Lab 2

Omar Aldawy Ibrahim Aldawy

2025-03-03

Part 1: Gene Expression Analysis

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyr)
library(ggplot2)
```

Task 1.1 Gene Expression Calculation

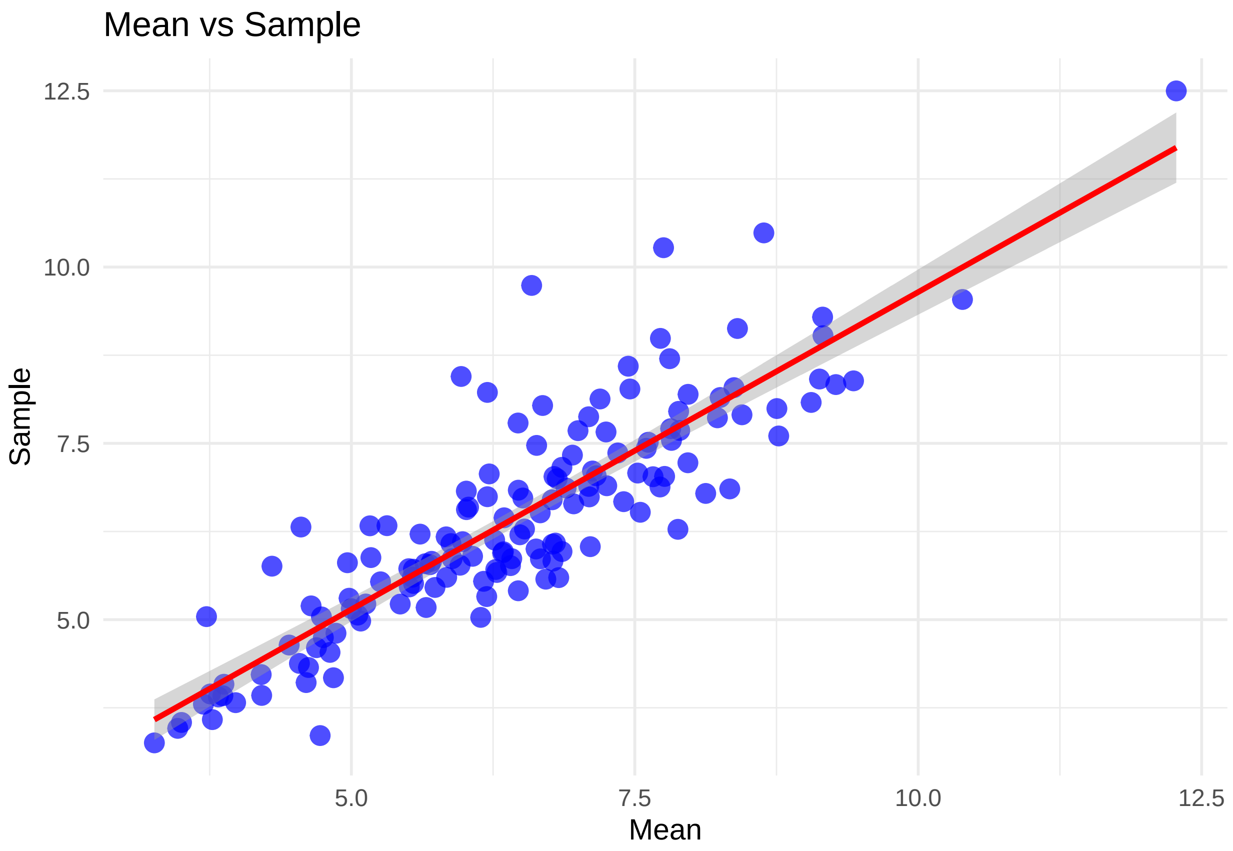
Calculate the mean gene expression for each gene across all types into a new dataframe.

```
brain_cancer_dataset <- read.csv("BrainCancerMin.csv")

genes_mean <- brain_cancer_dataset %>%
  gather(key = "Gene", value = "Mean", -c(1, 2)) %>%
  group_by(Gene) %>%
  summarise(Mean = mean(Mean))

first_row_values <- brain_cancer_dataset %>%
  slice(1) %>%
  select(-c(1:2)) %>%
  gather(key = "Gene", value = "Sample")

genes_mean <- left_join(genes_mean, first_row_values, by = "Gene")
```

The trend between the Mean Gene Expression and Sample Gene Expression

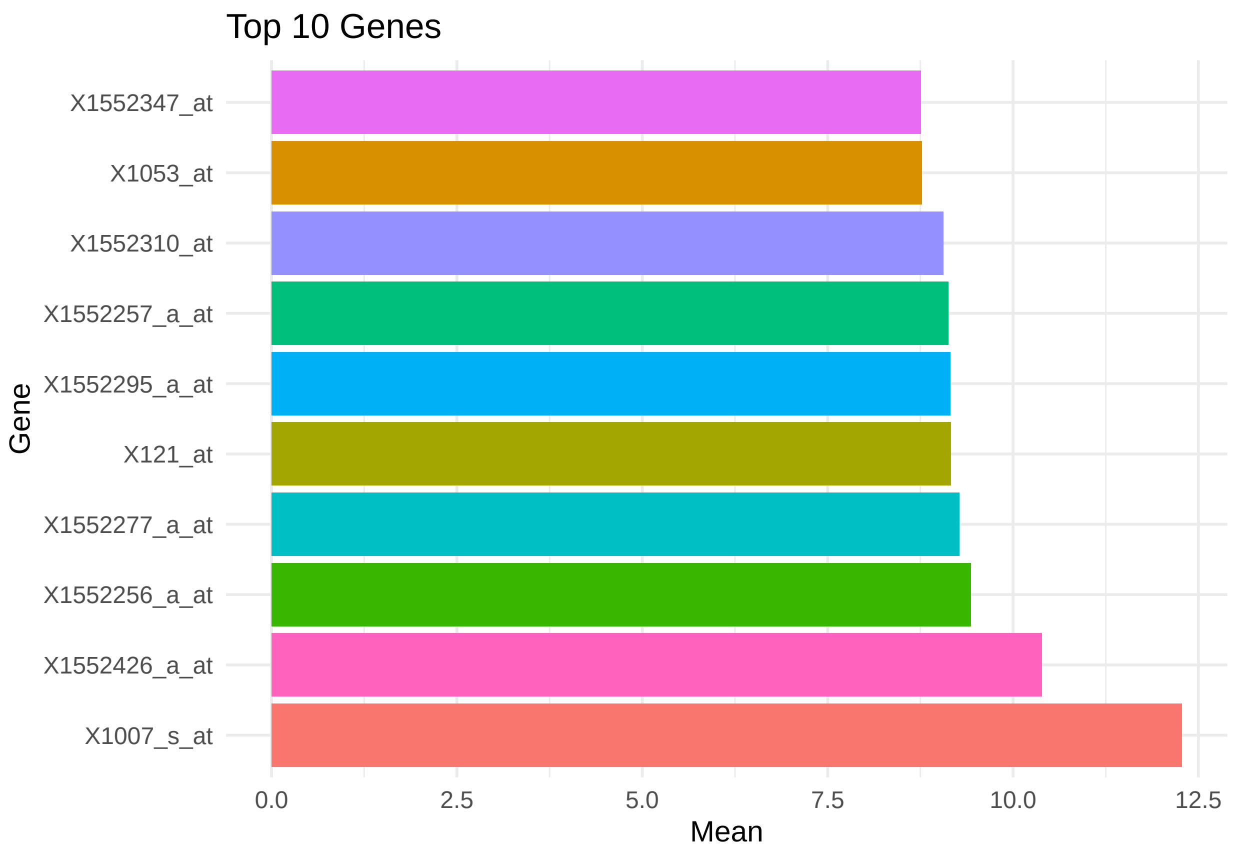
```
ggplot(genes_mean, aes(x = Mean, y = Sample)) +
  geom_point(color = "blue", size = 3, alpha = 0.7) +
  geom_smooth(method = "lm", color = "red", se = TRUE) +
  labs(title = "Mean vs Sample", x = "Mean", y = "Sample") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



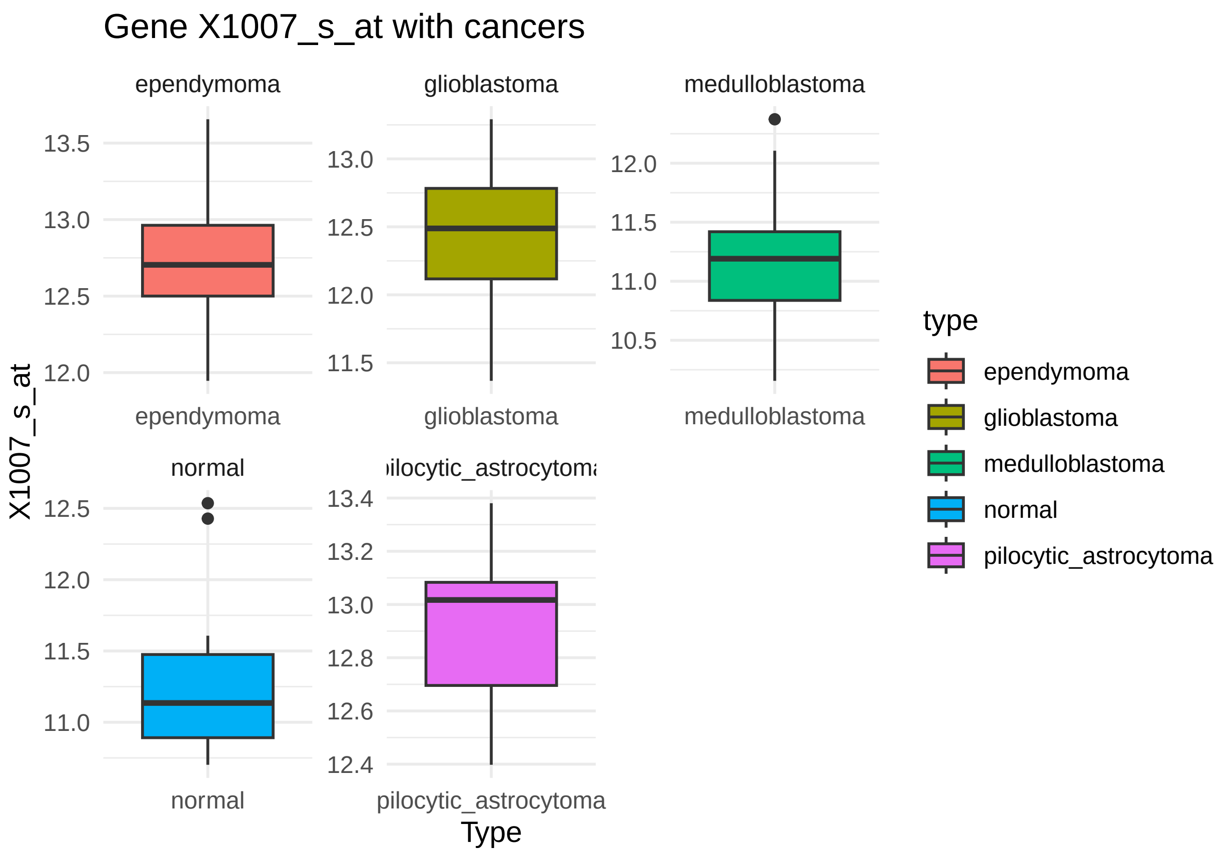
Sort the genes by the mean gene expression and plot the top 10 genes.

```
top_10_genes <- genes_mean %>%
  arrange(desc(Mean)) %>%
  slice(1:10)

ggplot(top_10_genes, aes(x = reorder(Gene, -Mean), y = Mean, fill = Gene)) +
  geom_bar(stat = "identity", show.legend = FALSE) +
  labs(title = "Top 10 Genes", x = "Gene", y = "Mean") +
  coord_flip() +
  theme_minimal()
```



Box plots showing the expression value based on the cancer type for the first gene in the dataset.

```
first_gene_with_cancers <- brain_cancer_dataset %>%
  select(2:3)

ggplot(first_gene_with_cancers, aes(x = type, y = X1007_s_at, fill = type)) +
  geom_boxplot() +
  facet_wrap(~type, scales = "free") +
  labs(title = "Gene X1007_s_at with cancers", x = "Type", y = "X1007_s_at") +
  theme_minimal()
```



Task 1.2 Principal Component Analysis

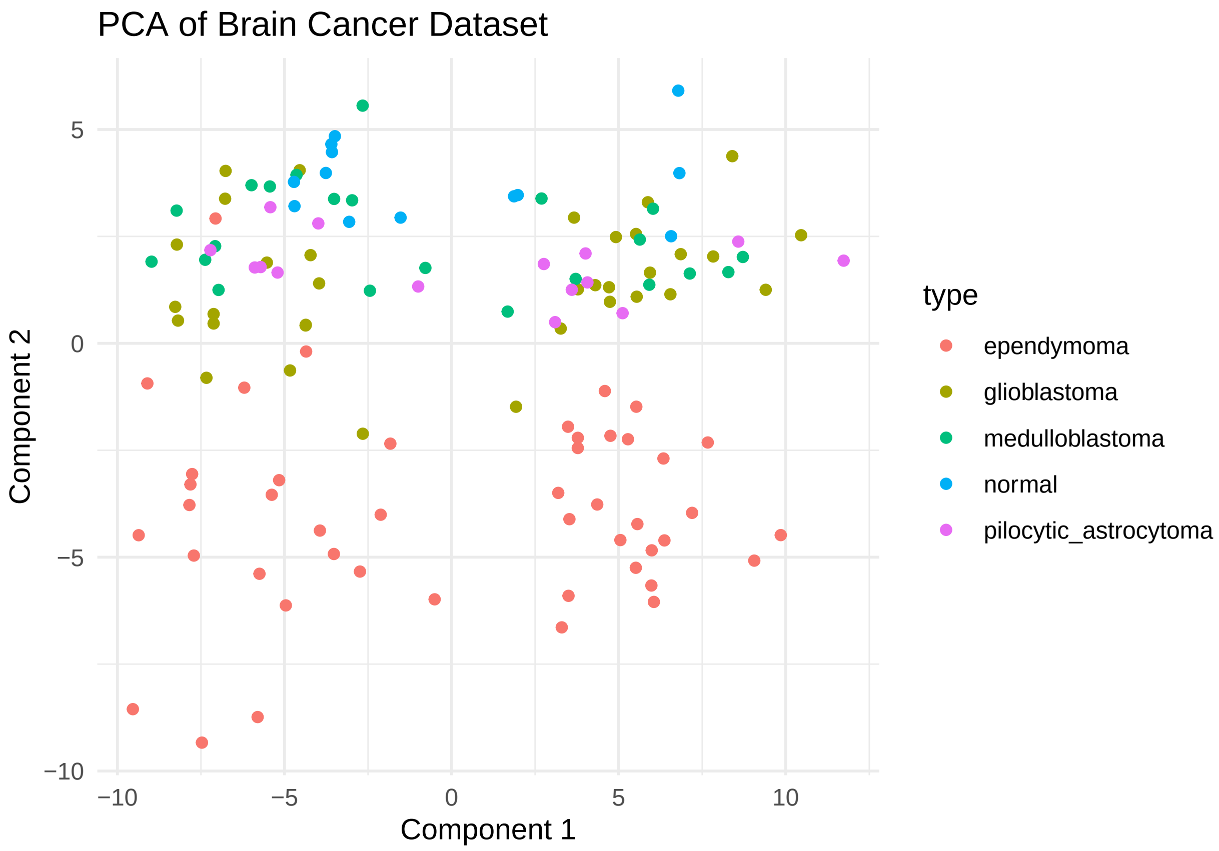
Performing PCA on the dataset and visualize the first three principal components combinations.

Component 1 and Component 2 will always give the best separation between the classes.

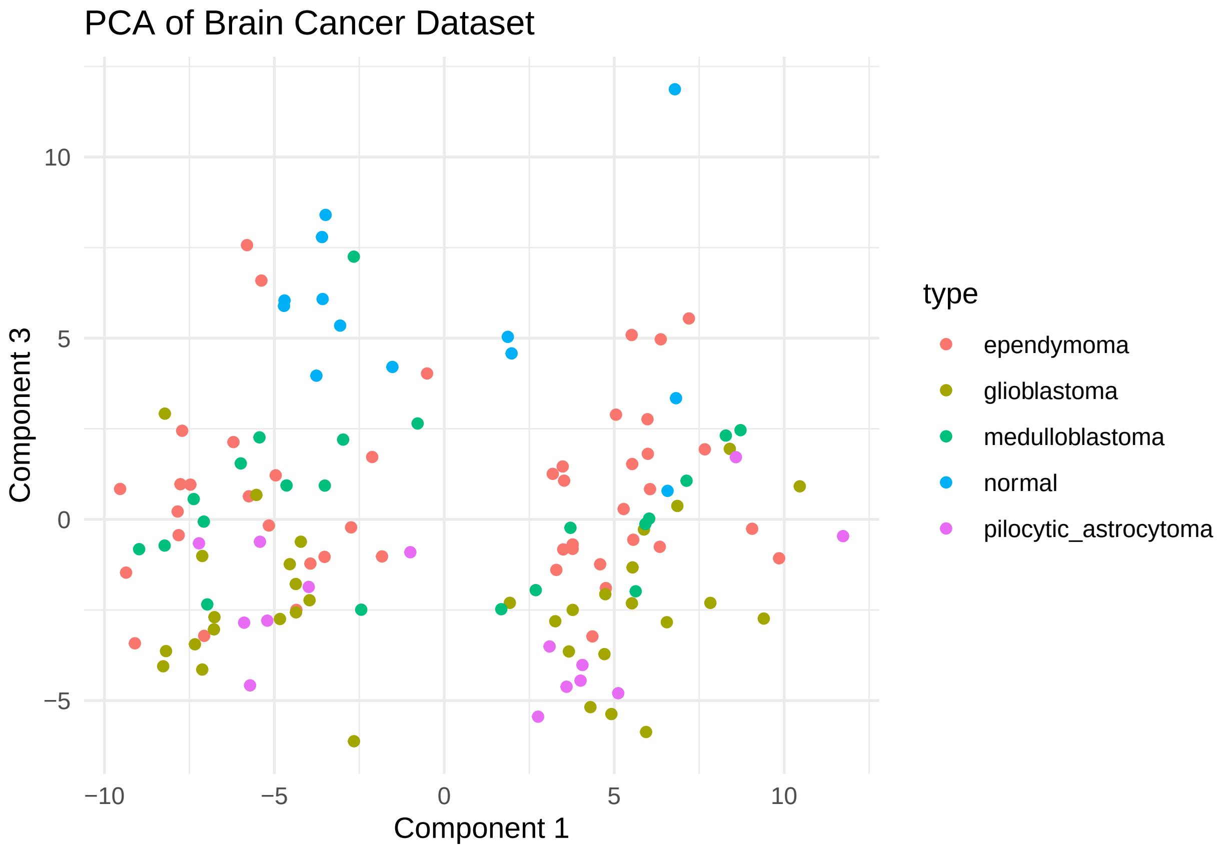
```
gene_samples <- brain_cancer_dataset %>%
  select(-c(1,2))

pca <- prcomp(gene_samples, scale. = TRUE)

components <- brain_cancer_dataset %>%
  select(type) %>%
  bind_cols(as.data.frame(pca$x[, 1:3])) %>%
  rename("component 1" = PC1, "component 2" = PC2, "component 3" = PC3)

print(components)
```

```
##                      type component 1 component 2 component 3
## 1              ependymoma   6.3700288  -4.6075590  4.96946914
## 2              ependymoma   4.3589894  -3.7677052 -3.22966229
## 3              ependymoma   4.5862941  -1.1148126 -1.23925109
## 4              ependymoma  -5.7487937  -5.3880115  0.63695970
## 5              ependymoma   3.1923752  -3.4969155  1.25572865
## 6              ependymoma   5.5137224  -5.2476805  5.08839758
## 7              ependymoma   9.8506841  -4.4836847 -1.07359330
## 8              ependymoma   5.0516748  -4.5997634  2.88951287
## 9              ependymoma   5.5289412  -1.4819157  1.52795650
## 10             ependymoma   7.6662498  -2.3186123  1.93313555
## 11             ependymoma  -7.8449082  -3.7797228  0.21810116
## 12             ependymoma   3.4847065  -1.9501462  1.46004915
## 13             ependymoma   5.9793881  -5.6618618  2.76391424
## 14             ependymoma  -4.3522414  -0.1912274 -2.49766349
## 15             ependymoma   3.2977167  -6.6406771 -1.39470220
## 16             ependymoma  -2.1212514  -4.0067359  1.72099258
## 17             ependymoma   9.0585516  -5.0805403 -0.25913576
## 18             ependymoma   5.2782351  -2.2422228  0.28535772
## 19             ependymoma  -3.9391625  -4.3775427 -1.21949722
## 20             ependymoma   3.4995967  -5.9033998 -0.82911697
## 21             ependymoma   4.7539905  -2.1621095 -1.89629696
## 22             ependymoma   3.7780936  -2.2094396 -0.69215501
## 23             ependymoma  -3.5215836  -4.9254065 -1.03501754
## 24             ependymoma  -2.7410882  -5.3359333 -0.21995335
## 25             ependymoma   6.3407359  -2.6924584 -0.75642765
## 26             ependymoma   6.0543896  -6.0463338  0.83438800
## 27             ependymoma   3.5266816  -4.1109428  1.06846788
## 28             ependymoma   5.9893668  -4.8398221  1.80885661
## 29             ependymoma  -1.8319189  -2.3452984 -1.02241151
## 30             ependymoma  -9.5374193  -8.5530553  0.83978542
## 31             ependymoma  -7.4708689  -9.3356734  0.95879340
## 32             ependymoma   5.5608395  -4.2254053 -0.56193158
## 33             ependymoma   7.1982189  -3.9636390  5.54497773
## 34             ependymoma  -0.5065207  -5.9847933  4.02623953
## 35             ependymoma   3.7771675  -2.4474407 -0.81327328
## 36             ependymoma  -7.7122978  -4.9618508  2.44557029
## 37             ependymoma  -9.1029214  -0.9372876 -3.41962662
## 38             ependymoma  -9.3629913  -4.4863737 -1.46933533
## 39             ependymoma  -7.7632542  -3.0567494  0.97156831
## 40             ependymoma  -7.8145967  -3.2989346 -0.43573358
## 41             ependymoma  -5.1604751  -3.1994298 -0.16952124
## 42             ependymoma  -6.2030776  -1.0367658  2.13317540
## 43             ependymoma  -7.0640771   2.9188436 -3.21436568
## 44             ependymoma  -4.9596810  -6.1270964  1.21544724
## 45             ependymoma  -5.8035022  -8.7380447  7.56769760
## 46             ependymoma  -5.3810630  -3.5407637  6.59077848
## 47           glioblastoma   4.9176894   2.4850825 -5.37185401
## 48           glioblastoma   5.5210241   2.5569174 -2.31558880
## 49           glioblastoma  -4.3681924   0.4181196 -1.78280059
## 50           glioblastoma  -5.5271996   1.8875364  0.67458260
## 51           glioblastoma  -4.2195722   2.0620588 -0.61668765
## 52           glioblastoma   5.9382231   1.6525297 -5.86700432
## 53           glioblastoma   6.5487518   1.1469539 -2.83833450
## 54           glioblastoma   9.4011489   1.2517746 -2.73698489
## 55           glioblastoma   7.8297684   2.0309678 -2.30478220
## 56           glioblastoma  10.4597747   2.5279246  0.91230165
## 57           glioblastoma   3.2663933   0.3456987 -2.81106477
## 58           glioblastoma   5.8744172   3.2987144 -0.27946930
## 59           glioblastoma   4.3005372   1.3603522 -5.18116018
## 60           glioblastoma   5.5401090   1.0903553 -1.32562371
## 61           glioblastoma   3.6664656   2.9391986 -3.64490127
## 62           glioblastoma  -2.6568266  -2.1139226 -6.12259856
## 63           glioblastoma  -6.7620605   4.0312353 -2.69828956
## 64           glioblastoma  -4.8340149  -0.6341334 -2.74858552
## 65           glioblastoma  -6.7767464   3.3815587 -3.03583884
## 66           glioblastoma  -4.5454116   4.0472740 -1.23658920
## 67           glioblastoma   6.8577873   2.0846091  0.37090541
## 68           glioblastoma   3.7794876   1.2630767 -2.49823253
## 69           glioblastoma   8.4009413   4.3764332  1.94601888
## 70           glioblastoma   4.7380420   0.9714239 -2.06455147
## 71           glioblastoma  -8.1864760   0.5314411 -3.63476764
## 72           glioblastoma   1.9296084  -1.4820636 -2.30308803
## 73           glioblastoma   4.7129984   1.3112919 -3.71836191
## 74           glioblastoma  -8.2696983   0.8535526 -4.05413264
## 75           glioblastoma  -8.2203959   2.3089152  2.91770553
## 76           glioblastoma  -7.1210805   0.4636077 -1.01059507
## 77           glioblastoma  -7.3358763  -0.8055284 -3.44681965
## 78           glioblastoma  -4.3629201   0.4328773 -2.56636405
## 79           glioblastoma  -7.1207578   0.6849910 -4.14385409
## 80           glioblastoma  -3.9635206   1.4014763 -2.23153307
## 81        medulloblastoma  -2.6611164   5.5576004  7.24970364
## 82        medulloblastoma   5.9173111   1.3700005 -0.12786745
## 83        medulloblastoma   7.1283342   1.6317965  1.06655448
## 84        medulloblastoma   6.0290712   3.1477738  0.02096172
## 85        medulloblastoma   8.7165919   2.0191987  2.46240529
## 86        medulloblastoma  -5.4373161   3.6678967  2.26303063
## 87        medulloblastoma   5.6323041   2.4264022 -1.98405434
## 88        medulloblastoma   8.2839165   1.6658943  2.31259731
## 89        medulloblastoma   1.6773875   0.7422202 -2.47879691
## 90        medulloblastoma  -6.9729799   1.2485172 -2.34671032
## 91        medulloblastoma   3.7129177   1.5060894 -0.23224060
## 92        medulloblastoma   2.6916904   3.3859723 -1.95221462
## 93        medulloblastoma  -7.3724350   1.9537824  0.56087845
## 94        medulloblastoma  -7.0744999   2.2715923 -0.06202595
## 95        medulloblastoma  -8.9782955   1.9095863 -0.82184142
## 96        medulloblastoma  -0.7838776   1.7630390  2.64584096
## 97        medulloblastoma  -3.5148846   3.3762788  0.93224824
## 98        medulloblastoma  -8.2278560   3.1034546 -0.72277768
## 99        medulloblastoma  -2.9759837   3.3442757  2.20171074
## 100       medulloblastoma  -4.6404401   3.9374885  0.93616886
## 101       medulloblastoma  -5.9874080   3.6973170  1.54373098
## 102       medulloblastoma  -2.4433158   1.2297037 -2.49252317
## 103                normal  -1.5257987   2.9392313  4.20649793
## 104                normal   6.8183003   3.9802356  3.34479920
## 105                normal  -3.7621468   3.9833433  3.96549611
## 106                normal   6.7846507   5.9077305 11.86885984
## 107                normal  -3.5791130   4.4715473  6.08022062
## 108                normal  -3.4915271   4.8429352  8.40253484
## 109                normal  -3.5975977   4.6555218  7.79061645
## 110                normal   6.5695836   2.5044802  0.78627623
## 111                normal  -3.0636164   2.8407966  5.34750973
## 112                normal   1.9803717   3.4663467  4.58020035
## 113                normal  -4.7002854   3.2073708  6.04078976
## 114                normal  -4.7153297   3.7745190  5.89192102
## 115                normal   1.8695382   3.4374287  5.03686077
## 116 pilocytic_astrocytoma   4.0090755   2.1012556 -4.45021442
## 117 pilocytic_astrocytoma   3.0983962   0.4954635 -3.50718182
## 118 pilocytic_astrocytoma   3.5973266   1.2520552 -4.61778914
## 119 pilocytic_astrocytoma  11.7336371   1.9332133 -0.46170186
## 120 pilocytic_astrocytoma   8.5801442   2.3792357  1.71670571
## 121 pilocytic_astrocytoma   4.0658096   1.4245808 -4.01887991
## 122 pilocytic_astrocytoma   5.1184421   0.7057000 -4.79645780
## 123 pilocytic_astrocytoma  -5.2072479   1.6552278 -2.79491057
## 124 pilocytic_astrocytoma  -5.7140301   1.7835162 -4.58077147
## 125 pilocytic_astrocytoma   2.7608578   1.8545696 -5.44391350
## 126 pilocytic_astrocytoma  -3.9870971   2.8040955 -1.86433578
## 127 pilocytic_astrocytoma  -0.9982557   1.3289709 -0.90668418
## 128 pilocytic_astrocytoma  -5.8886599   1.7736365 -2.85033081
## 129 pilocytic_astrocytoma  -7.2168430   2.1788045 -0.66122385
## 130 pilocytic_astrocytoma  -5.4230633   3.1829487 -0.61737747
```

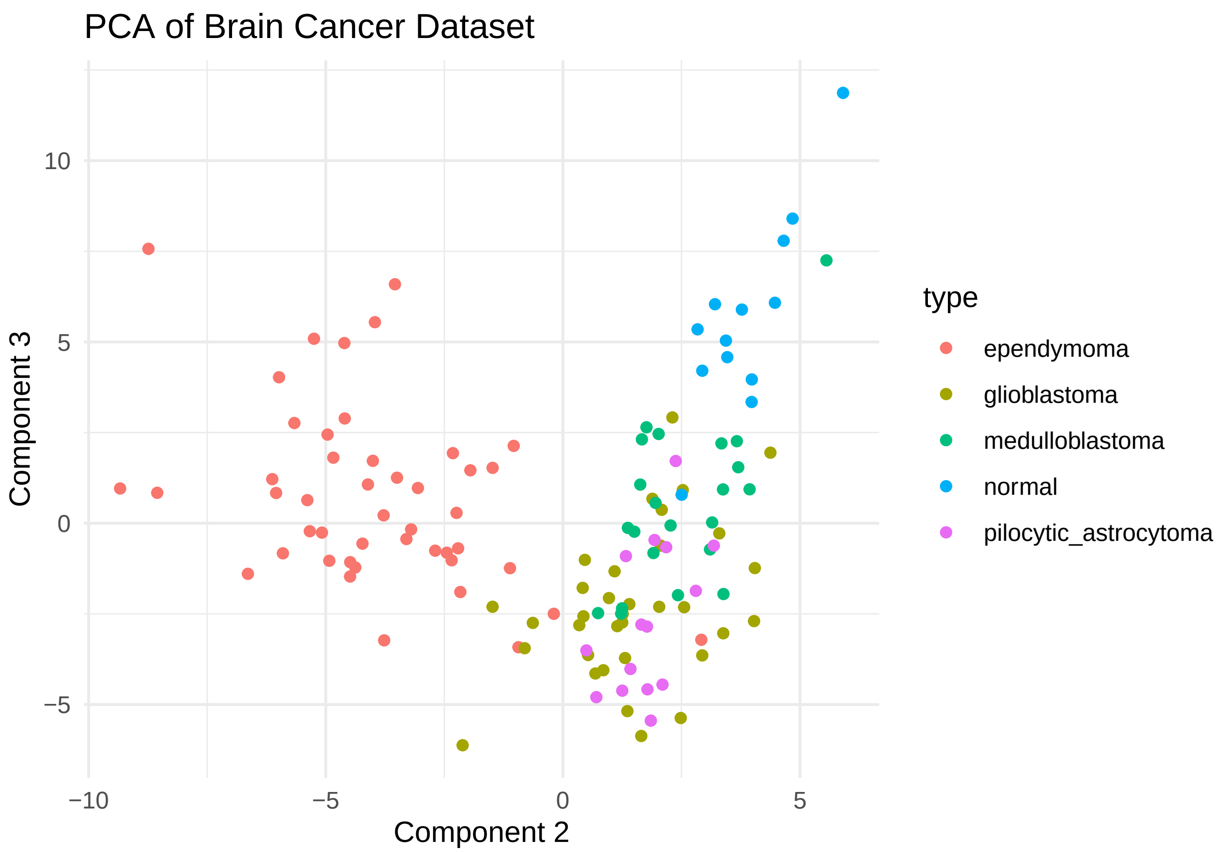
```r
ggplot(components, aes(x = `component 1`, y = `component 2`, color = type)) +
  geom_point() +
  labs(
    title = "PCA of Brain Cancer Dataset",
    x = "Component 1",
    y = "Component 2") +
  theme_minimal()
```



```r
ggplot(components, aes(x = `component 1`, y = `component 3`, color = type)) +
  geom_point() +
  labs(
    title = "PCA of Brain Cancer Dataset",
    x = "Component 1",
    y = "Component 3") +
  theme_minimal()
```



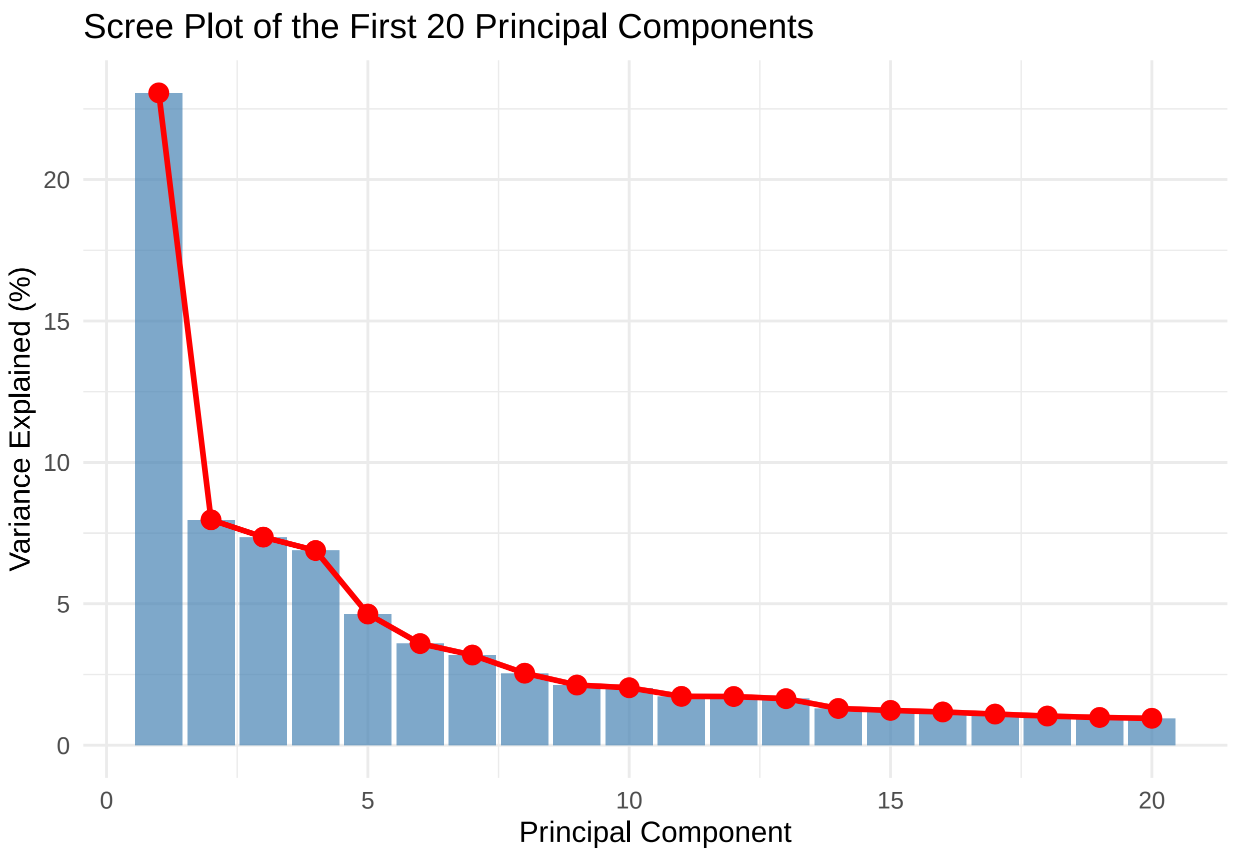
```
ggplot(components, aes(x = `component 2`, y = `component 3`, color = type)) +
  geom_point() +
  labs(
    title = "PCA of Brain Cancer Dataset",
    x = "Component 2",
    y = "Component 3") +
  theme_minimal()
```



Drawing a scree plot to show the variance explained by each principal component.

```r
explained_variance <- pca$sdev^2 / sum(pca$sdev^2)

scree_data <- data.frame(
  PC = 1:20,
  Variance_Explained = explained_variance[1:20] * 100
)

ggplot(scree_data, aes(x = PC, y = Variance_Explained)) +
  geom_bar(stat = "identity", fill = "steelblue", alpha = 0.7) +
  geom_line(aes(group = 1), color = "red", size = 1) +
  geom_point(size = 3, color = "red") +
  labs(
    title = "Scree Plot of the First 20 Principal Components",
    x = "Principal Component",
    y = "Variance Explained (%)"
  ) +
  theme_minimal()
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



Part 2: Sequence Alignment Intro

Task 2.1: Installing Biostrings

```
library(BiocManager)
BiocManager::install("Biostrings")
```

```
## 'getOption("repos")' replaces Bioconductor standard repositories, see
## 'help("repositories", package = "BiocManager")' for details.
## Replacement repositories:
##     CRAN: https://cloud.r-project.org

## Bioconductor version 3.20 (BiocManager 1.30.25), R 4.4.3 (2025-02-28)

## Warning: package(s) not installed when version(s) same as or greater than current; use
##   `force = TRUE` to re-install: 'Biostrings'

## Installation paths not writeable, unable to update packages
##   path: /usr/lib/R/library
##   packages:
##     lattice, MASS, spatial

## Old packages: 'cpp11', 'jsonlite'
```

```
BiocManager::install("pwalign")
```

```
## 'getOption("repos")' replaces Bioconductor standard repositories, see
## 'help("repositories", package = "BiocManager")' for details.
## Replacement repositories:
##     CRAN: https://cloud.r-project.org
```

```
## Bioconductor version 3.20 (BiocManager 1.30.25), R 4.4.3 (2025-02-28)
```

```
## Warning: package(s) not installed when version(s) same as or greater than current; use
##   `force = TRUE` to re-install: 'pwalign'
```

```
## Installation paths not writeable, unable to update packages
##   path: /usr/lib/R/library
##   packages:
##     lattice, MASS, spatial
```

```
## Old packages: 'cpp11', 'jsonlite'
```

```r
library(Biostrings)
```

```
## Loading required package: BiocGenerics
```

```
##
## Attaching package: 'BiocGenerics'
```

```
## 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, saveRDS, setdiff,
##     table, tapply, union, unique, unsplit, which.max, which.min
```

```
## Loading required package: S4Vectors
```

```
## Loading required package: stats4
```

```
##
## Attaching package: 'S4Vectors'
```

```
## The following object is masked from 'package:tidyr':
##
##     expand
```

```
## The following objects are masked from 'package:dplyr':
##
##     first, rename
```

```
## 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 objects are masked from 'package:dplyr':
##
##     collapse, desc, slice

## Loading required package: XVector

## Loading required package: GenomeInfoDb

##
## Attaching package: 'Biostrings'

## The following object is masked from 'package:base':
##
##     strsplit
```

Task 2.2: Run Pairwise Alignment

```r
seq_A <- DNAString("AGCTGAACTAGCTAGCTGACTGACTGACTAGCTAGCTGACTAGCTG")
seq_B <- DNAString("AGCGAACTAGCTGACTGACGACTGACTAGCTGACTAGCTGACTAGC")
```

Performing global pairwise alignment between the two sequences.

Observing the pattern, the subject, and the score of the alignment.

```r
global_alignment <- pwalign::pairwiseAlignment(seq_A,
                                               seq_B,
                                               type = "global")

cat("Score:", score(global_alignment), "\n")
```

```
## Score: -4.528324
```

```r
cat("Pattern:\n", as.character(pattern(global_alignment)), "\n")
```

```
## Pattern:
##  AGCTGAACTAGCTAGCTGACTGACTGACTAGCT----AGCTGACTAGC
```

```r
cat("Subject:\n", as.character(subject(global_alignment)), "\n")
```

```
## Subject:
##  AGC-GAACTAGCTGACTGAC-GACTGACTAGCTGACTAGCTGACTAGC
```

Here we change the substitution matrix and gap penalties.

```r
custom_matrix <- pwalign::nucleotideSubstitutionMatrix(match = 2,
                                              mismatch = -1,
                                              baseOnly = TRUE)

global_alignment_custom <- pwalign::pairwiseAlignment(seq_A, seq_B,
                                      substitutionMatrix = custom_matrix,
                                      gapOpening = -5, gapExtension = -2,
                                      type = "global")
```

Method to run the experiment with different parameters.

```r
run_experiment <- function(match,
                           mismatch,
                           gap_open,
                           gap_ext,
                           alignment_type = "global") {
    custom_matrix <- nucleotideSubstitutionMatrix(match = match,
                                                  mismatch = mismatch,
                                                  baseOnly = TRUE)

    alignment <- pairwiseAlignment(seq_A,
                                   seq_B,
                                   substitutionMatrix = custom_matrix,
                                   gapOpening = gap_open,
                                   gapExtension = gap_ext,
                                   type = alignment_type)

    cat("\n==========================================\n")
    cat("Experiment: Match =", match, "| Mismatch =", mismatch,
        "| Gap Opening =", gap_open, "| Gap Extension =", gap_ext,
        "| Type =", alignment_type, "\n")
    cat("Score:", score(alignment), "\n")
    cat("Pattern:\n", as.character(pattern(alignment)), "\n")
    cat("Subject:\n", as.character(subject(alignment)), "\n")
    cat("==========================================\n")
}
```

```r
run_experiment(match = 1, mismatch = -1, gap_open = -2, gap_ext = -1)
```

```
## Warning in .call_fun_in_pwalign("nucleotideSubstitutionMatrix", ...): nucleotideSubstitutionMatrix()
##   call pwalign::nucleotideSubstitutionMatrix() to get rid of this
##   warning.

## Warning in .call_fun_in_pwalign("pairwiseAlignment", ...): pairwiseAlignment() has moved to the pwal
##   pwalign::pairwiseAlignment() to get rid of this warning.

##
## ==========================================
## Experiment: Match = 1 | Mismatch = -1 | Gap Opening = -2 | Gap Extension = -1 | Type = global
## Score: 22
## Pattern:
##  AGCTGAACTAGCTAGCTGACTGACTGACTAGCT----AGCTGACTAGC
## Subject:
##  AGC-GAACTAGCTGACTGAC-GACTGACTAGCTGACTAGCTGACTAGC
## ==========================================
```

```r
run_experiment(match = 3, mismatch = -1, gap_open = -2, gap_ext = -1)
```

```
## Warning in .call_fun_in_pwalign("nucleotideSubstitutionMatrix", ...): nucleotideSubstitutionMatrix()
##   call pwalign::nucleotideSubstitutionMatrix() to get rid of this
##   warning.
## Warning in .call_fun_in_pwalign("nucleotideSubstitutionMatrix", ...): pairwiseAlignment() has moved
##   pwalign::pairwiseAlignment() to get rid of this warning.

##
## ==========================================
## Experiment: Match = 3 | Mismatch = -1 | Gap Opening = -2 | Gap Extension = -1 | Type = global
## Score: 102
## Pattern:
##  AGCTGAACTAGCTAGCTGACTGACTGACTAGCT----AGCTGACTAGC
## Subject:
##  AGC-GAACTAGCTGACTGAC-GACTGACTAGCTGACTAGCTGACTAGC
## ==========================================
```

```r
run_experiment(match = 1, mismatch = -3, gap_open = -2, gap_ext = -1)
```

```
## Warning in .call_fun_in_pwalign("nucleotideSubstitutionMatrix", ...): nucleotideSubstitutionMatrix()
##   call pwalign::nucleotideSubstitutionMatrix() to get rid of this
##   warning.
## Warning in .call_fun_in_pwalign("nucleotideSubstitutionMatrix", ...): pairwiseAlignment() has moved
##   pwalign::pairwiseAlignment() to get rid of this warning.

##
## ==========================================
## Experiment: Match = 1 | Mismatch = -3 | Gap Opening = -2 | Gap Extension = -1 | Type = global
## Score: 19
## Pattern:
##  AGCTGAACTAGCT-AGCTGACTGACTGACTAGCT----AGCTGACTAGC
## Subject:
##  AGC-GAACTAGCTGA-CTGAC-GACTGACTAGCTGACTAGCTGACTAGC
## ==========================================
```

```r
run_experiment(match = 1, mismatch = -1, gap_open = -8, gap_ext = -1)
```

```
## Warning in .call_fun_in_pwalign("nucleotideSubstitutionMatrix", ...): nucleotideSubstitutionMatrix()
##   call pwalign::nucleotideSubstitutionMatrix() to get rid of this
##   warning.
## Warning in .call_fun_in_pwalign("nucleotideSubstitutionMatrix", ...): pairwiseAlignment() has moved
##   pwalign::pairwiseAlignment() to get rid of this warning.

##
## ==========================================
## Experiment: Match = 1 | Mismatch = -1 | Gap Opening = -8 | Gap Extension = -1 | Type = global
## Score: 1
## Pattern:
##  AGCTGAACTAGCTAGCTGAC---TGACTGACTAGCTAGCTGACTAGC
## Subject:
##  AGC-GAACTAGCTGACTGACGACTGACTAGCTGACTAGCTGACTAGC
## ==========================================
```

```r
run_experiment(match = 1, mismatch = -1, gap_open = -2, gap_ext = -5)
```

```
## Warning in .call_fun_in_pwalign("nucleotideSubstitutionMatrix", ...): nucleotideSubstitutionMatrix()
##   call pwalign::nucleotideSubstitutionMatrix() to get rid of this
##   warning.
## Warning in .call_fun_in_pwalign("nucleotideSubstitutionMatrix", ...): pairwiseAlignment() has moved
##   pwalign::pairwiseAlignment() to get rid of this warning.

##
## ==========================================
## Experiment: Match = 1 | Mismatch = -1 | Gap Opening = -2 | Gap Extension = -5 | Type = global
## Score: 0
## Pattern:
##  AGCTGAACTAGCTAGCTGACTGACTGACTAGCTAGCTGACTAGCTG
## Subject:
##  AGC-GAACTAGCTGACTGAC-GACTGACTAGCTGACTAGCTGACTA
## ==========================================
```

```r
run_experiment(match = 2, mismatch = -5, gap_open = -7, gap_ext = -1)
```

```
## Warning in .call_fun_in_pwalign("nucleotideSubstitutionMatrix", ...): nucleotideSubstitutionMatrix()
##   call pwalign::nucleotideSubstitutionMatrix() to get rid of this
##   warning.
## Warning in .call_fun_in_pwalign("nucleotideSubstitutionMatrix", ...): pairwiseAlignment() has moved t
##   pwalign::pairwiseAlignment() to get rid of this warning.
```

```
##
## ==========================================
## Experiment: Match = 2 | Mismatch = -5 | Gap Opening = -7 | Gap Extension = -1 | Type = global
## Score: 34
## Pattern:
##  AGCTGAACTAGCTAGCTGACTGACTGACTAGCT----AGCTGACTAGC
## Subject:
##  AGC-GAACTAGCTGACTGAC-GACTGACTAGCTGACTAGCTGACTAGC
## ==========================================
```

Part 3: Sequence Alignment Advanced

Task 3.1: BLAST

Sequence alignment using BLAST web tool.

Using Nuccore.

Organism: Homo-sapiens INS-IGF2

Length: 39098 bp

Type: DNA

Organism: Homo-sapiens Human gene for insulin-like growth factor II

Length: 8837 bp

Type: DNA

Task 3.2: Running Locally - Retrieve Sequences

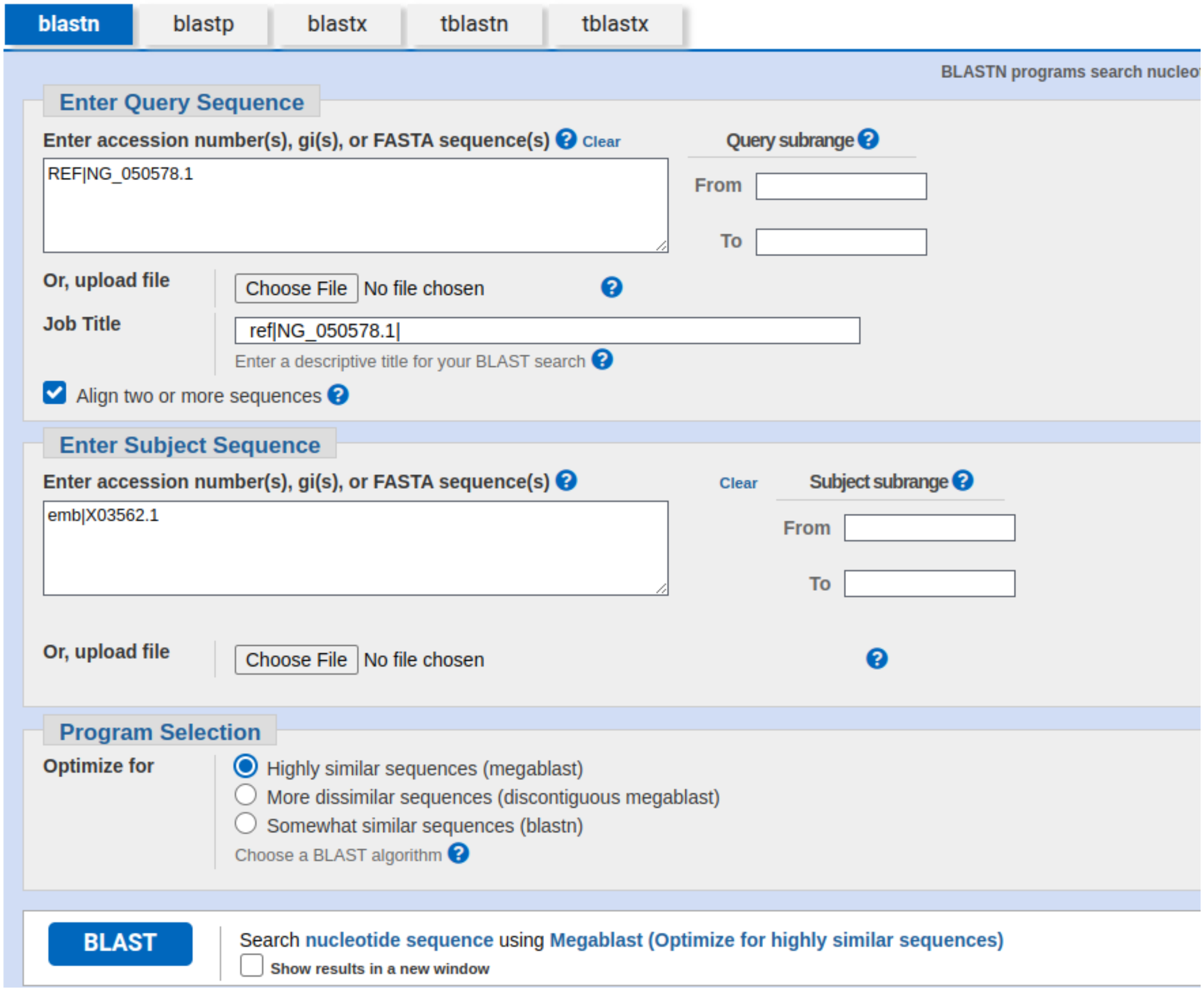
```r
install.packages("rentrez")
```

```
## Installing package into '/home/omar-aldawy/R/x86_64-pc-linux-gnu-library/4.4'
## (as 'lib' is unspecified)
```

```r
library(rentrez)
```

Fetching two sequences from GenBank using their accession numbers.

```r
accessions <- c("NG_050578.1", "X03562.1")
```



Enter Query Sequence

Enter accession number(s), gi(s), or FASTA sequence(s)

REF|NG_050578.1

Or, upload file: No file chosen

Job Title: ref|NG_050578.1|

Enter a descriptive title for your BLAST search

☑ Align two or more sequences

Enter Subject Sequence

Enter accession number(s), gi(s), or FASTA sequence(s)

emb|X03562.1

Or, upload file: No file chosen

Program Selection

Optimize for

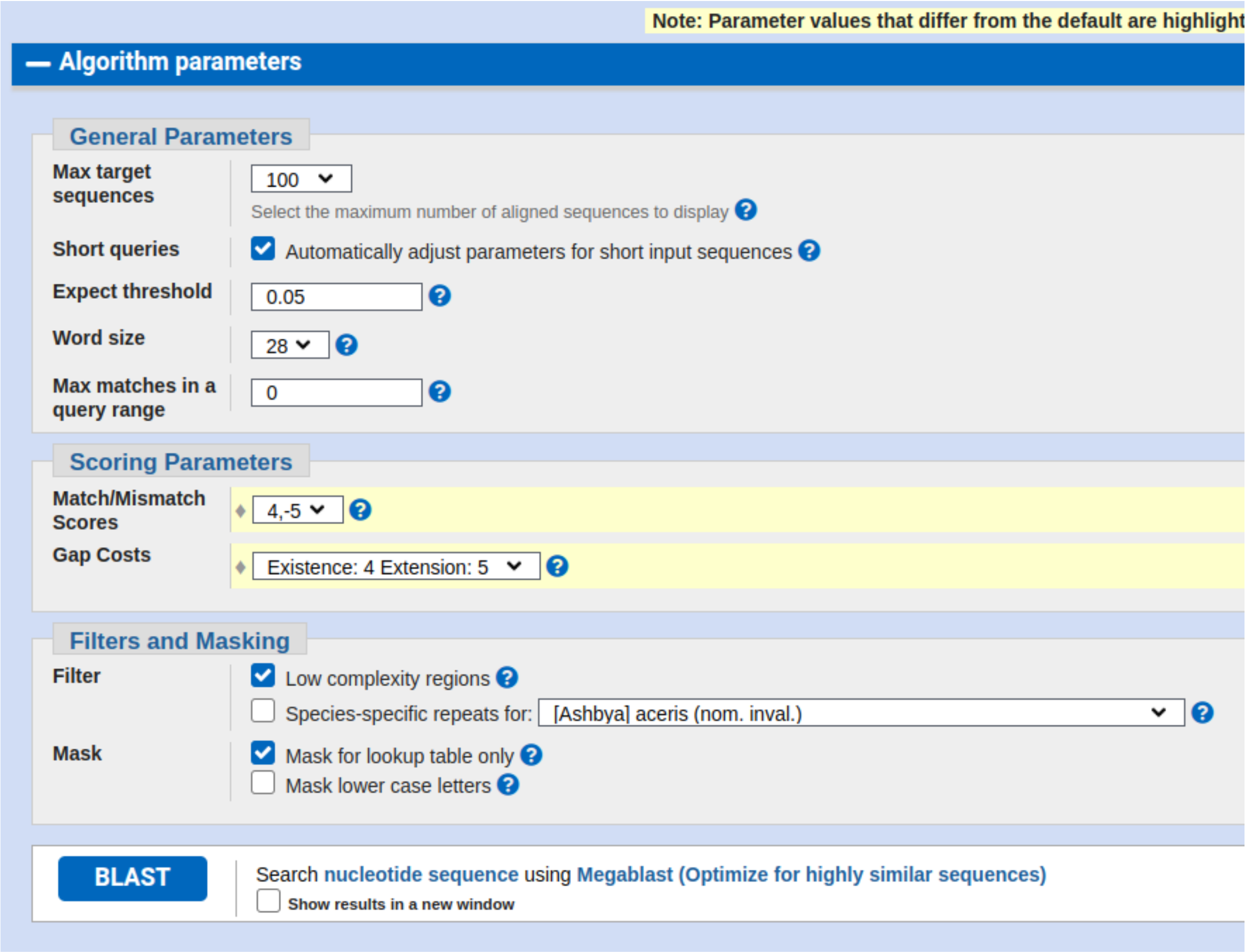
● Highly similar sequences (megablast)
○ More dissimilar sequences (discontiguous megablast)
○ Somewhat similar sequences (blastn)

Choose a BLAST algorithm

Search nucleotide sequence using Megablast (Optimize for highly similar sequences)

□ Show results in a new window

Figure 1: Input part



Note: Parameter values that differ from the default are highlight

— Algorithm parameters

General Parameters

Max target sequences: 100
Select the maximum number of aligned sequences to display

Short queries: ☑ Automatically adjust parameters for short input sequences

Expect threshold: 0.05

Word size: 28

Max matches in a query range: 0

Scoring Parameters

Match/Mismatch Scores: 4,-5

Gap Costs: Existence: 4 Extension: 5

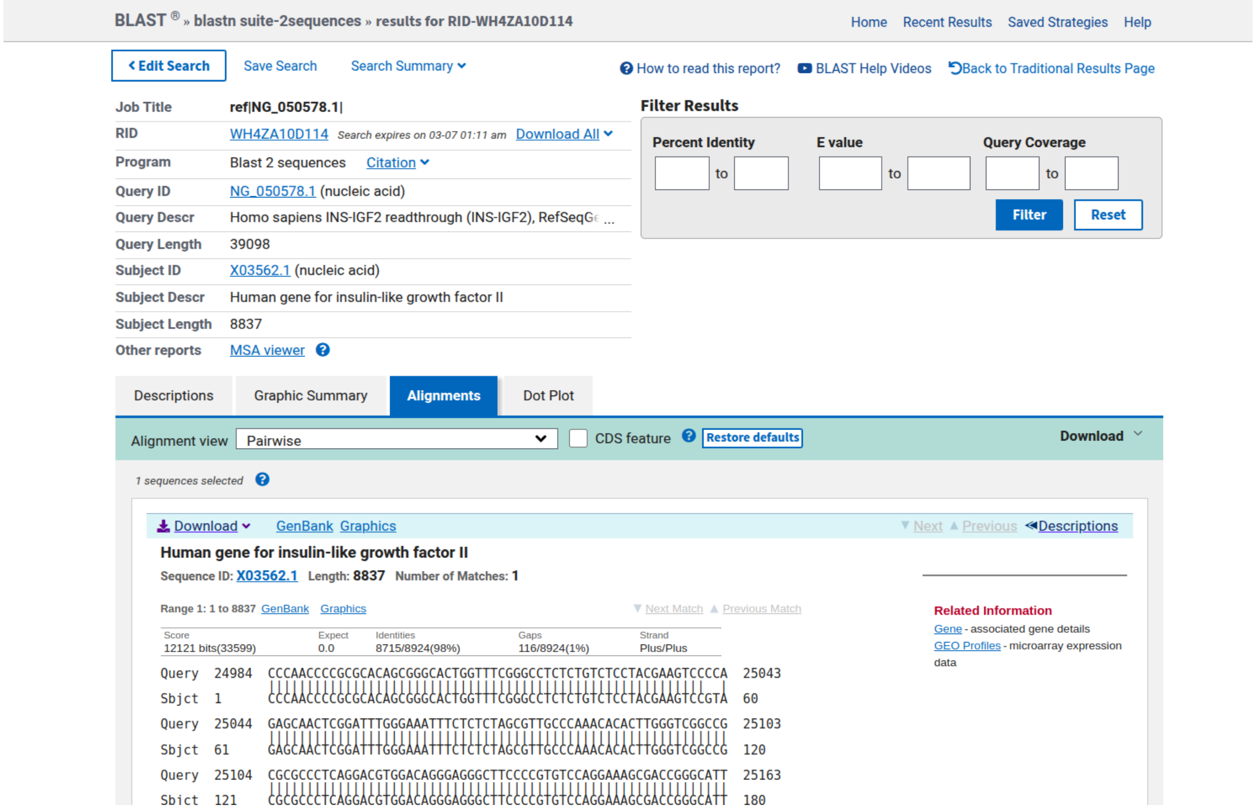
Filters and Masking

Filter: ☑ Low complexity regions
□ Species-specific repeats for: [Ashbya] aceris (nom. inval.)

Mask: ☑ Mask for lookup table only
□ Mask lower case letters

Search nucleotide sequence using Megablast (Optimize for highly similar sequences)
□ Show results in a new window

Figure 2: Scores



Job Title	ref|NG_050578.1|
RID	WH4ZA10D114 *Search expires on 03-07 01:11 am* Download All
Program	Blast 2 sequences Citation
Query ID	NG_050578.1 (nucleic acid)
Query Descr	Homo sapiens INS-IGF2 readthrough (INS-IGF2), RefSeqGe ...
Query Length	39098
Subject ID	X03562.1 (nucleic acid)
Subject Descr	Human gene for insulin-like growth factor II
Subject Length	8837
Other reports	MSA viewer

Alignment view: Pairwise

□ CDS feature

1 sequences selected

Human gene for insulin-like growth factor II

Sequence ID: X03562.1 **Length:** 8837 **Number of Matches:** 1

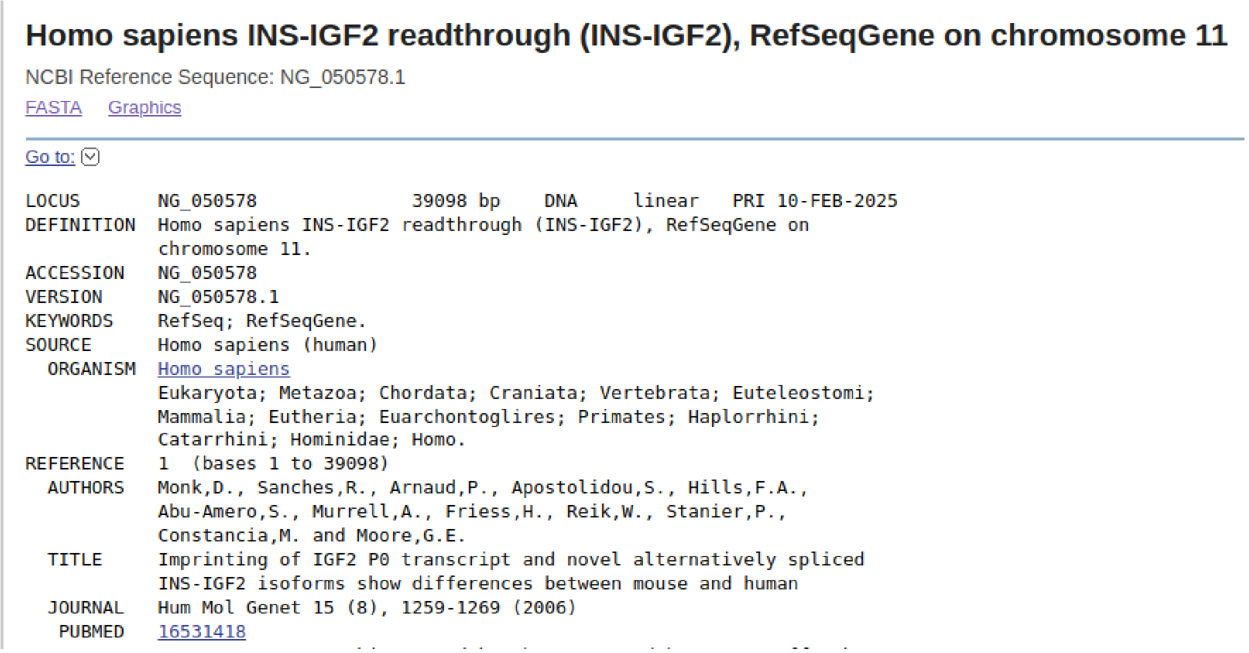
Range 1: 1 to 8837 GenBank Graphics

Score	Expect	Identities	Gaps	Strand
12121 bits(33599)	0.0	8715/8924(98%)	116/8924(1%)	Plus/Plus

```
Query  24984  CCCAACCCCGCGCACAGCGGGCACTGGTTTCGGGCCTCTCTGTCTCCTACGAAGTCCCCA  25043
              |||||||||||||||||||||||||||||||||||||||||||||||||||||||||  |
Sbjct  1      CCCAACCCCGCGCACAGCGGGCACTGGTTTCGGGCCTCTCTGTCTCCTACGAAGTCCGTA  60

Query  25044  GAGCAACTCGGATTTGGGAAATTTCTCTCTAGCGTTGCCCAAACACACTTGGGTCGGCCG  25103
              ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  61     GAGCAACTCGGATTTGGGAAATTTCTCTCTAGCGTTGCCCAAACACACTTGGGTCGGCCG  120

Query  25104  CGCGCCCTCAGGACGTGGACAGGGAGGGCTTCCCCGTGTCCAGGAAAGCGACCGGGCATT  25163
              ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  121    CGCGCCCTCAGGACGTGGACAGGGAGGGCTTCCCCGTGTCCAGGAAAGCGACCGGGCATT  180
```

Related Information
Gene - associated gene details
GEO Profiles - microarray expression data

Figure 3: Sequence Alignment

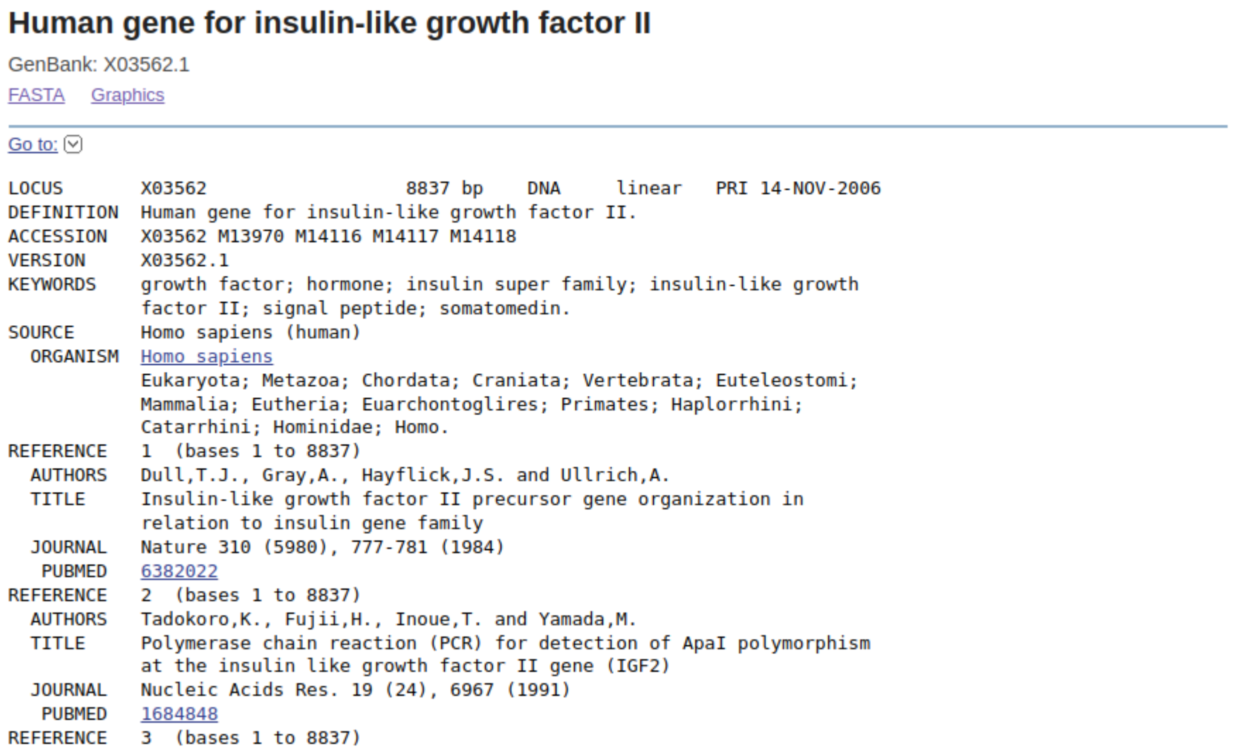


Homo sapiens INS-IGF2 readthrough (INS-IGF2), RefSeqGene on chromosome 11

NCBI Reference Sequence: NG_050578.1

```
LOCUS       NG_050578            39098 bp    DNA     linear   PRI 10-FEB-2025
DEFINITION  Homo sapiens INS-IGF2 readthrough (INS-IGF2), RefSeqGene on
            chromosome 11.
ACCESSION   NG_050578
VERSION     NG_050578.1
KEYWORDS    RefSeq; RefSeqGene.
SOURCE      Homo sapiens (human)
  ORGANISM  Homo sapiens
            Eukaryota; Metazoa; Chordata; Craniata; Vertebrata; Euteleostomi;
            Mammalia; Eutheria; Euarchontoglires; Primates; Haplorrhini;
            Catarrhini; Hominidae; Homo.
REFERENCE   1  (bases 1 to 39098)
  AUTHORS   Monk,D., Sanches,R., Arnaud,P., Apostolidou,S., Hills,F.A.,
            Abu-Amero,S., Murrell,A., Friess,H., Reik,W., Stanier,P.,
            Constancia,M. and Moore,G.E.
  TITLE     Imprinting of IGF2 P0 transcript and novel alternatively spliced
            INS-IGF2 isoforms show differences between mouse and human
  JOURNAL   Hum Mol Genet 15 (8), 1259-1269 (2006)
   PUBMED   16531418
```

Figure 4: seq 1



Human gene for insulin-like growth factor II

GenBank: X03562.1

FASTA Graphics

Go to:

```
LOCUS       X03562                  8837 bp    DNA     linear   PRI 14-NOV-2006
DEFINITION  Human gene for insulin-like growth factor II.
ACCESSION   X03562 M13970 M14116 M14117 M14118
VERSION     X03562.1
KEYWORDS    growth factor; hormone; insulin super family; insulin-like growth
            factor II; signal peptide; somatomedin.
SOURCE      Homo sapiens (human)
  ORGANISM  Homo sapiens
            Eukaryota; Metazoa; Chordata; Craniata; Vertebrata; Euteleostomi;
            Mammalia; Eutheria; Euarchontoglires; Primates; Haplorrhini;
            Catarrhini; Hominidae; Homo.
REFERENCE   1  (bases 1 to 8837)
  AUTHORS   Dull,T.J., Gray,A., Hayflick,J.S. and Ullrich,A.
  TITLE     Insulin-like growth factor II precursor gene organization in
            relation to insulin gene family
  JOURNAL   Nature 310 (5980), 777-781 (1984)
   PUBMED   6382022
REFERENCE   2  (bases 1 to 8837)
  AUTHORS   Tadokoro,K., Fujii,H., Inoue,T. and Yamada,M.
  TITLE     Polymerase chain reaction (PCR) for detection of ApaI polymorphism
            at the insulin like growth factor II gene (IGF2)
  JOURNAL   Nucleic Acids Res. 19 (24), 6967 (1991)
   PUBMED   1684848
REFERENCE   3  (bases 1 to 8837)
```

Figure 5: seq 2

```r
sequences <- lapply(accessions, function(acc) {
  entrez_fetch(db = "nucleotide", id = acc, rettype = "fasta")
})
```

```r
# Define a custom getSequence function to remove headers and return the nucleotide sequence
getSequence <- function(fasta_text) {
  # Split the FASTA text into lines
  lines <- strsplit(fasta_text, "\n")[[1]]
  # Remove header lines that start with '>'
  seq_lines <- lines[!grepl("^>", lines)]
  # Concatenate the remaining lines into one string
  sequence <- paste(seq_lines, collapse = "")
  return(sequence)
}

# Extract the nucleotide sequences from the FASTA text
sequences <- lapply(sequences, getSequence)
dna_1 <- DNAStringSet(sequences[[1]])
dna_2 <- DNAStringSet(sequences[[2]])
```

Task 3.3: Sequence Processing

Identifing sequences with gaps or ambiguous bases.

```r
freq_1 <- alphabetFrequency(dna_1)
freq_2 <- alphabetFrequency(dna_2)
```

```r
cat("Sequence 1 gaps count:", freq_1[1, "-"],
    " | ambiguous bases count:", freq_1[1, "N"], "\n")
```

```
## Sequence 1 gaps count: 0  | ambiguous bases count: 0
```

```r
cat("Sequence 2 gaps count:", freq_2[1, "-"],
    " | ambiguous bases count:", freq_2[1, "N"], "\n")
```

```
## Sequence 2 gaps count: 0  | ambiguous bases count: 30
```

Removeing gaps and ambiguous bases from sequences.

```r
clean_sequence <- function(dna_seq) {
  # Get the original length
  original_length <- width(dna_seq)

  # Remove 'N' and '-' from the sequence
  cleaned_seq <- DNAStringSet(gsub("[N-]", "", as.character(dna_seq)))

  # Get the cleaned length
  cleaned_length <- width(cleaned_seq)

  return(list(original = original_length,
              cleaned = cleaned_length,
              cleaned_seq = cleaned_seq))
}

cleaned_seq_1 <- clean_sequence(dna_1)
cleaned_seq_2 <- clean_sequence(dna_2)

cat("Sequence 1 ( Original Length:", cleaned_seq_1$original,
    ", Cleaned Length:", cleaned_seq_1$cleaned, ")\n")
```

```
## Sequence 1 ( Original Length: 39098 , Cleaned Length: 39098 )
```

```r
cat("Sequence 2 ( Original Length:", cleaned_seq_2$original,
    ", Cleaned Length:", cleaned_seq_2$cleaned, ")\n")
```

```
## Sequence 2 ( Original Length: 8837 , Cleaned Length: 8807 )
```

Performing local pairwise alignment on the cleaned sequences.

```r
sub_matrix <- pwalign::nucleotideSubstitutionMatrix(match = 4,
                                                    mismatch = -5,
                                                    baseOnly = TRUE)

alignment <- pwalign::pairwiseAlignment(
  cleaned_seq_1$cleaned_seq[[1]], cleaned_seq_2$cleaned_seq[[1]],
  type = "local",
  substitutionMatrix = sub_matrix,
  gapOpening = -4,
  gapExtension = -5,
)

# Extract alignment details
```

```r
alignment_score <- score(alignment)
num_matches <- nmatch(alignment)
num_mismatches <- nmismatch(alignment)

# Extract the aligned sequences
aligned_seq1 <- as.character(alignment@pattern)
aligned_seq2 <- as.character(alignment@subject)

# Count gaps in each sequence
gaps_in_seq1 <- sum(aligned_seq1 == "-")
gaps_in_seq2 <- sum(aligned_seq2 == "-")

# Total gaps in the alignment
total_gaps <- gaps_in_seq1 + gaps_in_seq2


# Print results
cat("Alignment Score:", alignment_score, "\n")
```

```
## Alignment Score: 33605
```

```r
cat("Matches:", num_matches, "\n")
```

```
## Matches: 8724
```

```r
cat("Mismatches:", num_mismatches, "\n")
```

```
## Mismatches: 55
```

```r
cat("Gaps in sequence 1:", gaps_in_seq1, "\n")
```

```
## Gaps in sequence 1: 0
```

```r
cat("Gaps in sequence 2:", gaps_in_seq2, "\n")
```

```
## Gaps in sequence 2: 0
```

```r
cat("Total gaps in alignment:", total_gaps, "\n")
```

```
## Total gaps in alignment: 0
```