

# DEG practice report

First things first. Read the data:

```
data_file <- read.csv("T2vsT1finalRsyn1.csv")  
View(data_file)
```

Now let's see what the data looks like:

```
print(which(grepl("Tt", names(data_file))))  
print(which(grepl("To", names(data_file))))
```

So, **Column names are not sorted by grouping...**

```
> print(which(grepl("Tt", names(data_file))))  
[1] 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 69 70  
[22] 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91  
[43] 92 93 94 95 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146  
[64] 147 148 149 150 151 152 153 154 223 224 225 226 227 228 229 230 231 232 233 234 235  
[85] 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256  
[106] 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 303 304  
[127] 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325  
[148] 326 327 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 401 402 403  
[169] 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 433 434 435 436  
[190] 437  
> print(which(grepl("To", names(data_file))))  
[1] 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22  
[22] 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 58 59 60 61 62  
[43] 63 64 65 66 67 68 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110  
[64] 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 155 156  
[85] 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177  
[106] 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198  
[127] 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219  
[148] 220 221 222 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293  
[169] 294 295 296 297 298 299 300 301 302 328 329 330 331 332 333 334 335 336 337 338 339  
[190] 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360  
[211] 361 362 363 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397  
[232] 398 399 400 421 422 423 424 425 426 427 428 429 430 431 432  
> View(data_file)
```

Let's reorder the data:

```
# Reorder the data base on control & case(controls come first):
```

```
# Find column indices
```

```
genes <- which(grepl("X", names(data_file)))
```

```
to_cols <- which(grepl("To", names(data_file)))
```

```
tt_cols <- which(grepl("Tt", names(data_file)))
```

```
# Combine the indices in the desired order
```

```
new_order <- c(genes,to_cols, tt_cols)
```

```
# Reorder the dataframe
```

```
data_reordered <- data_file[, new_order]
```

```
View(data_reordered)
```

```
print(which(grepl("Tt", names(data_reordered))))
```

```
print(which(grepl("To", names(data_reordered))))
```

Now the columns are coming in an order ...

```
> print(which(grepl("Tt", names(data_reordered))))
[1] 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268
[22] 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289
[43] 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310
[64] 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331
[85] 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352
[106] 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373
[127] 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394
[148] 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415
[169] 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436
[190] 437
> print(which(grepl("To", names(data_reordered))))
[1] 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22
[22] 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43
[43] 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64
[64] 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85
[85] 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106
[106] 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127
[127] 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148
[148] 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169
[169] 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190
[190] 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211
[211] 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232
[232] 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247
```

Now let's check whether we have duplication of gene symbol or not:

```
#check if having a duplicate of gene symbols or not:
```

```
anyDuplicated(data_reordered[[1]])
```

There are 0 duplicates. It's okay..

```
> anyDuplicated(data_reordered[[1]])
[1] 0
```

```
# Save the ordered data:  
write.csv(data_reordered, "data_reordered.csv", row.names = FALSE)
```

Now say hello to Limma:

```
# Extract expression matrix (remove the first column, which contains  
gene names)  
expr_data <- as.matrix(data_reordered[, -1])  
rownames(expr_data) <- data_reordered[[1]]
```

This is why we did that:

- The expression data input to `lmFit` must be numeric (expression levels).
- Gene names (text) cannot be part of the matrix itself, but are important for labeling rows so results can be linked back to gene identifiers.

```
# Check for 0 data
```

```
any(expr_data == 0)
```

```
summary(as.numeric(expr_data))
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
19.40	63.13	75.86	79.36	91.92	399.37

Based on the summary results, we need to perform `log2` on the data because the numbers are not suitable for limma.

```
# Log2 transform the expression data
```

```
log2_expr_data <- log2(expr_data)
```

```
# Define sample groups
```

```
# Suppose first 'n' columns are control (To), remaining are case (Tt)
```

```
num_controls <- length(which(grepl("To", names(data_reordered))))
```



```
num_cases <- length(which(grepl("Tt", names(data_reordered))))
```

```
group <- factor(c(rep("Control", num_controls), rep("case", num_cases)))
```

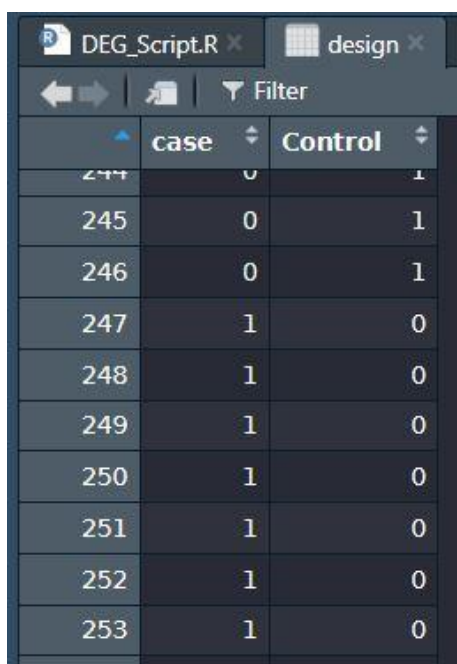
```
group
```

```
> group
[1] Control Control Control Control Control Control Control Control Control Control
[11] Control Control Control Control Control Control Control Control Control Control
[21] Control Control Control Control Control Control Control Control Control Control
[31] Control Control Control Control Control Control Control Control Control Control
[41] Control Control Control Control Control Control Control Control Control Control
[51] Control Control Control Control Control Control Control Control Control Control
[61] Control Control Control Control Control Control Control Control Control Control
[71] Control Control Control Control Control Control Control Control Control Control
[81] Control Control Control Control Control Control Control Control Control Control
[91] Control Control Control Control Control Control Control Control Control Control
[101] Control Control Control Control Control Control Control Control Control Control
[111] Control Control Control Control Control Control Control Control Control Control
[121] Control Control Control Control Control Control Control Control Control Control
[131] Control Control Control Control Control Control Control Control Control Control
[141] Control Control Control Control Control Control Control Control Control Control
[151] Control Control Control Control Control Control Control Control Control Control
[161] Control Control Control Control Control Control Control Control Control Control
[171] Control Control Control Control Control Control Control Control Control Control
[181] Control Control Control Control Control Control Control Control Control Control
[191] Control Control Control Control Control Control Control Control Control Control
[201] Control Control Control Control Control Control Control Control Control Control
[211] Control Control Control Control Control Control Control Control Control Control
[221] Control Control Control Control Control Control Control Control Control Control
[231] Control Control Control Control Control Control Control Control Control Control
[241] Control Control Control Control Control Control Control case case case case
[251] case case case case case case case case case case case
[261] case case case case case case case case case case case
```

```
design <- model.matrix(~0+group)
```

```
View(design)
```

```
colnames(design) <- c("case", "Control")
```



	case	Control
244	0	1
245	0	1
246	0	1
247	1	0
248	1	0
249	1	0
250	1	0
251	1	0
252	1	0
253	1	0

```
# Fit linear model
fit <- lmFit(log2_expr_data , design)
#Fit the lmFit with Contrasts
cont <- makeContrasts(contrasts = "case - Control" , levels = design)
fit2 <- contrasts.fit(fit = fit , contrasts = cont)
fit3 <- eBayes(fit2)
```

```
# Check top DEGs
```

```
top_table <- topTable(fit3, number=Inf, adjust.method="fdr")
```

```
View(top_table)
```

```
# Save results
```

```
write.csv(top_table, "limma_log2_results.csv")
```

	logFC	AveExpr	t	P.Value	adj.P.Val	B
MT2A	0.3805254	6.771343	22.93789	4.806628e-77	6.197667e-73	164.78080
IFI30	0.2693766	6.503477	21.22326	2.980644e-69	1.921621e-65	146.97580
CTHRC1	0.3421909	6.609149	21.16104	5.722549e-69	2.459552e-65	146.32843
C1QC	0.2994013	6.617091	20.39517	1.756857e-65	5.663230e-62	138.35889
ANXA5	0.2197726	6.739244	19.79101	9.859800e-63	2.542645e-59	132.07550
TYROBP	0.3045426	6.604740	19.70964	2.311292e-62	4.966967e-59	131.22984
MSN	0.2348258	6.732888	19.69060	2.821093e-62	5.196453e-59	131.03198
SULF1	0.4131044	6.396907	19.50535	1.960718e-61	3.160187e-58	129.10744
TUBB6	0.2718860	6.456414	19.41771	4.903976e-61	7.025763e-58	128.19742
PTRF	0.2342854	6.729746	19.33074	1.217726e-60	1.570136e-57	127.29455
ITGB2	0.2958258	6.632982	19.29784	1.717620e-60	2.013363e-57	126.95311
TNC	0.3528378	6.360275	19.27299	2.227191e-60	2.393117e-57	126.69521

Showing 1 to 14 of 12,894 entries, 6 total columns