

Class 12: RNASeq analysis

Berne Chu (A18608434)

Table of contents

Background	1
Data import	1
Toy differential gene expression	3
DESeq2 analysis	8
Volcano Plot	10
Save our results	10
Add gene annotation	11
Pathway analysis	13
Save our main results	15

Background

Today we will analyze some RNASeq data from Himes et al. on the effects of a common steroid (dexamethasone) on airway smooth muscle cells (ASM cells).

At the starting point is the “counts” data and “metadata” that contain the count values for each gene in their different experiments (i.e. cell lines with or without the drug).

Data import

```
# Complete the missing code
counts <- read.csv("airway_scaledcounts.csv", row.names=1)
metadata <- read.csv("airway_metadata.csv")
```

Let's have a wee peek at these objects:

```
head(counts)
```

	SRR1039508	SRR1039509	SRR1039512	SRR1039513	SRR1039516
ENSG000000000003	723	486	904	445	1170
ENSG000000000005	0	0	0	0	0
ENSG000000000419	467	523	616	371	582
ENSG000000000457	347	258	364	237	318
ENSG000000000460	96	81	73	66	118
ENSG000000000938	0	0	1	0	2

	SRR1039517	SRR1039520	SRR1039521
ENSG000000000003	1097	806	604
ENSG000000000005	0	0	0
ENSG000000000419	781	417	509
ENSG000000000457	447	330	324
ENSG000000000460	94	102	74
ENSG000000000938	0	0	0

Q1. How many genes are in the dataset?

```
nrow(counts)
```

```
[1] 38694
```

Q. How many different experiments (columns in counts or rows in metadata) are there?

```
ncol(counts)
```

```
[1] 8
```

```
nrow(metadata)
```

```
[1] 8
```

```
head(metadata)
```

	id	dex	celltype	geo_id
1	SRR1039508	control	N61311	GSM1275862
2	SRR1039509	treated	N61311	GSM1275863
3	SRR1039512	control	N052611	GSM1275866
4	SRR1039513	treated	N052611	GSM1275867
5	SRR1039516	control	N080611	GSM1275870
6	SRR1039517	treated	N080611	GSM1275871

Q2. How many ‘control’ cell lines do we have?

```
sum(metadata$dex=="control")
```

```
[1] 4
```

Toy differential gene expression

To start our analysis let’s calculate the mean counts for all genes in the “control” experiments

1. Extract all “control” columns from the `count` object
2. Calculate the mean for all rows (i.e. genes) of these “control” columns
- 3-4. Do the same for the “treated”
5. Compare these `control.mean` and `treated.mean` values.

Q3. How would you make the above code in either approach more robust? Is there a function that could help here?

```
control.inds <- metadata$dex == "control"
control.counts <- counts[, control.inds]
```

```
control.means <- rowMeans(control.counts)
```

Q4. Follow the same procedure for the treated samples (i.e. calculate the mean per gene across drug treated samples and assign to a labeled vector called `treated.mean`)

```
treated.means <- rowMeans(counts[,metadata$dex == "treated"])
```

Store these together for ease of bookkeeping as `meancounts`

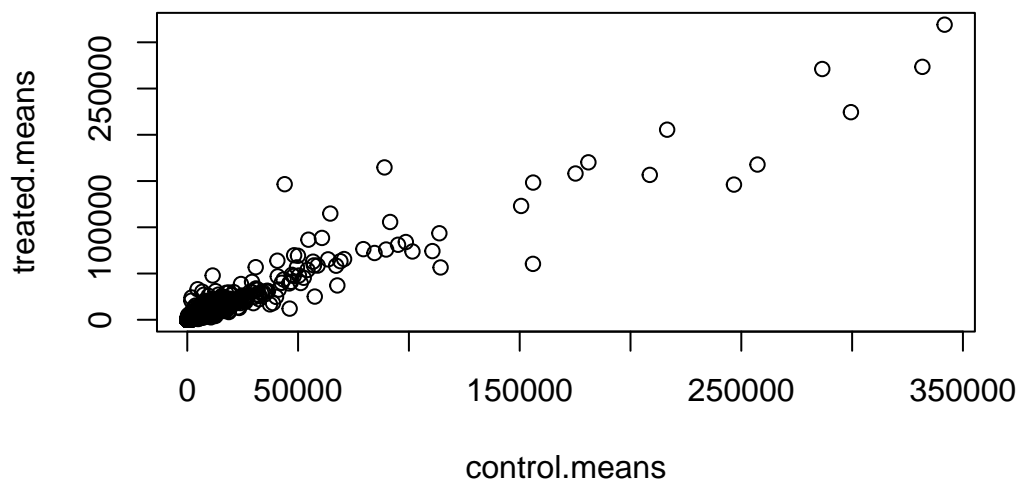
```
meancounts <- data.frame(control.means,treated.means)
head(meancounts)
```

	control.means	treated.means
ENSG000000000003	900.75	658.00
ENSG000000000005	0.00	0.00
ENSG000000000419	520.50	546.00
ENSG000000000457	339.75	316.50
ENSG000000000460	97.25	78.75
ENSG000000000938	0.75	0.00

Q5 (a). Create a scatter plot showing the mean of the treated samples against the mean of the control samples. Your plot should look something like the following.

Make a plot of control vs treated mean values for all genes

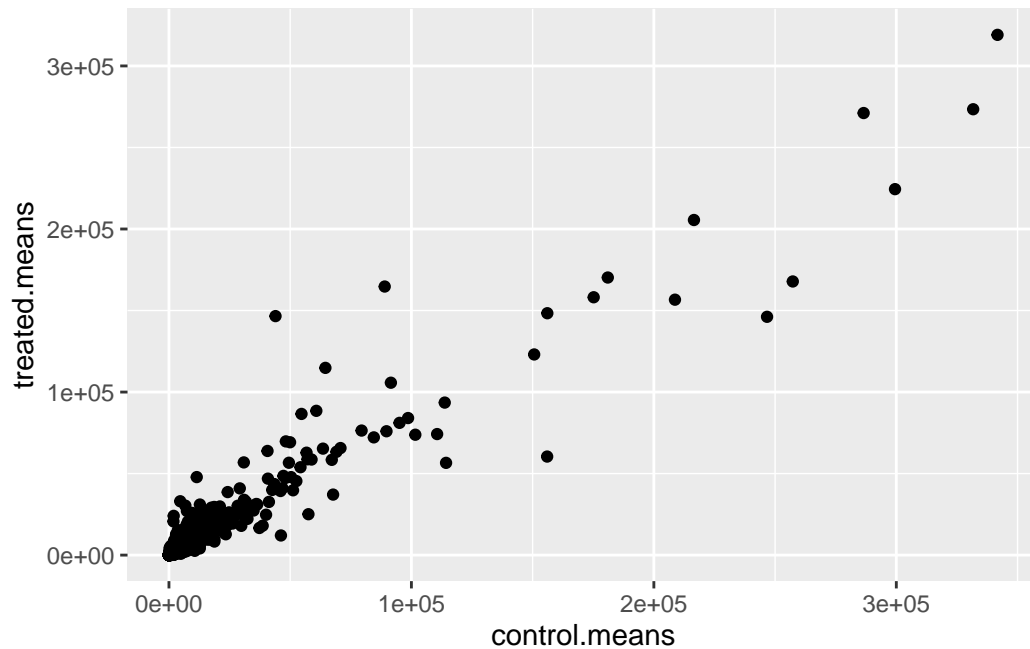
```
plot(meancounts)
```



Q5 (b). You could also use the ggplot2 package to make this figure producing the plot below. What geom_?() function would you use for this plot?

```
library(ggplot2)

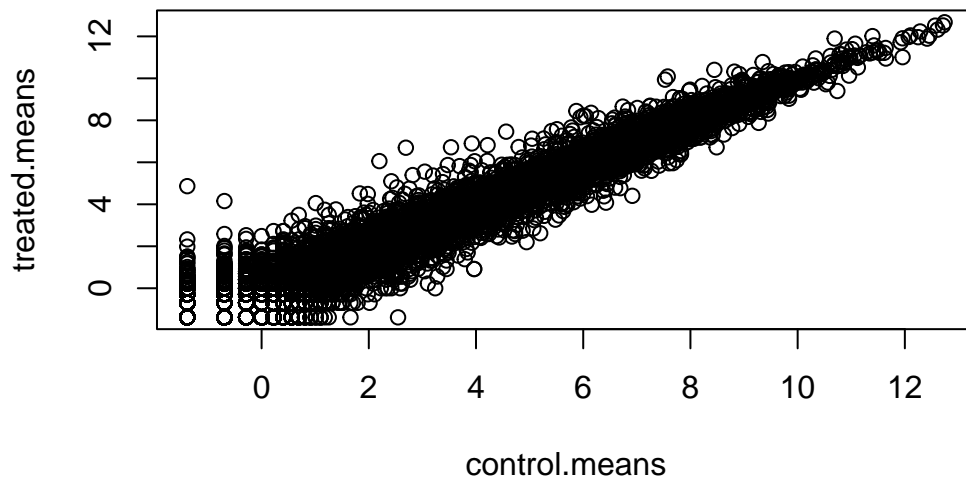
ggplot(meancounts) +
  aes(control.means,treated.means)+
  geom_point()
```



Q6. Try plotting both axes on a log scale. What is the argument to plot() that allows you to do this?

Make this a log log plot

```
plot(log(meancounts))
```



We often talk metrics like “log2 fold-change”

```
# control/treated  
log2(10/10)
```

```
[1] 0
```

```
log2(10/20)
```

```
[1] -1
```

```
log2(20/10)
```

```
[1] 1
```

```
log2(40/10)
```

```
[1] 2
```

```
log2(10/40)
```

```
[1] -2
```

Let's calculate the log2 fold change for our treated over control mean counts.

```
meancounts$log2fc <-  
log2(meancounts$control.means/  
      meancounts$treated.means)
```

```
head(meancounts)
```

	control.means	treated.means	log2fc
ENSG000000000003	900.75	658.00	0
ENSG000000000005	0.00	0.00	NaN
ENSG000000000419	520.50	546.00	0
ENSG000000000457	339.75	316.50	0
ENSG000000000460	97.25	78.75	0
ENSG000000000938	0.75	0.00	0

Q7. What is the purpose of the arr.ind argument in the which() function call above? Why would we then take the first column of the output and need to call the unique() function?

It returns the row and column coordinates instead of a single vector index. We then take the first column (row indices) and use unique() so we don't remove the same row more than once if it had zero in both columns.

A common "rule of thumb" is a log2 fold change cutoff of +2 and -2 to call genes "Up regulated" or "Down regulated"

Q8. Using the up.ind vector above can you determine how many up regulated genes we have at the greater than 2 fc level?

Number of "up" genes at +2 threshold

```
sum(meancounts$log2fc >= +2, na.rm=T)
```

```
[1] 0
```

Q9. Using the down.ind vector above can you determine how many down regulated genes we have at the greater than 2 fc level?

Number of “down” genes at -2 threshold

```
sum(meancounts$log2fc <= -2, na.rm =T)
```

```
[1] 0
```

Q10. Do you trust these results? Why or why not?

I don't trust these results because it considers the outliers and the values that are not statistical significant.

DESeq2 analysis

Let's do this analysis properly and keep our inner stats nerd happy - i.e. are the differences we see between drug and no drug significant given the replicated experiments.

```
library(DESeq2)
```

For DESeq analysis we need three things

- count values (`countData`)
- metadata telling us about the columns in the `countData` (`colData`)
- design of the experiment (i.e. what do you want to compare)

Our first function from DESeq2 will setup the input required for analysis by storing all these 3 things together.

```
dds <- DESeqDataSetFromMatrix(countData = counts,  
                              colData = metadata,  
                              design = ~dex)
```

converting counts to integer mode

```
Warning in DESeqDataSet(se, design = design, ignoreRank): some variables in  
design formula are characters, converting to factors
```

The main function in DESeq2 that runs the analysis is called `DESeq()`

```
dds <- DESeq(dds)
```

estimating size factors

estimating dispersions

gene-wise dispersion estimates

mean-dispersion relationship

final dispersion estimates

fitting model and testing

```
res <- results(dds)
head(res)
```

log2 fold change (MLE): dex treated vs control

Wald test p-value: dex treated vs control

DataFrame with 6 rows and 6 columns

	baseMean	log2FoldChange	lfcSE	stat	pvalue
	<numeric>	<numeric>	<numeric>	<numeric>	<numeric>
ENSG000000000003	747.194195	-0.3507030	0.168246	-2.084470	0.0371175
ENSG000000000005	0.000000	NA	NA	NA	NA
ENSG0000000000419	520.134160	0.2061078	0.101059	2.039475	0.0414026
ENSG0000000000457	322.664844	0.0245269	0.145145	0.168982	0.8658106
ENSG0000000000460	87.682625	-0.1471420	0.257007	-0.572521	0.5669691
ENSG0000000000938	0.319167	-1.7322890	3.493601	-0.495846	0.6200029
	padj				
	<numeric>				
ENSG0000000000003	0.163035				
ENSG0000000000005	NA				
ENSG0000000000419	0.176032				
ENSG0000000000457	0.961694				
ENSG0000000000460	0.815849				
ENSG0000000000938	NA				

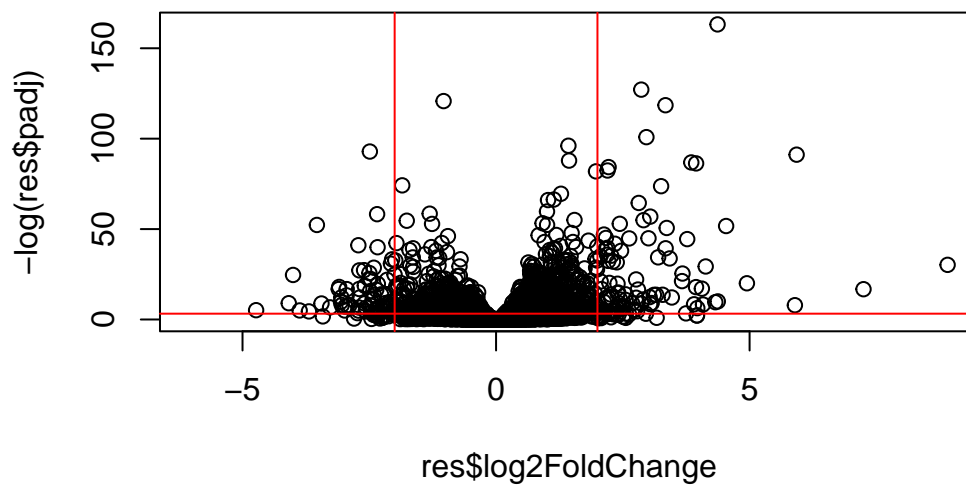
36000*0.05

[1] 1800

Volcano Plot

This is a common summary result figure from these types of experiments and plot the log2 fold-change vs the adjusted p-value.

```
plot(res$log2FoldChange, -log(res$padj))  
abline(v=c(-2,2), col="red")  
abline(h=-log(0.04), col="red")
```



```
log(0.1)
```

```
[1] -2.302585
```

```
log(0.000001)
```

```
[1] -13.81551
```

Save our results

```
write.csv(res,file="my_results.csv")
```

Add gene annotation

To help make sense of our results and communicate to other folks we need to add some more annotation to our main `res` object.

We will use two bioconductor packages to first map IDs to different formats including the classic gene “symbol” gene name.

I will install these with the following commands: `BiocManager::install("AnnotationDbi")`
`BiocManager::install("org.Hs.eg.db")`

```
library(AnnotationDbi)
library(org.Hs.eg.db)
```

Let’s see what is in `org.Hs.eg.db` with the `columns()` function

```
columns(org.Hs.eg.db)
```

[1]	"ACCNUM"	"ALIAS"	"ENSEMBL"	"ENSEMBLPROT"	"ENSEMBLTRANS"
[6]	"ENTREZID"	"ENZYME"	"EVIDENCE"	"EVIDENCEALL"	"GENENAME"
[11]	"GENETYPE"	"GO"	"GOALL"	"IPI"	"MAP"
[16]	"OMIM"	"ONTOLOGY"	"ONTOLOGYALL"	"PATH"	"PFAM"
[21]	"PMID"	"PROSITE"	"REFSEQ"	"SYMBOL"	"UCSCKG"
[26]	"UNIPROT"				

We can translate or “map” IDs between any of these 26 databases using the `mapIds()` function

```
res$symbol <- mapIds(keys=row.names(res), #current IDs
                     keytype = "ENSEMBL", #Format of our IDs
                     x=org.Hs.eg.db,      #where to get the mappings from
                     column="SYMBOL")     #format/DB to map to
```

'select()' returned 1:many mapping between keys and columns

```
head(res)
```

log2 fold change (MLE): dex treated vs control

Wald test p-value: dex treated vs control

DataFrame with 6 rows and 7 columns

	baseMean	log2FoldChange	lfcSE	stat	pvalue
	<numeric>	<numeric>	<numeric>	<numeric>	<numeric>
ENSG000000000003	747.194195	-0.3507030	0.168246	-2.084470	0.0371175
ENSG000000000005	0.000000	NA	NA	NA	NA
ENSG000000000419	520.134160	0.2061078	0.101059	2.039475	0.0414026
ENSG000000000457	322.664844	0.0245269	0.145145	0.168982	0.8658106
ENSG000000000460	87.682625	-0.1471420	0.257007	-0.572521	0.5669691
ENSG000000000938	0.319167	-1.7322890	3.493601	-0.495846	0.6200029
	padj	symbol			
	<numeric>	<character>			
ENSG000000000003	0.163035	TSPAN6			
ENSG000000000005	NA	TNMD			
ENSG000000000419	0.176032	DPM1			
ENSG000000000457	0.961694	SCYL3			
ENSG000000000460	0.815849	FIRRM			
ENSG000000000938	NA	FGR			

Add the maps for “GENENAME” and “ENTREZID” and store as `res$genename` and `res$entrez`

```
res$genename <- mapIds(keys=row.names(res),  
  keytype = "ENSEMBL",  
  x=org.Hs.eg.db,  
  column="GENENAME")
```

'select()' returned 1:many mapping between keys and columns

```
res$entrez <- mapIds(keys=row.names(res),  
  keytype = "ENSEMBL",  
  x=org.Hs.eg.db,  
  column="ENTREZID")
```

'select()' returned 1:many mapping between keys and columns

```
head(res)
```

log2 fold change (MLE): dex treated vs control

Wald test p-value: dex treated vs control

DataFrame with 6 rows and 9 columns

	baseMean	log2FoldChange	lfcSE	stat	pvalue
	<numeric>	<numeric>	<numeric>	<numeric>	<numeric>
ENSG000000000003	747.194195	-0.3507030	0.168246	-2.084470	0.0371175
ENSG000000000005	0.000000	NA	NA	NA	NA
ENSG0000000000419	520.134160	0.2061078	0.101059	2.039475	0.0414026
ENSG0000000000457	322.664844	0.0245269	0.145145	0.168982	0.8658106
ENSG0000000000460	87.682625	-0.1471420	0.257007	-0.572521	0.5669691
ENSG0000000000938	0.319167	-1.7322890	3.493601	-0.495846	0.6200029
	padj	symbol	genename	entrez	
	<numeric>	<character>	<character>	<character>	
ENSG0000000000003	0.163035	TSPAN6	tetraspanin 6	7105	
ENSG0000000000005	NA	TNMD	tenomodulin	64102	
ENSG0000000000419	0.176032	DPM1	dolichyl-phosphate m..	8813	
ENSG0000000000457	0.961694	SCYL3	SCY1 like pseudokina..	57147	
ENSG0000000000460	0.815849	FIRRM	FIGNL1 interacting r..	55732	
ENSG0000000000938	NA	FGR	FGR proto-oncogene, ..	2268	

Pathway analysis

There are lots of bioconductor packages to do this type of analysis. For now let's just try one called **gage** again we need to install this if we don't have it already

```
library(gage)
library(gageData)
library(pathview)
```

To use **gage** I need two things

- a named vector of fold-change values for our DEGs (our geneset of interest)
- a set of pathways or genesets to use for annotation.

```
x <- c("barry"=5, "lise"=10)
x
```

```
barry lise
    5    10
```

```
names(x) <- c("low", "high")
x
```

```
low high
5    10
```

```
foldchanges <- res$log2FoldChange
names(foldchanges) <- res$entrez
head(foldchanges)
```

```
          7105          64102          8813          57147          55732          2268
-0.35070302          NA  0.20610777  0.02452695 -0.14714205 -1.73228897
```

```
data(kegg.sets.hs)

keggres = gage(foldchanges, gsets=kegg.sets.hs)
```

In our results object we have:

```
attributes(keggres)
```

```
$names
[1] "greater" "less"    "stats"
```

```
head(keggres$less, 5)
```

	p.geomean	stat.mean
hsa05332 Graft-versus-host disease	0.0004250461	-3.473346
hsa04940 Type I diabetes mellitus	0.0017820293	-3.002352
hsa05310 Asthma	0.0020045888	-3.009050
hsa04672 Intestinal immune network for IgA production	0.0060434515	-2.560547
hsa05330 Allograft rejection	0.0073678825	-2.501419
	p.val	q.val
hsa05332 Graft-versus-host disease	0.0004250461	0.09053483
hsa04940 Type I diabetes mellitus	0.0017820293	0.14232581
hsa05310 Asthma	0.0020045888	0.14232581
hsa04672 Intestinal immune network for IgA production	0.0060434515	0.31387180
hsa05330 Allograft rejection	0.0073678825	0.31387180
	set.size	expl

Let's look at one of these pathways with our genes colored up so we can see the overlap

'select()' returned 1:1 mapping between keys and columns

Info: Writing image file hsa05310.pathview.png

ASTHMA

Diagram illustrating the KEGG pathway for Asthma, showing the progression from allergen exposure to bronchial obstruction.

Key Components and Interactions:

- Allergen:** Initiates the response, interacting with T cell receptor, B cell receptor, and Mast cell.
- Antigen Presenting Cell (APC):** Processes and presents antigen to T1 cell.
- T1 cell:** Interacts with T2 cell via T cell receptor signaling pathway.
- T2 cell:** Interacts with B cell via B cell receptor signaling pathway.
- B cell:** Produces IgE antibodies.
- Mast cell:** Interacts with IgE and allergen, leading to FeRI signaling pathway activation.
- Eosinophil:** Interacts with IgE and allergen, leading to cytokine release.
- Epithelial cells:** Interact with eosinophils and release cytokines.
- Smooth muscle cells:** Interact with eosinophils and release cytokines.
- Immediate reaction:** Includes Bronchospasm, Edema, and Airflow obstruction.
- Late reaction:** Includes Airway inflammation, Airway hyperresponsiveness, and Airway hyperventilation.

Signaling Pathways and Cytokines:

- T cell receptor signaling pathway:** Involves CD4, CD8, and TCR.
- B cell receptor signaling pathway:** Involves CD19, CD20, and BCR.
- FeRI signaling pathway:** Involves IgE and allergen.
- Cytokine-cytokine receptor interaction:** Involves IL-1, IL-2, IL-3, IL-4, IL-5, IL-6, IL-7, IL-8, IL-9, IL-10, IL-11, IL-12, IL-13, IL-14, IL-15, IL-16, IL-17, IL-18, IL-19, IL-20, IL-21, IL-22, IL-23, IL-24, IL-25, IL-26, IL-27, IL-28, IL-29, IL-30, IL-31, IL-32, IL-33, IL-34, IL-35, IL-36, IL-37, IL-38, IL-39, IL-40, IL-41, IL-42, IL-43, IL-44, IL-45, IL-46, IL-47, IL-48, IL-49, IL-50, IL-51, IL-52, IL-53, IL-54, IL-55, IL-56, IL-57, IL-58, IL-59, IL-60, IL-61, IL-62, IL-63, IL-64, IL-65, IL-66, IL-67, IL-68, IL-69, IL-70, IL-71, IL-72, IL-73, IL-74, IL-75, IL-76, IL-77, IL-78, IL-79, IL-80, IL-81, IL-82, IL-83, IL-84, IL-85, IL-86, IL-87, IL-88, IL-89, IL-90, IL-91, IL-92, IL-93, IL-94, IL-95, IL-96, IL-97, IL-98, IL-99, IL-100, IL-101, IL-102, IL-103, IL-104, IL-105, IL-106, IL-107, IL-108, IL-109, IL-110, IL-111, IL-112, IL-113, IL-114, IL-115, IL-116, IL-117, IL-118, IL-119, IL-120, IL-121, IL-122, IL-123, IL-124, IL-125, IL-126, IL-127, IL-128, IL-129, IL-130, IL-131, IL-132, IL-133, IL-134, IL-135, IL-136, IL-137, IL-138, IL-139, IL-140, IL-141, IL-142, IL-143, IL-144, IL-145, IL-146, IL-147, IL-148, IL-149, IL-150, IL-151, IL-152, IL-153, IL-154, IL-155, IL-156, IL-157, IL-158, IL-159, IL-160, IL-161, IL-162, IL-163, IL-164, IL-165, IL-166, IL-167, IL-168, IL-169, IL-170, IL-171, IL-172, IL-173, IL-174, IL-175, IL-176, IL-177, IL-178, IL-179, IL-180, IL-181, IL-182, IL-183, IL-184, IL-185, IL-186, IL-187, IL-188, IL-189, IL-190, IL-191, IL-192, IL-193, IL-194, IL-195, IL-196, IL-197, IL-198, IL-199, IL-200, IL-201, IL-202, IL-203, IL-204, IL-205, IL-206, IL-207, IL-208, IL-209, IL-210, IL-211, IL-212, IL-213, IL-214, IL-215, IL-216, IL-217, IL-218, IL-219, IL-220, IL-221, IL-222, IL-223, IL-224, IL-225, IL-226, IL-227, IL-228, IL-229, IL-230, IL-231, IL-232, IL-233, IL-234, IL-235, IL-236, IL-237, IL-238, IL-239, IL-240, IL-241, IL-242, IL-243, IL-244, IL-245, IL-246, IL-247, IL-248, IL-249, IL-250, IL-251, IL-252, IL-253, IL-254, IL-255, IL-256, IL-257, IL-258, IL-259, IL-260, IL-261, IL-262, IL-263, IL-264, IL-265, IL-266, IL-267, IL-268, IL-269, IL-270, IL-271, IL-272, IL-273, IL-274, IL-275, IL-276, IL-277, IL-278, IL-279, IL-280, IL-281, IL-282, IL-283, IL-284, IL-285, IL-286, IL-287, IL-288, IL-289, IL-290, IL-291, IL-292, IL-293, IL-294, IL-295, IL-296, IL-297, IL-298, IL-299, IL-300, IL-301, IL-302, IL-303, IL-304, IL-305, IL-306, IL-307, IL-308, IL-309, IL-310, IL-311, IL-312, IL-313, IL-314, IL-315, IL-316, IL-317, IL-318, IL-319, IL-320, IL-321, IL-322, IL-323, IL-324, IL-325, IL-326, IL-327, IL-328, IL-329, IL-330, IL-331, IL-332, IL-333, IL-334, IL-335, IL-336, IL-337, IL-338, IL-339, IL-340, IL-341, IL-342, IL-343, IL-344, IL-345, IL-346, IL-347, IL-348, IL-349, IL-350, IL-351, IL-352, IL-353, IL-354, IL-355, IL-356, IL-357, IL-358, IL-359, IL-360, IL-361, IL-362, IL-363, IL-364, IL-365, IL-366, IL-367, IL-368, IL-369, IL-370, IL-371, IL-372, IL-373, IL-374, IL-375, IL-376, IL-377, IL-378, IL-379, IL-380, IL-381, IL-382, IL-383, IL-384, IL-385, IL-386, IL-387, IL-388, IL-389, IL-390, IL-391, IL-392, IL-393, IL-394, IL-395, IL-396, IL-397, IL-398, IL-399, IL-400, IL-401, IL-402, IL-403, IL-404, IL-405, IL-406, IL-407, IL-408, IL-409, IL-410, IL-411, IL-412, IL-413, IL-414, IL-415, IL-416, IL-417, IL-418, IL-419, IL-420, IL-421, IL-422, IL-423, IL-424, IL-425, IL-426, IL-427, IL-428, IL-429, IL-430, IL-431, IL-432, IL-433, IL-434, IL-435, IL-436, IL-437, IL-438, IL-439, IL-440, IL-441, IL-442, IL-443, IL-444, IL-445, IL-446, IL-447, IL-448, IL-449, IL-450, IL-451, IL-452, IL-453, IL-454, IL-455, IL-456, IL-457, IL-458, IL-459, IL-460, IL-461, IL-462, IL-463, IL-464, IL-465, IL-466, IL-467, IL-468, IL-469, IL-470, IL-471, IL-472, IL-473, IL-474, IL-475, IL-476, IL-477, IL-478, IL-479, IL-480, IL-481, IL-482, IL-483, IL-484, IL-485, IL-486, IL-487, IL-488, IL-489, IL-490, IL-491, IL-492, IL-493, IL-494, IL-495, IL-496, IL-497, IL-498, IL-499, IL-500, IL-501, IL-502, IL-503, IL-504, IL-505, IL-506, IL-507, IL-508, IL-509, IL-510, IL-511, IL-512, IL-513, IL-514, IL-515, IL-516, IL-517, IL-518, IL-519, IL-520, IL-521, IL-522, IL-523, IL-524, IL-525, IL-526, IL-527, IL-528, IL-529, IL-530, IL-531, IL-532, IL-533, IL-534, IL-535, IL-536, IL-537, IL-538, IL-539, IL-540, IL-541, IL-542, IL-543, IL-544, IL-545, IL-546, IL-547, IL-548, IL-549, IL-550, IL-551, IL-552, IL-553, IL-554, IL-555, IL-556, IL-557, IL-558, IL-559, IL-560, IL-561, IL-562, IL-563, IL-564, IL-565, IL-566, IL-567, IL-568, IL-569, IL-570, IL-571, IL-572, IL-573, IL-574, IL-575, IL-576, IL-577, IL-578, IL-579, IL-580, IL-581, IL-582, IL-583, IL-584, IL-585, IL-586, IL-587, IL-588, IL-589, IL-590, IL-591, IL-592, IL-593, IL-594, IL-595, IL-596, IL-597, IL-598, IL-599,

```
write.csv(res,file="myresults_annotated.csv")
```