

```

import math
import sys
from pathlib import Path
import itertools
# import kaleido
import pickle
import pandas as pd
import plotly.graph_objects as go
import plotly.express as px
import plotly.express as px
import plotly.io as pio
from matplotlib import pyplot as plt
from matplotlib_venn import venn3, venn3_circles
from matplotlib_venn import venn2, venn2_circles
from venn import venn
from venn import generate_petal_labels, draw_venn, generate_colors
px.defaults.template = 'ggplot2'
px.defaults.template = 'simple_white'
pio.templates.default = 'simple_white'
cdr_colors = ["#74D2F5", "#5495D6", "#3763F5", "#CF81EB", "#E564F5", "#A04AD6",
              'Aqua', 'AquaMarine']

from collections import namedtuple, defaultdict, Counter

from abag_interactions_hydrophobic import *
from interactions_hydrophobic import *

data_dir = Path("/home/pbarletta/labo/22/migue/data")
expdata_dir = Path("/home/pbarletta/labo/22/migue/data/AB-Bind-Database-master")
rtdos_dir = Path("/home/pbarletta/labo/22/migue/rtdos")
hydro_dir = Path("/home/pbarletta/labo/22/migue/rtdos/hydro")
aux_dir = Path("/home/pbarletta/labo/22/migue/aux")
pdfs_dir = Path("/home/pbarletta/labo/22/migue/run/pdfs")
mutpdfs_dir = Path("/home/pbarletta/labo/22/migue/run/mut_pdfs")
bin_dir = Path("/home/pbarletta/labo/22/locuaz/bin")
evo_bin = Path(bin_dir, "evoef2/EvoEF2")

AA_LIST = ["ALA", "ARG", "ASN", "ASP", "CYS", "GLU", "GLN", "GLY", "HIS",
           "ILE", "LEU", "LYS", "MET", "PHE", "PRO", "SER", "THR", "TRP", "TYR", "VAL"]
AA_LIST = ("D", "E", "S", "T", "R", "N", "Q", "H", "K", "A", "G", "I",
           "M", "L", "V", "P", "F", "W", "Y", "C" )

```

```

NEG_AAS = ("D", "E", "S", "T")
POS_AAS = ("R", "N", "Q", "H", "K")
PHO_AAS = ("A", "G", "I", "M", "L", "V")
RIN_AAS = ("P", "F", "W", "Y")
CAT_AAS = (NEG_AAS, POS_AAS, PHO_AAS, RIN_AAS)
N_CAT = len(CAT_AAS)

```

Hydrophobic

```

expdata_df = pd.read_csv(Path(expdata_dir, "AB-Bind_experimental_data.csv"), encoding='latin1')
# Fix weird column name:
expdata_df = expdata_df.rename(columns = {"#PDB": "PDB"})
pdb_list = tuple(sorted(set(expdata_df['PDB'])))

abag_chains = {}
for partners, pdb_id in zip(expdata_df["Partners(A_B)"], expdata_df["PDB"]):
    target, binder = partners.split("_")
    abag_chains[pdb_id] = Chains(antibody=binder, antigen=target)

mut_list = []
with open(Path(mutpdbs_dir, "mut_list.txt"), "r") as f:
    for mute in f:
        mut_list.append(mute.strip())

pdbs_mut = defaultdict(list)
for mut in mut_list:
    this_pdb = mut.split('-')[0]
    pdbs_mut[this_pdb].append(mut)

with open(Path(joinpath(hydro_dir, "pdb_hydrophobic.pkl"), "rb") as file:
    pdb_hydrophobic = pickle.load(file)
with open(Path(joinpath(hydro_dir, "mut_hydrophobic.pkl"), "rb") as file:
    mut_hydrophobic = pickle.load(file)
with open(Path(rtdo_dir, "pdbs_interface_mut.pkl"), 'rb') as file:
    pdbs_interface_mut = pickle.load(file)
with open(Path(rtdo_dir, "pdbs_cluster_mut.pkl"), 'rb') as file:
    pdbs_cluster_mut = pickle.load(file)

```

```

mut_carbons = []
mut_ddgs = []
for pdb_id, mut in pdb_mut.items():
    pdb-Cs = sum(len(cluster) for cluster in pdb_hydrophobic[pdb_id])

    for mut in pdb_mut[pdb_id]:
        # Discard PDBs without mutations on the interface
        if mut in pdb_interface_mut[pdb_id]:
            mut-Cs = sum([ len(cluster) for cluster in mut_hydrophobic[mut] ])
            mut_carbons.append(pdb-Cs - mut-Cs)

            mut_string = mut.split('-')[1].replace('_', ',')
            ddg = expdata_df.query(f"PDB == '{pdb_id}'").query(
                f"Mutation == '{mut_string}'")["ddG(kcal/mol)"].values[0]
            mut_ddgs.append(float(ddg))
    mean_mut_carbons = np.mean(mut_carbons)

```

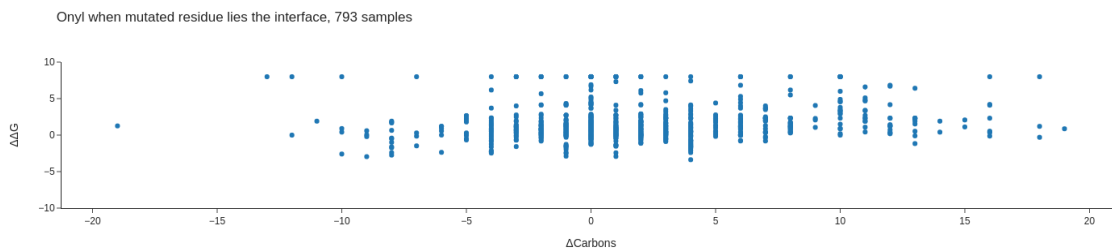
ddG de 8: “1MHP, 1N8Z, 3BN9, 3HFM”

```

figu = px.scatter(x = mut_carbons, y = mut_ddgs,
    labels={'x': 'ΔCarbons', 'y': 'ΔΔG'},
    title = f'Only when mutated residue lies the interface, {len(mut_carbons)} samples')
figu.update_yaxes(range = (-10, 10))

```

Unable to display output for mime type(s): text/html



```

mut_carbons = []
mut_ddgs = []
# Only check PDBs with mutations on the interface
for pdb_id, mut in pdb_mut.items():

```

```

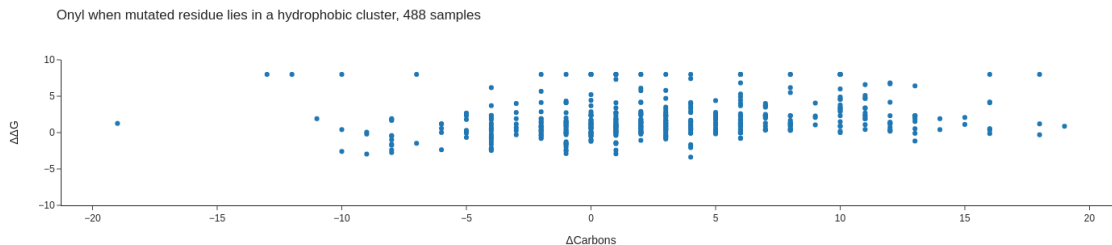
pdb-Cs = sum(len(cluster) for cluster in pdb_hydrophobic[pdb_id])

for mut in pddb_mut[pdb_id]:
    # Discard PDBs without mutations on a cluster
    if mut in pddb_cluster_mut[pdb_id]:
        mut-Cs = sum([ len(cluster) for cluster in mut_hydrophobic[mut] ])
        mut_carbons.append(pdb-Cs - mut-Cs)

        mut_string = mut.split('-')[1].replace('_', ',')
        ddg = expdata_df.query(f"PDB == '{pdb_id}'").query(
            f"Mutation == '{mut_string}'")["ddG(kcal/mol)"].values[0]
        mut_ddgs.append(float(ddg))
    mean_mut_carbons = np.mean(mut_carbons)

figu = px.scatter(x = mut_carbons, y = mut_ddgs,
    labels={'x': 'ΔCarbons', 'y': 'ΔΔG'},
    title = f'Only when mutated residue lies in a hydrophobic cluster, {len(mut_carbons)}'
figu.update_yaxes(range = (-10, 10))

```



```

mut_same_carbons = []
mut_same_ddgs = []
mut_diff_carbons = []
mut_diff_ddgs = []
# Only check PDBs with mutations on a cluster
for pdb_id, lista_mutaciones in pddb_cluster_mut.items():
    pdb-Cs = sum(len(cluster) for cluster in pdb_hydrophobic[pdb_id])

    for mutaciones in lista_mutaciones:
        mut-Cs = sum([ len(cluster) for cluster in mut_hydrophobic[mutaciones] ])

        mut_string = mutaciones.split('-')[1].replace('_', ',')

```

```

ddg = expdata_df.query(f"PDB == '{pdb_id}'").query(
    f"Mutation == '{mut_string}'")["ddG(kcal/mol)"].values[0]

mut_hydro_residues = { (atom.resSeq_str, atom.chain_ID)
    for cluster in mut_hydrophobic[mutaciones] for atom in cluster }

mut_strings = mutaciones.split('-')[1].split('_')
for mut_string in mut_strings:
    chainID, mut_residue = mut_string.split(':')
    resSeq_str = mut_residue[1:-1]

    if (resSeq_str, chainID) not in mut_hydro_residues:
        mut_diff_carbons.append(pdb-Cs - mut-Cs)
        mut_diff_ddgs.append(float(ddg))
        break
    else:
        mut_same_carbons.append(pdb-Cs - mut-Cs)
        mut_same_ddgs.append(float(ddg))
        if (pdb-Cs - mut-Cs) < -12:
            print(mutaciones)

```

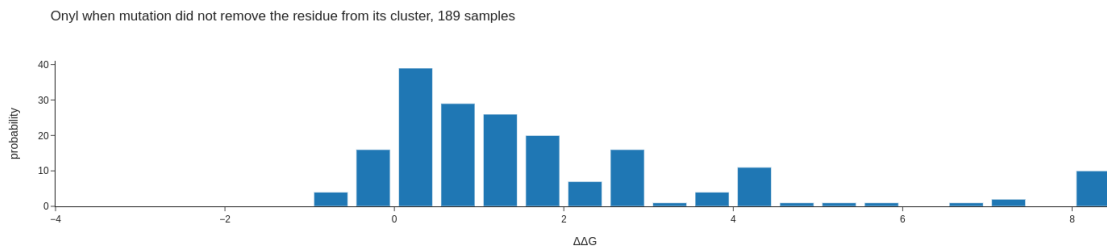
1MHP-H:G53W

1MHP-H:T33I_H:S52T_H:G53Q_H:G54F

```

pbbty, bins = np.histogram(mut_same_ddgs, bins=np.arange(-4., 9., .5))
bins = 0.5 * (bins[:-1] + bins[1:])
px.bar(x=bins, y=pbbty,
    labels={'x': 'ΔΔG', 'y': 'probability'},
    title = f'Only when mutation did not remove the residue from its cluster, {len(mut_sam

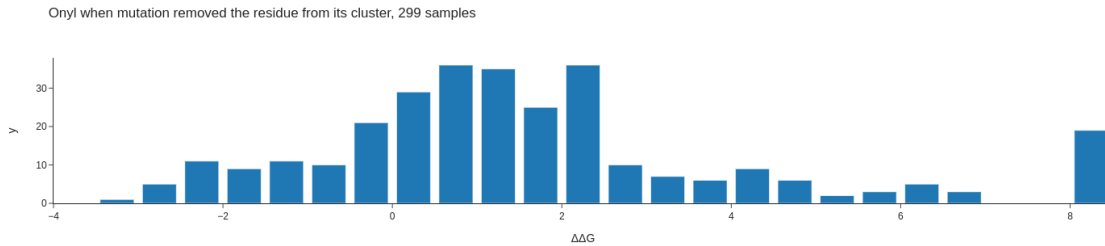
```



```

pbbty, bins = np.histogram(mut_diff_ddgs, bins=np.arange(-4., 9., .5))
bins = 0.5 * (bins[:-1] + bins[1:])
px.bar(x=bins, y=pbbty,
      labels={'x': ' $\Delta G$ '},
      title = f'Only when mutation removed the residue from its cluster, {len(mut_diff_ddgs)}

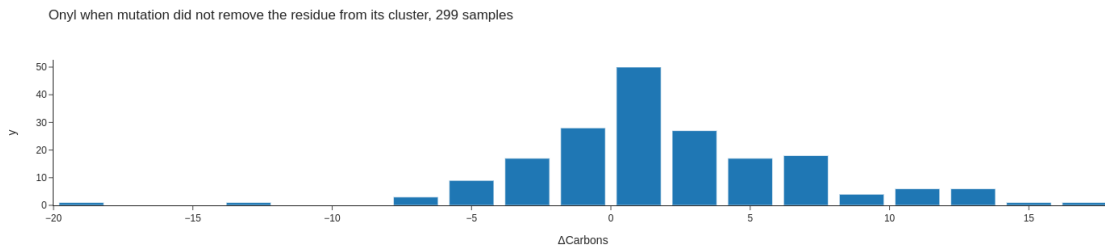
```



```

pbbty, bins = np.histogram(mut_same_carbons, bins=np.arange(-20., 20., 2))
bins = 0.5 * (bins[:-1] + bins[1:])
px.bar(x=bins, y=pbbty,
      labels={'x': ' $\Delta$ Carbons'},
      title = f'Only when mutation did not remove the residue from its cluster, {len(mut_diff_ddgs)}

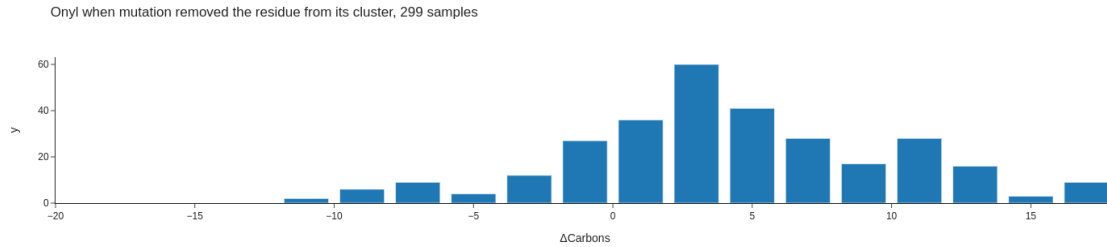
```



```

pbbty, bins = np.histogram(mut_diff_carbons, bins=np.arange(-20., 20., 2))
bins = 0.5 * (bins[:-1] + bins[1:])
px.bar(x=bins, y=pbbty,
      labels={'x': ' $\Delta$ Carbons'},
      title = f'Only when mutation removed the residue from its cluster, {len(mut_diff_ddgs)}

```



How do affinity improving mutations from the interface, relate to the clusters?

```

mut_siempre_carbons = []
mut_siempre_ddgs = []
mut_nunca_carbons = []
mut_nunca_ddgs = []
mut_gano_carbons = []
mut_gano_ddgs = []
mut_perdio_carbons = []
mut_perdio_ddgs = []
# Only check PDBs with mutations on a cluster
for pdb_id, lista_mutaciones in pdb_mut.items():
    pdb-Cs = sum(len(cluster) for cluster in pdb_hydrophobic[pdb_id])

    pdb_hydro_residues = { (atom.resSeq_str, atom.chain_ID)
                           for cluster in pdb_hydrophobic[pdb_id] for atom in cluster }

    for mutaciones in lista_mutaciones:
        mut-Cs = sum([ len(cluster) for cluster in mut_hydrophobic[mutaciones] ])

        mut_string = mutaciones.split('-')[1].replace('_', ',')
        ddg = expdata_df.query(f"PDB == '{pdb_id}'").query(
            f"Mutation == '{mut_string}'")["ddG(kcal/mol)"].values[0]

        # Only check those mutations that improve binding affinity
        if ddg > 0:
            continue

        mut_hydro_residues = { (atom.resSeq_str, atom.chain_ID)
                               for cluster in mut_hydrophobic[mutaciones] for atom in cluster }

        mut_strings = mutaciones.split('-')[1].split('_')

```

```

estaba_en_cluster = False
esta_en_cluster = False
for mut_string in mut_strings:
    chainID, mut_residue = mut_string.split(':')
    resSeq_str = mut_residue[1:-1]

    tmp_estaba_en_cluster = (resSeq_str, chainID) in pdb_hydro_residues
    tmp_esta_en_cluster = (resSeq_str, chainID) in mut_hydro_residues
    estaba_en_cluster |= tmp_estaba_en_cluster
    esta_en_cluster |= tmp_esta_en_cluster

if estaba_en_cluster:
    if esta_en_cluster:
        mut_siempre_carbons.append(pdb-Cs - mut-Cs)
        mut_siempre_ddgs.append(float(ddg))
    else:
        mut_perdio_carbons.append(pdb-Cs - mut-Cs)
        mut_perdio_ddgs.append(float(ddg))
else:
    if esta_en_cluster:
        mut_gano_carbons.append(pdb-Cs - mut-Cs)
        mut_gano_ddgs.append(float(ddg))
    else:
        mut_nunca_carbons.append(pdb-Cs - mut-Cs)
        mut_nunca_ddgs.append(float(ddg))

muts_and_clusters = {
    'kept': len(mut_siempre_ddgs),
    'never_been': len(mut_nunca_ddgs),
    'added': len(mut_gano_ddgs),
    'removed': len(mut_perdio_ddgs)
}

figu_muts_and_clusters = go.Figure(data = go.Pie(labels = list(muts_and_clusters.keys()),
    values = list(muts_and_clusters.values()),
    hole=0.6, pull=0.02, sort=True))

figu_muts_and_clusters.update_traces(marker={'colors': ['BurlyWood', 'Brown', 'DarkMagenta',
    'DarkGoldenRod']})

```



```
figu_muts_and_clusters.update_layout(annotations=[dict(
    text='', x=0.5, y=0.5, font_size=16, showarrow=False)],
    width=500, height=500, font_size=16)
```



Which mutations are these?

```
def get_key(old_AA, new_AA):
    if old_AA in PHO_AAS:
        if new_AA in NEG_AAS or new_AA in POS_AAS:
            return 'PHO_POL'
        elif new_AA in RIN_AAS:
            return 'PHO_RIN'
        else:
            return 'PHO_PHO'
    elif old_AA in NEG_AAS or old_AA in POS_AAS:
        if new_AA in PHO_AAS:
            return 'POL_PHO'
        elif new_AA in RIN_AAS:
            return 'POL_RIN'
        else:
            return 'POL_POL'
    elif old_AA in RIN_AAS:
        if new_AA in NEG_AAS or new_AA in POS_AAS:
            return 'RIN_POL'
        elif new_AA in PHO_AAS:
            return 'RIN_PHO'
        else:
            return 'RIN_RIN'
```

```

raise RuntimeError("REEEEE")

muts_never = {'PHO_POL': 0, 'PHO_RIN': 0, 'POL_PHO': 0, 'POL_RIN': 0, 'RIN_PHO': 0, 'RIN_PO
    'PHO_PHO': 0, 'RIN_RIN': 0, 'POL_POL': 0}
muts_kept = {'PHO_POL': 0, 'PHO_RIN': 0, 'POL_PHO': 0, 'POL_RIN': 0, 'RIN_PHO': 0, 'RIN_PO
    'PHO_PHO': 0, 'RIN_RIN': 0, 'POL_POL': 0}
muts_added = {'PHO_POL': 0, 'PHO_RIN': 0, 'POL_PHO': 0, 'POL_RIN': 0, 'RIN_PHO': 0, 'RIN_PO
    'PHO_PHO': 0, 'RIN_RIN': 0, 'POL_POL': 0}
muts_removed = {'PHO_POL': 0, 'PHO_RIN': 0, 'POL_PHO': 0, 'POL_RIN': 0, 'RIN_PHO': 0, 'RIN_
    'PHO_PHO': 0, 'RIN_RIN': 0, 'POL_POL': 0}
for pdb_id, lista_mutaciones in pdbbs_mut.items():
    pdb-Cs = sum(len(cluster) for cluster in pdb_hydrophobic[pdb_id])

    pdb_hydro_residues = { (atom.resSeq_str, atom.chain_ID)
        for cluster in pdb_hydrophobic[pdb_id] for atom in cluster }

for mutaciones in lista_mutaciones:
    mut-Cs = sum([ len(cluster) for cluster in mut_hydrophobic[mutaciones] ])

    mut_string = mutaciones.split('-')[1].replace('_', ',')
    ddg = expdata_df.query(f"PDB == '{pdb_id}'").query(
        f"Mutation == '{mut_string}'")["ddG(kcal/mol)"].values[0]

    # Only check those mutations that improve binding affinity
    if ddg > 0:
        continue

    mut_hydro_residues = { (atom.resSeq_str, atom.chain_ID)
        for cluster in mut_hydrophobic[mutaciones] for atom in cluster }

    mut_strings = mutaciones.split('-')[1].split('_')
    estaba_en_cluster = False
    esta_en_cluster = False
    for mut_string in mut_strings:
        chainID, mut_residue = mut_string.split(':')
        resSeq_str = mut_residue[1:-1]
        old_AA = mut_residue[0]
        new_AA = mut_residue[-1]

        tmp_estaba_en_cluster = (resSeq_str, chainID) in pdb_hydro_residues
        tmp_esta_en_cluster = (resSeq_str, chainID) in mut_hydro_residues

```

```

        estaba_en_cluster |= tmp_estaba_en_cluster
        esta_en_cluster |= tmp_esta_en_cluster

    if estaba_en_cluster:
        if esta_en_cluster:
            muts_kept[get_key(old_AA, new_AA)] += 1

            mut_siempre_carbons.append(pdb-Cs - mut-Cs)
            mut_siempre_ddgs.append(float(ddg))
        else:
            muts_removed[get_key(old_AA, new_AA)] += 1

            mut_perdio_carbons.append(pdb-Cs - mut-Cs)
            mut_perdio_ddgs.append(float(ddg))
    else:
        if esta_en_cluster:
            muts_added[get_key(old_AA, new_AA)] += 1

            mut_gano_carbons.append(pdb-Cs - mut-Cs)
            mut_gano_ddgs.append(float(ddg))
        else:
            muts_never[get_key(old_AA, new_AA)] += 1

            mut_nunca_carbons.append(pdb-Cs - mut-Cs)
            mut_nunca_ddgs.append(float(ddg))

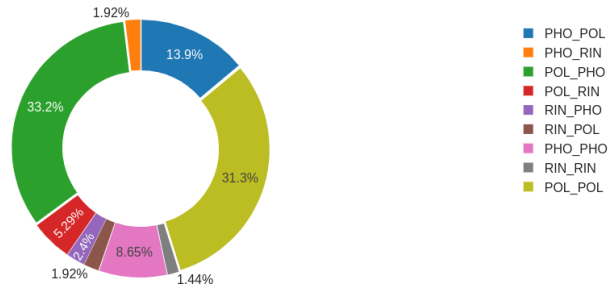
figu_muts_and_clusters = go.Figure(data = go.Pie(labels = list(muts_never.keys()),
    values = list(muts_never.values()),
    hole=0.6, pull=0.02, sort=False))

# figu_muts_and_clusters.update_traces(marker={'colors': ['BurlyWood', 'Brown', 'DarkMagen
#     'DarkGoldenRod', 'Bisque']})

figu_muts_and_clusters.update_layout(annotations=[dict(
    text='', x=0.5, y=0.5, font_size=16, showarrow=False)],
    width=500, height=500, font_size=16,
    title = f"total never been: {sum(muts_never.values())}")

```

total never been: 208

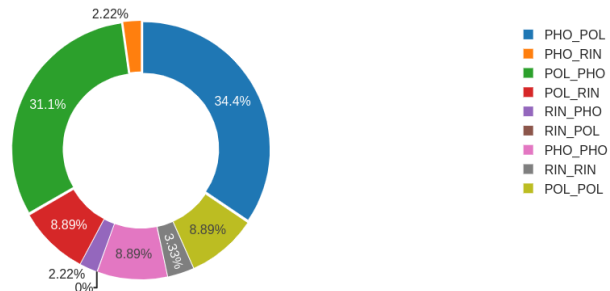


```
figu_muts_and_clusters = go.Figure(data = go.Pie(labels = list(muts_kept.keys()),
    values = list(muts_kept.values()),
    hole=0.6, pull=0.02, sort=False))

# figu_muts_and_clusters.update_traces(marker={'colors': ['BurlyWood', 'Brown', 'DarkMagen',
# 'DarkGoldenRod', 'Bisque']})

figu_muts_and_clusters.update_layout(annotations=[dict(
    text='', x=0.5, y=0.5, font_size=16, showarrow=False)],
    width=500, height=500, font_size=16,
    title = f"total kept: {sum(muts_kept.values())}")
```

total kept: 90

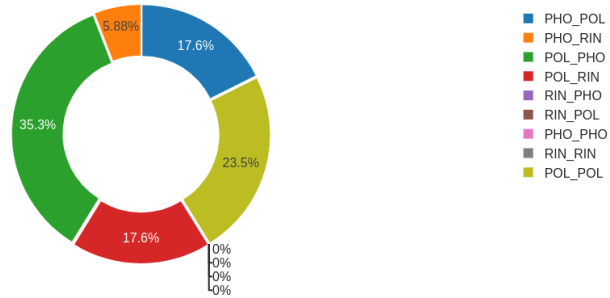


```
figu_muts_and_clusters = go.Figure(data = go.Pie(labels = list(muts_added.keys()),
    values = list(muts_added.values()),
    hole=0.6, pull=0.02, sort=False))
```

```
# figu_muts_and_clusters.update_traces(marker={'colors': ['BurlyWood', 'Brown', 'DarkMagen',
#               'DarkGoldenRod', 'Bisque']})
```

```
figu_muts_and_clusters.update_layout(annotations=[dict(
    text='', x=0.5, y=0.5, font_size=16, showarrow=False)],
    width=500, height=500, font_size=16, title = f"total added: {sum(muts_added.values())}")
```

total added: 17



```
figu_muts_and_clusters = go.Figure(data = go.Pie(labels = list(muts_removed.keys()),
    values = list(muts_removed.values()),
    hole=0.6, pull=0.02, sort=False))
```

```
# figu_muts_and_clusters.update_traces(marker={'colors': ['BurlyWood', 'Brown', 'DarkMagen',
#               'DarkGoldenRod', 'Bisque']})
```

```
figu_muts_and_clusters.update_layout(annotations=[dict(
    text='', x=0.5, y=0.5, font_size=16, showarrow=False)],
    width=500, height=500, font_size=16,
    title = f"total removed: {sum(muts_removed.values())}")
```

total removed: 10

