PCA And Sequence Alignment

Abdullah Taman

2024-03-09

```r
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```r
library(rentrez)
```

```
## Warning: package 'rentrez' was built under R version 4.3.3
```

```r
library(seqinr)
```

```
## Warning: package 'seqinr' was built under R version 4.3.3
```

```
##
## Attaching package: 'seqinr'
##
## The following object is masked from 'package:dplyr':
##
##     count
```

```r
library(Biostrings)
```

```
## Loading required package: BiocGenerics
##
## Attaching package: 'BiocGenerics'
##
## The following objects are masked from 'package:lubridate':
##
##     intersect, setdiff, union
##
## The following objects are masked from 'package:dplyr':
```

```
##
##     combine, intersect, setdiff, union
##
## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs
##
## The following objects are masked from 'package:base':
##
##     anyDuplicated, aperm, append, as.data.frame, basename, cbind,
##     colnames, dirname, do.call, duplicated, eval, evalq, Filter, Find,
##     get, grep, grepl, intersect, is.unsorted, lapply, Map, mapply,
##     match, mget, order, paste, pmax, pmax.int, pmin, pmin.int,
##     Position, rank, rbind, Reduce, rownames, sapply, setdiff, sort,
##     table, tapply, union, unique, unsplit, which.max, which.min
##
## Loading required package: S4Vectors
## Loading required package: stats4
##
## Attaching package: 'S4Vectors'
##
## The following objects are masked from 'package:lubridate':
##
##     second, second<-
##
## The following objects are masked from 'package:dplyr':
##
##     first, rename
##
## The following object is masked from 'package:tidyr':
##
##     expand
##
## The following object is masked from 'package:utils':
##
##     findMatches
##
## The following objects are masked from 'package:base':
##
##     expand.grid, I, unname
##
## Loading required package: IRanges
##
## Attaching package: 'IRanges'
##
## The following object is masked from 'package:lubridate':
##
##     %within%
##
## The following objects are masked from 'package:dplyr':
##
##     collapse, desc, slice
##
## The following object is masked from 'package:purrr':
```

```
##
##     reduce
##
## The following object is masked from 'package:grDevices':
##
##     windows
##
## Loading required package: XVector
##
## Attaching package: 'XVector'
##
## The following object is masked from 'package:purrr':
##
##     compact
##
## Loading required package: GenomeInfoDb
##
## Attaching package: 'Biostrings'
##
## The following object is masked from 'package:seqinr':
##
##     translate
##
## The following object is masked from 'package:base':
##
##     strsplit
```

```r
library(ggplot2)
library("FactoMineR")
```

```
## Warning: package 'FactoMineR' was built under R version 4.3.3
```

```r
library(ggcorrplot)
```

```
## Warning: package 'ggcorrplot' was built under R version 4.3.3
```

```r
library('corrr')
```

```
## Warning: package 'corrr' was built under R version 4.3.3
```

# Part One

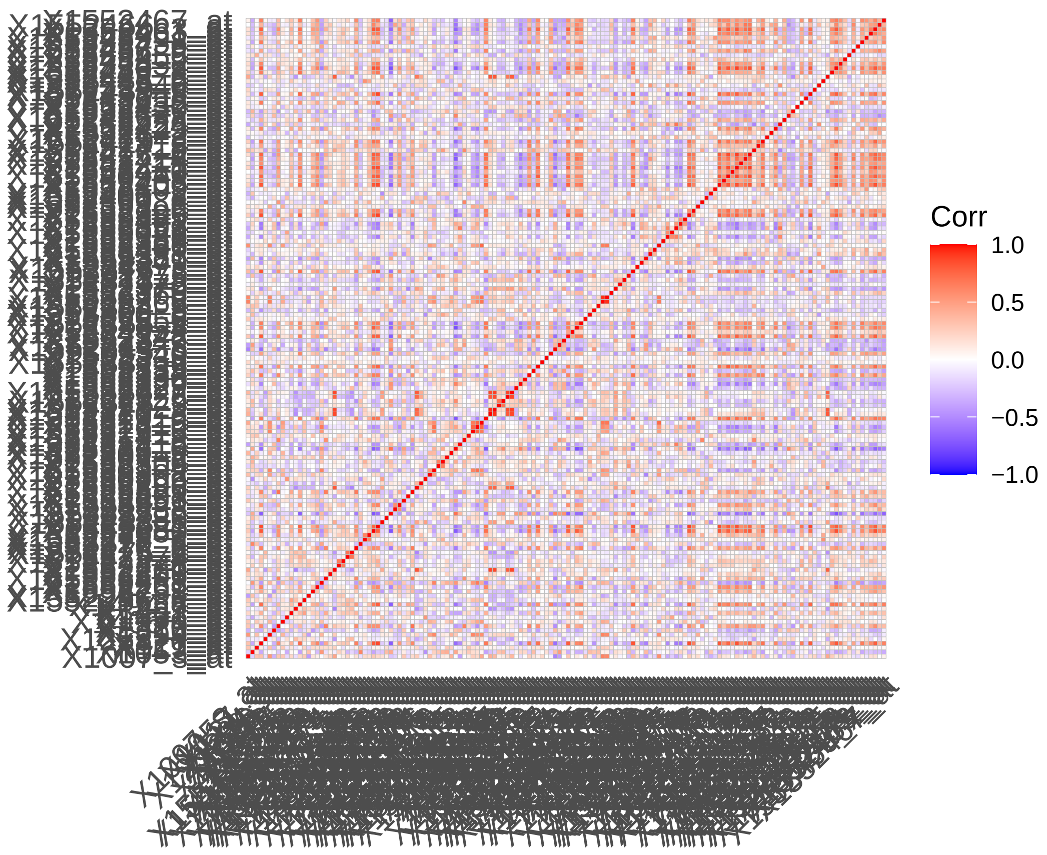
First performing the PCA

```r
cancer <- read.csv("J:/champions work/datasets/BrainCancerMin.csv", header = TRUE)
?princomp()
```

```
## starting httpd help server ... done
```

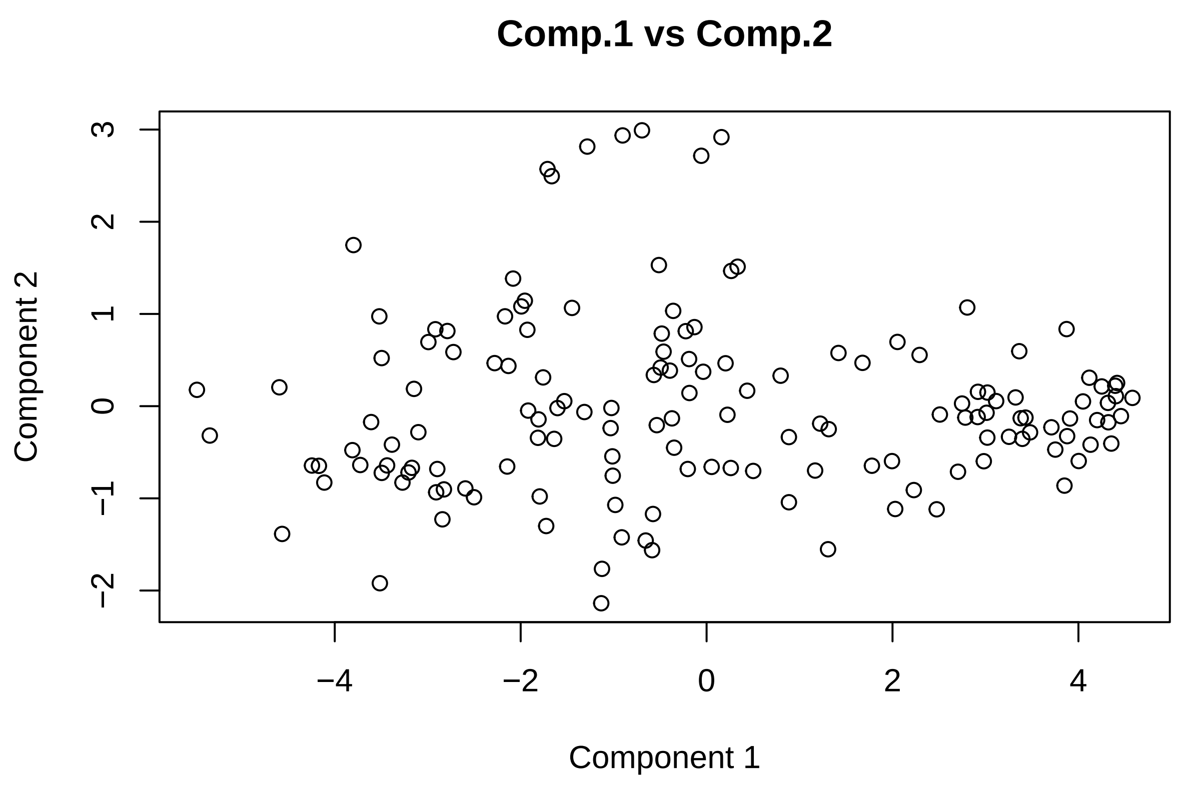
Now we'll perform the PCA I faced an issue with the code, This error: Error in princomp.default(remove_type, cor = TRUE, scores = TRUE) : 'princomp' can only be used with more units than variables

```
remove_type <- subset(cancer, select = c(3:ncol(cancer)))
data_normalized <- scale(remove_type)
corr_matrix <- cor(data_normalized)
ggcorrplot(corr_matrix)
```

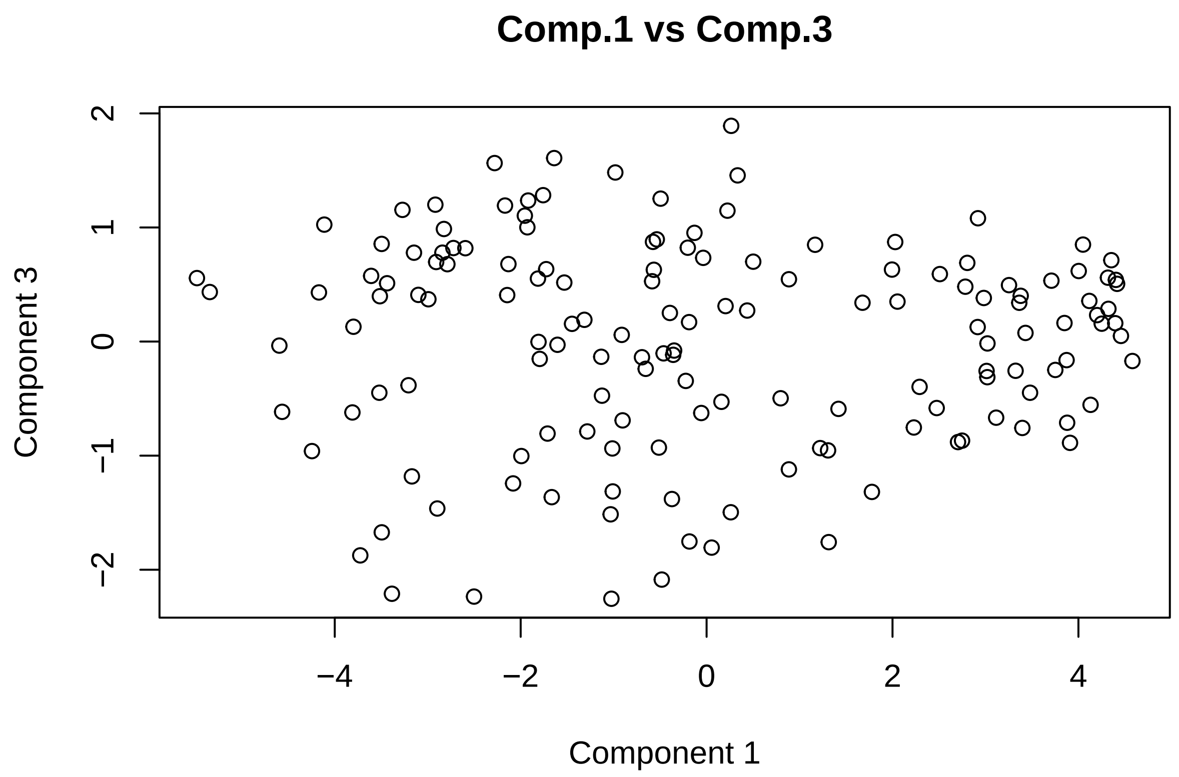


```
data.pca <- princomp(as.data.frame(corr_matrix))
#summary(data.pca)
```

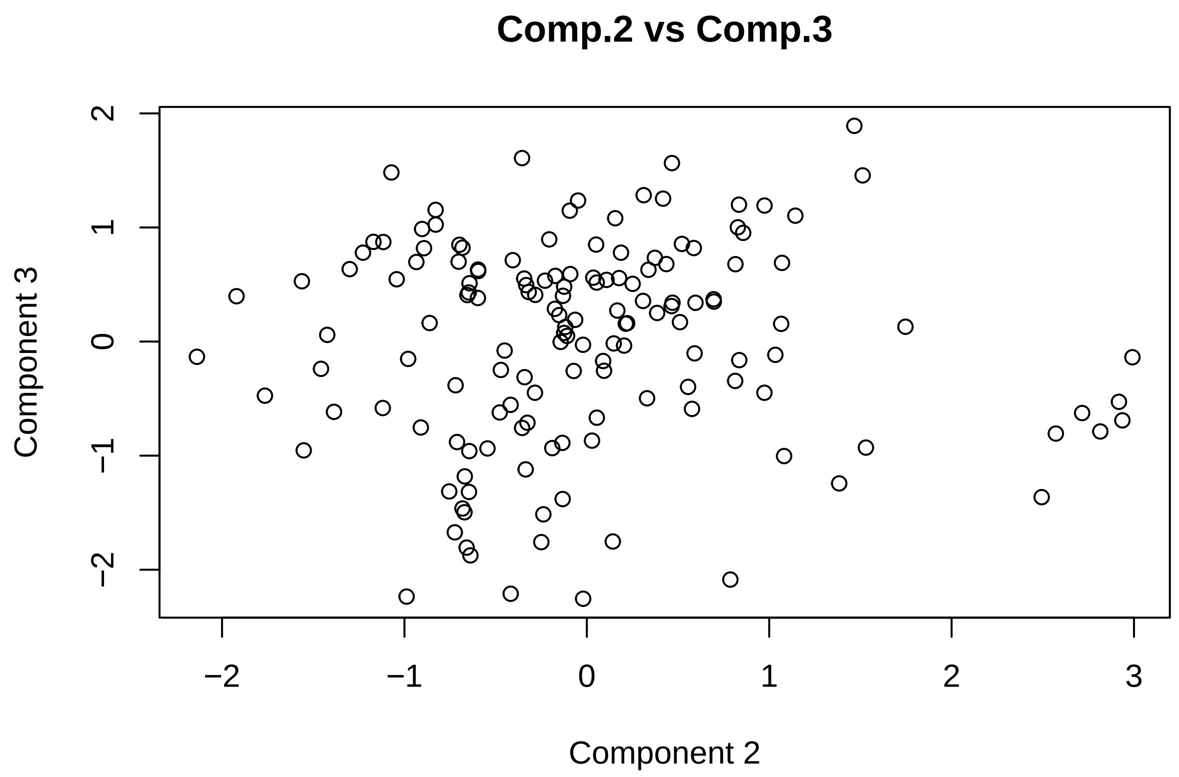
```
# Extract principal component scores
pc_scores <- data.pca$scores

# Plot Comp.1 vs Comp.2
plot(pc_scores[,1], pc_scores[,2], xlab = "Component 1", ylab = "Component 2",
     main = "Comp.1 vs Comp.2")
```



```r
# Plot Comp.1 vs Comp.3
plot(pc_scores[,1], pc_scores[,3], xlab = "Component 1", ylab = "Component 3",
     main = "Comp.1 vs Comp.3")
```

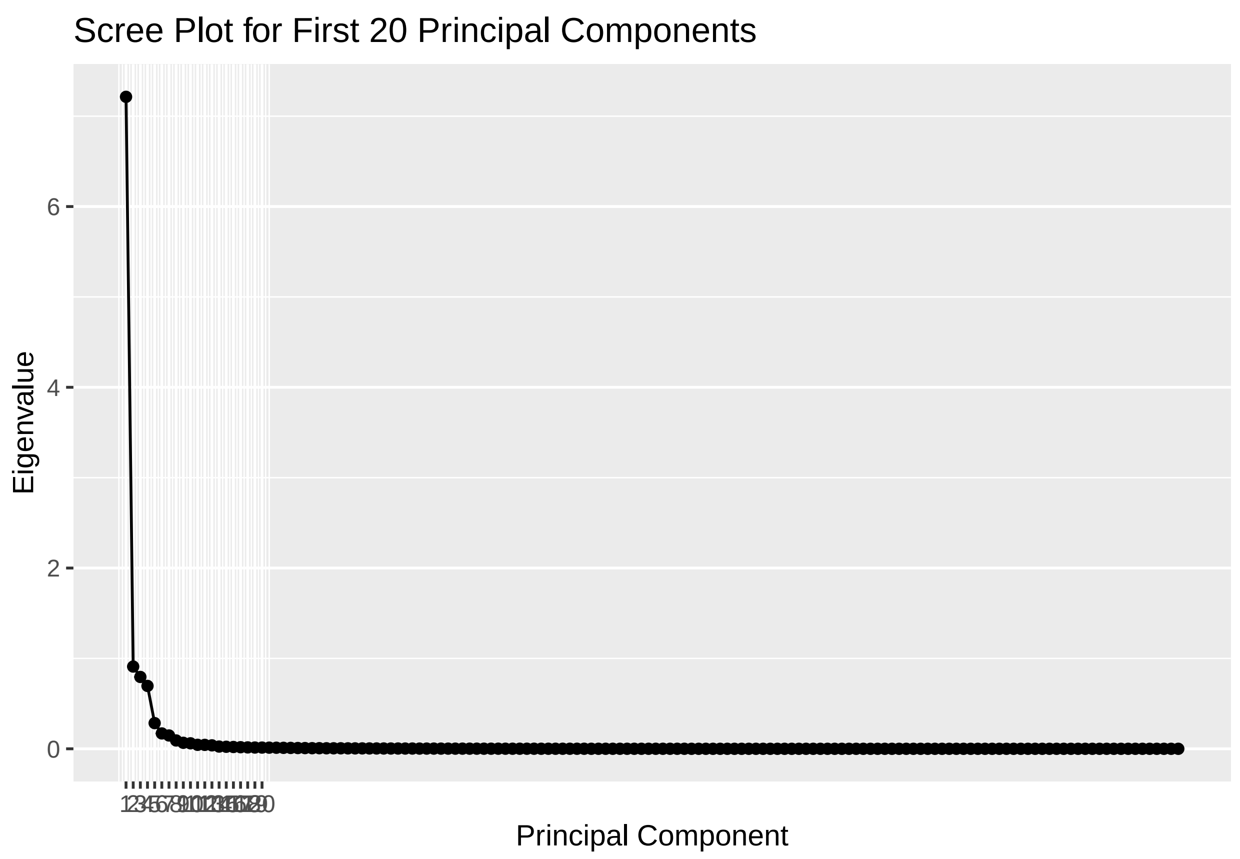


```r
# Plot Comp.2 vs Comp.3
plot(pc_scores[,2], pc_scores[,3], xlab = "Component 2", ylab = "Component 3",
     main = "Comp.2 vs Comp.3")
```



Component 1 vs Component 2 is the best, why ? You could see that the number of points that are overlapped is the least number in it.

```r
# Extract eigenvalues
eigenvalues <- data.pca$sdev^2

# Create data frame for plotting
scree_data <- data.frame(Component = 1:length(eigenvalues), Eigenvalue = eigenvalues)

# Plot scree plot
ggplot(scree_data, aes(x = Component, y = Eigenvalue)) +
  geom_point() +
  geom_line() +
  scale_x_continuous(breaks = seq(1, 20, by = 1)) +  # Adjust x-axis breaks for better readability
  labs(x = "Principal Component", y = "Eigenvalue", title = "Scree Plot for First 20 Principal Componen
```



In multivariate statistics, a scree plot is a line plot of the eigenvalues of factors or principal components in an analysis.[1] The scree plot is used to determine the number of factors to retain in an exploratory factor analysis (FA) or principal components to keep in a principal component analysis (PCA). We could see that starting from the 6th eigen value the eigen values are so small that they don't affect anything in the results.

# Part Two

```r
diab <- read.csv("J:/champions work/datasets/diabetes_prediction_dataset.csv")
```

```r
allele1 <- substr(diab$alleles, 1, 1)  # First character of alleles
allele2 <- substr(diab$alleles, 2, 2)  # Second character of alleles

allele_counts <- table(No_Diabetes = c(allele1, allele2), Diabetes = rep(diab$diabetes, 2))

# Convert table to data frame for easier manipulation
allele_counts_df <- as.data.frame(allele_counts)

# Print the resulting table
print(allele_counts_df)
```

```
##   No_Diabetes Diabetes  Freq
## 1           A        0 91660
## 2           C        0 91340
## 3           A        1  8499
## 4           C        1  8501
```

```r
allele_counts <- matrix(c(91660, 8499, 91340, 8501), ncol = 2, byrow = TRUE)

# Perform Fisher's exact test
fisher_result <- fisher.test(allele_counts)

# Extract p-value
p_value <- fisher_result$p.value

# Report the p-value
print(p_value)
```

```
## [1] 0.816139
```

It's not significant at all, .81 is very big. Which means that the association is not significant.

```r
# Extract BMI samples for each allele family
BMI_AA <- subset(diab, alleles == "AA")$bmi
BMI_AC <- subset(diab, alleles == "AC")$bmi
BMI_CC <- subset(diab, alleles == "CC")$bmi

# Perform t-test between BMI samples for AA and AC allele families
t_test_AA_AC <- t.test(BMI_AA, BMI_AC)

# Perform t-test between BMI samples for AA and CC allele families
t_test_AA_CC <- t.test(BMI_AA, BMI_CC)

# Perform t-test between BMI samples for AC and CC allele families
t_test_AC_CC <- t.test(BMI_AC, BMI_CC)

# Report p-values
p_value_AA_AC <- t_test_AA_AC$p.value
p_value_AA_CC <- t_test_AA_CC$p.value
p_value_AC_CC <- t_test_AC_CC$p.value

# Print p-values
print(p_value_AA_AC)
```

```
## [1] 0.5125191
```

```r
print(p_value_AA_CC)
```

```
## [1] 0.8228364
```

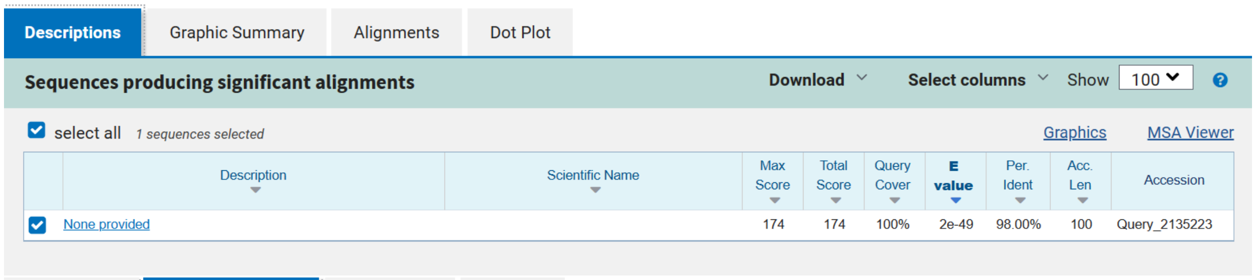
```r
print(p_value_AC_CC)
```

```
## [1] 0.6691533
```

It's clear that there is no real significant difference between the family of alleles when it's related to BMI phenotype.

Part Three

There are two mismatches.



Descriptions | Graphic Summary | Alignments | Dot Plot

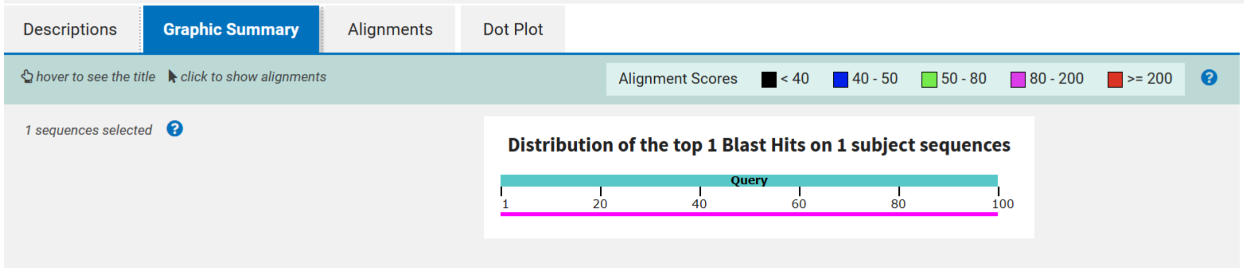
Sequences producing significant alignments

Download | Select columns | Show 100

✓ select all *1 sequences selected*

Graphics | MSA Viewer

	Description	Scientific Name	Max Score	Total Score	Query Cover	E value	Per. Ident	Acc. Len	Accession
✓	None provided		174	174	100%	2e-49	98.00%	100	Query_2135223



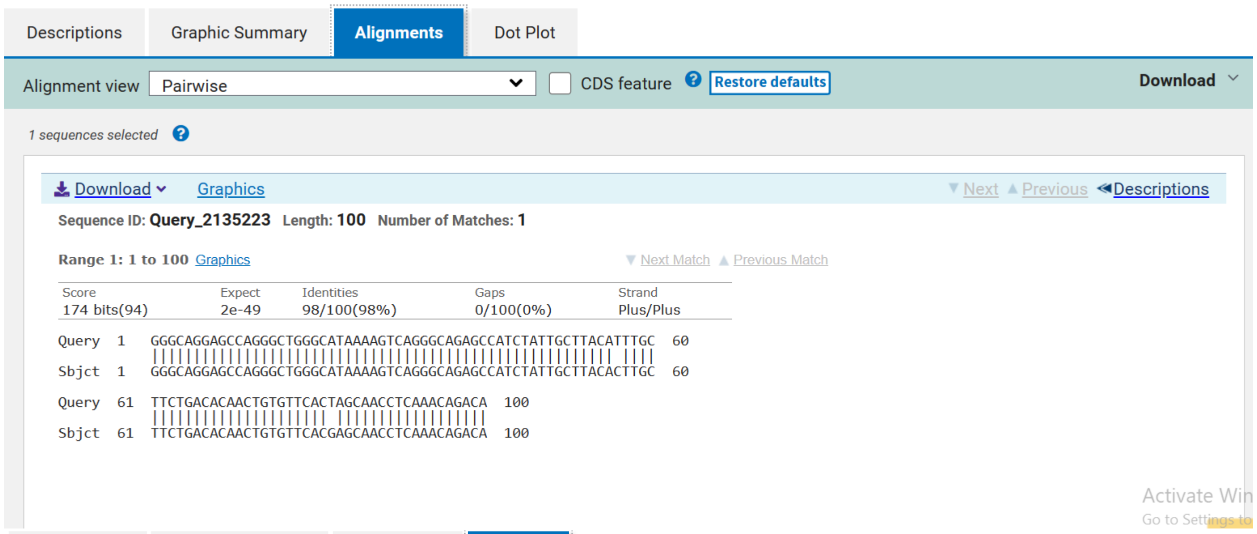
Descriptions | Graphic Summary | Alignments | Dot Plot

hover to see the title *click to show alignments*

Alignment Scores ■ < 40 🟦 40 - 50 🟩 50 - 80 🟪 80 - 200 🟥 >= 200

1 sequences selected

Distribution of the top 1 Blast Hits on 1 subject sequences



Descriptions | Graphic Summary | Alignments | Dot Plot

Alignment view: Pairwise | CDS feature | Restore defaults | Download

1 sequences selected

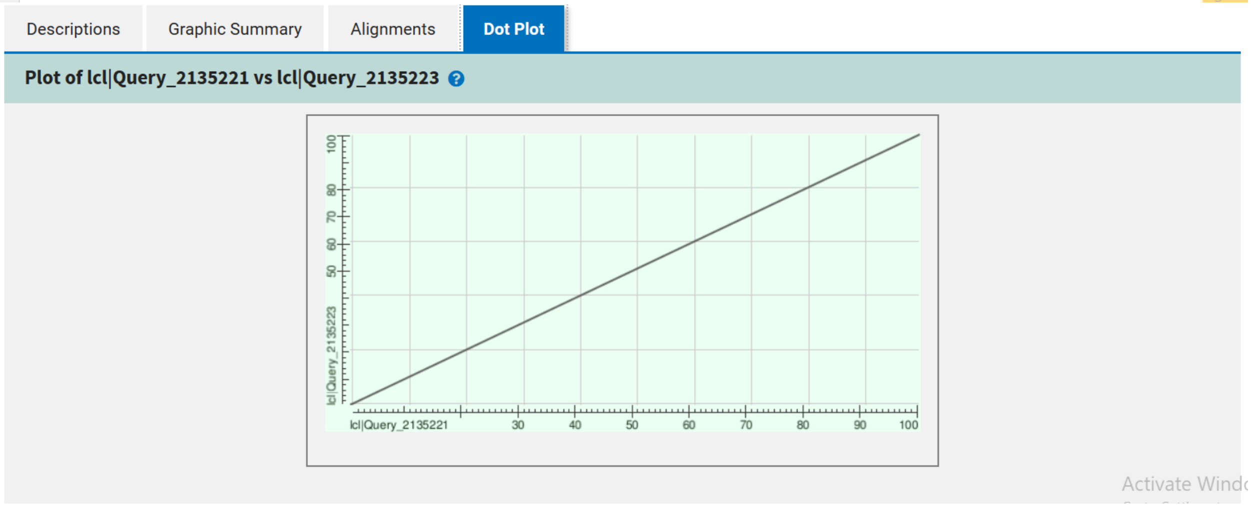
Download | Graphics | Next | Previous | Descriptions

Sequence ID: Query_2135223 Length: 100 Number of Matches: 1

Range 1: 1 to 100 Graphics | Next Match | Previous Match

Score	Expect	Identities	Gaps	Strand
174 bits(94)	2e-49	98/100(98%)	0/100(0%)	Plus/Plus

```
Query  1    GGGCAGGAGCCAGGGCTGGGCATAAAAGTCAGGGCAGAGCCATCTATTGCTTACATTTGC  60
            ||||||||||||||||||||||||||||||||||||||||||||||||||||||| ||||
Sbjct  1    GGGCAGGAGCCAGGGCTGGGCATAAAAGTCAGGGCAGAGCCATCTATTGCTTACACTTGC  60

Query  61   TTCTGACACAACTGTGTTCACTAGCAACCTCAAACAGACA  100
            ||||||||||||||||||||| ||||||||||||||||||
Sbjct  61   TTCTGACACAACTGTGTTCACGAGCAACCTCAAACAGACA  100
```



Descriptions | Graphic Summary | Alignments | Dot Plot

Plot of lcl|Query_2135221 vs lcl|Query_2135223

```
seq1 <- entrez_fetch(db = "nucleotide", id = "NG_050578.1", rettype = "fasta")
seq2 <- entrez_fetch(db = "nucleotide", id = "X03562.1", rettype = "fasta")
seq1
```

```
## [1] ">NG_050578.1 Homo sapiens INS-IGF2 readthrough (INS-IGF2), RefSeqGene on chromosome 11\nGAGGTGC
```

```
seq1_lines <- unlist(strsplit(seq1, "\n"))
seq1_lines <- seq1_lines[!grepl(">", seq1_lines)]
seq1_clean <- paste(seq1_lines, collapse = "")

seq2_lines <- unlist(strsplit(seq2, "\n"))
seq2_lines <- seq2_lines[!grepl(">", seq2_lines)]
seq2_clean <- paste(seq2_lines, collapse = "")

seq1 <- DNAStringSet(seq1_clean)
seq2 <- DNAStringSet(seq2_clean)

print(seq1)
```

```
## DNAStringSet object of length 1:
##     width seq
## [1] 39098 GAGGTGCGGATCCTGGGCGGCCAGGGAAGGTCTC...CCATCCCTCCACTCATCCATCCACTCATCCCTC
```

```
print(seq2)
```

```
## DNAStringSet object of length 1:
##     width seq
## [1]  8837 CCCAACCCCGCGCACAGCGGGCACTGGTTTCGGG...TCTCCCTTCTCACGGGAATTTTCAGGGTAAACT
```

```
freq_seq1 <- alphabetFrequency(seq1)
freq_seq2 <- alphabetFrequency(seq2)
print(freq_seq1)
```

```
##         A     C     G    T M R W S Y K V H D B N - + .
## [1,] 7355 12386 11660 7697 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

```
print(freq_seq2)
```

```
##         A    C    G    T M R W S Y K V H D B  N - + .
## [1,] 1388 3037 2697 1685 0 0 0 0 0 0 0 0 0 0 30 0 0 0
```

```
seq1_has_gaps_ambiguous <- any(names(freq_seq1) %in% c("-", "N"))
seq2_has_gaps_ambiguous <- any(names(freq_seq2) %in% c("-", "N"))
print(seq1_has_gaps_ambiguous)
```

```
## [1] FALSE
```

```r
print(seq2_has_gaps_ambiguous)
```

```
## [1] FALSE
```

this means that the two sequences are clean

```r
seq1_has_gaps_ambiguous <- any(grepl("[-N]", seq1))
seq2_has_gaps_ambiguous <- any(grepl("[-N]", seq2))

# 2. Remove gaps and ambiguous bases from sequences
seq1_cleaned <- gsub("[-N]", "", seq1)
seq2_cleaned <- gsub("[-N]", "", seq2)
seq1_length_before <- nchar(seq1)
seq2_length_before <- nchar(seq2)
seq1_length_after <- nchar(seq1_cleaned)
seq2_length_after <- nchar(seq2_cleaned)
print(paste("Sequence 1 length before cleaning:", seq1_length_before))
```

```
## [1] "Sequence 1 length before cleaning: 39098"
```

```r
print(paste("Sequence 2 length before cleaning:", seq2_length_before))
```

```
## [1] "Sequence 2 length before cleaning: 8837"
```

```r
print(paste("Sequence 1 length after cleaning:", seq1_length_after))
```

```
## [1] "Sequence 1 length after cleaning: 39098"
```

```r
print(paste("Sequence 2 length after cleaning:", seq2_length_after))
```

```
## [1] "Sequence 2 length after cleaning: 8807"
```

For confirmation, now we're 100% percent that they were clean. We'll work with the seq1, seq2.

```r
run_pairwise_alignment <- function(seq1, seq2) {
  # Run Pairwise Local Alignment
  alignment <- pairwiseAlignment(seq1, seq2, substitutionMatrix = "BLOSUM62", gapOpening = -10, gapExte

  # Extract alignment score
  alignment_score <- score(alignment)

  # Calculate width of each sequence before and after alignment
  seq1_width_before <- nchar(seq1)
  seq2_width_before <- nchar(seq2)
  seq1_width_after <- nchar(pattern(alignment))
  seq2_width_after <- nchar(subject(alignment))

  # Return results
  return(list(
    alignment_score = alignment_score,
    seq1_width_before = seq1_width_before,
    seq2_width_before = seq2_width_before,
    seq1_width_after = seq1_width_after,
    seq2_width_after = seq2_width_after,
    alignment = alignment
  ))
}
```

```
alignment_result <- run_pairwise_alignment(seq1, seq2)
```

```
# Report results
print("Alignment Score:")
```

```
## [1] "Alignment Score:"
```

```
print(alignment_result$alignment_score)
```

```
## [1] 25861
```

```
print("Width of Sequences Before Alignment:")
```

```
## [1] "Width of Sequences Before Alignment:"
```

```
print(paste("Sequence 1:", alignment_result$seq1_width_before))
```

```
## [1] "Sequence 1: 39098"
```

```
print(paste("Sequence 2:", alignment_result$seq2_width_before))
```

```
## [1] "Sequence 2: 8837"
```

```
print("Width of Sequences After Alignment:")
```

```
## [1] "Width of Sequences After Alignment:"
```

```
print(paste("Sequence 1:", alignment_result$seq1_width_after))
```

```
## [1] "Sequence 1: 8920"
```

```
print(paste("Sequence 2:", alignment_result$seq2_width_after))
```

```
## [1] "Sequence 2: 8920"
```

```r
print(nmismatch(alignment_result$alignment))
```

```
## [1] 98
```

```r
print(mismatchTable(alignment_result$alignment))
```

```
##    PatternId PatternStart PatternEnd PatternSubstring SubjectStart SubjectEnd
## 1          1        25041      25041                C           58         58
## 2          1        25042      25042                C           59         59
## 3          1        25221      25221                T          238        238
## 4          1        25222      25222                T          239        239
## 5          1        25449      25449                T          466        466
## 6          1        25727      25727                G          733        733
## 7          1        25759      25759                G          765        765
## 8          1        25780      25780                A          785        785
## 9          1        25794      25794                G          799        799
## 10         1        26198      26198                G         1200       1200
## 11         1        26262      26262                C         1264       1264
## 12         1        26263      26263                G         1265       1265
## 13         1        26533      26533                A         1535       1535
## 14         1        26755      26755                G         1756       1756
## 15         1        28038      28038                C         3036       3036
## 16         1        28184      28184                C         3178       3178
## 17         1        28185      28185                T         3179       3179
## 18         1        28186      28186                G         3180       3180
## 19         1        28187      28187                G         3181       3181
## 20         1        28188      28188                C         3182       3182
## 21         1        28189      28189                G         3183       3183
## 22         1        28190      28190                G         3184       3184
## 23         1        28191      28191                C         3185       3185
## 24         1        28192      28192                G         3186       3186
## 25         1        28193      28193                G         3187       3187
## 26         1        28194      28194                A         3188       3188
## 27         1        28195      28195                G         3189       3189
## 28         1        28198      28198                G         3190       3190
## 29         1        28199      28199                G         3191       3191
## 30         1        28200      28200                G         3192       3192
## 31         1        28201      28201                G         3193       3193
## 32         1        28202      28202                G         3194       3194
## 33         1        28203      28203                T         3195       3195
## 34         1        28204      28204                G         3196       3196
## 35         1        28205      28205                G         3197       3197
## 36         1        28206      28206                G         3198       3198
## 37         1        28207      28207                G         3199       3199
## 38         1        28208      28208                T         3200       3200
## 39         1        28209      28209                G         3201       3201
## 40         1        28210      28210                G         3202       3202
## 41         1        28211      28211                G         3203       3203
## 42         1        28213      28213                G         3205       3205
## 43         1        28218      28218                G         3210       3210
## 44         1        28249      28249                C         3217       3217
## 45         1        28383      28383                A         3341       3341
## 46          1       28426      28426                T         3386         3386
## 47          1       28483      28483                T         3443         3443
## 48          1       28626      28626                A         3588         3588
## 49          1       28627      28627                G         3589         3589
## 50          1       28628      28628                T         3590         3590
## 51          1       28629      28629                G         3591         3591
## 52          1       28634      28634                G         3596         3596
## 53          1       28638      28638                G         3600         3600
## 54          1       28695      28695                C         3661         3661
## 55          1       28829      28829                A         3793         3793
## 56          1       28856      28856                G         3819         3819
## 57          1       28930      28930                C         3891         3891
## 58          1       29345      29345                T         4309         4309
## 59          1       29346      29346                T         4310         4310
## 60          1       29467      29467                A         4431         4431
## 61          1       29510      29510                C         4478         4478
## 62          1       29511      29511                T         4479         4479
## 63          1       29647      29647                T         4614         4614
## 64          1       29662      29662                G         4629         4629
## 65          1       29728      29728                G         4694         4694
## 66          1       29834      29834                C         4798         4798
## 67          1       30198      30198                A         5160         5160
## 68          1       30210      30210                G         5172         5172
## 69          1       30212      30212                A         5174         5174
## 70          1       30213      30213                T         5175         5175
## 71          1       30284      30284                G         5245         5245
## 72          1       30396      30396                G         5356         5356
## 73          1       30853      30853                T         5813         5813
## 74          1       30905      30905                G         5866         5866
## 75          1       30910      30910                A         5871         5871
## 76          1       31090      31090                C         6051         6051
## 77          1       31091      31091                T         6052         6052
## 78          1       31092      31092                C         6053         6053
## 79          1       31227      31227                C         6188         6188
## 80          1       31587      31587                C         6548         6548
## 81          1       31897      31897                T         6855         6855
## 82          1       32135      32135                T         7096         7096
## 83          1       32165      32165                G         7126         7126
## 84          1       32323      32323                G         7282         7282
## 85          1       32325      32325                C         7284         7284
## 86          1       32342      32342                T         7301         7301
## 87          1       32395      32395                A         7354         7354
## 88          1       32434      32434                T         7393         7393
## 89          1       32473      32473                C         7432         7432
## 90          1       32474      32474                C         7433         7433
## 91          1       32475      32475                C         7434         7434
## 92          1       32476      32476                C         7435         7435
## 93          1       32477      32477                C         7436         7436
## 94          1       32481      32481                A         7440         7440
## 95          1       32501      32501                A         7461         7461
## 96          1       33806      33806                A         8765         8765
## 97          1       33835      33835                T         8794         8794
## 98          1       33836      33836                G         8795         8795
##    SubjectSubstring
## 1                 G
## 2                 T
## 3                 N
## 4                 N
## 5                 C
## 6                 N
## 7                 A
## 8                 G
## 9                 A
## 10                A
## 11                G
## 12                C
## 13                G
## 14                C
## 15                G
## 16                N
## 17                N
## 18                N
## 19                N
## 20                N
## 21                N
## 22                N
## 23                N
## 24                N
## 25                N
## 26                N
## 27                N
## 28                N
## 29                N
## 30                N
## 31                N
## 32                N
## 33                N
## 34                N
## 35                N
## 36                N
## 37                N
## 38                N
## 39                N
## 40                N
## 41                N
## 42                A
## 43                C
## 44                A
## 45                G
## 46                C
## 47                C
## 48                C
## 49                A
## 50                G
## 51                T
## 52                A
## 53                T
## 54                A
## 55                T
## 56                T
## 57                T
## 58                C
## 59                G
## 60                G
## 61                T
## 62                C
## 63                A
## 64                A
## 65                A
## 66                T
## 67                G
## 68                T
## 69                G
## 70                A
## 71                A
## 72                C
## 73                A
## 74                T
## 75                T
## 76                T
## 77                C
## 78                T
## 79                T
## 80                A
## 81                C
## 82                C
## 83                C
## 84                A
## 85                T
## 86                C
## 87                G
## 88                N
## 89                G
## 90                G
## 91                G
## 92                G
## 93                G
## 94                G
## 95                G
## 96                G
## 97                G
## 98                T
```