

MD-NPH analisis

Esto es una idea de Ester Esteban, así que toda la culpa es suya.

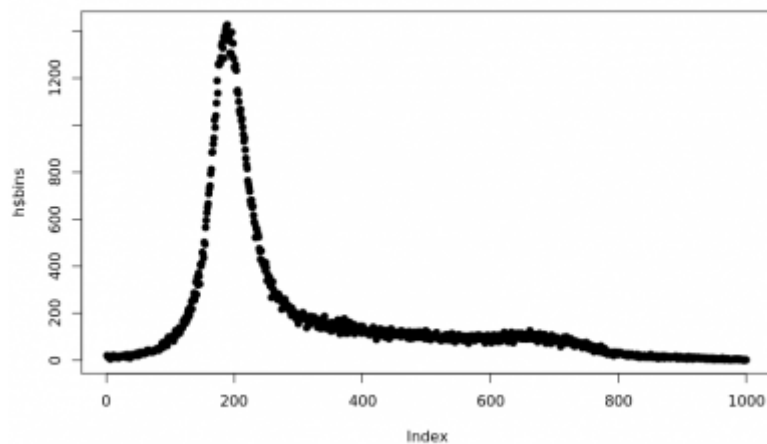
Principio

Siguiendo [Zeestraten 2017 Neurology](#) queremos determinar la altura del pico de los histogramas normalizados de MD.

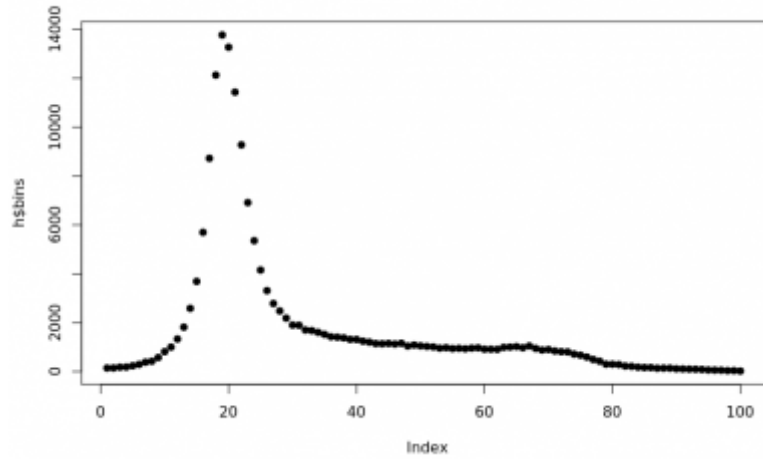
Se supone que ya tenemos la imagen MD por el preprocesamiento previo. Ahora hemos de sacar el histograma de esta imagen,

```
[osotolongo@brick03 nph]$ fslstats 0005_dti_MD.nii.gz -l 0 -u 0.004 -h 1000  
> hist.txt; sed -i "libins" hist.txt
```

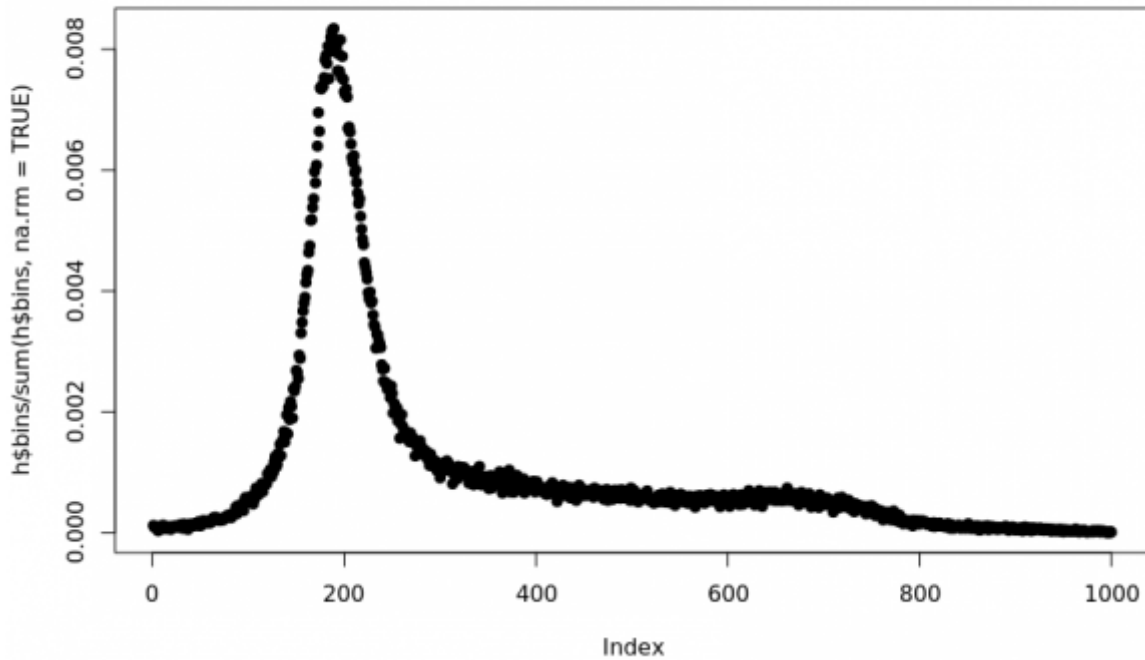
Esto produce un perfil de los valores de MD bastante claro, con un pico muy determinado.



Obviamente, si cambiamos el número de *bins*, la altura cambia,



Asi que vamos a normalizar por el area bajo la curva (A.K.A. densidad)



La altura de este pico es lo que vamos a tomar como el valor de MD-NPH,

```
> max(h$bins, na.rm = TRUE)/sum(h$bins, na.rm=TRUE)
[1] 0.08052718
```

Pero con una salvedad, no voy a normalizar por el area la curva del histograma individual sino por el valor medio de esta area entre todos los histogramas, para que todos los valores de densidad tengan un mismo nivel de referencia.

Implementación

Voy a,

1. reutilizar los componentes del pipeline
2. seguir el paper en cuanto a los valores de threshold y demas
3. usar List::Util para manipular los arrays

asi que todo deberia ser sencillo.

Ok. Pero no taaaan sencillo. 😬

Antes que nada, no voy a sacar el valor medio de toda la imagen MD sino de la porcion correspondiente a la WM. Es decir, que tengo que empezar por hacer una mascara de la WM. Es to no deberia ser dificil si reutilizo el aseg de FS,

[get_wm_mask.sh](#)

```
#!/bin/sh
subject=$1
shift

tmp_dir=$1
shift

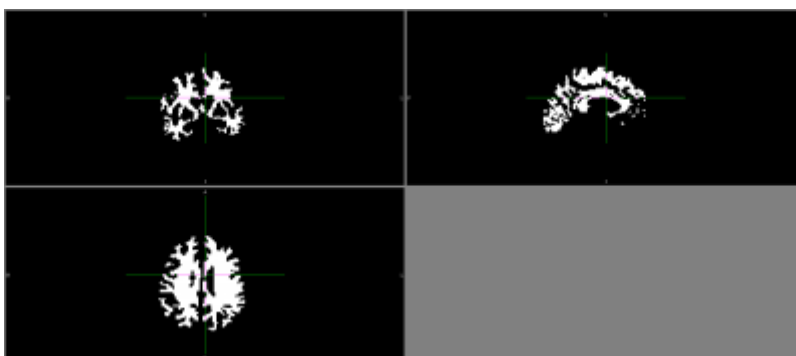
if [ ! -f ${tmp_dir}/register.dat ]; then
    if [ ! -d ${tmp_dir} ]; then mkdir -p ${tmp_dir}; fi;
    tkregister2 --mov $SUBJECTS_DIR/${subject}/mri/rawavg.mgz --
noedit --s ${subject} --regheader --reg ${tmp_dir}/register.dat;
fi
if [ ! -f ${tmp_dir}/all_aseg.nii.gz ]; then
    mri_label2vol --seg $SUBJECTS_DIR/${subject}/mri/aseg.mgz --
temp $SUBJECTS_DIR/${subject}/mri/rawavg.mgz --o
${tmp_dir}/all_aseg.nii.gz --reg ${tmp_dir}/register.dat;
fi
${FSLDIR}/bin/fslmaths ${tmp_dir}/all_aseg.nii.gz -uthr 2 -thr 2 -div
2 ${tmp_dir}/lhwm
${FSLDIR}/bin/fslmaths ${tmp_dir}/all_aseg.nii.gz -uthr 41 -thr 41 -
div 41 ${tmp_dir}/rhwm
${FSLDIR}/bin/fslmaths ${tmp_dir}/lhwm -add ${tmp_dir}/rhwm -bin
${tmp_dir}/wm_mask.nii.gz
~
```



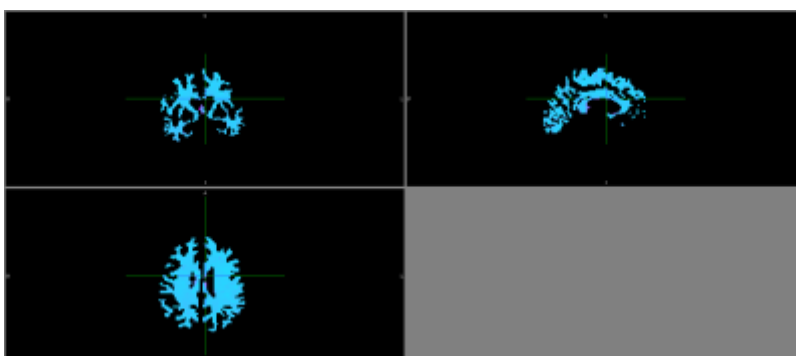
Pero ojo, que esa mascara esta en espacio nativo T1, asi que tendria que hacer algo como,

```
antsApplyTransforms -d 3 -i ${tmp_dir}/wm_mask.nii.gz -r  
${tmp_dir}/hifi_b0.nii.gz -t ${tmp_dir}/${subject_epi_reg_ANTs.mat} -n  
GenericLabel -o ${tmp_dir}/wm_tmp.nii.gz
```

La mascara resultante hay que aplicarla a la imagen MD,



y a esta ultima imagen,



es a la que hay que sacarle el histograma.

```
my $order = $ENV{'FSLDIR'}.'/bin/fslstats '.$dti_md_masked.'  
-l 0 -u '.$sup_threshold.' -h '.$nbins;  
my @shist = qx/$order/;
```

ahora, el valor para normalizar es el numero de voxels de la mascara de WM

```
$order = $ENV{'FSLDIR'}.'/bin/fslstats '.$dti_md_masked.'  
V';  
my $out = qx/$order/;
```

```
my @gout = split ' ', $oout;
my $norm = $gout[0];
```

y todo esto se guarda,

```
$nph{$subject}{'sum'} = sum(@shist);
$nph{$subject}{'norm'} = $norm;
$nph{$subject}{'max'} = max(@shist);
```

para escribirlo luego,

```
open ODF, ">$ofile";
print ODF "Subject,NPH\n";
foreach my $subject (sort keys %nph){
    my $nphv = $nph{$subject}{'max'}/$nph{$subject}{'norm'};
    print ODF "$subject,$nphv\n";
}
close ODF;
```

y el código completo online en <https://github.com/asqwerty666/acenip/blob/main/tools/nph.pl>

Uso

Esto es bastante fácil de usar,

```
[osotolongo@brick03 nph]$ ./nph.pl bioface
[osotolongo@brick03 nph]$ head ../bioface_normalised_peak_height.csv
Subject,NPH
0002,0.142459141831309
0003,0.101291664281762
0004,0.128800261523374
0005,0.131614509246088
0006,0.116625745352508
0007,0.142582104338399
0008,0.140059946569362
0009,0.129072904095975
0010,0.12497310864109
[osotolongo@brick03 nph]$ sed 's/;/,/;1iSubject,Subject_ID'
../bioface_mri.csv > bioface_codes.csv
[osotolongo@brick03 nph]$ join -t, -j 1 bioface_codes.csv
../bioface_normalised_peak_height.csv | awk -F"," '{print $2","$3}' |
csvsort -c 1 > nph_sorted.csv
[osotolongo@brick03 nph]$ head nph_sorted.csv
Subject_ID,NPH
B002,0.142459141831309
B003,0.101291664281762
B004,0.128800261523374
B005,0.131614509246088
B006,0.116625745352508
```

B007, 0.142582104338399
B008, 0.140059946569362
B009, 0.129072904095975
B010, 0.12497310864109

Nota: Para simplificar he usado *csvsort* de [csvkit](#) pero tambien se puede dar una vuelta y usar *gnu sort*.

From:

<https://imagen.fundacioace.com/wiki/> - **Detritus Wiki**

Permanent link:

https://imagen.fundacioace.com/wiki/doku.php?id=neuroimagen:md_nph

Last update: **2021/10/19 09:04**

