```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"]])
```

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

# Perform non-linear registration usin 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 <- ants_apply_transforms(fixed = eve_template, moving = fa,
    transformlist = reg$fwdtransforms)
md_eve <- ants_apply_transforms(fixed = eve_template, moving = md,
    transformlist = reg$fwdtransforms)
```

```r
# 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)
```

```r
# 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:
```

```r
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)
```

```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
```

```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)
```