# Why fixed renormalized gradient norm size = 200

Weixu Wang

2022/1/16

In benchmark analysis of maximum L2 norm model with or without gradient renormalization, we fixed the gradient norm size = 200 after performing gradient renormalization to make sure the comparison of different model would not be strongly biased by the gradient norm size. In order to show that this value is reasonable, we provide script to visualize the gradient norm size of the maximum L2 norm model without gradient renormalization.

First, load the script and R package

```
source("/mnt/data1/weixu/HiDe/revised/ENIGMA.R")
```

```
## Loading required package: sva
```

```
## Loading required package: mgcv
```

```
## Loading required package: nlme
```

```
## This is mgcv 1.8-33. For overview type 'help("mgcv-package")'.
```

```
## Loading required package: genefilter
```

```
## Loading required package: BiocParallel
```

```
## Loading required package: purrr
```

```
source("/mnt/data1/weixu/HiDe/revised/Model_Compare/L2MaxNorm/norm_size_illustrate.R")
##These scripts could be downloaded from
#https://github.com/WWXkenmo/ENIGMA/blob/main/ENIGMA_analysis/ENIGMA_Script/ENIGMA.R
#https://github.com/WWXkenmo/ENIGMA/blob/main/ENIGMA_analysis/ENIGMA_Script/norm_size_illustrate.R
library(ggplot2)
```

Then, we load the datasets and perform deconvolution

```
load("/mnt/data1/weixu/HiDe/revised/DEG_test_data_4.8")
Frac <- get_proportion(Bulk, Reference)
```

```
## Loading required package: MASS
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:genefilter':
##
##     area
```

```
## Sun Jan 16 21:23:35 2022 Calculating cell type proportion of bulk samples...
```

```
y <- gl(2, 100/2)
ENIGMA_l2max = cell_deconvolve(X=as.matrix(Bulk),
                                theta=Frac$theta,
                                R=Reference,
```

```
                                    epsilon=0.001,
                                    alpha=0.1,
                                    beta=50,
                                    tao_k=1,
                                    max.iter = 1000,
                                    verbose=TRUE,
                                    Normalize=FALSE,pos = FALSE)
```

```
## Sun Jan 16 21:23:36 2022 Optimizing cell type specific expression profile...
##      Ratio ranges from: 22497.586526 - 22513.012218
##      Ratio ranges from: 8325.482035 - 8459.817304
##      Ratio ranges from: 3370.277576 - 3473.884359
##      Ratio ranges from: 1364.267658 - 1427.576474
##      Ratio ranges from: 552.573918 - 587.573955
##      Ratio ranges from: 224.386843 - 242.819952
##      Ratio ranges from: 91.737316 - 101.380882
##      Ratio ranges from: 38.074732 - 43.308916
##      Ratio ranges from: 16.270779 - 19.333145
##      Ratio ranges from: 7.308764 - 9.265573
##      Ratio ranges from: 3.535916 - 4.877064
##      Ratio ranges from: 1.878249 - 2.833952
##      Ratio ranges from: 1.099774 - 1.790085
##      Ratio ranges from: 0.700288 - 1.198398
##      Ratio ranges from: 0.474116 - 0.830879
##      Ratio ranges from: 0.334037 - 0.587168
##      Ratio ranges from: 0.241094 - 0.423249
##      Ratio ranges from: 0.176506 - 0.312816
##      Ratio ranges from: 0.130322 - 0.231818
##      Ratio ranges from: 0.096735 - 0.172014
##      Ratio ranges from: 0.072060 - 0.127728
##      Ratio ranges from: 0.053819 - 0.094889
##      Ratio ranges from: 0.040278 - 0.070522
##      Ratio ranges from: 0.030197 - 0.052434
##      Ratio ranges from: 0.022674 - 0.039002
##      Ratio ranges from: 0.017048 - 0.029023
##      Ratio ranges from: 0.012834 - 0.021607
##      Ratio ranges from: 0.009673 - 0.016093
##      Ratio ranges from: 0.007298 - 0.011991
##      Ratio ranges from: 0.005511 - 0.008938
##      Ratio ranges from: 0.004166 - 0.006665
##      Ratio ranges from: 0.003135 - 0.004971
##      Ratio ranges from: 0.002272 - 0.003710
##      Ratio ranges from: 0.001646 - 0.002769
##      Ratio ranges from: 0.001194 - 0.002067
##      Ratio ranges from: 0.000866 - 0.001544
##      Ratio ranges from: 0.000628 - 0.001153
##      Ratio ranges from: 0.000456 - 0.000863
## Optimizing cell type proportions...
## Total delta_loss: 740834.134902, Sun Jan 16 21:23:45 2022
## part1:158048.106010954 part2:452794.433956531 part3:64.1726718309978
## Done!
## Converge in 38 steps
```
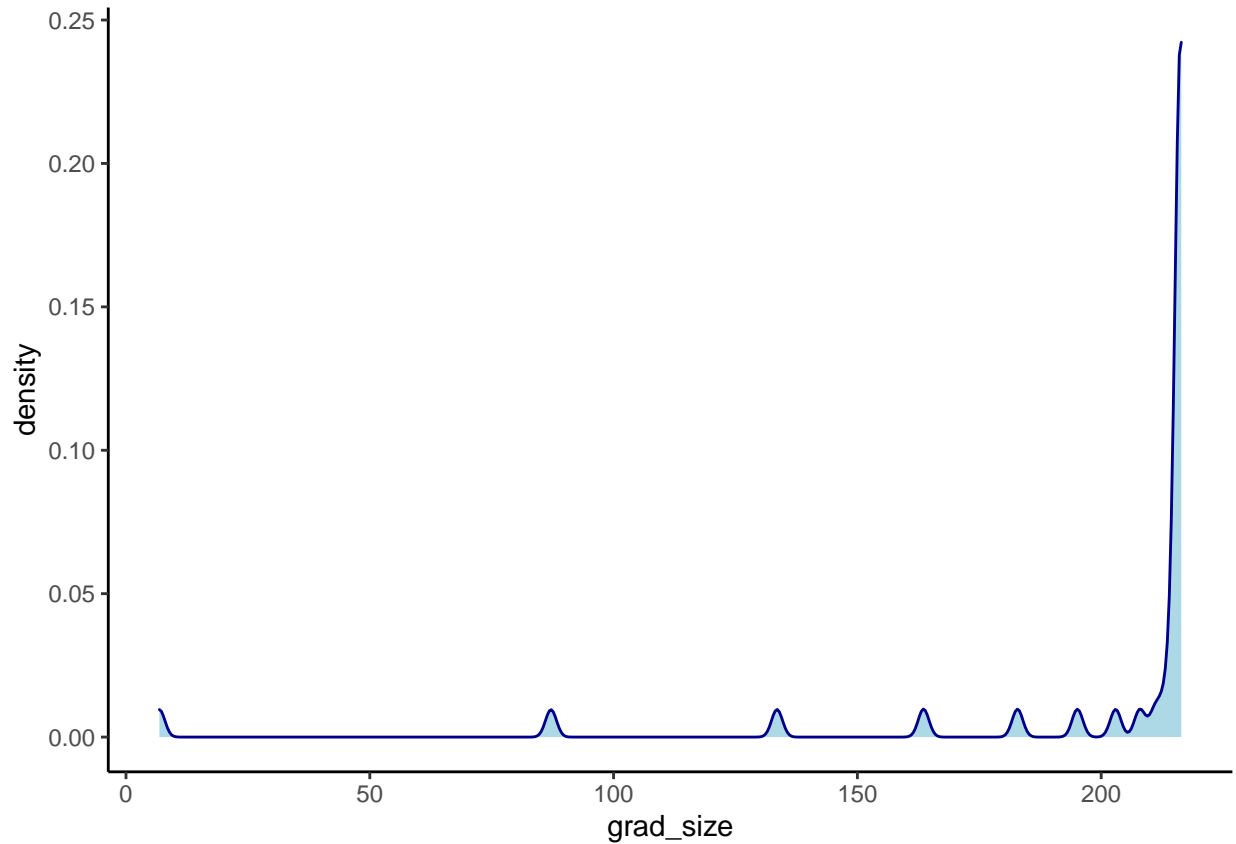
The new script function would return the norm size in the returned object, user could visualize the distribution

of gradient norm size through following code

```
grad_avg = rowMeans(ENIGMA_l2max$gradient_norm)

df = data.frame(grad_size = grad_avg)
ggplot(df, aes(x=grad_size))+
  geom_density(color="darkblue", fill="lightblue") + theme_classic()
```



We could find that majority of gradients have gradient norm size around 210, we calculated the mean value of all gradients.

```
mean(grad_avg)
```

```
## [1] 201.1309
```

So, setting the renormalized gradient norm size = 200 is reasonable.