Introduction)

Scientific question: Dermititus is known to go away with age. 10% of children are diagnosed with demritis then grow out of it. Only 1% of adults observe dermatitis. 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 is found in both humans and mice) CTIP2 shouldn't really differ in terms of structure between ages. CTIP2 is found to actual be expressed fewer in adult healthy mice then when developing embryos. So then how closely related and similar are the CTIP2 protein in mice is to humans in order to make any assumption that CTIP2 is also linked to dermatitis in humans?

##^^ How can I tighten up my scientific question? Shorten it and make it conciser I guess.

Scientific hypothesis: If the human and mouse CTIP2 protein are closely related, then we can assume their structures and links to the dermatitis pathway and other properties of their proteins to also be similar.

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.

Next maybe I could try using PDB to find the different proteins and use Homology Modeling and Structural Bioinformatics to compare them.

I could use analysis methods like P-values to test the significance of their similarity and 3D protein measurements to check their protein structures.

The code: Loading in Packages)

```
#Most of these packages you should already have pre-installed in your R-studio. If not you can use can
#BiocManager::install(BiocManager)
#BiocManager::install(Biostrings)
#BiocManager::install(Seqnir)

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## Bioconductor version '3.14' is out-of-date; the current release version '3.15'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
```

```
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()
library(seqinr)
```

```
##
## Attaching package: 'seqinr'

## The following object is masked from 'package:Biostrings':
##
##     translate
```

```
# ^^ Helps to be able to use sequences in fasta files.

BiocManager::install("DESeq2")
```

```
## Bioconductor version 3.14 (BiocManager 1.30.18), R 4.1.3 (2022-03-10)

## Warning: package(s) not installed when version(s) same as current; use ‘force = TRUE‘ to
##   re-install: 'DESeq2'

## Installation paths not writeable, unable to update packages
##   path: C:/Program Files/R/R-4.1.3/library
##   packages:
##     cluster, MASS, Matrix, mgcv, nlme, survival

## Old packages: 'AlgDesign', 'cli', 'openssl', 'segmented', 'xfun'
```

```
library("DESeq2")
```

```
## Loading required package: GenomicRanges

## Loading required package: SummarizedExperiment

## Loading required package: MatrixGenerics

## Loading required package: matrixStats

##
## Attaching package: 'matrixStats'

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

##
## Attaching package: 'MatrixGenerics'

## The following objects are masked from 'package:matrixStats':
##
##     colAlls, colAnyNAs, colAnys, colAvgsPerRowSet, colCollapse,
##     colCounts, colCummaxs, colCummins, colCumprods, colCumsums,
##     colDiffs, colIQRDiffs, colIQRs, colLogSumExps, colMadDiffs,
##     colMads, colMaxs, colMeans2, colMedians, colMins, colOrderStats,
##     colProds, colQuantiles, colRanges, colRanks, colSdDiffs, colSds,
##     colSums2, colTabulates, colVarDiffs, colVars, colWeightedMads,
##     colWeightedMeans, colWeightedMedians, colWeightedSds,
##     colWeightedVars, rowAlls, rowAnyNAs, rowAnys, rowAvgsPerColSet,
##     rowCollapse, rowCounts, rowCummaxs, rowCummins, rowCumprods,
##     rowCumsums, rowDiffs, rowIQRDiffs, rowIQRs, rowLogSumExps,
##     rowMadDiffs, rowMads, rowMaxs, rowMeans2, rowMedians, rowMins,
##     rowOrderStats, rowProds, rowQuantiles, rowRanges, rowRanks,
##     rowSdDiffs, rowSds, rowSums2, rowTabulates, rowVarDiffs, rowVars,
##     rowWeightedMads, rowWeightedMeans, rowWeightedMedians,
##     rowWeightedSds, rowWeightedVars

## Loading required package: Biobase
```

```
## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     'browseVignettes()'. To cite Bioconductor, see
##     'citation("Biobase")', and for packages 'citation("pkgname")'.

##
## Attaching package: 'Biobase'

## The following object is masked from 'package:MatrixGenerics':
##
##     rowMedians

## The following objects are masked from 'package:matrixStats':
##
##     anyMissing, rowMedians
```

```
#install readr if not done already, it helps provide statistical methods.
library(readr)

if(!require('BSDA')){
  install.packages('BSDA')
  library('BSDA')
}
```

```
## Loading required package: BSDA

## Loading required package: lattice

##
## Attaching package: 'BSDA'

## The following object is masked from 'package:datasets':
##
##     Orange
```

Performing Bioinformatics Analysis)

```
# Using the first dataset NCBI(Nucleotides) to find the human and mice genes of CTIP2
Human_CTIP2 <- readAAStringSet(file = "Human CTIP2.fasta")
Mice_CTIP2 <- readAAStringSet(file = "Mice CTIP2.fasta")
#^^ Creats a single long string all in uppercase letters

Human_CTIP2read <- read.fasta(file = "Human CTIP2.fasta")  #BP length of 8525
Mice_CTIP2read <- read.fasta(file = "Mice CTIP2.fasta")    #BP length of 2697
# ^^ Creates a vector of individual lowercase letters

#https://a-little-book-of-r-for-bioinformatics.readthedocs.io/en/latest/src/chapter4.html
nucleotidematrix = nucleotideSubstitutionMatrix(match = 2, mismatch = -1, baseOnly = TRUE)
nucleotidematrix
```

```
##    A  C  G  T
## A  2 -1 -1 -1
## C -1  2 -1 -1
## G -1 -1  2 -1
## T -1 -1 -1  2
```

```
globalAlignsequence1sequence2 = pairwiseAlignment(Human_CTIP2, Mice_CTIP2, substitutionMatrix = nucleoti
globalAlignsequence1sequence2
```

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

Such a low score means the two sequences are very similar. Especially with them given the big size difference between the two genes.

Next maybe I could try using PDB to find the different proteins and use Homology Modeling and Structural Bioinformatics to compare them.

^^ I ended up using NCBI(geo) and using differential expression

```
dataset2 = read.csv("Project 2 dataset2.tgz")
```

```
## Warning in read.table(file = file, header = header, sep = sep, quote = quote, :
## line 1 appears to contain embedded nulls
```

```
## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## embedded nul(s) found in input
```

```
dataset2
```

```
##
## 1
## 2
## 3
## 4
## 5
## 6
## 7
## 8
## 9
## 10
## 11
## 12
## 13
## 14
## 15
## 16
## 17
```

```
## 18
## 19
## 20
## 21
## 22
## 23
## 24
## 25
## 26
## 27
## 28
## 29
## 30
## 31
## 32
## 33
## 34
## 35
## 36
## 37
## 38
## 39
## 40
## 41
## 42
## 43
## 44
## 45
## 46
## 47
## 48
## 49
## 50
## 51
## 52
## 53
## 54
## 55
## 56
## 57
## 58
## 59
## 60
## 61
## 62
## 63
## 64
## 65
## 66
## 67
## 68
## 69
## 70
## 71
```

```
## 72
## 73
## 74
## 75
## 76
## 77
## 78
## 79
## 80
## 81
## 82
## 83
## 84
## 85
## 86
## 87
## 88
## 89
## 90
## 91
## 92
## 93
## 94
## 95
## 96
## 97
## 98
## 99
## 100
## 101
## 102
## 103
## 104
## 105
## 106
## 107
## 108
## 109
## 110
## 111
## 112
## 113
## 114                    Analysis of library complexity and per-base sequence quality (i.e. q &gt; 30)
## 115
## 116
## 117
## 118
## 119
## 120
## 121
## 122
## 123
## 124
## 125
```

```
## 126
## 127
## 128
## 129
## 130
## 131
## 132
## 133
## 134
## 135
## 136
## 137
## 138
## 139
## 140
## 141
## 142
## 143
## 144
## 145
## 146
## 147
## 148
## 149
## 150
## 151
## 152
## 153
## 154
## 155
## 156
## 157
## 158
## 159
## 160
## 161
## 162
## 163
## 164
## 165
## 166
## 167
## 168
## 169
## 170
## 171
## 172
## 173
## 174
## 175
## 176
## 177
## 178
## 179
```

```
## 180
## 181
## 182
## 183
## 184
## 185
## 186
## 187
## 188                    Analysis of library complexity and per-base sequence quality (i.e. q &gt; 30)
## 189
## 190
## 191
## 192
## 193
## 194
## 195
## 196
## 197
## 198
## 199
## 200
## 201
## 202
## 203
## 204
## 205
## 206
## 207
## 208
## 209
## 210
## 211
## 212
## 213
## 214
## 215
## 216
## 217
## 218
## 219
## 220
## 221
## 222
## 223
## 224
## 225
## 226
## 227
## 228
## 229
## 230
## 231
## 232
## 233
```

```
## 234
## 235
## 236
## 237
## 238
## 239
## 240
## 241
## 242
## 243
## 244
## 245
## 246
## 247
## 248
## 249
## 250
## 251
## 252
## 253
## 254
## 255
## 256
## 257
## 258
## 259
## 260
## 261
## 262                    Analysis of library complexity and per-base sequence quality (i.e. q &gt; 30)
## 263
## 264
## 265
## 266
## 267
## 268
## 269
## 270
## 271
## 272
## 273
## 274
## 275
## 276
## 277
## 278
## 279
## 280
## 281
## 282
## 283
## 284
## 285
## 286
## 287
```

```
## 288
## 289
## 290
## 291
## 292
## 293
## 294
## 295
## 296
## 297
## 298
## 299
## 300
## 301
## 302
## 303
## 304
## 305
## 306
## 307
## 308
## 309
## 310
## 311
## 312
## 313
## 314
## 315
## 316
## 317
## 318
## 319
## 320
## 321
## 322
## 323
## 324
## 325
## 326
## 327
## 328
## 329
## 330
## 331
## 332
## 333
## 334
## 335
## 336                    Analysis of library complexity and per-base sequence quality (i.e. q &gt; 30)
## 337
## 338
## 339
## 340
## 341
```

```
## 342
## 343
## 344
## 345
## 346
## 347
## 348
## 349
## 350
## 351
## 352
## 353
## 354
## 355
## 356
## 357
## 358
## 359
## 360
## 361
## 362
## 363
## 364
## 365
## 366
## 367
## 368
## 369
## 370
## 371
## 372
## 373
## 374
## 375
## 376
## 377
## 378
## 379
## 380
## 381
## 382
## 383
## 384
## 385
## 386
## 387
## 388
## 389
## 390
## 391
## 392
## 393
## 394
## 395
```

```
## 396
## 397
## 398
## 399
## 400
## 401
## 402
## 403
## 404
## 405
## 406
## 407
## 408
## 409
## 410                    Analysis of library complexity and per-base sequence quality (i.e. q &gt; 30)
## 411
## 412
## 413
## 414
## 415
## 416
## 417
## 418
## 419
## 420
## 421
## 422
## 423
## 424
## 425
## 426
## 427
## 428
## 429
## 430
## 431
## 432
## 433
## 434
## 435
## 436
## 437
## 438
## 439
## 440
## 441
## 442
## 443
## 444
## 445
## 446
## 447
## 448
## 449
```

```
## 450
## 451
## 452
## 453
## 454
## 455
## 456
## 457
## 458
## 459
## 460
## 461
## 462
## 463
## 464
## 465
## 466
## 467
## 468
## 469
## 470
## 471
## 472
## 473
## 474
## 475
## 476
## 477
## 478
## 479
## 480
## 481
## 482
## 483
## 484                    Analysis of library complexity and per-base sequence quality (i.e. q &gt; 30)
## 485
## 486
## 487
## 488
## 489
## 490
## 491
## 492
## 493
## 494
## 495
## 496
## 497
## 498
## 499
## 500
## 501
## 502
## 503
```

```
## 504
## 505
## 506
## 507
## 508
## 509
## 510
## 511
## 512
## 513
## 514
## 515
## 516
## 517
## 518
## 519
## 520
## 521
## 522
## 523
## 524
## 525
## 526
## 527
## 528
## 529
## 530
## 531
## 532
## 533
## 534
## 535
## 536
## 537
## 538
## 539
## 540
## 541                                                                    the cells formed a meshwork
## 542
## 543
## 544
## 545
## 546
## 547
## 548   the cells were rinsed with PBS and harvested 140 with ice-cold Trizol or fixed with 4% para-fo
## 549
## 550
## 551
## 552
## 553
## 554
## 555
## 556
## 557
```

```
## 558
## 559
## 560
## 561
## 562
## 563
## 564
## 565                    Analysis of library complexity and per-base sequence quality (i.e. q &gt; 30)
## 566
## 567
## 568
## 569
## 570
## 571
## 572
## 573
## 574
## 575
## 576
## 577
## 578
## 579
## 580
## 581
## 582
## 583
## 584
## 585
## 586
## 587
## 588
## 589
## 590
## 591
## 592
## 593
## 594
## 595
## 596
## 597
## 598
## 599
## 600
## 601
## 602
## 603
## 604
## 605
## 606
## 607
## 608
## 609
## 610
## 611
```

612
613
614
615
616
617
618
619
620
621
622 the cells formed a meshwork
623
624
625
626
627
628
629 the cells were rinsed with PBS and harvested 140 with ice-cold Trizol or fixed with 4% para-fo
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646 Analysis of library complexity and per-base sequence quality (i.e. q > 30)
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665

```
## 666
## 667
## 668
## 669
## 670
## 671
## 672
## 673
## 674
## 675
## 676
## 677
## 678
## 679
## 680
## 681
## 682
## 683
## 684
## 685
## 686
## 687
## 688
## 689
## 690
## 691
## 692
## 693
## 694
## 695
## 696
## 697
## 698
## 699
## 700
## 701
## 702
## 703                                                                    the cells formed a meshwork
## 704
## 705
## 706
## 707
## 708
## 709
## 710   the cells were rinsed with PBS and harvested 140 with ice-cold Trizol or fixed with 4% para-fo
## 711
## 712
## 713
## 714
## 715
## 716
## 717
## 718
## 719
```

```
## 720
## 721
## 722
## 723
## 724
## 725
## 726
## 727                    Analysis of library complexity and per-base sequence quality (i.e. q &gt; 30)
## 728
## 729
## 730
## 731
## 732
## 733
## 734
## 735
## 736
## 737
## 738
## 739
## 740
## 741
## 742
## 743
## 744
## 745
## 746
## 747
## 748
## 749
## 750
## 751
## 752
## 753
## 754
## 755
## 756
## 757
## 758
## 759
## 760
## 761
## 762
## 763
## 764
## 765
## 766
## 767
## 768
## 769
## 770
## 771
## 772
## 773
```

```
## 774
## 775
## 776
## 777
## 778
## 779
## 780
## 781
## 782
## 783
## 784                                                                    the cells formed a meshwork
## 785
## 786
## 787
## 788
## 789
## 790
## 791   the cells were rinsed with PBS and harvested 140 with ice-cold Trizol or fixed with 4% para-fo
## 792
## 793
## 794
## 795
## 796
## 797
## 798
## 799
## 800
## 801
## 802
## 803
## 804
## 805
## 806
## 807
## 808                    Analysis of library complexity and per-base sequence quality (i.e. q &gt; 30)
## 809
## 810
## 811
## 812
## 813
## 814
## 815
## 816
## 817
## 818
## 819
## 820
## 821
## 822
## 823
## 824
## 825
## 826
## 827
```

828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865 the cells formed a meshwork
866
867
868
869
870
871
872 the cells were rinsed with PBS and harvested 140 with ice-cold Trizol or fixed with 4% para-fo
873
874
875
876
877
878
879
880
881

```
## 882
## 883
## 884
## 885
## 886
## 887
## 888
## 889                    Analysis of library complexity and per-base sequence quality (i.e. q &gt; 30)
## 890
## 891
## 892
## 893
## 894
## 895
## 896
## 897
## 898
## 899
## 900
## 901
## 902
## 903
## 904
## 905
## 906
## 907
## 908
## 909
## 910
## 911
## 912
## 913
## 914
## 915
## 916
## 917
## 918
## 919
## 920
## 921
## 922
## 923
## 924
## 925
## 926
## 927
## 928
## 929
## 930
## 931
## 932
## 933
## 934
## 935
```

```
## 936
## 937
## 938
## 939
## 940
## 941
## 942
## 943
## 944
## 945
## 946                                                                    the cells formed a meshwork
## 947
## 948
## 949
## 950
## 951
## 952
## 953   the cells were rinsed with PBS and harvested 140 with ice-cold Trizol or fixed with 4% para-fo
## 954
## 955
## 956
## 957
## 958
## 959
## 960
## 961
## 962
## 963
## 964
## 965
## 966
## 967
## 968
## 969
## 970                    Analysis of library complexity and per-base sequence quality (i.e. q &gt; 30)
## 971
## 972
## 973
## 974
## 975
## 976
## 977
## 978
## 979
## 980
## 981
## 982
## 983
## 984
## 985
## 986
## 987
## 988
## 989
```

```
## 990
## 991
## 992
## 993
## 994
## 995
## 996
## 997
## 998
## 999
## 1000
## 1001
## 1002
## 1003
## 1004
## 1005
## 1006
## 1007
## 1008
## 1009
## 1010                                    as well as other locations throughout the genome. These tra
## 1011
## 1012
## 1013
## 1014
## 1015
## 1016                                  12 total polyA selected samples. 6 IPSC samples with 3 biolog
## 1017
## 1018
## 1019
## 1020
## 1021
## 1022
## 1023
## 1024
## 1025
## 1026
## 1027
## 1028
## 1029
## 1030
## 1031
## 1032
## 1033
## 1034
## 1035
## 1036
## 1037
## 1038
## 1039
## 1040
## 1041
## 1042
## 1043
```

```
## 1044
## 1045
## 1046
## 1047
## 1048
## 1049
## 1050
## 1051
## 1052
## 1053
```

```
## ^^ Having trouble finding the right second data set for my project.
```

```
dds = DESeqDataSetFromMatrix(dataset2)
```

```
## Error in h(simpleError(msg, call)): error in evaluating the argument 'x' in selecting a method for fu
```

```
dds = DESeq(dds)
```

```
## Error in is(object, "DESeqDataSet"): object 'dds' not found
```

```
res = results(dds)
```

```
## Error in is(object, "DESeqDataSet"): object 'dds' not found
```

Plotting The Results)

P-value

```
sd_Human_CTIP2read = sapply(Human_CTIP2read[], sd)
```

```
## Warning in is.data.frame(x): NAs introduced by coercion
```

```
sd_Human_CTIP2read
```

```
## NM_001282237.2
##             NA
```

```
sd_Mice_CTIP2read = sapply(Mice_CTIP2read[], sd)
```

```
## Warning in is.data.frame(x): NAs introduced by coercion
```

```
sd_Mice_CTIP2read
```

```
## AF186019.1
##         NA
```

```
z.test(Human_CTIP2read, Mice_CTIP2read, alternative = "two.sided", mu = 0, sigma.x =sd_Human_CTIP2read,
```

```
## Error in z.test(Human_CTIP2read, Mice_CTIP2read, alternative = "two.sided", : not enough x observati
```

3D protein

```
library(bio3d)
```

```
##
## Attaching package: 'bio3d'

## The following object is masked from 'package:SummarizedExperiment':
##
##     trim

## The following object is masked from 'package:GenomicRanges':
##
##     trim

## The following objects are masked from 'package:seqinr':
##
##     consensus, read.fasta, write.fasta

## The following object is masked from 'package:Biostrings':
##
##     mask

## The following object is masked from 'package:IRanges':
##
##     trim
```

```
pdb = read.pdb("3CJW") #human, mouse is 3TNQ
```

```
##   Note: Accessing on-line PDB file
##    PDB has ALT records, taking A only, rm.alt=TRUE
```

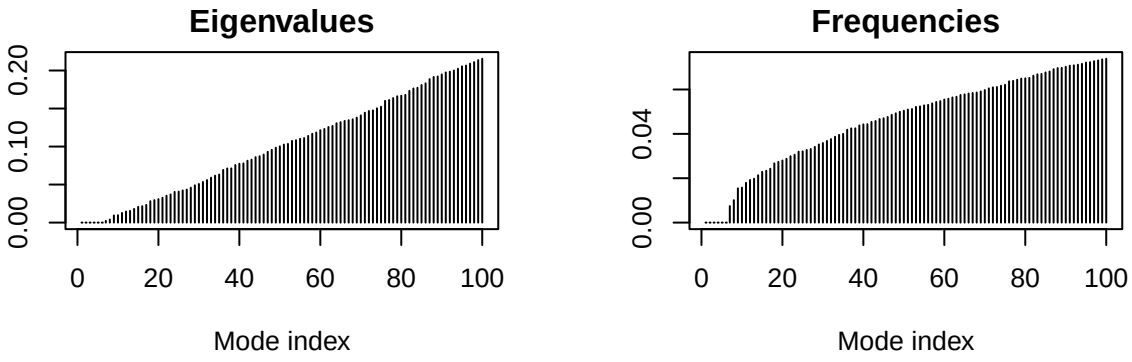
```
modes = nma(pdb)
```

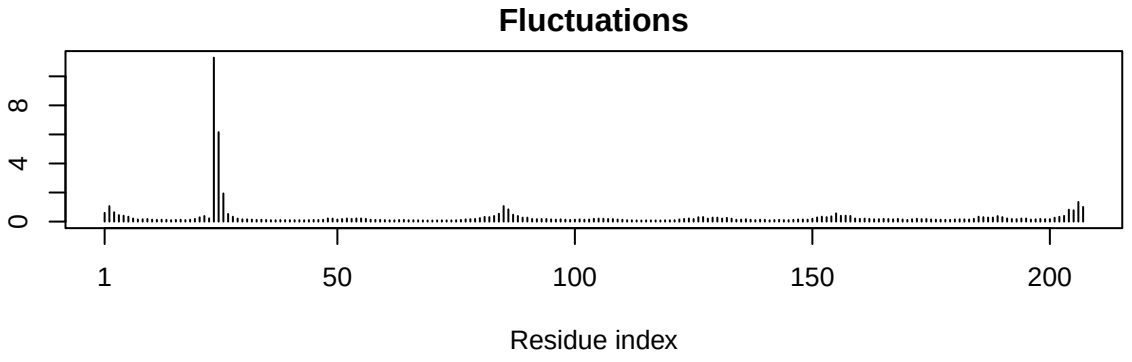
```
## Warning in nma.pdb(pdb): Possible multi-chain structure or missing in-structure residue(s) present
##   Fluctuations at neighboring positions may be affected.

##  Building Hessian...     Done in 0.03 seconds.
##  Diagonalizing Hessian...    Done in 0.36 seconds.
```

```
print(modes)
```

```
##
## Call:
##   nma.pdb(pdb = pdb)
##
## Class:
##   VibrationalModes (nma)
##
## Number of modes:
##   621 (6 trivial)
##
## Frequencies:
##   Mode 7:    0.007
##   Mode 8:    0.01
##   Mode 9:    0.015
##   Mode 10:   0.016
##   Mode 11:   0.018
##   Mode 12:   0.019
##
## + attr: modes, frequencies, force.constants, fluctuations,
##         U, L, xyz, mass, temp, triv.modes, natoms, call
```

```
plot(modes)
```

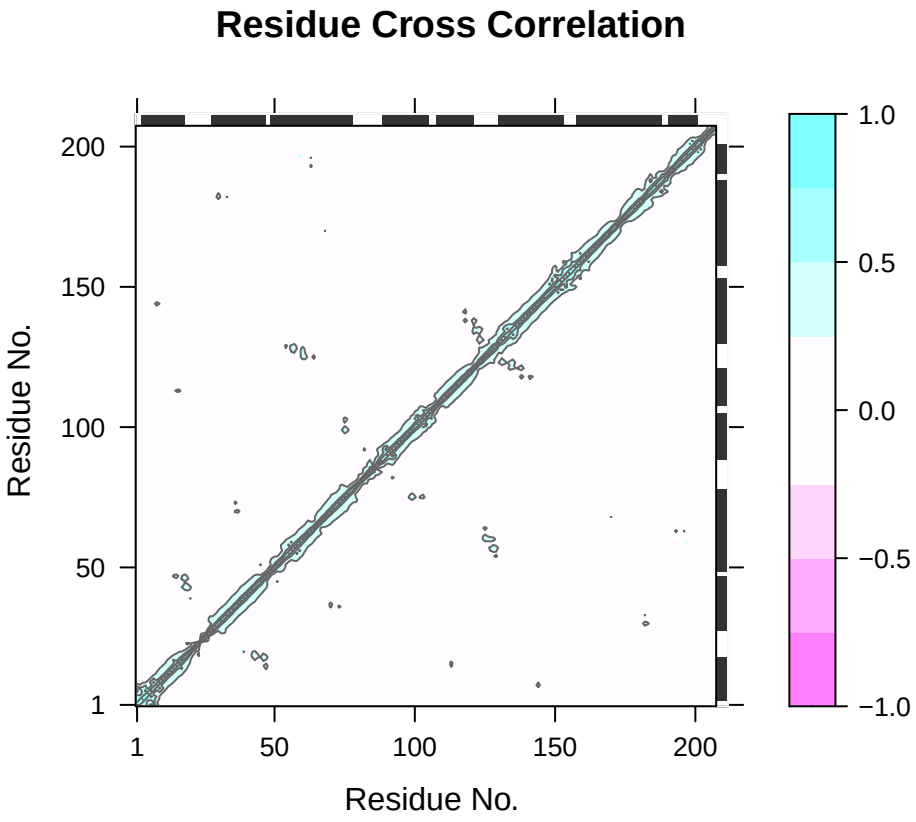





```
cm = dccm(modes)
```

```
##   |                                                                      |
```

```
plot(cm, sse=pdb)
```




```
modes.anhm = nma(pdb, ff="anm")
```

```
## Warning in nma.pdb(pdb, ff = "anm"): Possible multi-chain structure or missing in-structure residue(s
##   Fluctuations at neighboring positions may be affected.

##  Building Hessian...     Done in 0.02 seconds.
##  Diagonalizing Hessian...    Done in 0.43 seconds.
```

```
r = rmsip(modes, modes.anm)
```

```
## Error in .fetchmodes(modes.b, subset = subset): object 'modes.anm' not found
```

```
view.modes(modes, mode=7, launch=T)
```

```
## Error in view.modes(modes, mode = 7, launch = T): could not find function "view.modes"
```

```
a = mktrj(modes, mode = 7)
```

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 or pretty different. Especially with the great size difference between them.

At least from the previews on the PDB website the protein structures for human vs the mouse seem pretty different and disimilar.

Perhaps from this the linkage of the CTIP2 protein to dermatitis in mice can't also be said the same for humans.