Introduction)

Background: Dermatitis is a skin condition where one may have overly dry and weak skin which can then be prone to infections, redness, and other ailments. Dermatitis is known to go away with age. 10% of children are diagnosed with dermatitis then grow out of it, while only 1% of adults observe dermatitis (Golonzhka O, Leid M, Indra G, Indra AK. Expression of COUP-TF-interacting protein 2 (CTIP2) in mouse skin during development and in adulthood. Gene Expr Patterns. 2007 Aug;7(7):754-60. doi: 10.1016/j.modgep.2007.06.002. Epub 2007 Jun 13. PMID: 17631058; PMCID: PMC2063996.https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8383845/). So then the protein linked to dermatitis by being a component in building up the skin barrier (this linkage has currently only been tested in mice studies but the protein itself is found in both humans and mice). CTIP2 is found to actual be expressed fewer in adult healthy mice then when developing embryos based on research findings(Ganguli-Indra G, Liang X, Hyter S, Leid M, Hanifin J, Indra AK. Expression of COUP-TF-interacting protein 2 (CTIP2) in human atopic dermatitis and allergic contact dermatitis skin. Exp Dermatol. 2009 Nov;18(11):994-6. doi: 10.1111/j.1600-0625.2009.00876.x. Epub 2009 Mar 29. PMID: 19366371; PMCID: PMC2783464. https://pubmed.ncbi.nlm.nih.gov/19366371/). CTIP2 is also sometimes called as BC11B, so some data may also be named BC11B.

Scientific question: So then how closely similar are the CTIP2 protein in mice are to humans (in terms of DNA sequence and protein structure), in order to make an assumption that CTIP2 is also linked to dermatitis in humans?

Scientific hypothesis: If the human and mouse CTIP2 protein are closely similar, then we can assume their protein structures and DNA sequences to be similar as well.

The first database I'll be utilizing is the NCBI(Nucleotide database). I can find 2 different gene encoding for Ctip2, one for humans and mice. Then the method I'll be using from method list 1 is Pairwise sequence alignment to analyze the different genes to see how similar they are. I'll plot the results on a Sequence Logos.

Next I'll USE swiss to find the different proteins PDB files and use Homology Modeling and Structural Bioinformatics to compare them in PyMol. I'll plot the results with 3D protein.

The code: Loading in Packages)

```
#Most of these packages you should already have pre-installed in your R-studio. If not you can  uncomment the line below to
install said packages first and then you are able to load them in with the library function.

#install.packages("BiocManager")
#install.packages("Biostrings")
#install.packages("Seqnir")
#install.packages("DESeq2")
#install.packages('readr')
#install.packages("ggseqlogo")
#install.packages("vembedr)
#install.packages("NGLVieweR")


#When loading in the packages here through the library function the console may asked whether
#you want to "Update all/some/none"?
#You can just type n for none in the console:

library(BiocManager)
```

```
## Bioconductor version '3.14' is out-of-date; the current release version '3.15'
##   is available with R version '4.2'; see https://bioconductor.org/install
```

```
# BiocManager package allows for access to other bioinformatics packages within BiocManager. Able to access various statisti
cal analysis's and genomic data. It helps install other package aa well like MSA. https://cran.r-project.org/web/packages/Bi
ocManager/vignettes/BiocManager.html

library(Biostrings)
```

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

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

```
## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs
```

```
## The following objects are masked from 'package:base':
##
##     anyDuplicated, 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:base':
##
##     expand.grid, I, unname
```

```
## Loading required package: IRanges
```

```
##
## Attaching package: 'IRanges'
```

```
## The following object is masked from 'package:grDevices':
##
##     windows
```

```
## Loading required package: XVector
```

```
## Loading required package: GenomeInfoDb
```

```
##
## Attaching package: 'Biostrings'
```

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

```
# ^^ Allows for use of many sequence allighmnet functions, like nucleotideSubsitutionMatrix(). I used this packae with my me
thos 1 of pairwise allighment method. https://bioconductor.org/packages/release/bioc/html/Biostrings.html

library(readr)
#install readr if not done already, readr helps provide statistical methods. It's a way to quicly read in rectanglar data an
d other files like CSV and TSV files.Able to parse many datatypes.
#https://readr.tidyverse.org/

library(vembedr)
# The vembedr package lets you embed video into your HTML pages in R with the embed_url() function. Not much else to say. I
uses this to show my comparison movie of the two protein structures of CTIP2.
#https://cran.r-project.org/web/packages/vembedr/index.html

require(ggplot2)
```

```
## Loading required package: ggplot2
```

```
require(ggseqlogo)
```

```
## Loading required package: ggseqlogo
```

```
# ^^ These two packages help with the sequence logos method. Helping take in multiple sequence alignments and then spitting
out the derived seq logos from them. Can either take in pariwise or multiple sequence alignments but they must be in a chara
cter vector.
# https://omarwagih.github.io/ggseqlogo/

library(NGLVieweR)
# NGLVieweR can be used to visualize and interact with Protein Data Bank (PDB) and structural files in R. It includes a set
of API functions to manipulate the viewer after creation in Shiny. Able to apply filters like sticks and mesh on the 3d visu
alization.
#https://cran.r-project.org/web/packages/NGLVieweR/index.html
```

Performing Bioinformatics Analysis)

Reading in the data:

```
# Using the first dataset NCBI(Nucleotides) to find the human and mice genes of CTIP2 as fasta files and then reading them i
n. This is creating all global variables here which I access throughout my code at various points:
Human_CTIP2 <- readAAStringSet(file = "Human CTIP2.fasta") #BP length of 8525 #https://www.ncbi.nlm.nih.gov/nuccore/NM_00128
2237.2
Mice_CTIP2 <- readAAStringSet(file = "Mice CTIP2.fasta") #BP length of 2697 #https://www.ncbi.nlm.nih.gov/nuccore/AF186019.1
#^^ The code creates a single long string all in uppercase letters
```

Method 1 Pairwise sequence alignment:

```
# The pairwise sequence alignment mehod will allign the 2 sequences give to the best match, then calcualting a score based o
n that new allignment with it adding to the score everythime there is a match or subtracting from everytime their isn't.



#source: https://a-little-book-of-r-for-bioinformatics.readthedocs.io/en/latest/src/chapter4.html
nucleotidematrix = nucleotideSubstitutionMatrix(match = 2, mismatch = -1, baseOnly = TRUE)
# This function just helps to create a nucleotide matirix to compare different nucleotide, which we'll use later when prefor
ming pairwaise sequence allighment on the DNA of CTIP2 in both mice and humans down below.

# nucleotidematrix
# Uncomment the code above check to see if the nucleotide matrix is properly made and has ACTG.

globalAlignsequence1sequence2 = pairwiseAlignment(Human_CTIP2, Mice_CTIP2, substitutionMatrix = nucleotidematrix, gapOpening
= -2, gapExtension = -8, scoreOnly = FALSE)
globalAlignsequence1sequence2
```

```
## Global PairwiseAlignmentsSingleSubject (1 of 1)
## pattern: AGCCATAGAGAGACCGAGAGCTCCCAGAGAACCC...AAAAATAAATTGGACATTTAACTTGATCTCCAAA
## subject: -G---------------------------------...-----------------------------------
## score: -42444
```

```
# This code just compares the sequences given. It takes in a singular long string as the sequence and also the the nucelotid
e  as a matrix in order to compare them. The it spits out best alignment pattern and a score.

#Make your own function
test_alignment_score = function(x){ #X is a local variable made in my function and can only be accessed within the function.

  if (x  > 0){ #This if statement helps check whether the score is positive or not.
    print("It's a positive score and a good alighment because the sequences are similar")
  }

  else{ #The else statement helps check whether the score is positive or not.
    print("It's a negative score and so a bad alighment because the sequences aren't similar")
  }

}
```

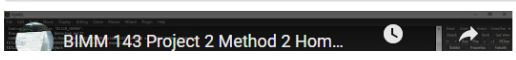
```
test_alignment_score(-42444)
```

```
## [1] "It's a negative score and so a bad alighment because the sequences aren't similar"
```

```
# Testing the aliment score we got from the pairwise Alignment function in the code chunk above. I input the score that was
give to me for the aliment score of the CTIP2 comparison.
```

Method 2, Homolgy model and structural bioinformatics.

```
# This method was comparing the structures of the proteins in PyMol. Making not of any observations for where their were dif
ferences or similarities.


embed_url("https://youtu.be/-wZdZdDbmi4")
```




```
# This code simply opens up the link to my movie comparing the protein structures of CTIP2 in both humans and mice
```

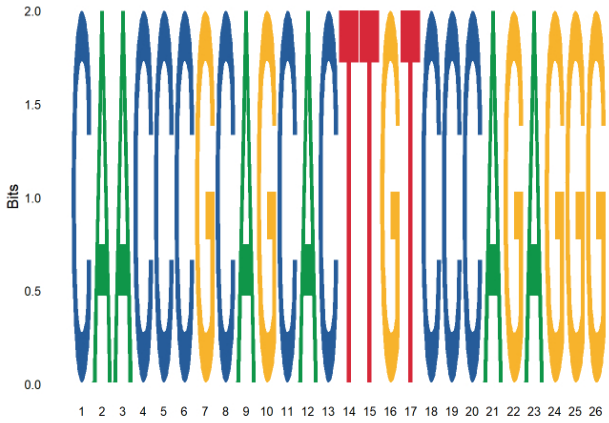
Plotting The Results)

Sequence logos data analysis https://omarwagih.github.io/ggseqlogo/#getting_started

```
#Sequence logo helps present the most fundamental information from a multiple alignment. With hight of the character correla
ting to conservation of that nucleotide in that alighment column and width of that character representing the abundance of t
hat nucelotide in that alignment column


#Checks to see if the ggseqlogo() functions properly work using the sample data given in the package.
#ggseqlogo( seqs_dna$MA0001.1)
#ggseqlogo( seqs_aa$AKT1 )

#You can also pass in a list of sequences directly into the ggseqlogo() function to get a seq logo so long as the sequences
are of the same length.
#multi = c(substring(Human_CTIP2,1025,1040), substring(Mice_CTIP2,1025,1040))
#ggseqlogo(multi)



logo_globalAlignsequence1sequence2 = as.character(globalAlignsequence1sequence2)
#^^ Converts a datatype into a character vector
trimlogo_globalAlignsequence1sequence2 = substring(logo_globalAlignsequence1sequence2,1000,1025)
#substring() is a built-in Bioconductor function. It takes in a character vector and helps trim it down to the specified par
ameters.

ggseqlogo(trimlogo_globalAlignsequence1sequence2)
```

```
## Warning in bits_method(seqs, decreasing = rev_stack_order, seq_type =
## seq_type, : All positions have zero information content perhaps due to too few
## input sequences. Setting all information content to 2.
```

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.
```




```
# Takes in a character to give you a sequence logo.
```

3D protein

```
# https://github.com/nvelden/NGLVieweR#:~:text=NGLvieweR%20provides%20an%20R%20interface,in%20R%20and%20Shiny%20application
s.
# Source: https://cran.r-project.org/web/packages/NGLVieweR/vignettes/NGLVieweR.html



NGLVieweR("BC11B_HUMAN.pdb") %>%
addRepresentation("cartoon")
```

```
# ^^ This code presents the 3D image of the protein as given by the PDB file. This once plots the protein of human CTIP2. Rm
ber CTIP2 is also sometimes referred to as BC11B.
# https://swissmodel.expasy.org/repository/uniprot/Q9C0K0
NGLViewR("BC11B_MOUSE.pdb") %>%
addRepresentation("cartoon")
```

```
# ^^ This one plots the  CTIP2 protein for mice.
# https://swissmodel.expasy.org/repository/uniprot/Q99PV8
```

Analyzing the Results)

Given the small number from the pairwise allignment, then we can probably deduce that the mouse gene and the human gene encoding for the CTIP2 are pretty different. Especially with the great size difference between them. Which makes sense that a human gene may be bigger then a mouse since the human genome is 2.9 billion base pairs long and the mouse is 2.5 billion basepairs long. This low alighement score makes sense with the sequence logos we got. Our sequence logo appears to only show single sequence when in fact it's possible we trimmed an area where there wasn't any alignment thus seq logo didn't see the mouse sequence and only the human sequence since the human sequence I used was 8000 bp long and the mouse sequence was 2000 bp long. Perhaps the human sequence has more introns then the mouse to bulk out it's sequence but still make the same protein as seen when we compare and visualize the protein models for the both of them.

Thus perhaps in humans they utilize a different sequence full of introns which bulk out the sequence to be longer but not affect the actual transcription and translation for the protein, resulting in the mouse and human CTIP2 protein being initally the same. But when it comes to the pathway, both proteins end up folding slightly differently to perhaps bind to different DNA binding domains. Thus the CTIP2 protein itself is conserved in both organism and maybe also it's function in dermatitis as well.