Pathogenic posIDs- adding grantham hi low scores to aa level df that has window burden info. from mod4

10/23/22 saved originaally as 'FIGURE4_CpDAA_missenseLevelScore..' jnb

3/13/23 updated name, added more score grantham columns, duplicated this jnb to make pathogenic CKY pos speicifc version

---

input data:

dir:

> /Users/mariapalafox/Desktop/BRIDGE/DISORDER/dbNSFPmapping/MAXMEAN_SCORIN(

- Originally downloaded filtered CpDAAoCATEGORY dbNSFP files from hoffman ```sh # INPUT FOR THIS MARKDOWN & FIGURE 3 scp mfpalafo@hoffman2.idre.ucla.edu:/u/home/m/mfpalafo/project-arboleda/patho_dbNSFP/MAXMEAN_SCORING/top10scores_MetaMerged_1323_MendelCpD .

```

### Input files made in:

- FIGURE_3c_missenseLevelScoresPt3_CpDAAoCATEGORY_MendelCpD.ipynb (pathogenic posIDs)

### Input file names for this markdown:

> **top10scores_Patho_positions_granthamHighOrLowCorrectedVersion_CKYonly_850_p**
>
> - highest or lowest grantham missense scores for pathogenic posID filtered data from OMIMCpD protein subset
> - could not make for OMIM protein subset becauses dont have all possible missense scores for every posID in OMIM, onnly downloaded OMIMCpD metamerge

---
```

output data:

dir:

/Users/mariapalafox/Desktop/BRIDGE/disorder/FIGURES/modules/STEP4/grantham/

In [13]:
```
# packages
import os
import sys
import numpy as np
import pandas as pd
import csv
from ast import literal_eval
from statistics import mean
%matplotlib inline
sys.path.append("/Users/mariapalafox/Desktop/TOOLBOXPY")
from all_funx import *

from IPython.display import display, HTML
from IPython.display import Image
display(HTML("<style>.container {width:90% !important;}</styl
e>"))
pd.set_option('display.max_columns', None)
pd.set_option('display.max_colwidth', 1000) # you can't use n
one here
pd.set_option('display.max_seq_items', 2000) # seq in column
pd.set_option('display.max_rows', 10)
%timeit
# no scientific notation display, format decimal
pd.options.display.float_format = '{:.4f}'.format

def mediumpic(name):
    display(Image(filename = name, width= 500, height=650))

def smallpic(name):
    display(Image(filename = name, width= 300, height=600))

def largepic(name):
    display(Image(filename = name, width= 800, height=1000))
```

filtered input data for CpDAA missense changes corresponding to the highest and lowest grantham score

- paper 1 si for K missense type odds ratio has different order than current project grantham reference excel **MISSENSE/GRANTHAM_CpDAA_MissenseTypes.csv**
- double check word doc and excel with other sources

highest grantham change:

```
C/W
K/I or K/M
Y/C
```

lowest grantham change:

```
C/S
K/R
Y/F
```

- Y missense types all = 2331, single codon
- C/S is most of any C missense type, with 14434 and other types at 7217
- K/R is similar to other missense type counts at 8908, but K/I and K/M are possible by only 1 of 2 codons, so these counts combined = 8908, why is K/N so high?

Grantham conservative range described below from this link (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3431198/)

title: In silico analysis of missense substitutions using sequence-alignment based methods, 2008

OVERVIEW OF IN SILICO APPROACHES TO MISSENSE SUBSTITUTION ANALYSIS

Fundamentally, there are four classes of amino acid, sequence, or structural attribute that have been used to try to distinguish between neutral and pathogenic missense substitutions in silico: (1) pairwise comparison of the physico-chemical characteristics or evolutionary substitution frequencies between the wild-type and variant amino acid, (2) evolutionary conservation at the position at which a missense substitution is observed, (3) comparison between the variant amino acid and the evolutionarily tolerated amino acid range of variation at its position in the protein, and (4) protein structural considerations. In silico missense analysis algorithms may use data from just one of these classes, or combine data from two, three, or all four of the classes.

Pairwise amino acid comparisons may be based on data from amino acid substitution scoring matrices (e.g., PAM250, BLOSUM62). These matrices were derived from the frequencies with which the 20 amino acids are observed to substitute for each other in multiple sequence alignments of related proteins [Dayhoff et al., 1978; Henikoff and Henikoff, 1992]. BLOSUM62 scores range from 4–11 for identities, from 0–3 for commonly observed substitutions, and from (–1) to (–4) for substitutions rarely observed in related proteins. The average BLOSUM62 score is

lower for pathogenic substitutions than for neutral substitutions [Ferrer-Costa et al., 2002; Balasubramanian et al., 2005]. Alternatively, comparisons may be based on amino acid physical or chemical properties.

A score called the Grantham Difference describes the difference in side chain atomic composition, polarity, and volume between two amino acids [Grantham, 1974]. Substitutions with Grantham Differences of:

- **5–60** are generally considered “conservative”,
- **60–100** “non-conservative”, and
- **> 100** “radical”

The average Grantham Difference for pathogenic substitutions is higher than for neutral substitutions [Miller et al., 2001; Abkevich et al., 2004; Balasubramanian et al., 2005]. However, these methods of pairwise amino acid comparisons alone have not led to popular missense substitution classification algorithms.

In [14]:
```
# v1 jnb PATH =  "/Users/mariapalafox/Desktop/ALL_DISORDER_NOTES/screenshots/"

largepic("/Users/mariapalafox/Desktop/ALL_DISORDER_NOTES/Dissertation/Defense/Screenshots/grantham_wiki_matrix.png")
```

Grantham's distance [edit]

Grantham's distance depends on three properties: composition, polarity and molecular volume.[4]

Distance difference *D* for each pair of amino acid *i* and *j* is calculated as:

$$D_{ij} = [\alpha(c_i - c_j)^2 + \beta(p_i - p_j)^2 + \gamma(v_i - v_j)^2]^{\frac{1}{2}}$$

where *c* = composition, *p* = polarity, and *v* = molecular volume; and are constants of squares of the inverses of the mean for each property, respectively equal to 1.833, 0.1018, 0.000399. According to Grantham's distance, most similar amino a leucine and isoleucine and the most distant are cysteine and tryptophan.

Difference *D* for amino acids[4]

Arg	Leu	Pro	Thr	Ala	Val	Gly	Ile	Phe	Tyr	Cys	His	Gln	Asn	Lys	Asp	Glu	Met	Trp	
110	145	74	58	99	124	56	142	155	144	112	89	68	46	121	65	80	135	177	Ser
	102	103	71	112	96	125	97	97	77	180	29	43	86	26	96	54	91	101	Arg
		98	92	96	32	138	5	22	36	198	99	113	153	107	172	138	15	61	Leu
			38	27	68	42	95	114	110	169	77	76	91	103	108	93	87	147	Pro
				58	69	59	89	103	92	149	47	42	65	78	85	65	81	128	Thr
					64	60	94	113	112	195	86	91	111	106	126	107	84	148	Ala
						109	29	50	55	192	84	96	133	97	152	121	21	88	Val
							135	153	147	159	98	87	80	127	94	98	127	184	Gly
								21	33	198	94	109	149	102	168	134	10	61	Ile
									22	205	100	116	158	102	177	140	28	40	Phe
										194	83	99	143	85	160	122	36	37	Tyr
											174	154	139	202	154	170	196	215	Cys
												24	68	32	81	40	87	115	His
													46	53	61	29	101	130	Gln
														94	23	42	142	174	Asn
															101	56	95	110	Lys
																45	160	181	Asp
																	126	152	Glu
																		67	Met

Wiki grantham difference table results

- cysteine change differences are all considered **radical >100** substitutions
- lysine and tyrosine have more conservative possible substitutions

--- missense Grantham -----

```
7           *K/R   26
6           K/Q   53
5           K/E   56
4           K/T   78
3           K/N   94
2           *K/M   95
1           *K/I   102
6           *C/S   112
5           C/G   159
4           C/R   180
3           C/Y   194
2           C/F   205
1           *C/W   215
6           *Y/F   22
5           Y/H   83
4           Y/N   143
3           Y/S   144
2           Y/D   160
1           *Y/C   194
```

Grantham difference article blurb at top of markdown:

- 5–60 are generally considered “conservative”,
- 60–100 “non-conservative”, and
- **> 100 “radical”**

[1] PATHO positions in OMIM&CpD proteins

In [15]:
```
os.chdir("/Users/mariapalafox/Desktop/BRIDGE/DISORDER/dbNSFPmapping/MAXMEAN_SCORING/")

# CHANGE
patho = "top10scores_Patho_positions_granthamHighOrLowCorrectedVersion_CKYonly_850_posIDs_415_proteins_OMIMCpD.csv"
patho = pd.read_csv(patho, low_memory=False)
```

```
uniqueCount(patho, 'posID')
uniqueCount2(patho, 'UKBID')

# only grantham change missense counts for category overlap o
f CpDAA codons  shown in  counts
checkColumnValue(patho, 'CATEGORY')


#making nonredundant posID version of df
dfposID =  patho.drop_duplicates(['posID'])

print( ' ----------- unique PATHO CKY in 415 proteins -------
----- ')
uniqueCount(dfposID, 'posID')
print()

print('---------------PATHO type counts in 415 proteins------
-----------')
checkColumnValue(dfposID, 'aaref')
print()

print('---------------total LP counts-----------------')
checkColumnValue(dfposID, 'LP')
crosstabit(dfposID, 'aaref', 'LP')
print()

print('---------------total RP counts-----------------')
checkColumnValue(dfposID, 'RP')
crosstabit(dfposID, 'aaref', 'RP')
print()
print('why are low and high grantham counts not the same, eac
h posID should have 1 high and 1 low  change..')
checkColumnValue(patho, 'missenseType')

# making seperate df's for highest grantham difference and lo
west grantham differences between aaref  and  aaalt in  misse
nse exchange

print('-------------------------------------------')
checkColumnValue(patho, 'Grantham.highest') # df.dtypes ==  b
ool
gHigh = patho[patho['Grantham.highest']].copy()
print('high grantham df shape: ', gHigh.shape)
checkColumnValue(gHigh, 'missenseType')
uniqueCount(gHigh, 'posID')
uniqueCount2(gHigh, 'UKBID')
print()

print('-------------------------------------------')
checkColumnValue(patho, 'Grantham.lowest') # df.dtypes ==  bo
ol
gLow = patho[patho['Grantham.lowest']].copy()
```

```
print('low grantham df shape: ', gLow.shape)
checkColumnValue(gLow, 'missenseType')
uniqueCount(gLow, 'posID')
uniqueCount2(gLow, 'UKBID')
print('-----------------------------------------')

kr = gLow[gLow['missenseType'] == 'K/R']
checkColumnValue(kr, 'codonpos')

yf = gLow[gLow['missenseType'] == 'Y/F']
checkColumnValue(yf, 'codonpos')
```

```
posID length: 1986
posID set length: 850

UKBID set length: 415

        CATEGORY  Count
0            NaN   1597
1     PATHOGENIC    314
2         RAREST     45
3            VUS     21
4           RARE      8
5  COMMON/BENIGN      1

 ----------- unique PATHO CKY in 415 proteins ------------
posID length: 850
posID set length: 850


---------------PATHO type counts in 415 proteins------------
-----
  aaref  Count
0     Y    320
1     C    286
2     K    244


---------------total LP counts-----------------
           LP  Count
0         NaN    786
1  Unliganded     56
2    Liganded      8

LP     Liganded  Unliganded  Total
aaref
C             7          34     41
K             1          22     23
Total         8          56     64


---------------total RP counts-----------------
```

```
       RP  Count
0     NaN    792
1     Low     42
2  Medium     11
3    High      5

RP     High  Low  Medium  Total
aaref
C         4   13       7     24
K         1   14       4     19
Y         0   15       0     15
Total     5   42      11     58


why are low and high grantham counts not the same, each posI
D should have 1 high and 1 low  change..
  missenseType  Count
0          C/S    572
1          Y/F    320
2          Y/C    320
3          C/W    286
4          K/R    244
5          K/M    151
6          K/I     93

-------------------------------------------
   Grantham.highest  Count
0             False   1136
1              True    850

high grantham df shape:  (850, 53)
  missenseType  Count
0          Y/C    320
1          C/W    286
2          K/M    151
3          K/I     93

posID length: 850
posID set length: 850

UKBID set length: 415


-------------------------------------------
   Grantham.lowest  Count
0             True   1136
1            False    850

low grantham df shape:  (1136, 53)
  missenseType  Count
0          C/S    572
1          Y/F    320
```

```
2          K/R    244

posID length: 1136
posID set length: 850

UKBID set length: 415

-----------------------------------------
   codonpos  Count
0         2    244

   codonpos  Count
0         2    320
```

[2] PATHO positions in OMIM&CpD proteins

In [16]:
```
# duplicate cysteine posID issue
cs = gLow[gLow['missenseType'] == 'C/S']
checkColumnValue(cs, 'codonpos')
checkColumnValue(cs, 'refcodon')

cs.groupby('codonpos').agg(
    mean_CADD_phred =('CADD_phred', mean),
    mean_phyloP100way_vertebrate =('phyloP100way_vertebrate',
mean)
)
# first position of codon  less constrained  than 2nd pos typ
ically
```

```
   codonpos  Count
0         1    286
1         2    286

  refcodon  Count
0      TGC    304
1      TGT    268
```

Out[16]:

codonpos	mean_CADD_phred	mean_phyloP100way_vertebrate
1	26.3442	7.4606
2	25.9940	7.7536

[3] PATHO positions in OMIM&CpD proteins

In [18]:
```
# Position 1 slightly higher cadd and less connserved than po
s 2
# going to count the C/S change thats at codon position 1
# arg for codonn pos 2, matches lysine and tyrosine substitut
ion

cs = cs[cs['codonpos'] == 1]
#print(cs.shape)

print('-----------high grantham-------------------------------
---')
uniqueCount(gHigh, 'posID')
checkColumnValue(gHigh, 'missenseType')
print('high grantham K missense 151 + 93: ', 151 + 93, '\n')
checkColumnValue(gHigh, 'codonpos')
print('-------------------------------------------')

print('------ low grantham-------')
gLow = pd.concat([cs, kr, yf], axis=0, ignore_index=True)
uniqueCount(gLow, 'posID')
checkColumnValue(gLow, 'missenseType')
checkColumnValue(gLow, 'codonpos')
print('-------------------------------------------')

checkColumnValue(gHigh, 'aaref')
checkColumnValue(gLow, 'aaref')
# check for exact count match
```

```
----------high grantham---------------------------------
posID length: 850
posID set length: 850

  missenseType  Count
0          Y/C    320
1          C/W    286
2          K/M    151
3          K/I     93

high grantham K missense 151 + 93:  244

   codonpos  Count
0         2    564
1         3    286

-----------------------------------------------
------ low grantham-------
posID length: 850
posID set length: 850

  missenseType  Count
0          Y/F    320
1          C/S    286
2          K/R    244

   codonpos  Count
0         2    564
1         1    286

-----------------------------------------------
  aaref  Count
0     Y    320
1     C    286
2     K    244

  aaref  Count
0     Y    320
1     C    286
2     K    244
```

[4] PATHO positions in OMIM&CpD proteins

10 raw scores:

- 8 were used to make mean and max raw/rankscore concordaance (MPC and phylop100way added later)

chose raw scores over rankscore for clustering input

In [19]:
```
# renaming grantham highessts and lowesst misesnse change df
columns
gLow = gLow[['posID',  'missenseType', 'codonType',
             'VEST4_score',
                'REVEL_score',
                'MutPred_score',
                'MVP_score',
                'BayesDel_addAF_score',
                'ClinPred_score',
                'LISTS2_score',
                'CADD_raw',
                'MPC_score',
                'phyloP100way_vertebrate']].copy()

lowColumns  =  ['posID', 'lowestGrantham.missenseType', 'lowe
stGrantham.codonType',
                'lowestGrantham.VEST4_score',
                'lowestGrantham.REVEL_score',
                'lowestGrantham.MutPred_score',
                'lowestGrantham.MVP_score',
                'lowestGrantham.BayesDel_addAF_score',
                'lowestGrantham.ClinPred_score',
                'lowestGrantham.LISTS2_score',
                'lowestGrantham.CADD_raw',
                'lowestGrantham.MPC_score',
                'lowestGrantham.phyloP100way_vertebrate']
# rename low granthaam
gLow.columns = lowColumns

gHigh = gHigh[['posID',  'missenseType', 'codonType',
               'VEST4_score',
                'REVEL_score',
                'MutPred_score',
                'MVP_score',
                'BayesDel_addAF_score',
                'ClinPred_score',
                'LISTS2_score',
                'CADD_raw',
                'MPC_score',
```

```
                'phyloP100way_vertebrate']].copy()

hiColumns  =  ['posID', 'highestGrantham.missenseType', 'high
estGrantham.codonType',
               'highestGrantham.VEST4_score',
                'highestGrantham.REVEL_score',
                'highestGrantham.MutPred_score',
                'highestGrantham.MVP_score',
                'highestGrantham.BayesDel_addAF_score',
                'highestGrantham.ClinPred_score',
                'highestGrantham.LISTS2_score',
                'highestGrantham.CADD_raw',
                'highestGrantham.MPC_score',
                'highestGrantham.phyloP100way_vertebrate']
# rename high granthaam
gHigh.columns = hiColumns
```

[5] PATHO positions in OMIM&CpD proteins

Importing window missense burden AA level df from module processing

In [20]:
```
dir4 = "/Users/mariapalafox/Desktop/BRIDGE/disorder/FIGURES/modules/STEP4/"
os.chdir(dir4)

# CHANGE
burden = pd.read_csv("CKY_level_rmd_Background_1D_3Dcounts_added_1234_MendelCpD_v2.csv", low_memory=False)

print('filtering for only positions in  high and low grantham dfs')
# CHANGE
# burden = burden[burden['CpDAA'] == 1].copy() # not necessaary because small omimcpd dataframe and merging

print(burden.shape, '\n')
checkColumnValue(burden, 'aaref')
uniqueCount2(burden, 'UKBID')
uniqueCount(burden, 'posID')

print('NOTE: all aa position rows are unique in burden df')
print('-------------------------------------------------------------')
print('merging grantham hi low dfs with aa-level window burden df')
final = pd.merge(burden, gHigh, on=['posID'], how='inner')
print(gHigh.shape)
print(final.shape)
final = pd.merge(final, gLow, on=['posID'], how='inner')
print(gLow.shape)
print(final.shape)
```

```
filtering for only positions in  high and low grantham dfs
(101845, 134)

  aaref  Count
0     K  60312
1     Y  25112
2     C  16421

UKBID set length: 1234

posID length: 101845
posID set length: 101845

NOTE: all aa position rows are unique in burden df
------------------------------------------------------------
---
merging grantham hi low dfs with aa-level window burden df
(850, 13)
(793, 146)
(850, 13)
(793, 158)
```

In [21]:
```
checkColumnValue(final, 'aaref')
uniqueCount2(final, 'UKBID')
uniqueCount(final, 'posID')
```

```
  aaref  Count
0     Y    304
1     C    260
2     K    229

UKBID set length: 389

posID length: 793
posID set length: 793
```

[6] SAVING PATHO only AA level with grantham hi low predictions from 10 scores for

1. OMIM&CpD proteins n = 389 proteins, 793 CKY pathogenic positions

In [22]:
```
dir4grantham = "/Users/mariapalafox/Desktop/BRIDGE/disorder/FIGURES/modules/STEP4/grantham/"
os.chdir(dir4grantham)

# accounting all cpdaa
uniqueCount(final, 'posID')
uniqueCount2(final, 'UKBID')
checkColumnValue(final, 'aaref')
crosstabit(final, 'aaref', 'RP')
crosstabit(final, 'aaref', 'LP')

final.to_csv("CKY_level_PATHO_HighLow_granthamScored_rmd_Background_1D_3Dcounts_included_389_OMIMCpD_793_posIDs_v3.csv", index=False)
```

```
posID length: 793
posID set length: 793

UKBID set length: 389

  aaref  Count
0     Y    304
1     C    260
2     K    229

RP      High  Low  Medium  Total
aaref
C          4   13       7     24
K          1   14       4     19
Y          0   15       0     15
Total      5   42      11     58

LP      Liganded  Unliganded  Total
aaref
C              7          34     41
K              1          22     23
Total          8          56     64
```

In [23]:
```
final.head()
```

Out[23]:

	posID	genename	UKBID	CpD_UKBID	Mendelian_UKBID	pos	aaref
0	A6NHR9_Y283	SMCHD1	A6NHR9	True	True	283	Y
1	O00329_C416	PIK3CD	O00329	True	True	416	C
2	O00429_C367	DNM1L	O00429	True	True	367	C
3	O00429_C431	DNM1L	O00429	True	True	431	C
4	O00429_C446	DNM1L	O00429	True	True	446	C

In [25]:
```
final.columns
```

Out[25]:
```
Index(['posID', 'genename', 'UKBID', 'CpD_UKBID', 'Mendelian
_UKBID', 'pos',
       'aaref', 'refcodon', 'CpDAA', 'LP', 'RP', 'R', 'PATHO
GENIC',
       'COMMONBENIGN', 'VUS', 'RARE', 'RAREST', 'BACKGROUN
D',
       'Interpro_domain', 'MPC_score.mean', 'MPC_score.max',
       'MPC_rankscore.mean', 'MPC_rankscore.max',
       'phyloP100way_vertebrate.mean', 'phyloP100way_vertebr
ate.max',
       'phyloP100way_vertebrate_rankscore.mean',
       'phyloP100way_vertebrate_rankscore.max', 'MAX.RANKSCO
RE.SUM',
       'MEAN.RANKSCORE.SUM', 'MAX.RAWSCORE.SUM', 'MEAN.RAWSC
ORE.SUM',
       'VEST4_rankscore.max', 'REVEL_rankscore.max', 'MutPre
d_rankscore.max',
       'MVP_rankscore.max', 'BayesDel_addAF_rankscore.max',
       'ClinPred_rankscore.max', 'LISTS2_rankscore.max',
       'CADD_raw_rankscore.max', 'binary.VEST4.max', 'binar
y.REVEL.max',
       'binary.MutPred.max', 'binary.MVP.max', 'binary.Bayes
Del.max',
       'binary.ClinPred.max', 'binary.LIST.max', 'binary.CAD
Dp.max',
       'VEST4_score.max', 'REVEL_score.max', 'MutPred_score.
max',
       'MVP_score.max', 'BayesDel_addAF_score.max', 'ClinPre
d_score.max',
       'LISTS2_score.max', 'CADD_phred.max', 'VEST4_rankscor
e.mean',
       'REVEL_rankscore.mean', 'MutPred_rankscore.mean', 'MV
P_rankscore.mean',
```

```
       'BayesDel_addAF_rankscore.mean', 'ClinPred_rankscore.
mean',
       'LISTS2_rankscore.mean', 'CADD_raw_rankscore.mean', '
binary.VEST4.mean',
       'binary.REVEL.mean', 'binary.MutPred.mean', 'binary.M
VP.mean',
       'binary.BayesDel.mean', 'binary.ClinPred.mean', 'bina
ry.LIST.mean',
       'binary.CADDp.mean', 'VEST4_score.mean', 'REVEL_scor
e.mean',
       'MutPred_score.mean', 'MVP_score.mean', 'BayesDel_add
AF_score.mean',
       'ClinPred_score.mean', 'LISTS2_score.mean', 'CADD_phr
ed.mean',
       'UKBID_VUSandPATHO', 'UKBID_BACKGROUNDandPATHO', 'UKB
ID_BENIGNandPATHO',
       'PATHOGENIC.withinRange0.count', 'PATHOGENIC.withinRa
nge3.count',
       'PATHOGENIC.withinRange6.count', 'PATHOGENIC.withinRa
nge15.count',
       'COMMONBENIGN.withinRange0.count', 'COMMONBENIGN.with
inRange3.count',
       'COMMONBENIGN.withinRange6.count', 'COMMONBENIGN.with
inRange15.count',
       'VUS.withinRange0.count', 'VUS.withinRange3.count',
       'VUS.withinRange6.count', 'VUS.withinRange15.count',
       'RARE.withinRange0.count', 'RARE.withinRange3.count',
       'RARE.withinRange6.count', 'RARE.withinRange15.coun
t',
       'RAREST.withinRange0.count', 'RAREST.withinRange3.cou
nt',
       'RAREST.withinRange6.count', 'RAREST.withinRange15.co
unt', 'Length',
       'posID.norm', 'posID.decile', 'PATHOGENIC.within0A.co
unt',
       'PATHOGENIC.within6A.count', 'PATHOGENIC.within8A.cou
nt',
       'PATHOGENIC.within10A.count', 'COMMONBENIGN.within0A.
count',
       'COMMONBENIGN.within6A.count', 'COMMONBENIGN.within8
A.count',
       'COMMONBENIGN.within10A.count', 'VUS.within0A.count',
       'VUS.within6A.count', 'VUS.within8A.count', 'VUS.with
in10A.count',
       'RARE.within0A.count', 'RARE.within6A.count', 'RARE.w
ithin8A.count',
       'RARE.within10A.count', 'RAREST.within0A.count',
       'RAREST.within6A.count', 'RAREST.within8A.count',
       'RAREST.within10A.count', 'BACKGROUND.withinRange0.co
unt',
       'BACKGROUND.withinRange3.count', 'BACKGROUND.withinRa
nge6.count',
```

```
       'BACKGROUND.withinRange15.count', 'BACKGROUND.within0
A.count',
       'BACKGROUND.within6A.count', 'BACKGROUND.within8A.cou
nt',
       'BACKGROUND.within10A.count', 'resolved3D.posID',
       'highestGrantham.missenseType', 'highestGrantham.codo
nType',
       'highestGrantham.VEST4_score', 'highestGrantham.REVEL
_score',
       'highestGrantham.MutPred_score', 'highestGrantham.MVP
_score',
       'highestGrantham.BayesDel_addAF_score',
       'highestGrantham.ClinPred_score', 'highestGrantham.LI
STS2_score',
       'highestGrantham.CADD_raw', 'highestGrantham.MPC_scor
e',
       'highestGrantham.phyloP100way_vertebrate',
       'lowestGrantham.missenseType', 'lowestGrantham.codonT
ype',
       'lowestGrantham.VEST4_score', 'lowestGrantham.REVEL_s
core',
       'lowestGrantham.MutPred_score', 'lowestGrantham.MVP_s
core',
       'lowestGrantham.BayesDel_addAF_score', 'lowestGrantha
m.ClinPred_score',
       'lowestGrantham.LISTS2_score', 'lowestGrantham.CADD_r
aw',
       'lowestGrantham.MPC_score', 'lowestGrantham.phyloP100
way_vertebrate'],
      dtype='object')
```

In [ ]:
```
# add describeprot data to this df for rmd clustering

"cluster1D_window3_mean_RAWSCORE_droppedOverlapBtw_Patho_Beni
gn_CpDAA_CKYfiltered_Mendelian.csv"
cluster1D_window3_MeanRAWSCOREsum_Pathogenic29324_Mendelian.c
sv
```