

MD-NPH analisis

Esto es una idea de Ester Esteban, asi que todo es culpa suya.

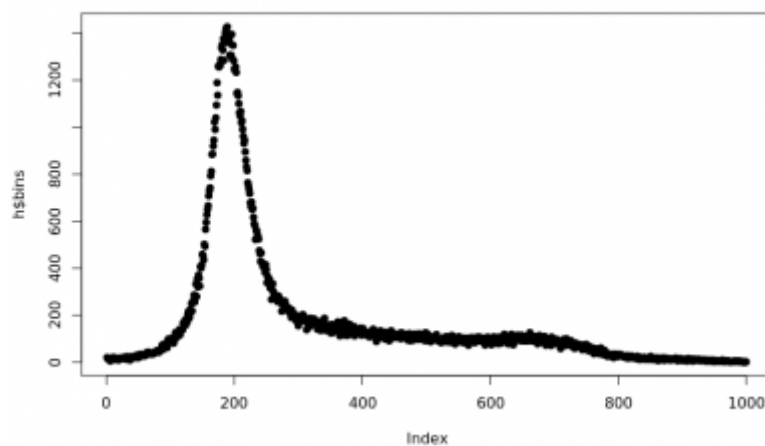
Principio

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

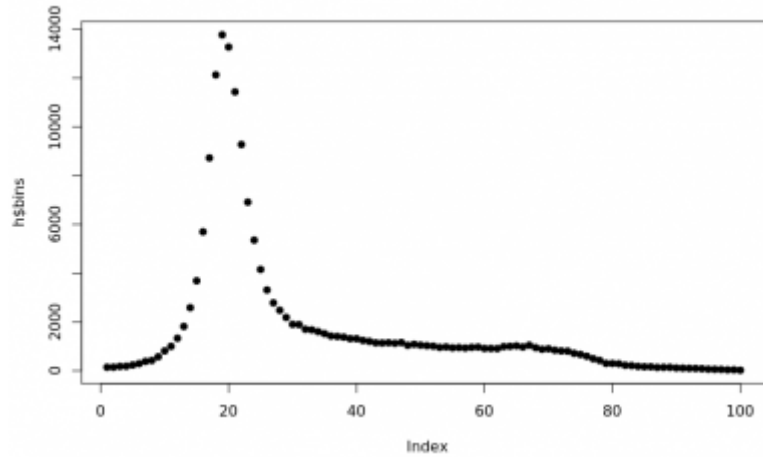
Se supone que ya tenemos la imagen MD por el preprocesamiento previo. Ahora hemos de sacar el histograma de esta imagen,

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

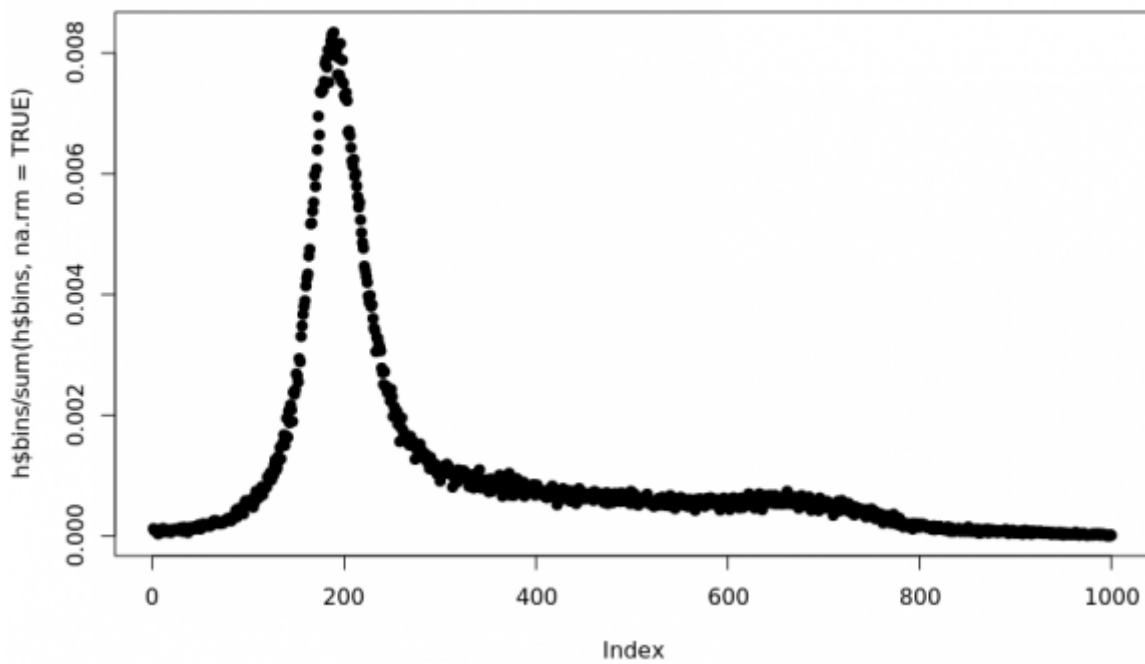
Esto produce un perfil de los valores de MD bastante claro, con un pico muy determinado.



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



Así 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.

Primero leo la DB del proyecto y por cada unos de las imagenes MD que he llegado a procesar ejecuto el procedimiento anterior,

```
foreach my $subject (@dtis){
    my $dti_md = $w_dir.'/'.$subject.'_dti_MD.nii.gz';
    if($subject and -e $dti_md){
        my $order = $ENV{'FSLDIR'}.'/bin/fsfstats '.$dti_md.' -l 0 -
u '.$sup_threshold.' -h '.$nbins;
        my @shist = qx/$order/;
        chomp @shist;
        push @allsums, sum(@shist);
        $nph{$subject} = max(@shist);
    }
}
```

ahora, el valor para normalizar es el medio de todas las areas,

```
my $norm = sum(@allsums)/scalar(@allsums);
```

y saco todo por *STDOUT*,

```
print "Subject,NPH\n";
foreach my $subject (sort keys %nph){
    my $nphv = $nph{$subject}/$norm;
    print "$subject,$nphv\n";
}
```

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

Uso

Esto es bastante facil de usar,

```
[osotolongo@brick03 nph]$ ./nph.pl bioface > nph.csv
[osotolongo@brick03 nph]$ sed 's/;/,/;1iSubject,Subject_ID'
../bioface_mri.csv > bioface_codes.csv
[osotolongo@brick03 nph]$ join -t, -j 1 bioface_codes.csv nph.csv | awk -
F", " '{print $2","$3}' | csvsort -c 1 > normalised_peak_height.csv
```

```
[osotolongo@brick03 nph]$ head normalised_peak_height.csv
Subject_ID,NPH
B002,0.0731249323687369
B003,0.0732767676009448
B004,0.0745456763272544
B005,0.0746812434988687
B006,0.0697899799470258
B007,0.0788675577583173
B008,0.0932810594443464
B009,0.0787428359604322
B010,0.0681414831401963
```

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

From:
<http://detritus.fundacioace.com/wiki/> - **Detritus Wiki**

Permanent link:
http://detritus.fundacioace.com/wiki/doku.php?id=neuroimagen:md_nph&rev=1634293796

Last update: **2021/10/15 10:29**

