Kidney Disease ML Analytics

Janis Corona

12/10/2019

## Kidney Disease Analysis from gene expression profiles of 12 healthy and 4 renal disease samples for day 1, 3, 6, and 9 days in culture

#### <https://www.ncbi.nlm.nih.gov/geo/download/?acc=GSE141257>

kidneyDisease <- read.csv('Samples16-downloaded-kidney-disease.csv',  
 sep=',', header=TRUE,  
 na.strings=c('',' ','NA'))  
head(kidneyDisease)

## X X1\_AK124p1\_Adh.count X2\_AK124p1\_SPH3d.count  
## 1 A1BG 16 8  
## 2 A1BG-AS1 2 2  
## 3 A1CF 1 0  
## 4 A2M 4 1  
## 5 A2M-AS1 5 9  
## 6 A2ML1 0 0  
## X3\_AK124p1\_SPH6d.count X4\_AK124p1\_SPH9d.count X5\_AK125p1\_Adh.count  
## 1 3 4 26  
## 2 1 2 0  
## 3 2 1 0  
## 4 4 5 3  
## 5 4 5 14  
## 6 2 0 0  
## X6\_AK125p1\_SPH3d.count X7\_AK125p1\_SPH6d.count X8\_AK125p1\_SPH9d.count  
## 1 15 18 17  
## 2 5 5 2  
## 3 2 1 2  
## 4 14 31 33  
## 5 12 20 17  
## 6 3 0 1  
## AK82p2Adh AK82p3SPH3d AK82p3SPH6d AK82p3SPH10d AK86p1Adh AK86p2.SPH3d  
## 1 16 9 9 11 36 25  
## 2 8 12 4 8 9 10  
## 3 12 11 9 8 20 8  
## 4 2 68 126 103 2 52  
## 5 31 45 26 25 24 18  
## 6 6 3 2 11 1 2  
## AK86p2.SPH6d AK86p2.SPH10d  
## 1 18 15  
## 2 10 5  
## 3 6 7  
## 4 142 119  
## 5 30 20  
## 6 6 6

tail(kidneyDisease)

## X X1\_AK124p1\_Adh.count X2\_AK124p1\_SPH3d.count  
## 25364 ZXDC 185 157  
## 25365 ZYG11A 13 12  
## 25366 ZYG11B 162 122  
## 25367 ZYX 2369 1408  
## 25368 ZZEF1 224 275  
## 25369 ZZZ3 279 198  
## X3\_AK124p1\_SPH6d.count X4\_AK124p1\_SPH9d.count X5\_AK125p1\_Adh.count  
## 25364 156 159 468  
## 25365 5 8 32  
## 25366 127 98 296  
## 25367 1171 1263 5406  
## 25368 267 250 507  
## 25369 216 189 490  
## X6\_AK125p1\_SPH3d.count X7\_AK125p1\_SPH6d.count X8\_AK125p1\_SPH9d.count  
## 25364 513 487 421  
## 25365 36 31 21  
## 25366 351 410 341  
## 25367 4665 4875 4272  
## 25368 821 886 669  
## 25369 561 566 508  
## AK82p2Adh AK82p3SPH3d AK82p3SPH6d AK82p3SPH10d AK86p1Adh  
## 25364 867 1174 1321 1169 944  
## 25365 186 122 98 67 151  
## 25366 889 1246 1398 1197 554  
## 25367 5560 6262 5831 5165 3920  
## 25368 1324 1834 2090 2006 1036  
## 25369 1304 1519 1889 1518 869  
## AK86p2.SPH3d AK86p2.SPH6d AK86p2.SPH10d  
## 25364 1522 1587 1122  
## 25365 106 113 65  
## 25366 1294 1337 1072  
## 25367 7483 6378 5735  
## 25368 2076 2299 2028  
## 25369 1691 1795 1389

SampleType <- read.csv('diseaseSampleType.csv', sep=',', header=TRUE,  
 na.strings=c('',' ', 'NA'))

SampleType

## sample sample\_ID Condition testCondition  
## 1 GSM4200015 AK82p2Adh healthy adherent  
## 2 GSM4200016 AK82p3SPH3d healthy 3daySpheres  
## 3 GSM4200017 AK82p3SPH6d healthy 6daySpheres  
## 4 GSM4200018 AK82p3SPH10d healthy 9daySpheres  
## 5 GSM4200019 AK86p1Adh healthy adherent  
## 6 GSM4200020 AK86p2-SPH3d healthy 3daySpheres  
## 7 GSM4200021 AK86p2-SPH6d healthy 6daySpheres  
## 8 GSM4200022 AK86p2-SPH10d healthy 9daySpheres  
## 9 GSM4200023 1\_AK124p1\_Adh.count renal disease adherent  
## 10 GSM4200024 2\_AK124p1\_SPH3d.count renal disease 3daySpheres  
## 11 GSM4200025 3\_AK124p1\_SPH6d.count renal disease 6daySpheres  
## 12 GSM4200026 4\_AK124p1\_SPH9d.count renal disease 9daySpheres  
## 13 GSM4200027 5\_AK125p1\_Adh.count healthy adherent  
## 14 GSM4200028 6\_AK125p1\_SPH3d.count healthy 3daySpheres  
## 15 GSM4200029 7\_AK125p1\_SPH6d.count healthy 6daySpheres  
## 16 GSM4200030 8\_AK125p1\_SPH9d.count healthy 9daySpheres

The sample\_IDs with AK124p1 in the name are the four renal disease samples

colnames(kidneyDisease)

## [1] "X" "X1\_AK124p1\_Adh.count"   
## [3] "X2\_AK124p1\_SPH3d.count" "X3\_AK124p1\_SPH6d.count"  
## [5] "X4\_AK124p1\_SPH9d.count" "X5\_AK125p1\_Adh.count"   
## [7] "X6\_AK125p1\_SPH3d.count" "X7\_AK125p1\_SPH6d.count"  
## [9] "X8\_AK125p1\_SPH9d.count" "AK82p2Adh"   
## [11] "AK82p3SPH3d" "AK82p3SPH6d"   
## [13] "AK82p3SPH10d" "AK86p1Adh"   
## [15] "AK86p2.SPH3d" "AK86p2.SPH6d"   
## [17] "AK86p2.SPH10d"

healthy <- kidneyDisease[,-c(2:5)]  
colnames(healthy)[1] <- 'Gene'  
renalDisease <- kidneyDisease[,c(1,2:5)]  
colnames(renalDisease) <- c('Gene','renal\_0','renal\_3','renal\_6','renal\_9')

colnames(healthy)

## [1] "Gene" "X5\_AK125p1\_Adh.count"   
## [3] "X6\_AK125p1\_SPH3d.count" "X7\_AK125p1\_SPH6d.count"  
## [5] "X8\_AK125p1\_SPH9d.count" "AK82p2Adh"   
## [7] "AK82p3SPH3d" "AK82p3SPH6d"   
## [9] "AK82p3SPH10d" "AK86p1Adh"   
## [11] "AK86p2.SPH3d" "AK86p2.SPH6d"   
## [13] "AK86p2.SPH10d"

dim(healthy)

## [1] 25369 13

colnames(renalDisease)

## [1] "Gene" "renal\_0" "renal\_3" "renal\_6" "renal\_9"

dim(renalDisease)

## [1] 25369 5

str(healthy)

## 'data.frame': 25369 obs. of 13 variables:  
## $ Gene : Factor w/ 25369 levels "A1BG","A1BG-AS1",..: 1 2 3 4 5 6 7 8 9 10 ...  
## $ X5\_AK125p1\_Adh.count : int 26 0 0 3 14 0 0 0 1031 2 ...  
## $ X6\_AK125p1\_SPH3d.count: int 15 5 2 14 12 3 0 0 1393 6 ...  
## $ X7\_AK125p1\_SPH6d.count: int 18 5 1 31 20 0 0 0 1365 3 ...  
## $ X8\_AK125p1\_SPH9d.count: int 17 2 2 33 17 1 0 0 1126 3 ...  
## $ AK82p2Adh : int 16 8 12 2 31 6 0 2 939 2 ...  
## $ AK82p3SPH3d : int 9 12 11 68 45 3 0 2 1043 1 ...  
## $ AK82p3SPH6d : int 9 4 9 126 26 2 0 2 982 2 ...  
## $ AK82p3SPH10d : int 11 8 8 103 25 11 1 1 1123 1 ...  
## $ AK86p1Adh : int 36 9 20 2 24 1 0 3 626 3 ...  
## $ AK86p2.SPH3d : int 25 10 8 52 18 2 0 2 1334 1 ...  
## $ AK86p2.SPH6d : int 18 10 6 142 30 6 0 2 1493 2 ...  
## $ AK86p2.SPH10d : int 15 5 7 119 20 6 0 2 909 7 ...

str(renalDisease)

## 'data.frame': 25369 obs. of 5 variables:  
## $ Gene : Factor w/ 25369 levels "A1BG","A1BG-AS1",..: 1 2 3 4 5 6 7 8 9 10 ...  
## $ renal\_0: int 16 2 1 4 5 0 0 0 536 0 ...  
## $ renal\_3: int 8 2 0 1 9 0 0 0 436 1 ...  
## $ renal\_6: int 3 1 2 4 4 2 0 0 378 1 ...  
## $ renal\_9: int 4 2 1 5 5 0 0 0 352 0 ...

library(dplyr)

## Assign no duplicate instances of genes

Renal\_df <- renalDisease[!duplicated(renalDisease$Gene),]

## Check that all the genes have 1 count each, they do

renalCounts <- Renal\_df %>% group\_by(Gene) %>%   
 summarise(counts = n())  
dim(renalCounts)

## [1] 25369 2

unique(renalCounts$counts)

## [1] 1

healthyCounts <- healthy %>% group\_by(Gene) %>%  
 summarise(counts = n())  
dim(healthyCounts)

## [1] 25369 2

unique(renalCounts$counts)

## [1] 1

#### Attach a field to the renal and healthy data frames for gene means

row.names(renalDisease) <- renalDisease$Gene  
renalDisease <- renalDisease[2:5]  
renalDisease$Gene\_Means <- rowMeans(renalDisease)

row.names(healthy) <- healthy$Gene  
healthy <- healthy[2:13]  
healthy$Gene\_Means <- round(rowMeans(healthy),3)

colnames(healthy)[13] <- "healthy\_Means"  
colnames(renalDisease)[5] <- "renal\_Means"

Combined <- cbind(renalDisease, healthy)  
Combined <- Combined[,c(5,18,1:4,6:17)]

#### Create the fold change field to compare the change in Renal diseased gene expression to healthy gene expression

Fold\_Change <- Combined %>% mutate(Fold\_Change = renal\_Means/healthy\_Means)  
row.names(Fold\_Change) <- row.names(Combined)  
Fold\_Change <- Fold\_Change[,c(19,1:18)]

#### Remove NaN’s or Not a number and Inf when dividing by zero or a very small value

Fold\_Change$Fold\_Change <- gsub('NaN',0,Fold\_Change$Fold\_Change)  
Fold\_Change$Fold\_Change <- gsub('Inf', 0, Fold\_Change$Fold\_Change)  
Fold\_Change$Fold\_Change <- round(as.numeric(Fold\_Change$Fold\_Change),3)

Top20\_FC <- Fold\_Change[order(Fold\_Change$Fold\_Change, decreasing=TRUE)[0:20],]

#### Create the Differential Expression in renal Disease compared to healthy genes

Differential <- Fold\_Change %>% mutate(Differential\_Expression = healthy\_Means-renal\_Means)  
Differential <- Differential[,c(20,1:19)]  
Differential$Differential\_Expression <- round(as.numeric(Differential$Differential\_Expression, 3))  
row.names(Differential) <- row.names(Fold\_Change)

#### Since this is healthy - diseased, positive values mean the diseased gene expression means are lower than the healthy gene expression levels

downgraded <- Differential[order(Differential$Differential\_Expression,   
 decreasing=TRUE),]

#### Top 20 downgraded genes

Top20\_down <- downgraded[0:20,]

#### Top 20 upgraded genes, diseased gene expression means were higher than the healthy gene expression means, hence negative values for differential expression

upgraded <- Differential[order(Differential$Differential\_Expression,  
 decreasing=FALSE),]

Top20\_up <- upgraded[0:20,]

Top20\_up

## Differential\_Expression Fold\_Change renal\_Means healthy\_Means  
## XIST -1109 1109.750 1109.75 1.000  
## CLDN2 -428 1.264 2052.25 1624.167  
## TDRD1 -70 7.523 80.25 10.667  
## CDKN1C -66 1.117 631.00 564.917  
## CEBPD -33 1.031 1101.00 1067.750  
## LOC101928796 -32 5.812 38.75 6.667  
## LBP -30 2.521 50.00 19.833  
## LOC389332 -29 1.143 234.25 205.000  
## CA9 -21 1.385 75.50 54.500  
## B4GALNT4 -19 1.447 61.00 42.167  
## IFI27 -18 1.149 142.25 123.833  
## NAT8 -16 2.169 30.00 13.833  
## OVCH1-AS1 -15 88.323 14.75 0.167  
## ANGPTL3 -14 1.933 29.00 15.000  
## RARRES1 -13 1.206 78.50 65.083  
## CADM3 -10 1.386 35.00 25.250  
## PRR7 -9 1.064 155.75 146.333  
## FMO1 -8 1.442 24.75 17.167  
## PDF -8 1.160 59.75 51.500  
## CFH -6 1.028 227.50 221.333  
## renal\_0 renal\_3 renal\_6 renal\_9 X5\_AK125p1\_Adh.count  
## XIST 1083 1181 1072 1103 1  
## CLDN2 695 1922 2640 2952 740  
## TDRD1 88 84 90 59 4  
## CDKN1C 1034 493 517 480 934  
## CEBPD 2333 616 721 734 4205  
## LOC101928796 30 53 32 40 3  
## LBP 101 30 39 30 3  
## LOC389332 122 179 306 330 158  
## CA9 191 24 37 50 118  
## B4GALNT4 70 43 58 73 40  
## IFI27 110 154 161 144 32  
## NAT8 0 13 42 65 3  
## OVCH1-AS1 21 12 11 15 1  
## ANGPTL3 0 25 64 27 0  
## RARRES1 124 51 78 61 75  
## CADM3 17 22 43 58 22  
## PRR7 262 134 111 116 439  
## FMO1 3 5 36 55 0  
## PDF 43 53 59 84 84  
## CFH 202 178 265 265 241  
## X6\_AK125p1\_SPH3d.count X7\_AK125p1\_SPH6d.count  
## XIST 1 0  
## CLDN2 2256 2530  
## TDRD1 3 2  
## CDKN1C 604 704  
## CEBPD 1926 1957  
## LOC101928796 12 15  
## LBP 5 4  
## LOC389332 213 310  
## CA9 15 20  
## B4GALNT4 32 44  
## IFI27 105 116  
## NAT8 22 34  
## OVCH1-AS1 0 0  
## ANGPTL3 27 47  
## RARRES1 57 75  
## CADM3 27 56  
## PRR7 319 319  
## FMO1 9 16  
## PDF 118 160  
## CFH 139 220  
## X8\_AK125p1\_SPH9d.count AK82p2Adh AK82p3SPH3d AK82p3SPH6d  
## XIST 1 2 1 1  
## CLDN2 2901 880 876 1553  
## TDRD1 3 21 14 25  
## CDKN1C 583 170 483 483  
## CEBPD 1832 301 255 344  
## LOC101928796 6 6 6 3  
## LBP 16 0 0 3  
## LOC389332 343 67 115 208  
## CA9 34 154 12 12  
## B4GALNT4 44 32 20 24  
## IFI27 95 94 180 175  
## NAT8 70 0 4 7  
## OVCH1-AS1 0 0 0 0  
## ANGPTL3 47 3 7 10  
## RARRES1 92 23 35 103  
## CADM3 74 4 5 15  
## PRR7 240 83 29 16  
## FMO1 26 0 10 10  
## PDF 147 10 10 2  
## CFH 194 604 230 145  
## AK82p3SPH10d AK86p1Adh AK86p2.SPH3d AK86p2.SPH6d  
## XIST 1 2 2 0  
## CLDN2 2970 706 1420 2156  
## TDRD1 29 5 2 11  
## CDKN1C 376 427 890 928  
## CEBPD 264 588 476 435  
## LOC101928796 4 3 13 6  
## LBP 10 61 33 101  
## LOC389332 321 149 221 262  
## CA9 25 177 27 35  
## B4GALNT4 32 89 43 57  
## IFI27 196 71 187 153  
## NAT8 15 0 3 3  
## OVCH1-AS1 0 0 0 1  
## ANGPTL3 22 3 9 3  
## RARRES1 115 66 35 63  
## CADM3 55 11 3 17  
## PRR7 43 82 79 69  
## FMO1 96 0 7 9  
## PDF 11 7 38 24  
## CFH 132 322 131 185  
## AK86p2.SPH10d  
## XIST 0  
## CLDN2 502  
## TDRD1 9  
## CDKN1C 197  
## CEBPD 230  
## LOC101928796 3  
## LBP 2  
## LOC389332 93  
## CA9 25  
## B4GALNT4 49  
## IFI27 82  
## NAT8 5  
## OVCH1-AS1 0  
## ANGPTL3 2  
## RARRES1 42  
## CADM3 14  
## PRR7 38  
## FMO1 23  
## PDF 7  
## CFH 113

Top20\_down

## Differential\_Expression Fold\_Change renal\_Means healthy\_Means  
## EEF1A1 210936 0.187 48649.75 259585.25  
## SPP1 168566 0.310 75657.25 244223.67  
## GAPDH 64730 0.284 25728.50 90458.83  
## ACTB 63054 0.189 14693.25 77747.08  
## PKM 49623 0.254 16907.00 66529.58  
## ACTG1 48854 0.200 12176.50 61030.50  
## TPT1 44349 0.252 14955.25 59304.58  
## APP 43083 0.205 11095.50 54178.33  
## ITGB1 42873 0.172 8891.50 51764.17  
## ITGA3 42182 0.223 12093.25 54275.08  
## CD24 41714 0.216 11462.00 53175.58  
## FTL 40830 0.414 28790.25 69619.75  
## ENO1 39477 0.180 8674.75 48151.50  
## FTH1 34238 0.413 24129.25 58366.92  
## AHNAK 32651 0.112 4135.00 36786.25  
## ITM2B 32607 0.298 13858.25 46465.58  
## FN1 32014 0.579 44045.00 76058.50  
## CTSD 30978 0.301 13351.75 44329.58  
## S100A6 30770 0.092 3111.75 33881.25  
## EEF2 30682 0.335 15444.25 46126.50  
## renal\_0 renal\_3 renal\_6 renal\_9 X5\_AK125p1\_Adh.count  
## EEF1A1 90272 38853 34048 31426 239248  
## SPP1 65298 79054 80888 77389 127128  
## GAPDH 50985 17807 16654 17468 99960  
## ACTB 23576 13526 10654 11017 55506  
## PKM 24223 14980 14042 14383 44159  
## ACTG1 22573 9354 7980 8799 56936  
## TPT1 20852 16071 12141 10757 50400  
## APP 10122 10025 11406 12829 22980  
## ITGB1 13689 7848 7235 6794 35480  
## ITGA3 16989 12031 9847 9506 34065  
## CD24 15392 10237 10015 10204 32246  
## FTL 43755 25884 22577 22945 102383  
## ENO1 15733 5745 6459 6762 32386  
## FTH1 41911 21204 16988 16414 82156  
## AHNAK 6527 2812 3750 3451 11620  
## ITM2B 12999 15913 13127 13394 27992  
## FN1 110993 30854 19374 14959 106221  
## CTSD 14332 11041 12405 15629 37448  
## S100A6 6739 2077 1763 1868 15292  
## EEF2 26275 14205 11337 9960 61685  
## X6\_AK125p1\_SPH3d.count X7\_AK125p1\_SPH6d.count  
## EEF1A1 104936 103173  
## SPP1 145794 164672  
## GAPDH 54280 52301  
## ACTB 44955 48192  
## PKM 47754 45858  
## ACTG1 33260 36404  
## TPT1 45227 36541  
## APP 34506 41493  
## ITGB1 25353 28416  
## ITGA3 37068 37976  
## CD24 30924 33000  
## FTL 70698 76114  
## ENO1 20727 22796  
## FTH1 56716 54788  
## AHNAK 9726 12685  
## ITM2B 43048 42279  
## FN1 70244 63233  
## CTSD 40079 52291  
## S100A6 9160 9083  
## EEF2 37262 31831  
## X8\_AK125p1\_SPH9d.count AK82p2Adh AK82p3SPH3d AK82p3SPH6d  
## EEF1A1 96223 369732 358958 354492  
## SPP1 168254 207451 271193 405106  
## GAPDH 49922 197908 88206 80886  
## ACTB 41221 141949 84291 93010  
## PKM 41527 98241 83227 80446  
## ACTG1 33062 87108 63832 74141  
## TPT1 31277 70837 73093 68178  
## APP 40545 29785 64443 84487  
## ITGB1 24352 71893 58946 75348  
## ITGA3 31792 66552 67193 67940  
## CD24 31193 32022 61796 88364  
## FTL 70852 54045 86606 74025  
## ENO1 20486 115134 52682 56422  
## FTH1 48194 72329 69510 53322  
## AHNAK 10344 43848 44861 51700  
## ITM2B 39884 25175 59702 69200  
## FN1 41505 62691 181963 76237  
## CTSD 53841 18419 57626 53975  
## S100A6 7603 57942 38977 42301  
## EEF2 28247 51676 54630 45025  
## AK82p3SPH10d AK86p1Adh AK86p2.SPH3d AK86p2.SPH6d AK86p2.SPH10d  
## EEF1A1 298852 255621 339466 310284 284038  
## SPP1 399705 149488 264813 374388 252692  
## GAPDH 74397 137540 91967 81636 76503  
## ACTB 79206 85586 86125 82400 90524  
## PKM 73785 36763 84009 77840 84746  
## ACTG1 70191 49178 67353 71902 88999  
## TPT1 58600 70156 85145 68374 53827  
## APP 86396 17193 70066 82498 75748  
## ITGB1 61731 39034 57480 68742 74395  
## ITGA3 64517 40550 71168 66258 66222  
## CD24 86975 22302 62035 78315 78935  
## FTL 73476 48715 80936 56773 40814  
## ENO1 49390 52554 48857 52189 54195  
## FTH1 47670 50331 72315 50154 42918  
## AHNAK 46990 27703 48435 58478 75045  
## ITM2B 64698 25088 58812 57211 44498  
## FN1 43674 27776 127619 38869 72670  
## CTSD 62983 8925 57473 47544 41351  
## S100A6 44065 52298 35948 37213 56693  
## EEF2 43864 34303 67288 53282 44425

#### Write these two top 20 genes being expressed more in diseased as Top 20 up-expressed, and the top 20 genes being expressed less as the Top 20 down-expressed genes as csv files. Also, write the top 20 fold change genes to its own csv file

write.csv(Top20\_down, 'Down-regulated-20.csv', row.names=TRUE)  
write.csv(Top20\_up, 'Up-regulated-20.csv', row.names=TRUE)  
write.csv(Top20\_FC, 'Fold-Change-20.csv', row.names=TRUE)

### What are the top 20 genes up-regulated in renal disease compared to healthy?

Up <- as.data.frame(row.names(Top20\_up))  
colnames(Up) <- 'Gene'  
Up

## Gene  
## 1 XIST  
## 2 CLDN2  
## 3 TDRD1  
## 4 CDKN1C  
## 5 CEBPD  
## 6 LOC101928796  
## 7 LBP  
## 8 LOC389332  
## 9 CA9  
## 10 B4GALNT4  
## 11 IFI27  
## 12 NAT8  
## 13 OVCH1-AS1  
## 14 ANGPTL3  
## 15 RARRES1  
## 16 CADM3  
## 17 PRR7  
## 18 FMO1  
## 19 PDF  
## 20 CFH

### What are the top 20 genes down-regulated in renal disease compared to healthy?

Down <- as.data.frame(row.names(Top20\_down))  
colnames(Down) <- 'Gene'  
Down

## Gene  
## 1 EEF1A1  
## 2 SPP1  
## 3 GAPDH  
## 4 ACTB  
## 5 PKM  
## 6 ACTG1  
## 7 TPT1  
## 8 APP  
## 9 ITGB1  
## 10 ITGA3  
## 11 CD24  
## 12 FTL  
## 13 ENO1  
## 14 FTH1  
## 15 AHNAK  
## 16 ITM2B  
## 17 FN1  
## 18 CTSD  
## 19 S100A6  
## 20 EEF2

### What are the top 20 genes that have the most fold change in the ratio of healthy to renal disease gene expression? Even the inverse fold change of disease to healthy would

FC <- as.data.frame(row.names(Top20\_FC))  
colnames(FC) <- 'Gene'  
FC

## Gene  
## 1 XIST  
## 2 OVCH1-AS1  
## 3 LGALS12  
## 4 HLA-DQA2  
## 5 TDRD1  
## 6 AIRN  
## 7 CD3D  
## 8 CXCR3  
## 9 LINC01272  
## 10 OR51B2  
## 11 SLA  
## 12 OXT  
## 13 LOC101928796  
## 14 CD302  
## 15 TDRD12  
## 16 LINC00668  
## 17 RNASE1  
## 18 ABCB11  
## 19 ATP2B3  
## 20 AVP

Up$RenalType <- rep('Up',20)  
Down$RenalType <- rep('Down',20)  
FC$RenalType <- rep('foldChange', 20)

Up <- Up[order(Up$Gene,decreasing=FALSE),]  
FC <- FC[order(FC$Gene,decreasing=FALSE),]  
common <- merge(Up,FC, by.x='Gene', by.y='Gene')

#### Common genes to most fold change and up regulated gene expressions are:

common

## Gene RenalType.x RenalType.y  
## 1 LOC101928796 Up foldChange  
## 2 OVCH1-AS1 Up foldChange  
## 3 TDRD1 Up foldChange  
## 4 XIST Up foldChange

#### We should go to NCBI Gene get the gene information on these up regulated genes in Renal disease.

chr <- as.data.frame(c('21','12','10','X'))  
direction <- as.data.frame(c('+','+','+','-'))  
start <- as.data.frame(c('45972970','29389294','114174442','73820651'))  
end <- as.data.frame(c('45974953','29487324','114232669','73852753'))  
TissueMostExpressed <- as.data.frame(c('testis','testis','testis','thyroid'))  
fullName <- as.data.frame(c('uncharacterized LOC101928796','OVCH1 antisense RNA 1','tudor domain containing 1','X inactive specific transcript'))  
GeneFunction <- as.data.frame(c('ncRNA.','ncRNA.','protein coding. This gene encodes a protein containing a tudor domain that is thought to function in the suppression of transposable elements during spermatogenesis. The related protein in mouse forms a complex with piRNAs and Piwi proteins to promote methylation and silencing of target sequences. This gene was observed to be upregulated by ETS transcription factor ERG in prostate tumors.','ncRNA. X inactivation is an early developmental process in mammalian females that transcriptionally silences one of the pair of X chromosomes, thus providing dosage equivalence between males and females. The process is regulated by several factors, including a region of chromosome X called the X inactivation center (XIC). The XIC comprises several non-coding and protein-coding genes, and this gene was the first non-coding gene identified within the XIC. This gene is expressed exclusively from the XIC of the inactive X chromosome, and is essential for the initiation and spread of X-inactivation. The transcript is a spliced RNA. Alternatively spliced transcript variants have been identified, but their full length sequences have not been determined. Mutations in the XIST promoter cause familial skewed X inactivation.'))

info <- cbind(fullName, TissueMostExpressed,GeneFunction, chr,direction, start, end)  
colnames(info) <- c('geneName','TissueMostExpressed','geneFunction','chromosome','strandDirection','startBP','endBP')  
  
information <- cbind(common, info)  
information

## Gene RenalType.x RenalType.y geneName  
## 1 LOC101928796 Up foldChange uncharacterized LOC101928796  
## 2 OVCH1-AS1 Up foldChange OVCH1 antisense RNA 1  
## 3 TDRD1 Up foldChange tudor domain containing 1  
## 4 XIST Up foldChange X inactive specific transcript  
## TissueMostExpressed  
## 1 testis  
## 2 testis  
## 3 testis  
## 4 thyroid  
## geneFunction  
## 1 ncRNA.  
## 2 ncRNA.  
## 3 protein coding. This gene encodes a protein containing a tudor domain that is thought to function in the suppression of transposable elements during spermatogenesis. The related protein in mouse forms a complex with piRNAs and Piwi proteins to promote methylation and silencing of target sequences. This gene was observed to be upregulated by ETS transcription factor ERG in prostate tumors.  
## 4 ncRNA. X inactivation is an early developmental process in mammalian females that transcriptionally silences one of the pair of X chromosomes, thus providing dosage equivalence between males and females. The process is regulated by several factors, including a region of chromosome X called the X inactivation center (XIC). The XIC comprises several non-coding and protein-coding genes, and this gene was the first non-coding gene identified within the XIC. This gene is expressed exclusively from the XIC of the inactive X chromosome, and is essential for the initiation and spread of X-inactivation. The transcript is a spliced RNA. Alternatively spliced transcript variants have been identified, but their full length sequences have not been determined. Mutations in the XIST promoter cause familial skewed X inactivation.  
## chromosome strandDirection startBP endBP  
## 1 21 + 45972970 45974953  
## 2 12 + 29389294 29487324  
## 3 10 + 114174442 114232669  
## 4 X - 73820651 73852753

Read in a table of the gene summaries for these genes with some summaries missing for certain genes in the up/down/fold change top 20 genes

summaries <- read.csv('GeneDescriptionsNCBIgene.csv', sep=',',header=TRUE,  
 na.strings=c('',' ','NA'))  
summaries <- summaries[,c(1:3)]

The gene summaries by complete cases

summ <- summaries[complete.cases(summaries$geneFunction),]

Merge the up, down, and fold change genes with their gene summaries

Top20\_up$gene <- row.names(Top20\_up)  
Top20\_down$gene <- row.names(Top20\_down)  
Top20\_FC$gene <- row.names(Top20\_FC)  
  
up\_summ <- merge(summ, Top20\_up, by.x='gene',by.y='gene')  
down\_summ <- merge(summ, Top20\_down, by.x='gene',by.y='gene')  
fc\_summ <- merge(summ, Top20\_FC, by.x='gene',by.y='gene')

Use lemmatization on the available top 20 down regulated gene summaries

library(tm)  
library(SnowballC)  
library(wordcloud)  
library(ggplot2)  
library(textstem)

lemma <- lemmatize\_strings(up\_summ$geneFunction, dictionary=lexicon::hash\_lemmas)  
  
Lemma <- as.data.frame(lemma)  
Lemma <- cbind(Lemma, up\_summ)  
  
colnames(Lemma)[1] <- 'lemmatized\_summary'  
  
write.csv(Lemma, 'Lemmatized\_upreg20.csv', row.names=FALSE)

dir.create('./upreg20-Lemma')  
  
ea <- as.character(Lemma$lemmatized\_summary)  
setwd('./upreg20-Lemma')  
  
for (j in 1:length(ea)){  
 write(ea[j], paste(paste('up',j, sep='.'), '.txt', sep=''))  
}  
setwd('../')

KidneyDisease <- Corpus(DirSource("upreg20-Lemma"))  
  
KidneyDisease

## <<SimpleCorpus>>  
## Metadata: corpus specific: 1, document level (indexed): 0  
## Content: documents: 16

KidneyDisease <- tm\_map(KidneyDisease, removePunctuation)  
KidneyDisease <- tm\_map(KidneyDisease, removeNumbers)  
KidneyDisease <- tm\_map(KidneyDisease, tolower)  
KidneyDisease <- tm\_map(KidneyDisease, removeWords, stopwords("english"))  
KidneyDisease <- tm\_map(KidneyDisease, stripWhitespace)  
  
dtmKidneyDisease <- DocumentTermMatrix(KidneyDisease)  
dtmKidneyDisease

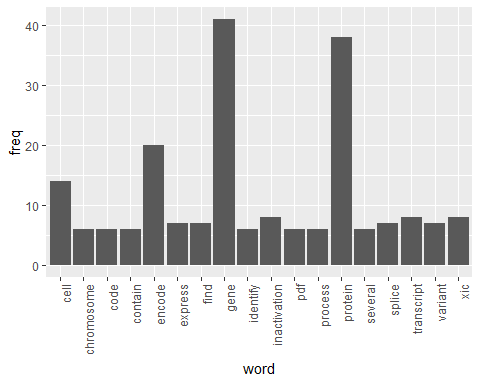
## <<DocumentTermMatrix (documents: 16, terms: 398)>>  
## Non-/sparse entries: 667/5701  
## Sparsity : 90%  
## Maximal term length: 23  
## Weighting : term frequency (tf)

freq <- colSums(as.matrix(dtmKidneyDisease))  
  
FREQ <- data.frame(freq)  
ord <- order(freq, decreasing=TRUE)  
  
freq[head(ord, 25)]

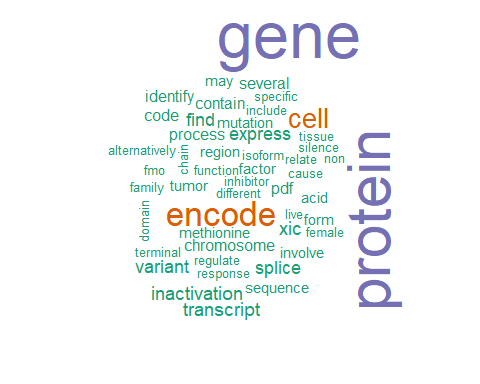
## gene protein encode cell transcript   
## 41 38 20 14 8   
## inactivation xic express find splice   
## 8 8 7 7 7   
## variant contain process pdf identify   
## 7 6 6 6 6   
## chromosome code several involve may   
## 6 6 6 5 5   
## mutation acid sequence methionine factor   
## 5 5 5 5 5

### Up regulated genes

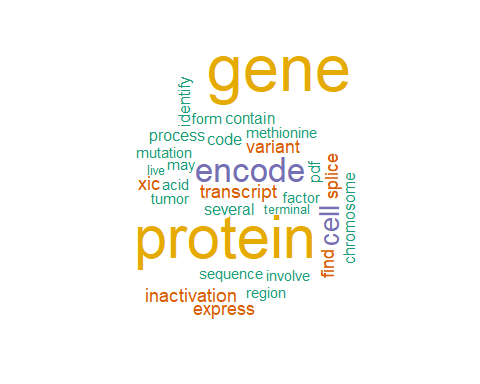
wf <- data.frame(word=names(freq), freq=freq)  
p <- ggplot(subset(wf, freq>5), aes(word, freq))  
p <- p + geom\_bar(stat= 'identity')   
p <- p + theme(axis.text.x=element\_text(angle=90, hjust=1))   
p



wordcloud(names(freq), freq, min.freq=4,colors=brewer.pal(3,'Dark2'))



wordcloud(names(freq), freq, max.words=30,colors=brewer.pal(6,'Dark2'))



### Now for the down regulated available summaries for the top 20 down regulated genes

lemma <- lemmatize\_strings(down\_summ$geneFunction, dictionary=lexicon::hash\_lemmas)  
  
Lemma <- as.data.frame(lemma)  
Lemma <- cbind(Lemma, down\_summ)  
  
colnames(Lemma)[1] <- 'lemmatized\_summary'  
  
write.csv(Lemma, 'Lemmatized\_downreg20.csv', row.names=FALSE)

dir.create('./downreg20-Lemma')  
  
ea <- as.character(Lemma$lemmatized\_summary)  
setwd('./downreg20-Lemma')  
  
for (j in 1:length(ea)){  
 write(ea[j], paste(paste('down',j, sep='.'), '.txt', sep=''))  
}  
setwd('../')

KidneyDisease <- Corpus(DirSource("downreg20-Lemma"))  
  
KidneyDisease

## <<SimpleCorpus>>  
## Metadata: corpus specific: 1, document level (indexed): 0  
## Content: documents: 20

KidneyDisease <- tm\_map(KidneyDisease, removePunctuation)  
KidneyDisease <- tm\_map(KidneyDisease, removeNumbers)  
KidneyDisease <- tm\_map(KidneyDisease, tolower)  
KidneyDisease <- tm\_map(KidneyDisease, removeWords, stopwords("english"))  
KidneyDisease <- tm\_map(KidneyDisease, stripWhitespace)  
  
dtmKidneyDisease <- DocumentTermMatrix(KidneyDisease)  
dtmKidneyDisease

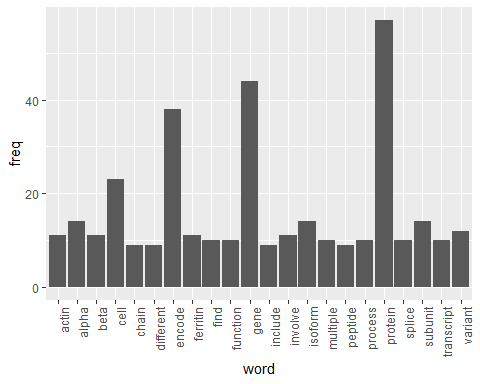
## <<DocumentTermMatrix (documents: 20, terms: 501)>>  
## Non-/sparse entries: 940/9080  
## Sparsity : 91%  
## Maximal term length: 20  
## Weighting : term frequency (tf)

freq <- colSums(as.matrix(dtmKidneyDisease))  
  
FREQ <- data.frame(freq)  
ord <- order(freq, decreasing=TRUE)  
  
freq[head(ord, 25)]

## protein gene encode cell isoform subunit   
## 57 44 38 23 14 14   
## alpha variant actin involve ferritin beta   
## 14 12 11 11 11 11   
## process splice transcript function multiple find   
## 10 10 10 10 10 10   
## different include chain peptide form disease   
## 9 9 9 9 8 8   
## may   
## 8

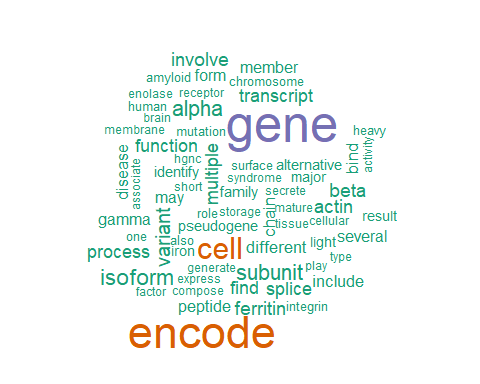
### Down regulated

wf <- data.frame(word=names(freq), freq=freq)  
p <- ggplot(subset(wf, freq>8), aes(word, freq))  
p <- p + geom\_bar(stat= 'identity')   
p <- p + theme(axis.text.x=element\_text(angle=90, hjust=1))   
p

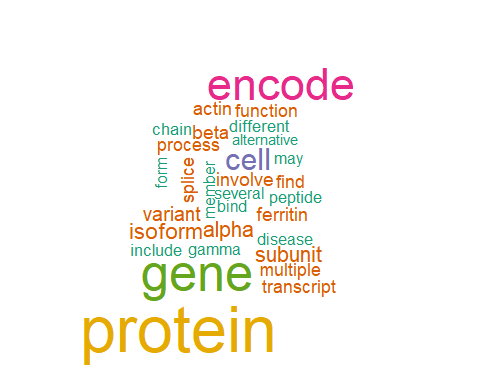


wordcloud(names(freq), freq, min.freq=4,colors=brewer.pal(3,'Dark2'))

## Warning in wordcloud(names(freq), freq, min.freq = 4, colors =  
## brewer.pal(3, : protein could not be fit on page. It will not be plotted.



wordcloud(names(freq), freq, max.words=30,colors=brewer.pal(6,'Dark2'))



### Now for the fold change top 20 available gene summaries

lemma <- lemmatize\_strings(fc\_summ$geneFunction, dictionary=lexicon::hash\_lemmas)  
  
Lemma <- as.data.frame(lemma)  
Lemma <- cbind(Lemma, fc\_summ)  
  
colnames(Lemma)[1] <- 'lemmatized\_summary'  
  
write.csv(Lemma, 'Lemmatized\_fcreg20.csv', row.names=FALSE)

dir.create('./fcreg20-Lemma')  
  
ea <- as.character(Lemma$lemmatized\_summary)  
setwd('./fcreg20-Lemma')  
  
for (j in 1:length(ea)){  
 write(ea[j], paste(paste('fc',j, sep='.'), '.txt', sep=''))  
}  
setwd('../')

KidneyDisease <- Corpus(DirSource("fcreg20-Lemma"))  
  
KidneyDisease

## <<SimpleCorpus>>  
## Metadata: corpus specific: 1, document level (indexed): 0  
## Content: documents: 16

KidneyDisease <- tm\_map(KidneyDisease, removePunctuation)  
KidneyDisease <- tm\_map(KidneyDisease, removeNumbers)  
KidneyDisease <- tm\_map(KidneyDisease, tolower)  
KidneyDisease <- tm\_map(KidneyDisease, removeWords, stopwords("english"))  
KidneyDisease <- tm\_map(KidneyDisease, stripWhitespace)  
  
dtmKidneyDisease <- DocumentTermMatrix(KidneyDisease)  
dtmKidneyDisease

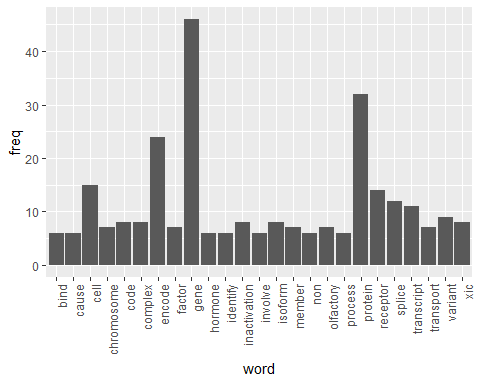
## <<DocumentTermMatrix (documents: 16, terms: 415)>>  
## Non-/sparse entries: 728/5912  
## Sparsity : 89%  
## Maximal term length: 17  
## Weighting : term frequency (tf)

freq <- colSums(as.matrix(dtmKidneyDisease))  
  
FREQ <- data.frame(freq)  
ord <- order(freq, decreasing=TRUE)  
  
freq[head(ord, 25)]

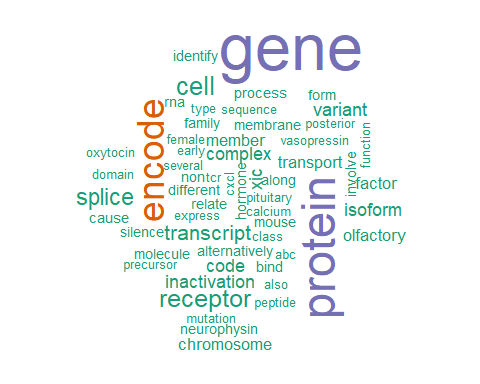
## gene protein encode cell receptor   
## 46 32 24 15 14   
## splice transcript variant code complex   
## 12 11 9 8 8   
## inactivation xic isoform member transport   
## 8 8 8 7 7   
## olfactory factor chromosome bind cause   
## 7 7 7 6 6   
## involve hormone non process identify   
## 6 6 6 6 6

### Fold Change genes

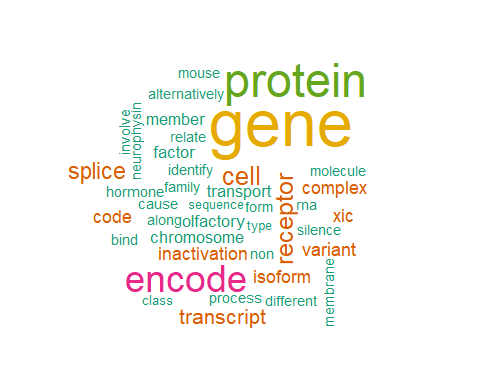
wf <- data.frame(word=names(freq), freq=freq)  
p <- ggplot(subset(wf, freq>5), aes(word, freq))  
p <- p + geom\_bar(stat= 'identity')   
p <- p + theme(axis.text.x=element\_text(angle=90, hjust=1))   
p



wordcloud(names(freq), freq, min.freq=4,colors=brewer.pal(3,'Dark2'))



wordcloud(names(freq), freq, max.words=40,colors=brewer.pal(6,'Dark2'))



## This script takes articles from the abstracts on Kidney Disease articles from NCBI’s PubMed, PLOS, and the summary of the NCBI GEO sample pages

This creates a directory to stem the abstracts and preprocess from the csv file into a corpus of 20 files in a folder called KidneyDisease.

Auto <- read.csv('NIH\_PLOS\_articles\_kidney\_disease.csv', sep=',',  
 header=FALSE, na.strings=c('',' '))

colnames(Auto) <- c('abstract','source')  
auto <- Auto[complete.cases(Auto$abstract),]  
  
  
dir.create('./KidneyDisease')  
  
ea <- as.character(auto$abstract)  
setwd('./KidneyDisease')  
  
for (j in 1:length(ea)){  
 write(ea[j], paste(paste('EA',j, sep='.'), '.txt', sep=''))  
}  
setwd('../')

This code preprocesses and stems the corpus

KidneyDisease <- Corpus(DirSource("KidneyDisease"))  
  
  
KidneyDisease

## <<SimpleCorpus>>  
## Metadata: corpus specific: 1, document level (indexed): 0  
## Content: documents: 43

#KidneyDisease <- tm\_map(KidneyDisease, removePunctuation)  
#KidneyDisease <- tm\_map(KidneyDisease, removeNumbers)  
KidneyDisease <- tm\_map(KidneyDisease, tolower)  
KidneyDisease <- tm\_map(KidneyDisease, removeWords, stopwords("english"))  
KidneyDisease <- tm\_map(KidneyDisease, stripWhitespace)  
KidneyDisease <- tm\_map(KidneyDisease, stemDocument)  
  
dtmKidneyDisease <- DocumentTermMatrix(KidneyDisease)  
  
freq <- colSums(as.matrix(dtmKidneyDisease))

This code orders words stemmed by frequency and finds input correlations

FREQ <- data.frame(freq)  
ord <- order(freq, decreasing=TRUE)  
  
freq[head(ord, 25)]

## kidney medium associ cell serum supplement   
## 223 128 112 110 102 98   
## sodium concentr diseas univers egfr depart   
## 97 82 77 77 75 74   
## declin use purchas function renal medicine,   
## 71 68 64 64 63 61   
## sampl risk growth incid tissu rapid   
## 57 54 51 51 50 50   
## san   
## 50

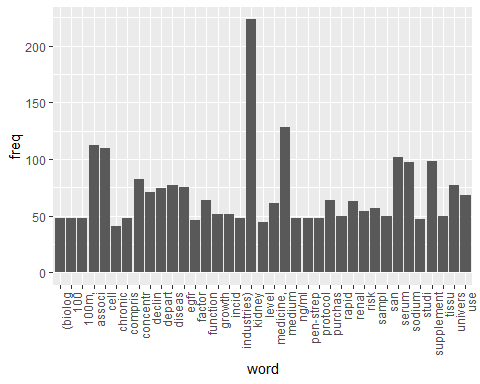
findAssocs(dtmKidneyDisease, "renal", corlimit=0.5)

## $renal  
## mice calcul (b) .e.   
## 0.70 0.69 0.68 0.68   
## accomplish area area. ascertain   
## 0.68 0.68 0.68 0.68   
## axi axial axis, biochem   
## 0.68 0.68 0.68 0.68   
## biopsy. can ckd-relat collagen   
## 0.68 0.68 0.68 0.68   
## content, content. coron deposit   
## 0.68 0.68 0.68 0.68   
## distance, easili ellips ellipsoid   
## 0.68 0.68 0.68 0.68   
## extend extent formula imag   
## 0.68 0.68 0.68 0.68   
## interstiti invasive, just make   
## 0.68 0.68 0.68 0.68   
## minor noninvas now often   
## 0.68 0.68 0.68 0.68   
## organ. parenchym pelvi picrosirius   
## 0.68 0.68 0.68 0.68   
## polar red remark risky,   
## 0.68 0.68 0.68 0.68   
## scar scarring, size size,   
## 0.68 0.68 0.68 0.68   
## sometim stain techniqu today   
## 0.68 0.68 0.68 0.68   
## treat true tubulointerstiti ultrasound,   
## 0.68 0.68 0.68 0.68   
## underestim via visual major   
## 0.68 0.68 0.68 0.65   
## obtain involv   
## 0.52 0.51

findAssocs(dtmKidneyDisease, "pain", corlimit=0.69)

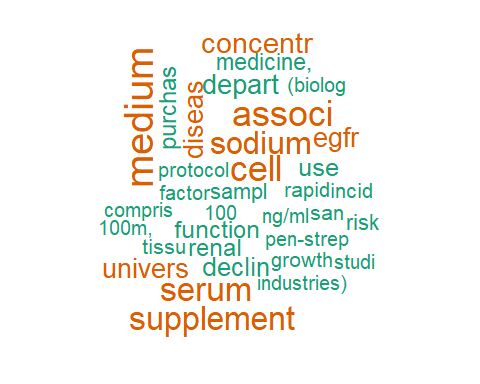
## $pain  
## (12) (23.0-28.0 (25%) (92%)   
## 0.7 0.7 0.7 0.7   
## (health (sd)ml/min/1.73 0.13-0.97) 0.97-3.07)   
## 0.7 0.7 0.7 0.7   
## 1.72; 1.9 2-fold 25.2   
## 0.7 0.7 0.7 0.7   
## 252 70-79 989 [8%])   
## 0.7 0.7 0.7 0.7   
## ab9, abc abc) aging,   
## 0.7 0.7 0.7 0.7   
## aging; alkalosis; analyzer. anesthesiolog   
## 0.7 0.7 0.7 0.7   
## bethesda, california city, composit   
## 0.7 0.7 0.7 0.7   
## de4, egfr0.55 elders: forest   
## 0.7 0.7 0.7 0.7   
## fri give harri inception.   
## 0.7 0.7 0.7 0.7   
## insight interven investigators. jh12;   
## 0.7 0.7 0.7 0.7   
## kritchevski kv5, lake least   
## 0.7 0.7 0.7 0.7   
## lf3, lost m(2), mg11,   
## 0.7 0.7 0.7 0.7   
## mj10, mmol/l mmol/l), mmol/l.   
## 0.7 0.7 0.7 0.7   
## newman pa. pa; patel   
## 0.7 0.7 0.7 0.7   
## persons. pittsburgh, predomin progression,   
## 0.7 0.7 0.7 0.7   
## ratio. rh6, rifkin salt   
## 0.7 0.7 0.7 0.7   
## sb8, separ sticht tb7,   
## 0.7 0.7 0.7 0.7   
## th2, ut. utah, venous   
## 0.7 0.7 0.7 0.7   
## wake well-funct winston-salem, yenchek   
## 0.7 0.7 0.7 0.7   
## (b) .e. accomplish area   
## 0.7 0.7 0.7 0.7   
## area. ascertain axi axial   
## 0.7 0.7 0.7 0.7   
## axis, biochem biopsy. can   
## 0.7 0.7 0.7 0.7   
## ckd-relat collagen content, content.   
## 0.7 0.7 0.7 0.7   
## coron deposit distance, easili   
## 0.7 0.7 0.7 0.7   
## ellips ellipsoid extend extent   
## 0.7 0.7 0.7 0.7   
## formula imag interstiti invasive,   
## 0.7 0.7 0.7 0.7   
## just make minor noninvas   
## 0.7 0.7 0.7 0.7   
## now often organ. parenchym   
## 0.7 0.7 0.7 0.7   
## pelvi picrosirius polar red   
## 0.7 0.7 0.7 0.7   
## remark risky, scar scarring,   
## 0.7 0.7 0.7 0.7   
## size size, sometim stain   
## 0.7 0.7 0.7 0.7   
## techniqu today treat true   
## 0.7 0.7 0.7 0.7   
## tubulointerstiti ultrasound, underestim via   
## 0.7 0.7 0.7 0.7   
## visual   
## 0.7

wf <- data.frame(word=names(freq), freq=freq)  
p <- ggplot(subset(wf, freq>40), aes(word, freq))  
p <- p + geom\_bar(stat= 'identity')   
p <- p + theme(axis.text.x=element\_text(angle=90, hjust=1))   
p

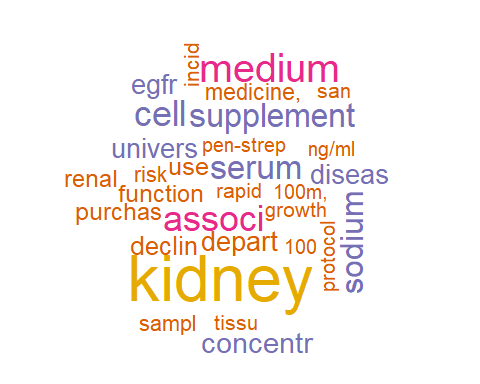


wordcloud(names(freq), freq, min.freq=45,colors=brewer.pal(3,'Dark2'))

## Warning in wordcloud(names(freq), freq, min.freq = 45, colors =  
## brewer.pal(3, : kidney could not be fit on page. It will not be plotted.



wordcloud(names(freq), freq, max.words=30,colors=brewer.pal(6,'Dark2'))



### The above stemmed the corpus, this will lemmatize the original csv file

and add the field to the table and write out to csv, followed by plot the word count frequencies that were lemmatized and the word clouds

#library(textstem)  
  
lemma <- lemmatize\_strings(auto$abstract, dictionary=lexicon::hash\_lemmas)  
  
Lemma <- as.data.frame(lemma)  
Lemma <- cbind(Lemma, auto)  
  
colnames(Lemma) <- c('lemmatizedAbstract','abstract', 'source')  
  
write.csv(Lemma, 'LemmatizedKidneyDisease.csv', row.names=FALSE)

dir.create('./KidneyDisease-Lemma')  
  
ea <- as.character(Lemma$lemmatizedAbstract)  
setwd('./KidneyDisease-Lemma')  
  
for (j in 1:length(ea)){  
 write(ea[j], paste(paste('EAL',j, sep='.'), '.txt', sep=''))  
}  
setwd('../')

KidneyDisease <- Corpus(DirSource("KidneyDisease-Lemma"))  
  
KidneyDisease

## <<SimpleCorpus>>  
## Metadata: corpus specific: 1, document level (indexed): 0  
## Content: documents: 43

#KidneyDisease <- tm\_map(KidneyDisease, removePunctuation)  
#KidneyDisease <- tm\_map(KidneyDisease, removeNumbers)  
KidneyDisease <- tm\_map(KidneyDisease, tolower)  
KidneyDisease <- tm\_map(KidneyDisease, removeWords, stopwords("english"))  
KidneyDisease <- tm\_map(KidneyDisease, stripWhitespace)  
  
dtmKidneyDisease <- DocumentTermMatrix(KidneyDisease)  
dtmKidneyDisease

## <<DocumentTermMatrix (documents: 43, terms: 2418)>>  
## Non-/sparse entries: 7417/96557  
## Sparsity : 93%  
## Maximal term length: 116  
## Weighting : term frequency (tf)

freq <- colSums(as.matrix(dtmKidneyDisease))  
  
FREQ <- data.frame(freq)  
ord <- order(freq, decreasing=TRUE)  
  
freq[head(ord, 25)]

## kidney cell medium serum sodium supplement   
## 223 142 128 102 97 96   
## egfr 100 invitrogen disease university department   
## 93 80 80 78 77 74   
## decline use function 4mg aldrich biological   
## 71 67 65 64 64 64   
## industry poly purchase sfm sigma renal   
## 64 64 64 64 64 63   
## associate   
## 62

pain <- as.data.frame(findAssocs(dtmKidneyDisease, "pain", corlimit=0.99))  
  
kidney <- as.data.frame(findAssocs(dtmKidneyDisease, "kidney", corlimit=0.65))  
  
  
treatment <- as.data.frame(findAssocs(dtmKidneyDisease, "treatment", corlimit=0.81))  
  
pain

## pain  
## 1.9 1  
## 23. 1  
## 25.2 1  
## 252 1  
## 28. 1  
## 72; 1  
## 989 1  
## ab9, 1  
## abc 1  
## age; 1  
## alkalosis; 1  
## analyzer. 1  
## anesthesiology 1  
## arterial 1  
## arterialized 1  
## bethesda, 1  
## california 1  
## city, 1  
## collaborator 1  
## collection 1  
## composition 1  
## de4, 1  
## egfr0.55 1  
## elder 1  
## elder: 1  
## forest 1  
## fry 1  
## give 1  
## harris 1  
## inception. 1  
## insight 1  
## intervene 1  
## investigator. 1  
## jh12; 1  
## kritchevsky 1  
## kv5, 1  
## lake 1  
## less 1  
## lf3, 1  
## lose 1  
## mg11, 1  
## mj10, 1  
## mmol 1  
## newman 1  
## pa. 1  
## pa; 1  
## patel 1  
## person. 1  
## pittsburgh, 1  
## predominantly 1  
## prevalent 1  
## progression, 1  
## ratio. 1  
## rh6, 1  
## rifkin 1  
## salem, 1  
## salt 1  
## sb8, 1  
## separate 1  
## sticht 1  
## tb7, 1  
## th2, 1  
## ut. 1  
## utah, 1  
## venous 1  
## wake 1  
## winston 1  
## yenchek 1

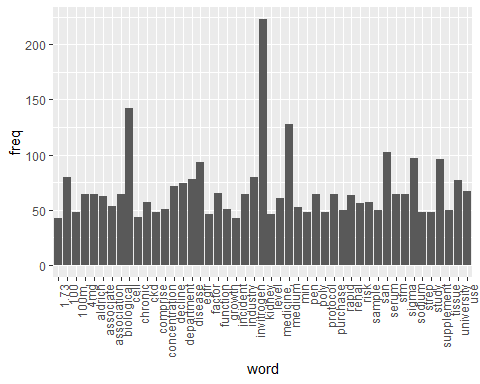
kidney

## kidney  
## function 0.72  
## albuminuria 0.67  
## ethnic 0.65  
## katz 0.65  
## washington, 0.65

treatment

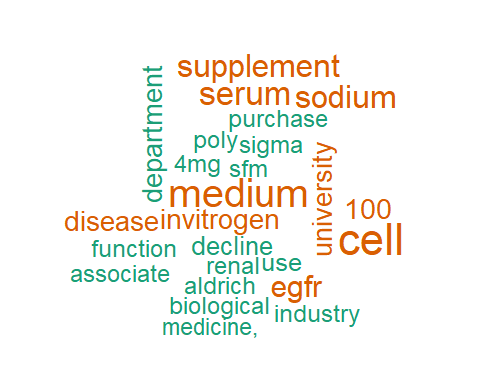
## treatment  
## cell 0.83  
## lipid 0.83  
## total 0.83

wf <- data.frame(word=names(freq), freq=freq)  
p <- ggplot(subset(wf, freq>40), aes(word, freq))  
p <- p + geom\_bar(stat= 'identity')   
p <- p + theme(axis.text.x=element\_text(angle=90, hjust=1))   
p

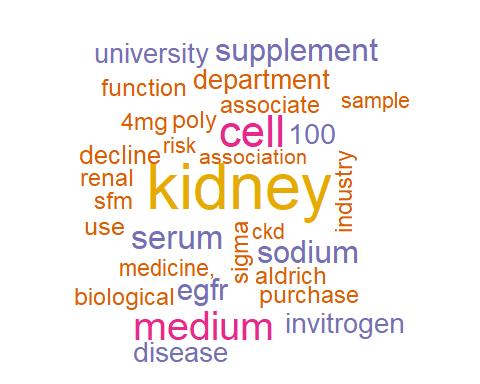


wordcloud(names(freq), freq, min.freq=60,colors=brewer.pal(3,'Dark2'))

## Warning in wordcloud(names(freq), freq, min.freq = 60, colors =  
## brewer.pal(3, : kidney could not be fit on page. It will not be plotted.



wordcloud(names(freq), freq, max.words=30,colors=brewer.pal(6,'Dark2'))



### Now for some machine learning on predicting renal or kidney type from these samples of 12 healthy and 4 renal disease using the top 20 genes of up/down/fold change genes

FC <- Top20\_FC[,4:20]  
UP <- Top20\_up[,5:20]  
DOWN <- Top20\_down[,5:20]  
  
FC$gene <- row.names(FC)  
UP$gene <- row.names(UP)  
DOWN$gene <- row.names(DOWN)

t\_tall <- rbind(FC,UP,DOWN)  
t\_tall <- t\_tall[!duplicated(t\_tall$gene),]

#remove the statistical observations

t\_tall <- t(t\_tall)  
row.names(t\_tall)

## [1] "renal\_0" "renal\_3"   
## [3] "renal\_6" "renal\_9"   
## [5] "X5\_AK125p1\_Adh.count" "X6\_AK125p1\_SPH3d.count"  
## [7] "X7\_AK125p1\_SPH6d.count" "X8\_AK125p1\_SPH9d.count"  
## [9] "AK82p2Adh" "AK82p3SPH3d"   
## [11] "AK82p3SPH6d" "AK82p3SPH10d"   
## [13] "AK86p1Adh" "AK86p2.SPH3d"   
## [15] "AK86p2.SPH6d" "AK86p2.SPH10d"   
## [17] "gene"

dim(t\_tall)

## [1] 17 56

t\_tall <- t\_tall[1:16,] #remove gene row

renal2 <- as.data.frame(rep('renal disease',4))  
healthy2 <- as.data.frame(rep('healthy',12))  
colnames(renal2) <- 'type'  
colnames(healthy2) <- 'type'  
  
type <- rbind(renal2, healthy2)  
  
ML\_set <- cbind(type,t\_tall)  
dim(ML\_set)

## [1] 16 57

ML\_set2 <- ML\_set[,2:57]  
  
for (i in 1:ncol(ML\_set2)){  
 ML\_set2[,i] <- as.numeric(as.character(ML\_set2[,i]))  
}  
  
ML\_set2$type <- ML\_set$type  
ML\_set <- ML\_set2[,c(57,1:56)]

The data set that will be used for Machine Learning will predict if the sample is renal disease or healthy. The samples will have to be randomized into 80% train and 20% test

library(caret)  
library(randomForest)  
library(MASS)  
library(gbm)  
library(dplyr)

set.seed(189678345)

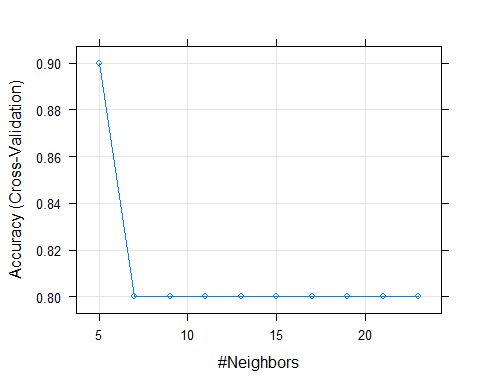
inTrain <- createDataPartition(y=ML\_set$type, p=0.8, list=FALSE)  
  
trainingSet <- ML\_set[inTrain,]  
testingSet <- ML\_set[-inTrain,]

### KNN

system.time(knnMod <- train(type ~ .,  
 method='knn', preProcess=c('center','scale'),  
 tuneLength=10, trControl=trainControl(method='cv'), data=trainingSet))

## user system elapsed   
## 2.39 0.02 2.55

plot(knnMod)



The predicted results with KNN

predKNN <- predict(knnMod, testingSet)  
predKNN

## [1] healthy healthy  
## Levels: renal disease healthy

The actual values in the testing set

testingSet$type

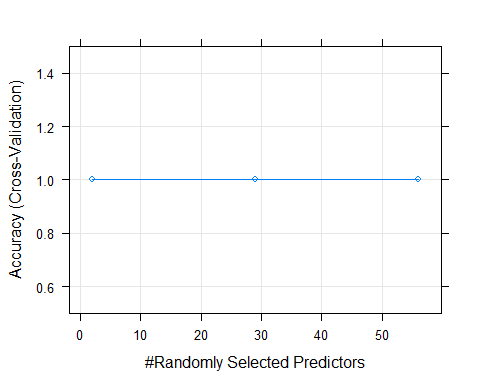
## [1] healthy healthy  
## Levels: renal disease healthy

### Random Forest

system.time(rfMod <- train(type ~., method='rf', data=(trainingSet),   
 trControl=trainControl(method='cv'), number=5))

## user system elapsed   
## 1.43 0.00 1.51

plot(rfMod)

 The predicted Random Forest results and the actual results

predRF <- predict(rfMod, testingSet)  
predRF

## [1] healthy healthy  
## Levels: renal disease healthy

testingSet$type

## [1] healthy healthy  
## Levels: renal disease healthy