dedicated to developers of `R` packages for image analysis. Neuroconductor contains extensive specific training materials and includes packages that do not integrate directly with CRAN or may require additional checks with external dependencies. Although Bioconductor is a framework that provides additional checks and has similar goals to Neuroconductor, Bioconductor packages were developed for bioinformatics. As one goal for Neuroconductor is to centralize imaging work in `R`, a separate platform is necessary. Neuroconductor is based on Git, GitHub, and continuous integration (CI) services via Travis CI and AppVeyor.

Many developers use Git, a version control system, for their projects. To distribute, host, and collaborate on projects, many use online tools, with GitHub (https://github.com/) being one of the most popular. As GitHub is an online site, distribution of a package is performed by downloading or cloning a repository. Users can easily install R packages directly from GitHub using functions from add-on R packages, most notably *devtools* [77]. Along with a detailed and revertible timeline for each package, GitHub provides a page that allows users to flag issues for developers, tag stable releases of the package, and obtain information on package activity. Along with GitHub, Travis CI (https://travis-ci.org/) and AppVeyor (https://www.appveyor.com) provide a system to check packages as they are updated. These tools provide a cloud-based infrastructure that: 1) can host and distribute packages; 2) check packages for specific requirements; 3) provide a platform for bug reporting and feature requests; and 4) check how frequently issues are addressed.

Travis CI will check packages on Linux and Mac OSX distributions. Appveyor will check packages on Windows platforms; a small percentage of packages will not be applicable for Windows machine due to intrinsic nature of the non-R components of the software. Windows 10 currently has a Linux subsystem that may be used in these few exceptions. Therefore, a package submitted to Neuroconductor does not need to pass checks for Windows to be incorporated into the platform. As some software in medical imaging has only implemented versions for *nix-based systems, we will allow users to submit Unix-only R packages, but will encourage them to refactor their code if possible to enable all of Neuroconductor to be cross-platform.

To submit a package to Neuroconductor, the author/maintainer of the package provides the GitHub link for the package. Once the package is submitted several initial checks are conducted (Figure 3). These checks ensure that the package has been created correctly. After initial checks are complete, the package must be verified by email. This verification is designed to prevent spam and allow the developer to stop a package if they would like to revise the package before re-submitting.

Once the verification is complete, the package is processed according to the workflow described in Figure 4. Overall, the package is copied/cloned to a remote server. Standardized Travis CI and Appveyor configuration files, specific to Neuroconductor, are added (see https://neuroconductor.org/package-changes for links to examples). These are to ensure that the checks performed on these services are consistent for each package. For example, any warnings when checking the package (using `R CMD check`) will be treated as an error. Some parameters of the package DESCRIPTION file are changed. These parameters ensure that when a package is downloaded from Neuroconductor, the correct versions of the dependent packages are used.

Next, the package is pushed to the central Neuroconductor GitHub (https://github.com/neuroconductor) and submitted to Travis CI and AppVeyor to be built and checked on multiple systems. Parameters are set to ensure that Travis CI and AppVeyor use the correct versions of Neuroconductor packages for checking and external dependencies are installed. The author of the package receives an automatic email indicating whether the package was built successfully and is integrated with Neuroconductor together with a description file containing pertinent information about the process. The code coverage, the percentage of the code in the package run when checked, is computed using the *covr* package [30] and the Coveralls.io platform. This coverage is displayed on the Neuroconductor package page.

Stable and Current Package Versions

We use the terminology "Stable" and "Current" to differentiate a different status of development for a Neuroconductor package. On the initial submission, after all checks are passed, the package is incorporated into Neuroconductor and deemed the Stable version. The Current version of the package is the result of nightly pulls and mirror the latest package version from the developer's GitHub repository. This provides Neuroconductor users with a way to use the latest versions of a package and at the same time it provides the Neuroconductor platform with a safe way of checking new versions of a package against the existing set of Current Neuroconductor packages. If a Current version of a package passes all the required Neuroconductor tests, we contact the developer of the package and suggest an official re-submission to Neuroconductor. If the newly re-submitted version of the package passes the checks against the Stable Neuroconductor packages, this version is incorporated to the Stable version of Neuroconductor.

The package author can ask for help from the maintainers of Neuroconductor (https://neuroconductor.org/contact-us or email *neuroconductor@gmail.com*) both for compatibility issues as well as for `R`-specific questions. All of these steps are automated after the verification has been completed.

Conclusion

Neuroconductor is a platform for developing and distributing `R` packages for medical image analysis. Neuroconductor aims to provide similar resources and content to Bioconductor, but is primarily focused on imaging. Neuroconductor can leverage all tools available in CRAN, Bioconductor, and `R` to provide complete image analytic pipelines. This framework is likely to increase the reliability and reproducibility of medical image analysis software and enable a larger number of users to perform image analyses using state-of-the-art tools.

Neuroconductor currently accepting any packages that fit within this framework, even if they are hosted on other platforms such as CRAN and Bioconductor. As the platform matures, we hope to centralize these packages in only one repository to reduce inconsistencies that may occur.

References

[1] Daniel Adler, Duncan Murdoch, and others. *rgl: 3D Visualization Using OpenGL*, 2016. URL https://CRAN.R-project.org/package=rgl. R package version 0.96.0.

[2] J Allaire, J Cheng, Yihui Xie, J McPherson, W Chang, Jeff Allen, H Wickham, and R Hyndman. rmarkdown: Dynamic documents for r. *R package version 0.5*, 2015.

[3] JJ Allaire, Kevin Ushey, Yuan Tang, and Dirk Eddelbuettel. *reticulate: R Interface to Python*, 2017. URL https://github.com/rstudio/reticulate.

[4] B. B. Avants, C. L. Epstein, M. Grossman, and J. C. Gee. Symmetric diffeomorphic image registration with cross-correlation: Evaluating automated labeling of elderly and neurodegenerative brain. 12(1):26–41. ISSN 1361-8415. doi: 10.1016/j.media.2007.06.004. URL http://www.sciencedirect.com/science/article/pii/S1361841507000606. 00548.

[5] B. B. Avants, N. J. Tustison, G. Song, P. A. Cook, A. Klein, and J. C. Gee. A reproducible evaluation of ants similarity metric performance in brain image registration. *NeuroImage*, 54(3):2033—2044, 2011. URL http://doi.org/10.1016/j.neuroimage.2010.09.025.

[6] Brian B Avants, Nicholas J Tustison, Gang Song, Philip A Cook, Arno Klein, and James C Gee. A reproducible evaluation of ants similarity metric performance in brain image registration. *Neuroimage*, 54(3):2033–2044, 2011.

[7] Brian B Avants, Nicholas J Tustison, Jue Wu, Philip A Cook, and James C Gee. An open source multivariate framework for n-tissue segmentation with evaluation on public data. *Neuroinformatics*, 9(4):381–400, 2011.

[8] Brian B Avants, David J Libon, Katya Rascovsky, Ashley Boller, Corey T McMillan, Lauren Massimo, H Branch Coslett, Anjan Chatterjee, Rachel G Gross, and Murray Grossman. Sparse canonical correlation analysis relates network-level atrophy to multivariate cognitive measures in a neurodegenerative population. *Neuroimage*, 84:698–711, 2014.

[9] Brian B Avants, Nicholas J Tustison, Michael Stauffer, Gang Song, Baohua Wu, and James C Gee. The insight toolkit image registration framework. *Frontiers in neuroinformatics*, 8:44, 2014.

[10] Peter J Basser, James Mattiello, and Denis LeBihan. Mr diffusion tensor spectroscopy and imaging. *Biophysical journal*, 66(1):259–267, 1994.

[11] Douglas Bates, Martin Mächler, Ben Bolker, and Steve Walker. Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, 67(1):1–48, 2015. doi: 10.18637/jss.v067.i01.

[12] Carlos J. Gil Bellosta. *rPython: Package Allowing R to Call Python*, 2015. URL https://CRAN.R-project.org/package=rPython. R package version 0.0-6.

[13] Winston Chang, Joe Cheng, J Allaire, Yihui Xie, and Jonathan McPherson. shiny: Web application framework for r, 2015. *URL http://CRAN. R-project. org/package= shiny. R package version 0.11.*

[14] Jonathan D Clayden and Christopher Rorden. *divest: Get images out of DICOM format quickly*, 2017. URL https://CRAN.R-project.org/package=divest. R package version 0.2.0.

[15] Jonathan D Clayden, Susana Muñoz Maniega, Amos J Storkey, Martin D King, Mark E Bastin, and Chris A Clark. TractoR: Magnetic resonance imaging and tractography with R. *Journal of Statistical Software*, 44(8):1–18, 2011.

[16] Jonathan D Clayden, Bob Cox, Mark Jenkinson, Rick Reynolds, Kate Fissell, Jean-loup Gailly, and Mark Adler. *RNifti: Fast R and C++ Access to NIfTI Images*, 2016. URL https://CRAN.R-project.org/package=RNifti. R package version 0.3.0.

[17] Jonathan D Clayden, Marc Modat, Pankaj Daga, et al. *RNiftyReg: Image Registration Using the NiftyReg Library*, 2016. URL https://CRAN.R-project.org/package=RNiftyReg. R package version 2.4.1.

[18] PA Cook, Y Bai, SKKS Nedjati-Gilani, KK Seunarine, MG Hall, GJ Parker, and DC Alexander. Camino: open-source diffusion-mri reconstruction and processing. In *14th scientific meeting of the international society for magnetic resonance in medicine*, volume 2759. Seattle WA, USA, 2006.

[19] Philip A Cook, Corey T McMillan, Brian B Avants, Jonathan E Peelle, James C Gee, and Murray Grossman. Relating brain anatomy and cognitive ability using a multivariate multimodal framework. *NeuroImage*, 99:477–486, 2014.

[20] Dirk Eddelbuettel, Romain François, J Allaire, John Chambers, Douglas Bates, and Kevin Ushey. Rcpp: Seamless R and C++ integration. *Journal of Statistical Software*, 40(8):1–18, 2011.

[21] Bruce Fischl. Freesurfer. *Neuroimage*, 62(2):774–781, 2012.

[22] Jean-Philippe Fortin, John Muschelli, and R. Taki Shinohara. *EveTemplate: JHU-MNI-ss (Eve) template*, 2016. URL https://github.com/Jfortin1/EveTemplate. R package version 0.99.14.

[23] Jean-Philippe Fortin, John Muschelli, and R. Taki Shinohara. *MNITemplate: MNI152 template*, 2016. URL https://github.com/Jfortin1/MNITemplate. R package version 0.99.4.

[24] Jean-Philippe Fortin, Elizabeth M Sweeney, John Muschelli, Ciprian M Crainiceanu, Russell T Shinohara, Alzheimer's Disease Neuroimaging Initiative, et al. Removing inter-subject technical variability in magnetic resonance imaging studies. *NeuroImage*, 132:198–212, 2016.

[25] J A Gagnon-Bartsch and T P Speed. Using control genes to correct for unwanted variation in microarray data. *Biostatistics*, 13(3):539–552, 2012. doi: 10.1093/biostatistics/kxr034.

[26] Angel Garcia de la Garza, Simon Vandekar, David Roalf, Kosha Ruparel, Ruben Gur, Raquel Gur, Theodore Satterthwaite, and R. Taki Shinohara. *voxel: Mass-Univariate Voxelwise Analysis of Medical Imaging Data*, 2016. URL https://CRAN.R-project.org/package=voxel. R package version 1.2.1.

[27] Robert C Gentleman, Vincent J Carey, Douglas M Bates, Ben Bolstad, Marcel Dettling, Sandrine Dudoit, Byron Ellis, Laurent Gautier, Yongchao Ge, Jeff Gentry, et al. Bioconductor: open software development for computational biology and bioinformatics. *Genome biology*, 5(10):1, 2004.

[28] Matthew F Glasser, Stamatios N Sotiropoulos, J Anthony Wilson, Timothy S Coalson, Bruce Fischl, Jesper L Andersson, Junqian Xu, Saad Jbabdi, Matthew Webster, Jonathan R Polimeni, et al. The minimal preprocessing pipelines for the human connectome project. *Neuroimage*, 80:105–124, 2013.

[29] Günther Grabner, Andrew L Janke, Marc M Budge, David Smith, Jens Pruessner, and D Louis Collins. Symmetric atlasing and model based segmentation: an application to the hippocampus in older adults. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 58–66. Springer, 2006.

[30] Jim Hester. *covr: Test Coverage for Packages*, 2017. URL https://CRAN.R-project.org/package=covr. R package version 3.0.0.

[31] Kurt Hornik. R FAQ, 2016. URL https://CRAN.R-project.org/doc/FAQ/R-FAQ.html.

[32] Wolfgang Huber, Vincent J Carey, Robert Gentleman, Simon Anders, Marc Carlson, Benilton S Carvalho, Hector Corrada Bravo, Sean Davis, Laurent Gatto, Thomas Girke, et al. Orchestrating high-throughput genomic analysis with bioconductor. *Nature methods*, 12(2): 115–121, 2015.

[33] W Evan Johnson, Cheng Li, and Ariel Rabinovic. Adjusting batch effects in microarray expression data using empirical bayes methods. *Biostatistics*, 8(1):118–127, 2007.

[34] Benjamin M Kandel, Danny JJ Wang, John A Detre, James C Gee, and Brian B Avants. Decomposing cerebral blood flow mri into functional and structural components: A non-local approach based on prediction. *NeuroImage*, 105:156–170, 2015.

[35] Cheng Guan Koay, Lin-Ching Chang, John D Carew, Carlo Pierpaoli, and Peter J Basser. A unifying theoretical and algorithmic framework for least squares methods of estimation in diffusion tensor imaging. *Journal of Magnetic Resonance*, 182(1):115–125, 2006.

[36] Bennett A Landman, Alan J Huang, Aliya Gifford, Deepti S Vikram, Issel Anne L Lim, Jonathan AD Farrell, John A Bogovic, Jun Hua, Min Chen, Samson Jarso, et al. Multi-parametric neuroimaging reproducibility: a 3-t resource study. *Neuroimage*, 54(4):2854–2866, 2011.

[37] Denis Le Bihan, Jean-François Mangin, Cyril Poupon, Chris A Clark, Sabina Pappata, Nicolas Molko, and Hughes Chabriat. Diffusion tensor imaging: concepts and applications. *Journal of magnetic resonance imaging*, 13(4):534–546, 2001.

[38] Jeffrey T Leek and John D Storey. Capturing heterogeneity in gene expression studies by surrogate variable analysis. *PLoS Genetics*, 3(9):1724–1735, 2007. doi: 10.1371/journal.pgen.0030161.

[39] Jeffrey T Leek and John D Storey. A general framework for multiple testing dependence. *Proceedings of the National Academy of Sciences*, 105(48):18718–18723, 2008. doi: 10.1073/pnas.0808709105.

[40] Jeffrey T Leek, Robert B Scharpf, Héctor Corrada Bravo, David Simcha, Benjamin Langmead, W Evan Johnson, Donald Geman, Keith Baggerly, and Rafael A Irizarry. Tackling

the widespread and critical impact of batch effects in high-throughput data. *Nature Reviews Genetics*, 11(10):733–739, 2010. doi: 10.1038/nrg2825.

[41] Xiangrui Li, Paul S Morgan, John Ashburner, Jolinda Smith, and Christopher Rorden. The first step for neuroimaging data analysis: DICOM to NIfTI conversion. *Journal of Neuroscience Methods*, 264:47–56, 2016.

[42] Amanda F Mejia, Mary Beth Nebel, Ani Eloyan, Brian Caffo, and Martin A Lindquist. Pca leverage: outlier detection for high-dimensional functional magnetic resonance imaging data. *arXiv preprint arXiv:1509.00882*, 2015.

[43] John Muschelli. *dcm2niir: Conversion of 'DICOM' to 'NIfTI' Imaging Files Through R*, 2016. URL https://www.nitrc.org/plugins/mwiki/index.php/dcm2nii:MainPage. R package version 0.3.3.

[44] John Muschelli. *ichseg: Intracerebral Hemorrhage Segmentation of X-Ray Computed Tomography (CT) Images*, 2016. R package version 0.5.1.

[45] John Muschelli. *neurobase: 'Neuroconductor' Base Package with Helper Functions for 'nifti' Objects*, 2016. R package version 1.9.1.

[46] John Muschelli. *rcamino: Port of the Camino Software*, 2016. R package version 0.3.1.

[47] John Muschelli. *spm12r: Wrapper Functions for 'SPM' (Statistical Parametric Mapping) Version 12 from the 'Wellcome' Trust Centre for 'Neuroimaging'*, 2016. R package version 2.1.

[48] John Muschelli, Elizabeth Sweeney, Martin Lindquist, and Ciprian Crainiceanu. fslr: Connecting the fsl software with r. *The R Journal*, 7(1):163–175, 2015.

[49] John Muschelli, Elizabeth M. Sweeney, Natalie L. Ullman, Paul Vespa, Daniel F. Hanley, and Ciprian M. Crainiceanu. PItcHPERFeCT: Primary intracranial hemorrhage probability estimation using random forests on CT. *NeuroImage: Clinical*, page Submitted, 2016.

[50] Kenichi Oishi, Andreia Faria, Hangyi Jiang, Xin Li, Kazi Akhter, Jiangyang Zhang, John T Hsu, Michael I Miller, Peter CM van Zijl, Marilyn Albert, et al. Atlas-based whole brain

white matter analysis using large deformation diffeomorphic metric mapping: application to normal elderly and alzheimer's disease participants. *Neuroimage*, 46(2):486–499, 2009.

[51] Yangming Ou, Aristeidis Sotiras, Nikos Paragios, and Christos Davatzikos. Dramms: Deformable registration via attribute matching and mutual-saliency weighting. *Medical image analysis*, 15(4):622–639, 2011.

[52] Lauren J O'Donnell and Carl-Fredrik Westin. An introduction to diffusion tensor image analysis. *Neurosurgery clinics of North America*, 22(2):185–196, 2011.

[53] William D Penny, Karl J Friston, John T Ashburner, Stefan J Kiebel, and Thomas E Nichols. *Statistical parametric mapping: the analysis of functional brain images*. Academic press, 2011.

[54] Dorian Pustina, H Coslett, Peter E Turkeltaub, Nicholas Tustison, Myrna F Schwartz, and Brian Avants. Automated segmentation of chronic stroke lesions using LINDA: lesion identification with neighborhood data analysis. *Human brain mapping*, 2016.

[55] R Core Team. R: A language and environment for statistical computing. 2016. URL https://www.r-project.org.

[56] Matthew E Ritchie, Belinda Phipson, Di Wu, Yifang Hu, Charity W Law, Wei Shi, and Gordon K Smyth. limma powers differential expression analyses for rna-sequencing and microarray studies. *Nucleic acids research*, page gkv007, 2015.

[57] Volker J Schmid, Brandon Whitcher, Anwar R Padhani, N Jane Taylor, and Guang-Zhong Yang. Bayesian methods for pharmacokinetic models in dynamic contrast-enhanced magnetic resonance imaging. *IEEE transactions on medical imaging*, 25(12):1627–1636, 2006.

[58] Florian Schwendinger. *PythonInR: Use Python from Within R*, 2015. URL https://CRAN.R-project.org/package=PythonInR. R package version 0.1-3.

[59] Russell T Shinohara, Elizabeth M Sweeney, Jeff Goldsmith, Navid Shiee, Farrah J Mateen, Peter A Calabresi, Samson Jarso, Dzung L Pham, Daniel S Reich, Ciprian M Crainiceanu,

et al. Statistical normalization techniques for magnetic resonance imaging. *NeuroImage: Clinical*, 6:9–19, 2014.

[60] John G. Sled, Alex P. Zijdenbos, and Alan C. Evans. A nonparametric method for automatic correction of intensity nonuniformity in MRI data. *Medical Imaging, IEEE Transactions on*, 17(1):87–97, 1998. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=668698.

[61] Stephen M Smith, Mark Jenkinson, Mark W Woolrich, Christian F Beckmann, Timothy EJ Behrens, Heidi Johansen-Berg, Peter R Bannister, Marilena De Luca, Ivana Drobnjak, David E Flitney, et al. Advances in functional and structural mr image analysis and implementation as fsl. *Neuroimage*, 23:S208–S219, 2004.

[62] Gordon K Smyth. Linear models and empirical bayes methods for assessing differential expression in microarray experiments. *Stat Appl Genet Mol Biol*, 3:Article3, 2004.

[63] Gordon K Smyth. Limma: linear models for microarray data. In *Bioinformatics and computational biology solutions using R and Bioconductor*, pages 397–420. Springer, 2005.

[64] Elizabeth M Sweeney, Russell T Shinohara, Navid Shiee, Farrah J Mateen, Avni A Chudgar, Jennifer L Cuzzocreo, Peter A Calabresi, Dzung L Pham, Daniel S Reich, and Ciprian M Crainiceanu. Oasis is automated statistical inference for segmentation, with applications to multiple sclerosis lesion segmentation in mri. *NeuroImage: clinical*, 2:402–413, 2013.

[65] Karsten Tabelow, Jonathan D Clayden, Pierre Lafaye de Micheaux, Jörg Polzehl, Volker J Schmid, and Brandon J Whitcher. Image analysis and statistical inference in neuroimaging with R. *NeuroImage*, 55(4):1686–1693, 2011.

[66] Terry M. Therneau and Patricia M. Grambsch. *Modeling Survival Data: Extending the Cox Model*. Springer, New York, 2000. ISBN 0-387-98784-3.

[67] Terry M Therneau. *A Package for Survival Analysis in S*, 2015. URL https://CRAN.R-project.org/package=survival. version 2.38.

[68] Vincent J Carey. Ported to R by Thomas Lumley and Brian Ripley. Note that maintainers are not available to give advice on using a package they did not author. *gee: General-*

ized Estimation Equation Solver, 2015. URL http://CRAN.R-project.org/package=gee. R package version 4.13-19.

[69] Nicholas J. Tustison, Brian B. Avants, Philip A. Cook, Yuanjie Zheng, Alexander Egan, Paul A. Yushkevich, and James C. Gee. N4itk: improved N3 bias correction. *Medical Imaging, IEEE Transactions on*, 29(6):1310–1320, 2010. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5445030.

[70] David C Van Essen, Stephen M Smith, Deanna M Barch, Timothy EJ Behrens, Essa Yacoub, Kamil Ugurbil, WU-Minn HCP Consortium, et al. The wu-minn human connectome project: an overview. *Neuroimage*, 80:62–79, 2013.

[71] Richard L Wahl, Heather Jacene, Yvette Kasamon, and Martin A Lodge. From RECIST to PERCIST: evolving considerations for PET response criteria in solid tumors. *Journal of nuclear medicine*, 50(Suppl 1):122S–150S, 2009.

[72] Hongzhi Wang and Paul Yushkevich. Multi-atlas segmentation with joint label fusion and corrective learning—an open source implementation. *Frontiers in neuroinformatics*, 7:27, 2013.

[73] Matthew J Weber, John A Detre, Sharon L Thompson-Schill, and Brian B Avants. Reproducibility of functional network metrics and network structure: a comparison of task-related BOLD, resting ASL with BOLD contrast, and resting cerebral blood flow. *Cognitive, Affective, & Behavioral Neuroscience*, 13(3):627–640, 2013.

[74] Brandon Whitcher, Volker J Schmid, David J Collins, Matthew R Orton, Dow-Mu Koh, Isabela Diaz de Corcuera, Marta Parera, Josep M del Campo, Martin O Leach, Kevin Harrington, et al. A Bayesian hierarchical model for DCE-MRI to evaluate treatment response in a phase II study in advanced squamous cell carcinoma of the head and neck. *Magnetic Resonance Materials in Physics, Biology and Medicine*, 24(2):85–96, 2011.

[75] Brandon Whitcher, Volker J. Schmid, and Andrew Thornton. Working with the DICOM and NIfTI data standards in R. *Journal of Statistical Software*, 44(6):1–28, 2011. URL http://www.jstatsoft.org/v44/i06/.

[76] Hadley Wickham. *ggplot2: elegant graphics for data analysis*. Springer Science & Business Media, 2009.

[77] Hadley Wickham and Winston Chang. *devtools: Tools to Make Developing R Packages Easier*. URL https://github.com/hadley/devtools. R package version 1.12.0.9000.

[78] Yihui Xie. knitr: A general-purpose package for dynamic report generation in r. *R package version*, 1(7):1, 2013.

[79] Terry S Yoo, Michael J Ackerman, William E Lorensen, Will Schroeder, Vikram Chalana, Stephen Aylward, Dimitris Metaxas, and Ross Whitaker. Engineering and algorithm design for an image processing api: a technical report on itk-the insight toolkit. *Studies in health technology and informatics*, pages 586–592, 2002.

[80] Yongyue Zhang, Michael Brady, and Stephen Smith. Segmentation of brain MR images through a hidden Markov random field model and the expectation-maximization algorithm. *Medical Imaging, IEEE Transactions on*, 20(1):45–57, 2001. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=906424.

List of Figures

Figure 1: Typical processing pipeline for multi-sequence structural MRI data.

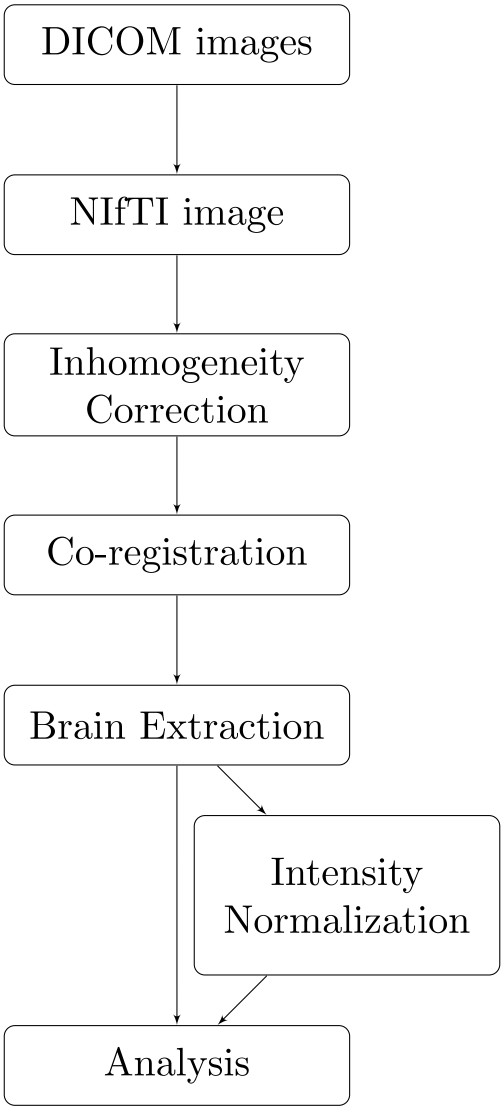


Figure 2: **Visualization of the diffusion tensor imaging (DTI) analysis results in R.** (a) Visualization of the anatomical structures in template space (Eve template, 1-mm isotropic T1-weighted modality) in coronal, sagittal and axial planes (coordinates x=77, y=114 and z=84). (b) T-statistics characterizing the differences between males and females in the mean diffusivity (MD) values for grey matter (GM) and white matter (WM) voxels, in template space. The DTI analysis was performed using data from the Human Connectome Project (HCP). Blue and red regions represent higher and lower values of MD in males, respectively. (c) Visualization of the Eve atlas white matter parcellation map (WMPM) with selected structures.

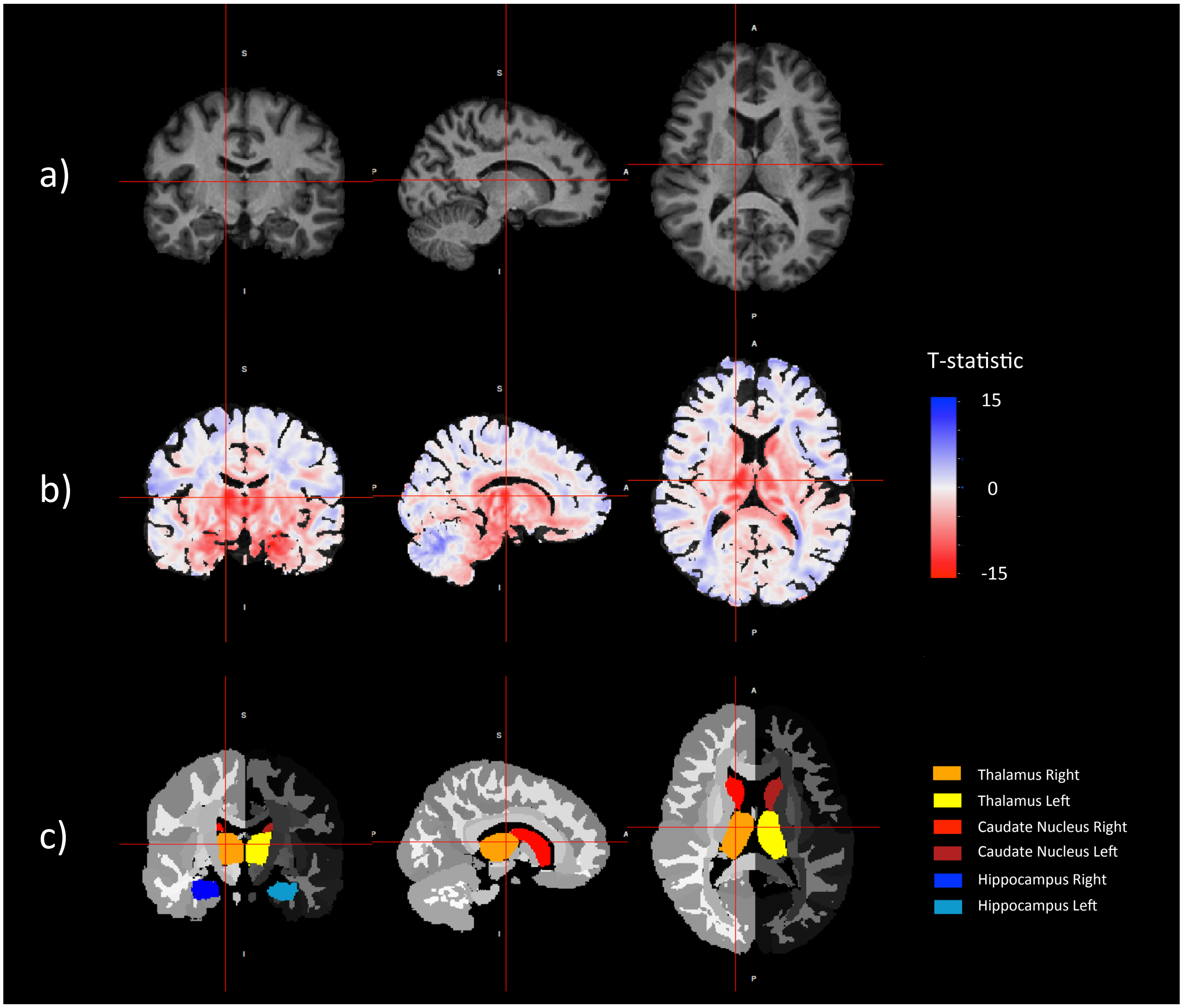


Figure 3: **Neuroconductor initial package submission.** If the DESCRIPTION file is present it will parsed and the submitted version of the package is checked against existing Neuroconductor packages. If this version is not already in Neuroconductor the email-based verification process is started. Once the maintainer verifies the submission the package is ready to be tested.

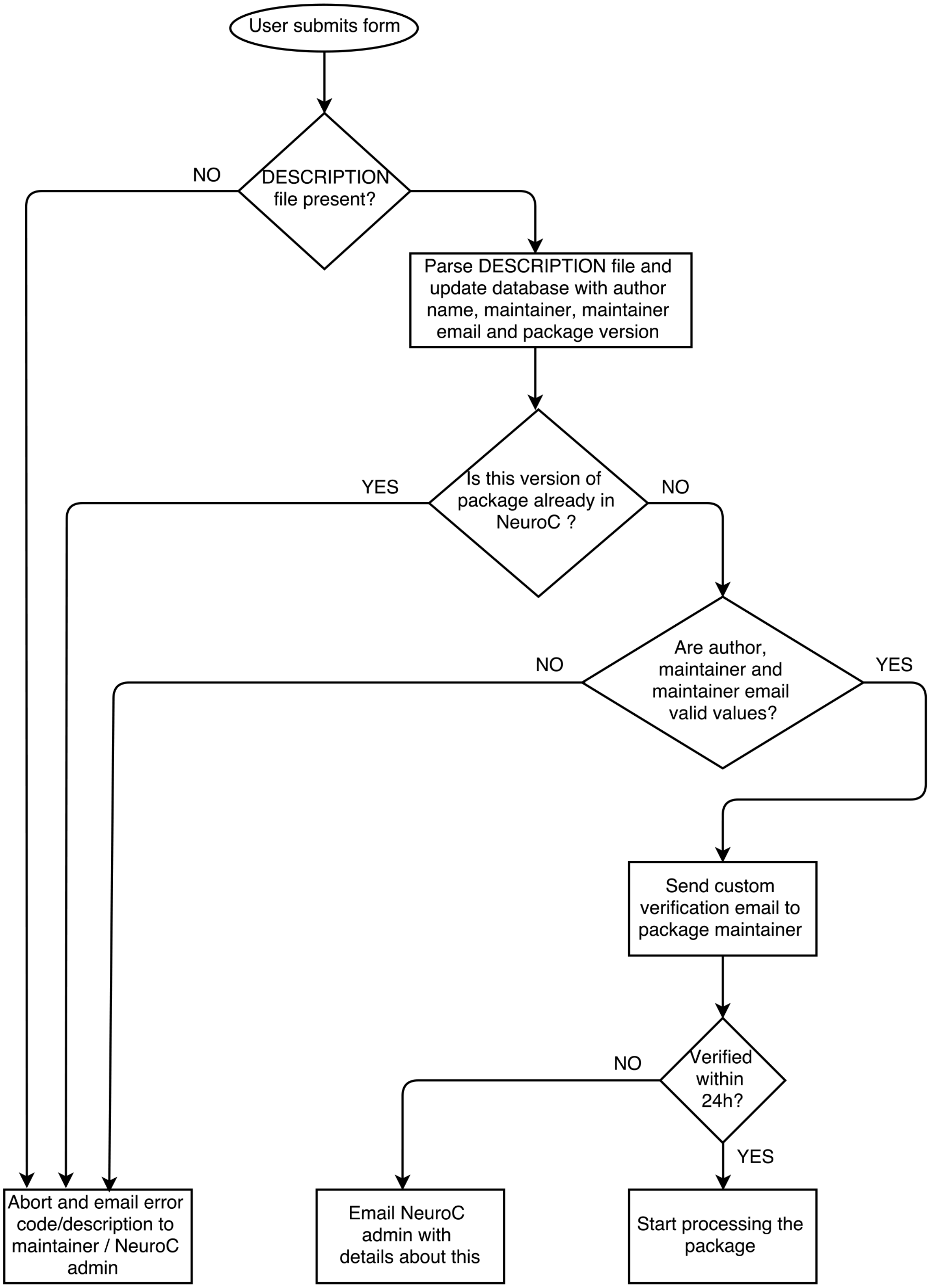
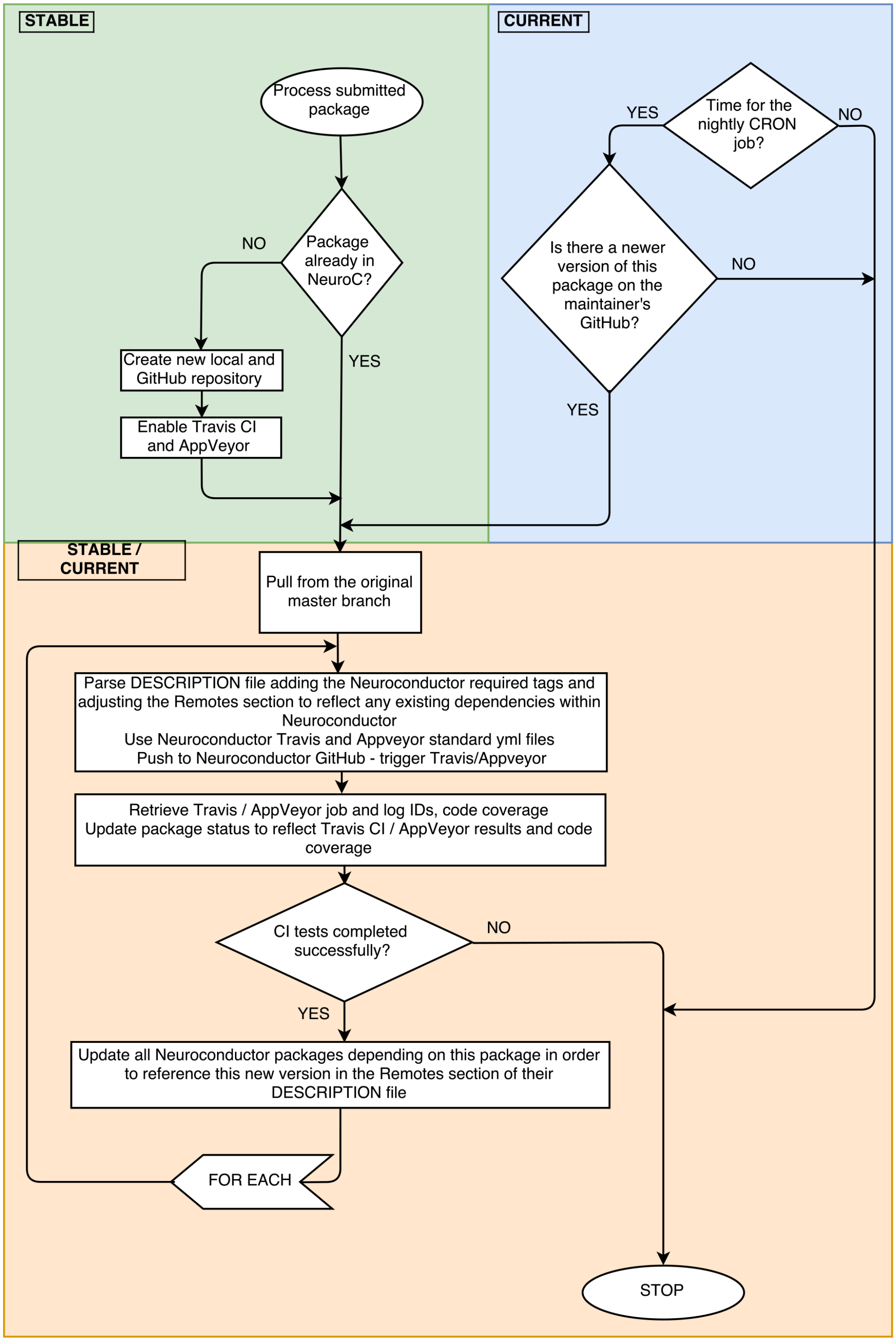

User submits form
DESCRIPTION file present?
NO
Parse DESCRIPTION file and update database with author name, maintainer, maintainer email and package version
Is this version of package already in NeuroC ?
YES
NO
Are author, maintainer and maintainer email valid values?
NO
YES
Send custom verification email to package maintainer
Verified within 24h?
NO
YES
Abort and email error code/description to maintainer / NeuroC admin
Email NeuroC admin with details about this
Start processing the package

Figure 4: **Neuroconductor package code testing.** The package is cloned/updated on the Neuroconductor server and Travis CI / AppVeyor checks are initiated for the original version of the package together with a stable and current version of this package. The DESCRIPTION, travis.yml and appveyor.yml files are updated to reflect the stable/current package status. Once the Travis CI / AppVeyor checks are done the package listing is updated to reflect the continuous integration test results.


STABLE
CURRENT
Process submitted package
Package already in NeuroC?
NO
YES
Create new local and GitHub repository
Enable Travis CI and AppVeyor
Time for the nightly CRON job?
YES
NO
Is there a newer version of this package on the maintainer's GitHub?
NO
YES
STABLE / CURRENT
Pull from the original master branch
Parse DESCRIPTION file adding the Neuroconductor required tags and adjusting the Remotes section to reflect any existing dependencies within Neuroconductor
Use Neuroconductor Travis and Appveyor standard yml files
Push to Neuroconductor GitHub - trigger Travis/Appveyor
Retrieve Travis / AppVeyor job and log IDs, code coverage
Update package status to reflect Travis CI / AppVeyor results and code coverage
CI tests completed successfully?
NO
YES
Update all Neuroconductor packages depending on this package in order to reference this new version in the Remotes section of their DESCRIPTION file
FOR EACH
STOP

List of Tables

Table 1: **Referenced R packages available on Neuroconductor**

Package	Description	References
Software packages		
neurobase	Base functions for Neuroconductor	
oro.nifti	Working with the DICOM and NIfTI Data Standards in R	[75]
oro.dicom	Working with the DICOM and NIfTI Data Standards in R	[75]
oro.asl	oro.asl: Rigorous - Aterial Spin Labelling	
oro.pet	oro.pet: Rigorous - Positron Emission Tomography	
ANTsR	Advanced Normalization Tools	[6]
extrantsr	Extensions for ANTsR	
fslr	R package for FSL	[48, 61]
freesurfer	R package for FreeSurfer	[21]
oasis	OASIS lesion segmentation	[64]
WhiteStripe	White Stripe intensity normalization	[59]
RAVEL	Statistical analysis of structural MRIs	[24]
neurohcp	R interface for the Human Connectome Project database	
Data packages		
kirby21.t1	Example T1 Structural Data from the Kirby21 Dataset	[36]
kirby21.t2	Example T2 Structural Data from the Kirby21 Dataset	[36]
kirby21.flair	Example FLAIR Structural Data from the Kirby21 Dataset	[36]
kirby21.dti	Example DTI Data from the Kirby21 Dataset	[36]
kirby21.fmri	Example fMRI Data from the Kirby21 Dataset	[36]
kirby21.mt	Example MT Structural Data from the Kirby21 Dataset	[36]
kirby21.vaso	Example VASO Structural Data from the Kirby21 Dataset	[36]
kirby21.asl	Example ASL Structural Data from the Kirby21 Dataset	[36]
Template packages		
MNITemplate	MNI152 template	[23]
EveTemplate	Eve Atlas and White Matter parcellation map	[22, 50]