In [1]:
```
import sys, os
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_rows', 1000)
pd.set_option('display.max_colwidth', 2000) # you can't use n
one here
pd.options.display.max_seq_items = 2000 # seq in column
pd.options.display.float_format = '{:,.4f}'.format

from IPython.display import Image
from IPython.core.display import HTML
def mediumpic(name):
    display(Image(filename = name, width= 500, height=650))
```

Checking La Cognata NCL gene list against OMIM universe data from thesis project

===

PPT1
TPP1
CLN3
DNAJC5
CLN5
CLN6
MFSD8
CLN8
CTSD
GRN
ATP13A2
CTSF
KCTD7

source:

In [6]:
```
omim = pd.read_csv("CarrierCalc/data/REF_merged_dbNSFPgenelev
el_Pei_macarthur_codon_AA_abun_16812proteinsgenes.csv")
print(omim.columns)
print(omim.shape)
```

```
Index(['UKBID.HGNC', 'UKBID', 'HGNCsymbol', 'HGNC.ID',
       'HGNC.approved.genename', 'ENSGv92', 'UKBIDmapsToMult
ipleGeneSymbols',
       'gene.primary.uniprot', 'gene.synonyms.uniprot',
       'protein.names.uniprot', 'CpD.protein', 'CpDC.count',
'CpDK.count',
       'CpDY.count', 'CpDC.protein', 'CpDK.protein', 'CpDY.p
rotein',
       'CpDCKY.protein', 'FDAtarget.HPA2021', 'ClinVar2021.a
nyPATHO',
       'ClinVar2021miss.PATHO', 'ClinVar2021miss.VUS', 'BENI
GN', 'PATHO',
       'VUS', 'VUS.and.Detected', 'PATHO.and.Detected', 'Men
delian2021',
       'Mendelian.and.Detected', 'MIM Number', 'phenotypeCou
nt',
       'phenotypesParsed', 'phenoKeysParsed', 'MimNumberPars
ed',
       'inheritanceParsed', 'inheritanceParsedSet',
       'Homo.LoF.tolerant.Lek2016', 'Essential.CRISPR.Hart20
17',
       'NonEssential.CRISPR.Hart2017', 'obs.mis', 'exp.mis',
'oe.mis', 'MOEUF',
       'SOEUF', 'obs.lof', 'exp.lof', 'oe.lof', 'LOEUF', 'pL
I',
       'HaploinsuffLv3.ClinGen2021', 'Essential.mouse.Blake2
011',
       'GWAS.peak.MacArthur2017', 'Olfactory.Mainland2015',
       'Kinase.Uniprot2018', 'GPCR.union.Uniprot2018',
       'GPI.anchored.Uniprot2017', 'DRG.union.WoodKang', 'ch
r', 'CCDS.id',
       'Refseq.id', 'ucsc.id', 'Function.description',
       'Tissue.specificity.Uniprot', 'Expression.egenetics',
       'TissueExpression.GNF.Atlas', 'Interactions.IntAct',
       'Interactions.BioGRID', 'Interactions.ConsensusPathD
B',
       'Pathway.ConsensusPathDB', 'GO.biological.process',
       'GO.molecular.function', 'GO.cellular.component',
       'Orphanet.disorder.id', 'Orphanet.disorder',
       'Orphanet.association.type', 'Trait.association.GWA
S', 'HPO.id',
       'HPO.name', 'Known.rec.info', 'RVIS.EVS', 'RVIS.perce
ntile.EVS',
       'gnomAD.pRec', 'gnomAD.pNull', 'LoFtool.score', 'P.H
I', 'HIPred.score',
       'HIPred', 'GHIS', 'GDI', 'GDIPhred',
```

```
       'SORVA.LOF.MAF0.005.HomOrCompoundHet', 'SORVA.LOF.MAF
0.001.HetOrHom',
       'SORVA.LOF.MAF0.001.HomOrCompoundHet',
       'SORVA.LOForMissense.MAF0.005.HetOrHom',
       'SORVA.LOForMissense.MAF0.005.HomOrCompoundHet',
       'SORVA.LOForMissense.MAF0.001.HetOrHom',
       'SORVA.LOForMissense.MAF0.001.HomOrCompoundHet', 'Ess
ential.gene',
       'Essential.gene.CRISPR', 'Essential.gene.CRISPR2',
       'Essential.gene.genetrap', 'Gene.indispensability.sco
re',
       'Gene.indispensability.pred', 'MGI.mouse.gene', 'MGI.
mouse.phenotype',
       'No.function.descript', 'interaction.count', 'autis
m', 'DDD', 'cosmic',
       'PathVar', 'disgenet.path', 'virus.interacting', 'met
abolic.enzymes',
       'ribosomal.protein.mitochondrial', 'ribosomal.protei
n.cytoplasmic',
       'mitochondrial', 'LOEUF.decile.percent', 'MOEUF.decil
e.percent',
       'SOEUF.decile.percent', 'LOEUF.0decile', 'MOEUF.0deci
le',
       'SOEUF.0decile', 'LOEUF.lessthan0.35', 'MOEUF.lesstha
n0.35',
       'SOEUF.lessthan0.35', 'TissueCount.GNF.Atlas', 'Tissu
eCount.level',
       'interaction.count.levels', 'Length', 'CodonNumber',
'GCpercent',
       'GC1percent', 'GC2percent', 'GC3percent', 'A', 'C', '
D', 'E', 'F', 'G',
       'H', 'I', 'K', 'L', 'M', 'N', 'P', 'Q', 'R', 'S', '
T', 'V', 'W', 'Y',
       'F.TTT', 'F.TTC', 'L.TTA', 'L.TTG', 'L.CTT', 'L.CTC',
'L.CTA', 'L.CTG',
       'I.ATT', 'I.ATC', 'I.ATA', 'M.ATG', 'V.GTT', 'V.GTC',
'V.GTA', 'V.GTG',
       'Y.TAT', 'Y.TAC', 'STOP.TAA', 'STOP.TAG', 'H.CAT', '
H.CAC', 'Q.CAA',
       'Q.CAG', 'N.AAT', 'N.AAC', 'K.AAA', 'K.AAG', 'D.GAT',
'D.GAC', 'E.GAA',
       'E.GAG', 'S.TCT', 'S.TCC', 'S.TCA', 'S.TCG', 'P.CCT',
'P.CCC', 'P.CCA',
       'P.CCG', 'T.ACT', 'T.ACC', 'T.ACA', 'T.ACG', 'A.GCT',
'A.GCC', 'A.GCA',
       'A.GCG', 'C.TGT', 'C.TGC', 'STOP.TGA', 'W.TGG', 'R.CG
T', 'R.CGC',
       'R.CGA', 'R.CGG', 'S.AGT', 'S.AGC', 'R.AGA', 'R.AGG',
'G.GGT', 'G.GGC',
       'G.GGA', 'G.GGG', 'CRISPR.essential.3studies'],
      dtype='object')
(16812, 220)
```

In [23]:
```
stable = pd.read_csv("CarrierCalc/data/Homo_sapiens.GRCh38.92.uniprot.tsv", delimiter="\t")
stable = stable[['gene_stable_id', 'transcript_stable_id', 'protein_stable_id', 'xref']].copy()
stable.columns = ['ENSGv92', 'ENSTv92', 'ENSPv92', 'UKBID']
uniqueCount(stable, 'ENSGv92')
uniqueCount(stable, 'UKBID')
uniqueCount(stable, 'ENSTv92')
#stable[stable['UKBID'].str.contains(';')]
# seems like all rows have only 1 ukbid, checked for , : ; .
stable['UKBID.ENSG'] = stable['UKBID'] + '_' + stable['ENSGv92']
stable.to_csv("CarrierCalc/data/Homo_sapiens.GRCh38.92.stable_ids_and_ukbidxref.csv", index=False)
```

```
ENSGv92 length: 113729
ENSGv92 set length: 22878

UKBID length: 113729
UKBID set length: 75494

ENSTv92 length: 113729
ENSTv92 set length: 104019
```

In [24]:
```
stable.head()
```

Out[24]:

	ENSGv92	ENSTv92	ENSPv92	UKBID	
0	ENSG00000186092	ENST00000641515	ENSP00000493376	Q8NH21	Q8NH
1	ENSG00000186092	ENST00000335137	ENSP00000334393	Q8NH21	Q8NH
2	ENSG00000284733	ENST00000426406	ENSP00000409316	Q6IEY1	Q6IE
3	ENSG00000284733	ENST00000426406	ENSP00000409316	A0A126GV92	A0A126GV
4	ENSG00000284662	ENST00000332831	ENSP00000329982	Q6IEY1	Q6IE

In [18]:
```
#omim['UKBID.ENSG'] = omim['UKBID'] + '_' + omim['ENSGv92']

#print(omim.shape)
#omim2 = pd.merge(stable, omim, how='inner', on=['UKBID.ENSG'])
#print(omim2.shape)
```

```
(16812, 221)
(38375, 225)
```

In [25]:
```
omim.head(2)
```

Out[25]:

	UKBID.HGNC	UKBID	HGNCsymbol	HGNC.ID	HGNC.approved.genenam
0	Q5T2S8_ARMC4	Q5T2S8	ARMC4	HGNC:25583	armadillo repeat containing

1	Q9H3J6_C12orf65	Q9H3J6		C12orf65	HGNC:26784	chromosome 12 ope reading frame 6

In [4]:
```
ncl_list = ["PPT1",
        "TPP1",
        "CLN3",
        "DNAJC5",
        "CLN5",
        "CLN6",
        "MFSD8",
        "CLN8",
        "CTSD",
        "GRN",
        "ATP13A2",
        "CTSF",
        "KCTD7"]
ncl = addcolumnconditionalDropFalse(ncl_list, omim, 'HGNCsymbol')

print(ncl.shape)
ncl.to_csv("NCL_13genes_from_universe_merge.csv", index=False)
```

```
dropping rows False for containing value in mapList...
filtered out False rows df shape:  (13, 220)
(13, 220)
```

In [7]:
```
# from official list website

lysosomalStorageOfficialList = ["GM2A",
"MAN2B1",
"MAN2B1",
"MAN2B1",
"MANBA",
"AGA",
"LIPA",
"LIPA",
"CTNS",
"CTNS",
"CTNS",
"CGI58",
"PNPLA2",
"LAMP2",
"GLA",
"GLA",
"ASAH1",
```

```
"FUCA1",
"CTSA",
"GBA",
"GBA",
"GBA",
"GBA",
"PSAP",
"GLB1",
"GLB1",
"GLB1",
"GALC",
"GALC",
"GALC",
"GALC",
"PSAP",
"ARSA",
"ARSA",
"ARSA",
"ARSA",
"ARSA",
"PSAP",
"IDUA",
"IDUA",
"IDUA",
"IDS",
"IDS",
"SGSH",
"NAGLU",
"HGSNAT",
"GNS",
"GALNS",
"GLB1",
"HYAL1",
"ARSB",
"GUSB",
"NEU1",
"NEU1",
"GNPTAB",
"GNPTAB",
"GNPTG",
"MCOLN1",
"SUMF1",
"SMPD1",
"SMPD1",
"NPC1",
"NPC2",
"NPC1",
"CLN6",
"CLN3",
"CLN5",
"TPP1",
"CLN6",
```

```
"CLN6",
"CLN8",
"PPT1",
"GAA",
"GAA",
"CTSK",
"HEXB",
"NAGA",
"NAGA",
"NAGA",
"SLC17A5",
"SLC17A5",
"ASAH1",
"HEXA",
"HEXA",
"HEXA",
"SLC9A6",
"OCRL",
"FIG4",
"FIG4",
"FIG4",
"CLCN5",
"OCRL"]
lysosomalStorageOfficialList = list(set(lysosomalStorageOffic
ialList))
len(lysosomalStorageOfficialList)
overlapqc = lysosomalStorageOfficialList and ncl_list
print(len(overlapqc), overlapqc)
```

```
13 ['PPT1', 'TPP1', 'CLN3', 'DNAJC5', 'CLN5', 'CLN6', 'MFSD
8', 'CLN8', 'CTSD', 'GRN', 'ATP13A2', 'CTSF', 'KCTD7']
```

Table of LIysosomal Storage Disorders

gcr from Zhu 2022 study, Data table S6

Official List of Lysosomal Diseases with added cateogory column info from La Cognata 2020, google

In [69]:
```
lysosomalCat = pd.read_csv("Official list Lysosomal Storage D
isease formatted names added category.csv")
print(lysosomalCat.shape)

gcr = pd.read_csv("Zhu2022_GCR_S6.csv")
print(gcr.shape)
```

```
(84, 4)
(2675, 9)
```

In [70]:
```
print(lysosomalCat.columns)
uniqueCount(lysosomalCat, 'Gene')
checkColumnValue(lysosomalCat, 'Category')
```

```
Index(['Gene', 'Category', 'Disease', 'Subtype'], dtype='obj
ect')
Gene length: 84
Gene set length: 47
```

	Category	Count
0	Sphingolipidoses	30
1	Mucopolysaccharidoses	14
2	Glycoproteinoses	12
3	Integral membrane protein disorders	10
4	Neuronal ceroid lipofuscinoses	8
5	Lipid storage diseases	4
6	Post-translational modification defects	4
7	Glycogen storage disease	2

In [71]:
```
print(gcr.columns)
gcr = renameit(gcr, 'symbol', 'Gene')
uniqueCount(gcr, 'Gene')
```

```
Index(['symbol', 'ALL', 'AFR', 'NFE', 'ASJ', 'EAS', 'FIN', '
SAS', 'AMR'], dtype='object')
Gene length: 2675
Gene set length: 2675
```

In [72]:
```
lsd =  pd.merge(lysosomalCat, gcr, how="inner", on=['Gene'])
uniqueCount(lsd,  'Gene')
describeMe(lsd)

replace_col_value(lsd, 'Subtype', np.nan, '.')

lsd['DiseaseFullName'] = lsd['Disease'] + ";" + lsd['Subtyp
e']
#lsd = lsd[['Gene', 'DiseaseFullName',  'Category',  'ALL', '
AFR', 'NFE',  'ASJ',  'EAS', 'FIN', 'SAS',  'AMR']].copy()
describeMe(lsd)
```

```
Gene length: 78
Gene set length: 44

(78, 12)
Gene          0
Category      0
Disease       0
Subtype      19
ALL           0
AFR           0
NFE           0
ASJ           0
EAS           0
FIN           0
SAS           0
AMR           0
dtype: int64
(78, 13)
Gene                0
Category            0
Disease             0
Subtype             0
ALL                 0
AFR                 0
NFE                 0
ASJ                 0
EAS                 0
FIN                 0
SAS                 0
AMR                 0
DiseaseFullName     0
dtype: int64
```

lost 3 genes from merge with official list and gcr scores from Zhu 2022

Caveat:

- Zhu et al gene carrier rate may refer to different phenotype? I dont think it matters since all pathogenic alleles were counted
- assumes subtypes of disease have same gene carrier rate
- calculations are an over estimate

In [73]:
```
lsd.sample(6)
```

Out[73]:

	Gene	Category	Disease	Subtype	ALL	AFR	NFE	ASJ
39	GLB1	Sphingolipidoses	GM1-gangliosidosis	adult/chronic GM1-gangliosidosis	0.00	0.00	0.00	0.00
18	NPC1	Integral membrane protein disorders	Niemann-Pick disease	type C1 / chronic neuronopathic form	0.00	0.00	0.00	0.00
57	GLA	Sphingolipidoses	Fabry disease	type II late-onset	0.00	0.00	0.00	0.00
69	ARSA	Sphingolipidoses	Metachromatic Leukodystrophy	adult	0.01	0.00	0.01	0.00
5	MANBA	Glycoproteinoses	beta-mannosidosis	.	0.00	0.00	0.00	0.00
52	GNPTAB	Post-translational modification defects	Pseudo-Hurler polydystrophy / mucolipidosis type III	.	0.01	0.01	0.01	0.00

In [74]:
```
uniqueCount(lsd, 'Gene')
uniqueCount(lsd, 'DiseaseFullName')
```

```
Gene length: 78
Gene set length: 44

DiseaseFullName length: 78
DiseaseFullName set length: 78
```

In [75]:
```
lsd.to_csv("GCR_merged_LSDcategories_44genes_78diseaseSubtypes_full.csv", index=False)
```

In [76]:
```
genelsd = lsd[['Gene', 'Category','ALL', 'AFR', 'NFE',  'ASJ',  'EAS', 'FIN', 'SAS',  'AMR']].copy()
```

In [77]:
```
genelsd.drop_duplicates(inplace=True)
```

In [78]:
```
genelsd
```

Out[78]:

	Gene	Category	ALL	AFR	NFE	ASJ	EAS	FIN	SAS	AMR
0	GAA	Glycogen storage disease	0.01	0.01	0.02	0.01	0.01	0.00	0.01	0.01
2	MAN2B1	Glycoproteinoses	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00
5	MANBA	Glycoproteinoses	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	AGA	Glycoproteinoses	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00
7	FUCA1	Glycoproteinoses	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	CTSA	Glycoproteinoses	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	NEU1	Glycoproteinoses	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
11	NAGA	Glycoproteinoses	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
14	CTNS	Integral membrane protein disorders	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
17	MCOLN1	Integral membrane protein disorders	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00
18	NPC1	Integral membrane protein disorders	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01
20	NPC2	Integral membrane protein disorders	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00
21	SLC17A5	Integral membrane protein disorders	0.00	0.00	0.00	0.01	0.00	0.01	0.00	0.00
23	LIPA	Lipid storage diseases	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
25	PNPLA2	Lipid storage diseases	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00
26	IDUA	Mucopolysaccharidoses	0.01	0.00	0.01	0.00	0.00	0.01	0.00	0.00
29	IDS	Mucopolysaccharidoses	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
31	SGSH	Mucopolysaccharidoses	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00
32	NAGLU	Mucopolysaccharidoses	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

33	HGSNAT	Mucopolysaccharidoses	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
34	GNS	Mucopolysaccharidoses	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
35	GALNS	Mucopolysaccharidoses	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
36	GLB1	Mucopolysaccharidoses	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
37	GLB1	Sphingolipidoses	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
40	HYAL1	Mucopolysaccharidoses	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
41	ARSB	Mucopolysaccharidoses	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
42	GUSB	Mucopolysaccharidoses	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
43	CLN6	Neuronal ceroid lipofuscinoses	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
46	CLN3	Neuronal ceroid lipofuscinoses	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
47	CLN5	Neuronal ceroid lipofuscinoses	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
48	TPP1	Neuronal ceroid lipofuscinoses	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
49	CLN8	Neuronal ceroid lipofuscinoses	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
50	PPT1	Neuronal ceroid lipofuscinoses	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00
51	GNPTAB	Post-translational modification defects	0.01	0.01	0.01	0.00	0.01	0.01	0.01	0.01
53	GNPTG	Post-translational modification defects	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
54	SUMF1	Post-translational modification defects	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
55	GM2A	Sphingolipidoses	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
56	GLA	Sphingolipidoses	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
58	ASAH1	Sphingolipidoses	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00
60	PSAP	Sphingolipidoses	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
63	GALC	Sphingolipidoses	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00
67	ARSA	Sphingolipidoses	0.01	0.00	0.01	0.00	0.00	0.00	0.00	0.00
72	SMPD1	Sphingolipidoses	0.00	0.01	0.00	0.01	0.00	0.00	0.00	0.01
74	HEXB	Sphingolipidoses	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00
75	HEXA	Sphingolipidoses	0.00	0.00	0.00	0.03	0.00	0.00	0.00	0.00

In [79]:
```
uniqueCount(genelsd, 'Gene')
uniqueCount(genelsd, 'Category')
showDuplicateRows(genelsd, 'Gene')
```

```
Gene length: 45
Gene set length: 44

Category length: 45
Category set length: 8

shape of duplicate df :  (2, 10)
```

Out[79]:

	Gene	Category	ALL	AFR	NFE	ASJ	EAS	FIN	SAS	AMR
37	GLB1	Sphingolipidoses	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
36	GLB1	Mucopolysaccharidoses	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

In [80]:
```
checkColumnValue(genelsd, 'Category')
```

	Category	Count
0	Mucopolysaccharidoses	11
1	Sphingolipidoses	10
2	Glycoproteinoses	7
3	Neuronal ceroid lipofuscinoses	6
4	Integral membrane protein disorders	5
5	Post-translational modification defects	3
6	Lipid storage diseases	2
7	Glycogen storage disease	1

In [81]:
```
genelsd.to_csv("GCR_merged_LSDcategories_44genes_8categories_
GLB1duplicated_45rows.csv", index=False)
```

Guo et al 2019 methods-

In [84]:
```
mediumpic("Guo_methods_screenshot.png")
```

The *GCR* for a gene *g* can then be estimated as:

$$GCR_g = 1 - \prod_{i=1}^{v} (1 - VCR_i)$$

Here VCR_i is the variant carrier rate for variant *i*, and *v* is the number of variants of interest in gene *g*.

These calculations were performed separately for each ancestry.

Estimation of cumulative carrier rate (CCR)

The *CCR* for a set of genes *s* can be estimated as:

$$CCR = 1 - \prod_{i=1}^{s} (1 - GCR_i)$$

Here GCR_i is the gene carrier rate for gene *i* in a set of *s* genes. These calculations were performed separately for each ancestry.

In [86]:
```
genelsd.Category.unique()
```

Out[86]:
```
array(['Glycogen storage disease', 'Glycoproteinoses',
       'Integral membrane protein disorders', 'Lipid storage
diseases',
       'Mucopolysaccharidoses', 'Sphingolipidoses',
       'Neuronal ceroid lipofuscinoses',
       'Post-translational modification defects'], dtype=obj
ect)
```

In [ ]: