

To gain access to the input data in an easy way you can check the references [5] source which is data from the author of the paper. You can also use the link in the GitHub repo's README file.

LINK TO GITHUB REPO https://github.com/jkroberts1/CS598_DLH-Spring24-Group47.git

LINK TO VIDEO PRESENTATION <https://youtu.be/hkvhK3Jqo6s>

Introduction

There are many previous studies that support the claim that anticancer drug sensitivity in cell lines are not tissue-specific, because of this tissue-specific drugs can be of benefit to other tissues. This project aims to develop a drug synergy prediction model for understudied tissues in an attempt to overcome data scarcity (Kim et al, 2021). With this synergy prediction model we can rank synergistic drug combinations in understudied tissues.

This problem is a data processing problem for diseases. There is simply not enough data available to be processed for cancer in areas that are hard to perform medical analysis on. Therefore what this paper aims to do is, using drugs that work well on a given tissue in the body, predict the effectiveness on the drug on a different yet similar tissue in the body. Medical study already indicates what kinds of tissue are similar to highly studied tissues and in what ways they are similar.

The innovations of the paper is that this will not only overcome the challenge of not having enough data on tissues that get cancer, once successful this will add to the data on those tissues. Therefore this project works to single-handedly solve the issues surrounding data-poor tissues.

The paper reports it's own accuracy as follows: "Our drug response prediction model was able to successfully bypass the posttreatment features while achieving competitive accuracy (Kim et al, 2021)". It's major contribution to the research regime is the transfer of learning between data-rich and data-poor in-silico experiments. Outside of the goals of the model there is much to be said about how this paper sets a precedent for future papers to tackle harder to experiment parts of the body.

Scope of Reproducibility

The hypothesis we will test from the paper in this project is as follows

1. If a model is trained to predict drug effectiveness on well studied tissues, then that model can be transferred to a "student" model trained on understudied tissues while retaining drug effectiveness prediction accuracy.

What this hypothesis is stating is a direct overview of the goals of the paper and project. The project is aiming to build a model that aims to show a direct synergy between well studied and understudied tissues and how effective drugs are on them.

This is feasible because we know from previous studies that cancer medication does not show higher efficacy in targeted cell lines (Jaeger et al., 2015). To improve the performance of training with tissues that don't have as much data we can use transfer learning where we build a teacher model that utilizes the well suited data and then transfers those output parameters as inputs to a model with less data.

We aim to reproduce the results of the paper and then also perform ablations on the model to test it's resiliency. We will be using the original code in order to replicate the results of the project and then test out the ablations and any improvements.

ENVIRONMENT

Python version: 3.10

Packages needed:

1. numpy
2. pickle
3. utilities
4. matplotlib
5. mpl_toolkits
6. rdkit
7. re
8. seaborn
9. math
10. pandas
11. torch

- 12. sklearn
- 13. time
- 14. logging
- 15. pdb

Methodology

Data Overview

To access the data, one can reference the link to the google drive in the References section [5]. The data for running this notebook can be downloaded from there. However, if one wants to gain access to the original data referenced in the paper then can follow the links in each section of Data Preprocessing.

Data Preprocessing (1)

The following section(s) will go over the data that we use and load into the model. It will cover the datasets we use, and what we gain from them.

First we start with the DrugComb data which can be accessed here: <https://drugcomb.fimm.fi/>

This dataset has records on 4150 drugs and 112 cell lines. We show a description of the data set with the `.head()` command.

Refer to comments in the code to see how the data is manipulated.

```
In [2]: %pip install rdkit
import pandas as pd
import numpy as np
import pickle
from utilities import Mapping

import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from matplotlib import cm

import rdkit
from rdkit import Chem
from rdkit.Chem import MACCSkeys

import re

import seaborn as sns
from matplotlib.colors import LogNorm
import math

data_path=''
external_data_path=''

summary=pd.read_csv(data_path+'summary_v_1_5.csv')
#Block to StudyID mapping
source=pd.read_csv(data_path+'source.csv')

# Create a dictionary mapping study names to study IDs
study_id_map = dict(zip(source['sname'], source['id']))

summary['study_name'] = summary['study_name'].str.strip()
summary['study_id'] = summary['study_name'].map(study_id_map)

summary_study = summary.sort_values(by='block_id')

cell_line=pd.read_csv(data_path+'cell_line.csv')
drug=pd.read_csv(data_path+'drug.csv')

summary_study = summary_study[summary_study['cell_line_name'].isin(cell_line['name'])]
summary_study = summary_study[summary_study['drug_row'].isin(drug['dname'])]
summary_study = summary_study[summary_study['drug_col'].isin(drug['dname'])]

study_ids=[ '', 'ONEIL', 'CLOUD', 'ALMANAC', 'FORCINA', 'NCATS_ATL', 'Mathews', 'NCATS_DIPG', 'NCATS_ES(FAKi/AURKi)', 'NCATS_ES(Nam
summary_study = summary_study[summary_study['study_name'].isin(study_ids)]

summary_study.head()
```

```

Collecting rdkit
  Downloading rdkit-2023.9.6-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (34.9 MB)
    34.9/34.9 MB 34.7 MB/s eta 0:00:00
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from rdkit) (1.25.2)
Requirement already satisfied: Pillow in /usr/local/lib/python3.10/dist-packages (from rdkit) (9.4.0)
Installing collected packages: rdkit
Successfully installed rdkit-2023.9.6
<ipython-input-2-4d7085a0b5bb>:25: DtypeWarning: Columns (2,7,19,22,23) have mixed types. Specify dtype option on import or set low_memory=False.
  summary=pd.read_csv(data_path+'summary_v_1_5.csv')

```

Out[2]:

	block_id	drug_row	drug_col	cell_line_name	study_name	tissue_name	conc_row_unit	conc_col_unit	ic50_row	ic50_col	.
1012548	1	5-Fluorouracil	Veliparib	A2058	ONEIL	skin	uM	uM	5.126836	3.267734	.
1012549	2	5-Fluorouracil	Veliparib	A2058	ONEIL	skin	uM	uM	5.126836	3.267734	.
1012550	3	5-Fluorouracil	Veliparib	A2058	ONEIL	skin	uM	uM	5.126836	3.267734	.
1012551	4	5-Fluorouracil	Veliparib	A2058	ONEIL	skin	uM	uM	5.126836	3.267734	.
1012552	5	5-Fluorouracil	MK-1775	A2058	ONEIL	skin	uM	uM	5.126836	0.266027	.

5 rows × 27 columns

In [3]:

```
summary_study.head()
```

Out[3]:

	block_id	drug_row	drug_col	cell_line_name	study_name	tissue_name	conc_row_unit	conc_col_unit	ic50_row	ic50_col	.
1012548	1	5-Fluorouracil	Veliparib	A2058	ONEIL	skin	uM	uM	5.126836	3.267734	.
1012549	2	5-Fluorouracil	Veliparib	A2058	ONEIL	skin	uM	uM	5.126836	3.267734	.
1012550	3	5-Fluorouracil	Veliparib	A2058	ONEIL	skin	uM	uM	5.126836	3.267734	.
1012551	4	5-Fluorouracil	Veliparib	A2058	ONEIL	skin	uM	uM	5.126836	3.267734	.
1012552	5	5-Fluorouracil	MK-1775	A2058	ONEIL	skin	uM	uM	5.126836	0.266027	.

5 rows × 27 columns

In [4]:

```
# take average of replicates experiments
#summary_study = summary_study.dropna()

columns_to_convert = ['ri_row', 'ri_col', 'synergy_loewe']

# Convert columns to numeric, coercing errors to NaN
for column in columns_to_convert:
    summary_study[column] = pd.to_numeric(summary_study[column], errors='coerce')

# Now perform the aggregation
summary_mean = summary_study.groupby(['drug_row', 'drug_col', 'cell_line_name', 'study_id'], as_index=False).agg({
    'ri_row': 'mean',
    'ri_col': 'mean'})
```

```

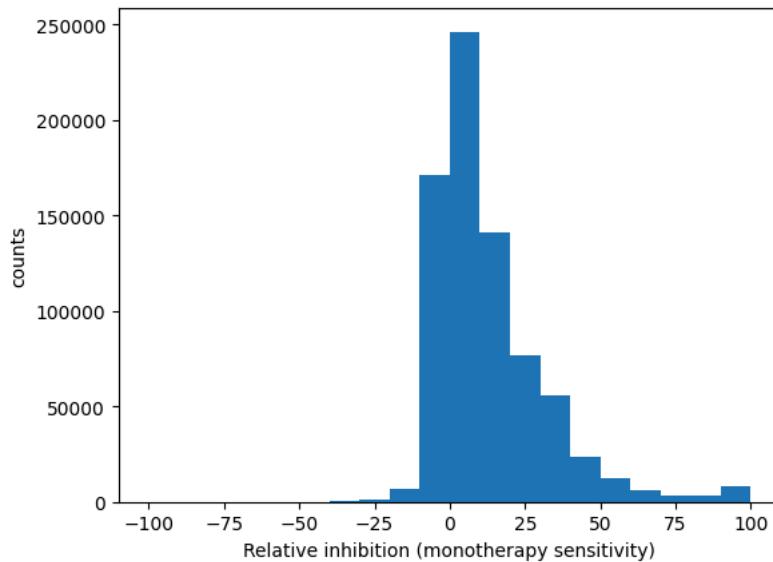
        'synergy_loewe': 'mean'
    })

sensitivity=pd.concat([summary_mean.loc[:, ['drug_row', 'cell_line_name', 'ri_row']].rename(columns={'drug_row':'drug', 'ri_row':'ri'}), summary_mean.loc[:, ['drug_col', 'cell_line_name', 'ri_col']].rename(columns={'drug_col':'drug', 'ri_col':'ri'})])

plt.hist(sensitivity['ri'], bins=20, range=[-100,100])
plt.xlabel("Relative inhibition (monotherapy sensitivity)")
plt.ylabel("counts")
#plt.savefig(data_path+'ri_cnt.png', bbox_inches='tight', dpi=200)

```

Out[4]: Text(0, 0.5, 'counts')

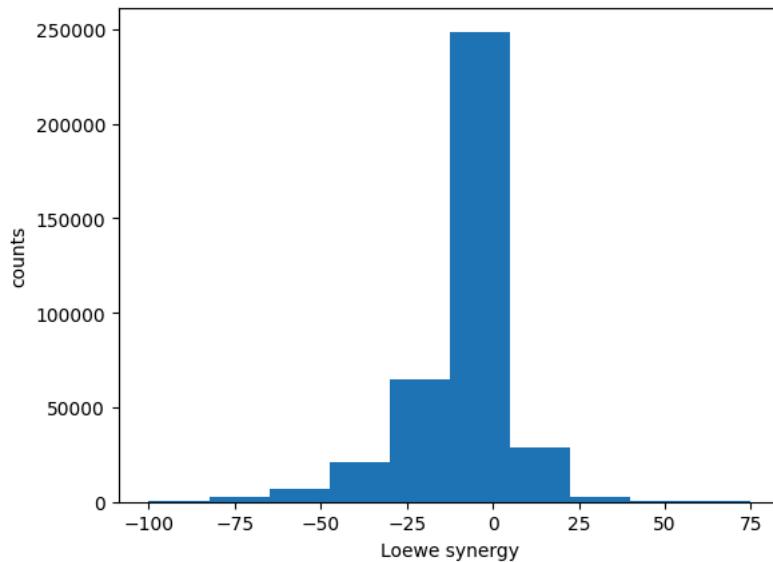


```

In [5]: plt.hist((summary_mean.loc[:, 'synergy_loewe'].values.reshape(-1)), range=[-100, 75])
plt.xlabel("Loewe synergy")
plt.ylabel("counts")
#plt.savefig(data_path+'syn_cnt.png', bbox_inches='tight', dpi=200)

```

Out[5]: Text(0, 0.5, 'counts')



In [6]:

```

X=summary_mean.loc[:, 'ri_row'].to_numpy()
Y=summary_mean.loc[:, 'ri_col'].to_numpy()
Z=summary_mean.loc[:, 'synergy_loewe'].to_numpy()

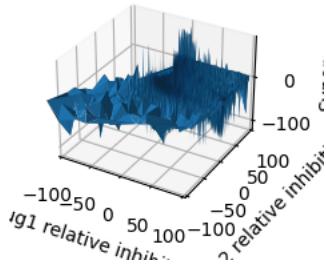
```

```
Z = np.nan_to_num(Z)

fig = plt.figure()
ax = fig.add_subplot(2, 1, 2, projection='3d')
ax.plot_trisurf(X,Y,Z)

ax.set_xlabel('Drug1 relative inhibition')
ax.set_ylabel('Drug2 relative inhibition')
ax.set_zlabel('Synergy')
ax.set_xlim(-100, 100)
ax.set_ylim(-100, 100)
#fig.savefig(data_path+'ri_synergy_3d.png', dpi=200)
```

Out[6]: (-100.0, 100.0)



In [7]:

```
codes={'drugs': Mapping(set(summary_mean['drug_row'].unique()).union(set(summary_mean['drug_col'].unique()))),
       'cell':Mapping(summary_mean['cell_line_name'].unique())}
```

In [8]:

```
summary_mean['drug_col']=summary_mean['drug_col'].apply(lambda x: codes['drugs'].item2idx[x])
summary_mean['drug_row']=summary_mean['drug_row'].apply(lambda x: codes['drugs'].item2idx[x])
summary_mean['cell_line_name']=summary_mean['cell_line_name'].apply(lambda x: codes['cell'].item2idx[x])
```

In [9]:

```
summary_mean.head()
```

Out[9]:

	drug_row	drug_col	cell_line_name	study_id	ri_row	ri_col	synergy_loewe	
0	2617	3003		0	4	-21.079400	17.392589	4.436431
1	1515	3003		0	4	-4.051616	17.392589	10.755529
2	904	1413		1	11	-71.949000	9.755000	-6.188270
3	904	3003		0	4	-9.231525	17.392589	0.739056
4	63	1413		1	11	-12.272000	15.133000	-18.680123

In [10]:

```
summary_mean.to_pickle(data_path+'summary_mean.p')
```

Data Preprocessing (2)

Drug's molecule structure

This data comes from SMILES. This is a linear notation to express the chemical compounds in a unique manner. The data can be accessed here: <https://www.dna.bio.keio.ac.jp/smiles/>

This will be used in the language processing model called Transformer to encode the sequences.

In [11]:

```
#FILTERING and ERROR CORRECTION
drug=pd.read_csv(data_path+'drug.csv')
drug.drop('id', axis=1, inplace=True)
drug['id']=drug['dname'].apply(lambda x: codes['drugs'].item2idx.get(x))
drug=drug.loc[~drug['id'].isna(),:]

drug.loc[drug['cid']==57519530,'smiles']='CC(C)CC1C(=O)NC(C(=O)NC(CCCC=CCCCCCC(=O)NC(C(=O)NC(C(=O)NC(C(=O)N1)CC
drug.loc[drug['cid']==73265323,'smiles']='CC(C)CCCCCCCC(=O)NC1(CC(C(O)C2=C3C=C4C=C20C5=C(C=C(C=C5)C(C6C(=O)NC(C7=C(C(=CC
drug.loc[drug['cid']==16131923,'smiles']='CCCCCCCCC(=O)NC1C(C(C(O)C2=C3C=C4C=C20C5=C(C=C(C=C5)C(C6C(=O)NC(C7=C(C(=CC(=C
print(drug)
drug.to_csv(data_path+'drug_clean.csv',index=False)
```

```

dname          chembl_id \
1   5-Fluorouracil      CHEMBL185
2   Veliparib          CHEMBL506871
3   MK-1775            CHEMBL1976040
4   915019-65-7        CHEMBL1879463
5   Bortezomib         CHEMBL325041
...
8165 BENZTROPINE MESYLATE      CHEMBL558805
8333 artemisinin           NaN
8335 quinidine             CHEMBL512340
8369 alfacalcidol          CHEMBL2362510
8390 chlorprothixene      CHEMBL90125; CHEMBL90125

inchikey \
1   GHASV$INZRGABV-UHFFFAOYSA-N
2   JNAHV$VVRKWKWKKQ-CYBMUJFWSA-N
3   BKWJAKQVGHWELA-UHFFFAOYSA-N
4   JOGKUKXHTYWRGZ-UHFFFAOYSA-N
5   GXJABQQUPOEUTA-RDJZCZTQSA-N
...
8165 GIJXKZJWITVLHI-RTBURBONSA-N
8333 BLUAFFEHZUWYNDE-GEURJLFSSA-N
8335 LOUPRKONTZGTKE-VPCNSNALSA-N
8369 OFHCOWSQAMBJIW-GITLSTMWSA-N
8390 WSPOMRSOLSGNFJ-VGOFMYFVSA-N

smiles      cid \
1   C1=C(C(=O)NC(=O)N1)F      3385.0
2   CC1(CCCN1)C2=NC3=C(C=CC=C3N2)C(=O)N 11960529.0
3   CC(C)(C1=NC(=CC=C1)N2C3=NC(=NC=C3C(=O)N2CC=C)C... 24856436.0
4   CC(C)(C#)C1=CC=C(C=C1)N2C3=C4C=C(C=C4C=NC=C3N... 11977753.0
5   B(C(CC(C)C)NC(=O)C(CC1=CC=CC=C1)NC(=O)C2=NC=CN... 387447.0
...
8165 CN1[C@H]2CC[C@H]1CC(OC(c1ccccc1)c1ccccc1)C2      8584.0
8333 CC1CCC2C(C(=O)O)C3C24C1CCC(O3)(O04)C)C 5458876.0
8335 COC1=C2C=C(C=CN=C2C=C1)(C3CC4CCN3CC4C=C)O 6604605.0
8369 CC(C)CCCC(C)C1CCC2C1(CCCCC2=CC=C3CC(CC(C3=C)O)O)C 5284515.0
8390 CN(C)CCC=C1C2=CC=C=C=C2SC3=C1C=C(C=C3)C1 667466.0

molecular_formula clinical_phase cid_m      cid_s \
1   C4H3FN2O2          4.0 CIDm00003385  CIDs00003385
2   C13H16N40          3.0 CIDm11842604  CIDs11960529
3   C27H32N8O2          2.0 CIDm24856436  CIDs24856436
4   C30H23N5O          2.0 CIDm11977753  CIDs11977753
5   C19H25BN4O4          4.0 CIDm00093860  CIDs00387447
...
8165 C21H25N0          ...  ...
8333 C15H22O5          3.0 CIDm00002240  CIDs05458876
8335 C20H24N2O2          3.0 CIDm00001065  CIDs06604605
8369 C27H44O2          4.0 CIDm00002091  CIDs05284515
8390 C18H18C1NS          3.0 CIDm00002729  CIDs00667466

stitch_name drugbank_id kegg_id \
1   5-fluorouracil      DB00544  D00584
2   veliparib          DB07232  D09692
3   MK-1775            DB11740  NaN
4   NVP-BEZ235         DB11651  D10552
5   bortezomib         DB00188  D03150
...
8165 benzotropine      NaN  C06846
8333 artemisinin       NaN  NaN
8335 quinine           NaN  NaN
8369 alfacalcidol      NaN  NaN
8390 chlorprothixene  DB01239  NaN

synonyms \
1   5-Fluorouracil; fluorouracil; 51-21-8; 5-FU; F...
2   Veliparib; 912444-00-9; ABT-888; ABT 888; ABT-...
3   MK-1775; 955365-80-7; Adavosertib; MK1775; MK ...
4   915019-65-7; dactolisib; NVP-BEZ235; BEZ235; N...
5   Bortezomib; 179324-69-7; Velcade; PS-341; LDP-...
...
8165 BENZTROPINE MESYLATE; Benzotropine mesylate [US...
8333 artemisinin; Artemisinine; NCGC00160207-01; DS...
8335 quinidine; Conquinine; 56-54-2; Quindine; CHEM...
8369 alfacalcidol; Etaalpha; 41294-56-8; 1alpha-Hydr...
8390 chlorprothixene; Chlorprothixene; (E)-chlorpro...

target_name \
1   Prelamin-A\;C; Survival motor neuron protein; ...
2   Poly [ADP-ribose] polymerase-1; Poly [ADP-ribo...
3   Serine\;/threonine-protein kinase TBK1; Protein...

```

```

4     ALK tyrosine kinase receptor; Macrophage colon...
5     Cathepsin G; Beta-chymotrypsin; Proteasome sub...
...
8165
8333
8335
8369
8390

           target_type      id
1     single protein; nucleic acid 1188.0
2     single protein; protein family 1112.0
3           single protein    882.0
4   single protein; protein complex; protein compl...  246.0
5   single protein; protein complex group; protein... 1172.0
...
8165
8333
8335
8369
8390

```

[3248 rows x 16 columns]

```
In [12]: #GET MACCS FINGERPRINT
drug=pd.read_csv(data_path+'drug_clean.csv')
drug['fps'] = drug['smiles'].apply(lambda x: list(MACCSKeys.GenMACCSKeys(Chem.MolFromSmiles(x.replace("\\", "")).replace(";", ""))))
drug.to_pickle(data_path+'drug_clean_fp.p')

< [REDACTED] >
```

```
In [13]: drug=pickle.load(open(data_path+'drug_clean_fp.p', 'rb'))
drug['id']=drug['id'].apply(lambda x: int(x))
```

```
In [14]: drug
```

```
Out[14]:
```

	dname	chembl_id	inchikkey	smiles	cid	molecul
0	5-Fluorouracil	CHEMBL185	GHASVSVNZRGABV-UHFFFAOYSA-N	C1=C(C(=O)NC(=O)N1)F	3385.0	(
1	Veliparib	CHEMBL506871	JNAHVYVRKWKWKQ-CYBMUJFWSA-N	CC1(CCCN1)C2=NC3=C(C=CC=C3N2)C(=O)N	11960529.0	(

0	5-Fluorouracil	CHEMBL185	GHASVSVNZRGABV-UHFFFAOYSA-N	C1=C(C(=O)NC(=O)N1)F	3385.0	(
----------	----------------	-----------	-----------------------------	----------------------	--------	---

1	Veliparib	CHEMBL506871	JNAHVYVRKWKWKQ-CYBMUJFWSA-N	CC1(CCCN1)C2=NC3=C(C=CC=C3N2)C(=O)N	11960529.0	(
----------	-----------	--------------	-----------------------------	-------------------------------------	------------	---

2	MK-1775	CHEMBL1976040	<chem>BKVVJAKQVGHWELA-UHFFFAOYSA-N</chem>	<chem>(C1=NC(=CC=C1)N2C3=NC(=NC=C3C(=O)N2CC=C)N...</chem>	24856436.0	C:
3	915019-65-7	CHEMBL1879463	<chem>JOGUKUXHTYWRGZ-UHFFFAOYSA-N</chem>	<chem>(C#N)C1=CC=C(C=C1)N2C3=C4C=C(C=CC4=NC=C3N...</chem>	11977753.0	C:
4	Bortezomib	CHEMBL325041	<chem>GXJABQQUPOEUTA-RDJZCZTQSA-N</chem>	<chem>B(C(CC(C)C)NC(=O)C(CC1=CC=CC=C1)NC(=O)C2=NC=CN...</chem>	387447.0	C1S
...
3243	BENZTROPINE MESYLATE	CHEMBL558805	<chem>GIJXKZJWITVLHI-RTBURBONSA-N</chem>	<chem>CN1[C@@H]2CC[C@H]1CC(OC(c1ccccc1)c1ccccc1)C2</chem>	8584.0	
3244	artemisinin	NaN	<chem>BLUAFEHZUWYNDE-GEURJLFFSA-N</chem>	<chem>CC1CCC2C(C(=O)OC3C24C1CCC(O3)(OO4)C)C</chem>	5458876.0	

3245 quinidine CHEMBL512340 LOUPRKONTZGTKE-
VPCNSNALSA-N COC1=CC2=C(C=CN=C2C=C1)C(C3CC4CCN3CC4C=C)O 6604605.0 C

3246 alfacalcidol CHEMBL2362510 OFHCOWSQAMBJIW-
GITLSTMWSA-N CC(C)CCCC(C)C1CCC2C1(CCCC2=CC=C3CC(CC(C3=C)O)O)C 5284515.0

3247 chlorprothixene CHEMBL90125; CHEMBL90125 WSPOMRSOLSGNFJ-
VGOFMYFVSA-N CN(C)CCC=C1C2=CC=CC=C2SC3=C1C=C(C=C3)Cl 667466.0 C

3248 rows × 17 columns

```
In [15]: #Convert SMILES's character into index
drug['smiles'] = drug['smiles'].astype(str)
seqs=drug['smiles'].to_list()
chars=set([char for seq in seqs for char in seq])
chars = ['']+list(chars) #for zero embedding
codes['mole']=Mapping(chars)
drug['smiles']=drug['smiles'].apply(lambda x : [codes['mole'].item2idx[char] for char in x])
```

Data Preprocessing (3)

Drug's target gene

This section will incorporate 3 different drug databases in order to get genomic features. These 3 databases are DrugBank, Therapeutic Target Database (TTD), and NIH-LINC.

DrugBank:<https://go.drugbank.com/releases/latest>

TTD: <https://db.idrblab.net/ttd/full-data-download>

NIH-LINC: <https://lincs.hms.harvard.edu/db/datasets/20316/main>

```
In [16]: #Merge target genes from DrugBank
drug_DrugBank_target=pd.read_csv(external_data_path+'genetic_features/drug/drug_DrugBank_target.csv')
drug_DrugBank_target=drug_DrugBank_target[['Gene', 'drugbank_id']].groupby('drugbank_id').agg(lambda x: set(x)).applymap(lambda
```

```

drug=pd.merge(drug, drug_DrugBank_target, how='left', on='drugbank_id')
drug.rename(columns={'Gene':'gene_drugbank'}, inplace=True)
drug['gene_drugbank']=drug['gene_drugbank'].apply(lambda x: [] if type(x) is float else x)

In [17]: #Merge target genes from TTD
drug_TTD_target=pd.read_csv(external_data_path+'genetic_features/drug/drug_TTD_target.csv')
drug_TTD_target['TTD_TARGETS']=drug_TTD_target[['TTD_TARGETS']].apply(lambda x: re.split(',|;', x))
drug=pd.merge(drug, drug_TTD_target.loc[:,['TTD_TARGETS', 'cid']], how='left', on='cid')
drug.rename(columns={'TTD_TARGETS':'gene_ttd'}, inplace=True)
drug['gene_ttd']=drug['gene_ttd'].apply(lambda x: [] if type(x) is float else x)

In [18]: #Merge target genes from LINC
drug_lincs_target=pd.read_csv(external_data_path+'genetic_features/drug/drug_lincs_target.csv')
drug_lincs_target['target_genes']=drug_lincs_target['target_genes'].apply(lambda x: x.split(',') if type(x) is not float else [])
drug=pd.merge(drug, drug_lincs_target.loc[:, ['target_genes', 'cid']], how='left', on='cid')
drug.rename(columns={'target_genes':'gene_linc'}, inplace=True)

In [19]: # Ensure all relevant columns are filled with Lists, and handle None or non-iterable types
drug['gene_drugbank'] = drug['gene_drugbank'].apply(lambda x: x if isinstance(x, list) else [])
drug['gene_ttd'] = drug['gene_ttd'].apply(lambda x: x if isinstance(x, list) else [])
drug['gene_linc'] = drug['gene_linc'].apply(lambda x: x if isinstance(x, list) else [])

# Combine target genes, safely handling non-list types
drug['gene'] = drug.apply(lambda row: list(set(row['gene_drugbank']) + row['gene_ttd'] + row['gene_linc'])), axis=1

In [20]: #Gene mapping
unique_genes_drug=[l for lst in drug['gene'].values for l in lst]

```

Data Preprocessing (4)

Cell Line Information

This cell line information is also coming from DrugComb which has a link to it in previous sections

```

In [21]: cell_line=pd.read_csv(data_path+'cell_line.csv')

In [22]: cell_line.replace({'large intestine':'colon'}, inplace=True)
cell_line.drop('id',axis=1, inplace=True)

In [23]: cell_line['cell_id']=cell_line['name'].apply(lambda x: codes['cell'].item2idx.get(x))

In [24]: #subset of cell line features
cell_line_tissue = cell_line.loc[~cell_line['cell_id'].isna(), ['cell_id', 'tissue_name', 'disease_name']]
cell_line_tissue['cell_id']=cell_line_tissue['cell_id'].astype(int)
cell_line_tissue=cell_line_tissue.groupby('cell_id').first()

In [25]: #tissue id to name mapping
codes['tissue']=Mapping(cell_line_tissue['tissue_name'].unique())
cell_line_tissue['tissue_name']=cell_line_tissue['tissue_name'].apply(lambda x: codes['tissue'].item2idx[x])
cell_line_tissue.rename(columns={'tissue_name':'tissue_id'}, inplace=True)

In [26]: #disease id to name mapping
codes['disease']=Mapping(cell_line_tissue['disease_name'].apply(lambda x: x.split(' ')[-1].lower() if type(x) is not float else x))
cell_line_tissue['disease_name']=cell_line_tissue['disease_name'].apply(lambda x: codes['disease'].item2idx[x.split(' ')[-1]])
cell_line_tissue.rename(columns={'disease_name':'disease_id'}, inplace=True)

In [27]: codes['tissue'].item2idx['haematopoietic_and_lymphoid']=1
codes['tissue'].idx2item[1]='hem&lymp'

```

```
In [28]: codes['tissue'].item2idx
```

```
Out[28]: {'brain': 0,
 'haematopoietic_and_lymphoid': 1,
 'bone': 2,
 'kidney': 3,
 'lung': 4,
 'breast': 5,
 'large_intestine': 6,
 'prostate': 7,
 'colon': 8,
 'ovary': 9,
 'skin': 10,
 'endometrium': 11,
 'pancreas': 12}
```

```
In [29]: #disease name to id mapping
codes['disease'].item2idx
```

```
Out[29]: {'(gbm)': 0,
 'hl': 1,
 'neoplasm': 2,
 'leukemia': 3,
 'kidney': 4,
 'adenocarcinoma': 5,
 'nst': 6,
 'ccrcc': 7,
 'colon': 8,
 'prostate': 9,
 'carcinoma': 10,
 '(aml)': 11,
 'ovary': 12,
 'melanoma': 13,
 'malignant': 14,
 'breast': 15,
 'lung': 16,
 'bac': 17,
 'disease': 18,
 'astrocytoma': 19,
 'component': 20,
 'cm': 21,
 'cystadenocarcinoma': 22,
 'lymphoma': 23,
 'cancer': 24,
 'pleura': 25,
 'rectum': 26,
 'pancreas': 27}
```

```
In [30]: tissue_cell_cnts = cell_line_tissue.groupby('tissue_id').count().sort_values(by='disease_id', ascending=False)[['disease_id']]

# Create a figure and axis for the plot
plt.figure(figsize=(8, 5))

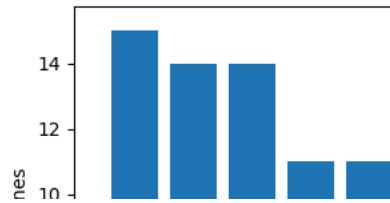
# Generate the plot with numerical x-values
plt.bar(range(len(tissue_cell_cnts)), tissue_cell_cnts) # Use numerical indices as x-values

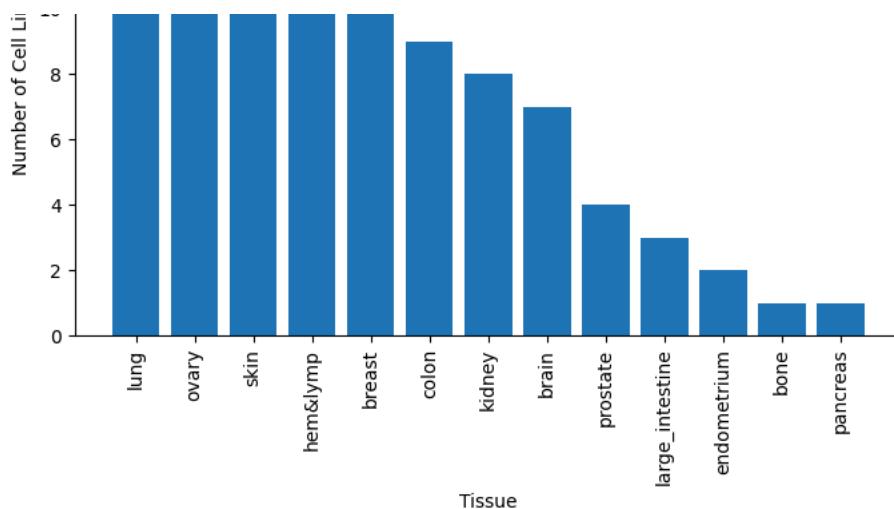
# Set the x-tick labels to the corresponding tissue names, rotating them for better visibility
plt.xticks(range(len(tissue_cell_cnts)), [codes['tissue'].idx2item[tissue_id] for tissue_id in tissue_cell_cnts.index], rotation=45)

# Set the labels for the axes
plt.xlabel('Tissue')
plt.ylabel('Number of Cell Lines')

# Optionally, save the figure
# plt.savefig(data_path + 'tissue_num_cell.png', bbox_inches='tight', dpi=200)

# Show the plot
plt.show()
```





Data Preprocessing (5)

Cell line's gene expression

Cell lines are also a part of this project and we will synthesize two datasets. This first dataset is CCLE and can be found here: <https://depmap.org/portal/download/all/>

The second dataset used is called COSMIC and can be found here: <https://cancer.sanger.ac.uk/cosmic/download/cosmic>

In [31]:

```
#Mapping
gene_identifiers=pd.read_csv(external_data_path+'genetic_features/cell/gene_identifiers_latest.csv')
cell_line_meta = pd.read_csv(external_data_path+'genetic_features/cell/cell_line_meta.csv')

_codes={}
_codes['gene_id2hgnc_symbol']=gene_identifiers[['gene_id', 'hgnc_symbol']].set_index('gene_id').to_dict('index')
_codes['SANGER_ID2cell_name']=cell_line_meta.loc[cell_line_meta['name'].isin(codes['cell'].idx2item) & ~cell_line_meta['SA
#Broad institute
fpkm_broad=pd.read_csv(external_data_path+'genetic_features/cell/fpkm_broad.csv')

fpkm_broad=fpkm_broad.loc[fpkm_broad['model_id'].isin(_codes['SANGER_ID2cell_name'])]#apply(lambda x: _codes['SANGER_ID2ce
fpkm_broad['model_id']=fpkm_broad['model_id'].apply(lambda x: codes['cell'].item2idx[_codes['SANGER_ID2cell_name'][x]['nam
fpkm_broad.rename(columns={col:_codes['gene_id2hgnc_symbol'][col]['hgnc_symbol']} for col in fpkm_broad.columns[1:]}, inplace=True)

fpkm_broad.rename(columns={'model_id':'cell_id'}, inplace=True)
fpkm_broad.set_index('cell_id', inplace=True)
(fpkm_broad.sum(>0).sum()
```

<ipython-input-31-941d48da97aa>:12: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view
-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view
-versus-a-copy)
fpkm_broad['model_id']=fpkm_broad['model_id'].apply(lambda x: codes['cell'].item2idx[_codes['SANGER_ID2cell_name'][x]['nam
e']])
<ipython-input-31-941d48da97aa>:13: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view
-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view
-versus-a-copy)
fpkm_broad.rename(columns={col:_codes['gene_id2hgnc_symbol'][col]['hgnc_symbol']} for col in fpkm_broad.columns[1:]), inplace=True)
<ipython-input-31-941d48da97aa>:15: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view
-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view
-versus-a-copy)
fpkm_broad.rename(columns={'model_id':'cell_id'}, inplace=True)

Out[31]: 21494

In [32]:

```
#Sangar
fpkm_sangar=pd.read_csv(external_data_path+'genetic_features/cell/fpkm_sangar.csv')
```

```

fpkm_sangar=fpkm_sangar.loc[fpkm_sangar['model_id'].isin(_codes['SANGAR_ID2cell_name'])]#apply(Lambda x: _codes['SANGAR_ID'].  
fpkm_sangar['model_id']=fpkm_sangar['model_id'].apply(lambda x: codes['cell'].item2idx[_codes['SANGAR_ID2cell_name'][x]])  
fpkm_sangar.rename(columns={col:_codes['gene_id2hgnc_symbol'][col]['hgnc_symbol'] for col in fpkm_sangar.columns[1:]}, inplace=True)  
  
fpkm_sangar.rename(columns={'model_id':'cell_id'}, inplace=True)  
fpkm_sangar.set_index('cell_id', inplace=True)  
(fpkm_sangar.sum()>0).sum()

```

```
<ipython-input-32-afdf32554896>:5: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row indexer,col indexer] = value instead
```

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
fpkmsangar['model_id']=fpkmsangar['model_id'].apply(lambda x: codes['cell'].item2idx[_codes['SANGAR_ID2cell_name'][x]['name']])
```

```
<ipython-input-32-afdf32554896>:6: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame
```

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
fpkm_sangar.rename(columns={col:_codes['gene_id2hgnc_symbol'][col]['hgnc_symbol'] for col in fpkm_sangar.columns[1:]}, inplace=True)
```

```
<ipython-input-32-afdf32554896>:8: SettingWithCopyWarning:
```

```
[ipython-Input-32-and152554890]:8: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame
```

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
fpkmsangar.rename(columns={'model_id':'cell_id'}, inplace=True)
```

Out[32]: 18421

```
In [33]: cell_gene_fpkm= pd.concat([fpkm_broad,fpkm_sangar], sort=True) #integrate BROAD and SANGAR  
cell_gene_fpkm=cell_gene_fpkm.loc[:,(cell_gene_fpkm.std(skipna=True)!=0)] #drop genes with std=0  
cell_gene_fpkm.fillna(0, inplace=True)  
  
cell_gene_fpkm_norm=((cell_gene_fpkm-cell_gene_fpkm.mean())/cell_gene_fpkm.std()) #z-score, gene-wise  
  
unique_genes_cell = cell_gene_fpkm_norm.columns.to_list()  
print(unique_genes_cell)  
len(unique_genes_cell)
```

```
<ipython-input-33-2874de292690>:3: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame
```

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```

cell_gene_fpkm.fillna(0, inplace=True)
['A1BG-AS1', 'A1CF', 'A2M', 'A2M-AS1', 'A2ML1', 'A2ML1-AS2', 'A4GALT', 'A4GNT', 'AAAS', 'AACS', 'AACSP1', 'AADAC', 'AADACL2-AS1', 'AADACP1', 'AADAT', 'AAED1', 'AAAGB', 'AAK1', 'AAMDC', 'AAMP', 'AAR2', 'AARD', 'AARS', 'AARS2', 'AARSD1', 'AASDH', 'AA SDHPP1', 'AASS', 'AATBC', 'AATF', 'AATK', 'ABALON', 'ABAT', 'ABCA1', 'ABCA10', 'ABCA11P', 'ABCA12', 'ABCA13', 'ABCA17P', 'AB CA2', 'ABCA3', 'ABCA4', 'ABCAS5', 'ABCAT7', 'ABCA8', 'ABCB1', 'ABCB10', 'ABCB10P1', 'ABCB10P3', 'ABCB10P4', 'ABCB4', 'ABCB5', 'ABCB6', 'ABCB7', 'ABCB8', 'ABCB9', 'ABCC1', 'ABCC10', 'ABCC11', 'ABCC12', 'ABCC2', 'ABCC3', 'ABCC4', 'ABCC5', 'ABCC5-AS1', 'ABCC6', 'ABCC6P1', 'ABCC6P2', 'ABCC8', 'ABCC9', 'ABCD1', 'ABCD2', 'ABCD3', 'ABCD4', 'ABCE1', 'ABCF1', 'ABCF2', 'ABCF3', 'AB CG1', 'ABCG2', 'ABCG5', 'ABHD1', 'ABHD10', 'ABHD11', 'ABHD11-AS1', 'ABHD12', 'ABHD12B', 'ABHD13', 'ABHD14A', 'ABHD14A-ACY1', 'ABHD14B', 'ABHD15', 'ABHD16A', 'ABHD17A', 'ABHD17AP3', 'ABHD17AP4', 'ABHD17AP5', 'ABHD17B', 'ABHD17C', 'ABHD18', 'ABHD2', 'ABHD3', 'ABHD4', 'ABHD5', 'ABHD6', 'ABHD8', 'ABI1', 'ABI2', 'ABI3', 'ABI3BP', 'ABL1', 'ABL2', 'ABLIM1', 'ABLIM2', 'ABLIM3', 'ABO', 'ABR', 'ABRACL', 'ABRAXAS1', 'ABRAXAS2', 'ABT1', 'ABTB1', 'ABTB2', 'ACAA1', 'ACAA2', 'ACACA', 'ACACB', 'ACAD10', 'ACA D11', 'ACAD8', 'ACAD9', 'ACADL', 'ACADM', 'ACADS', 'ACADSB', 'ACADVL', 'ACAN', 'ACAP1', 'ACAP2', 'ACAP2-IT1', 'ACAP3', 'ACAT 1', 'ACAT2', 'ACBD3', 'ACBD3-AS1', 'ACBD4', 'ACBD5', 'ACBD6', 'ACBD7', 'ACCS', 'ACD', 'ACE', 'ACE2', 'ACER1', 'ACER2', 'ACER 3', 'ACHE', 'ACIN1', 'ACKR1', 'ACKR2', 'ACKR3', 'ACKR4', 'ACLY', 'ACO1', 'ACO2', 'ACOT1', 'ACOT11', 'ACOT12', 'ACOT13', 'ACO T2', 'ACOT4', 'ACOT6', 'ACOT7', 'ACOT8', 'ACOT9', 'ACOX1', 'ACOX2', 'ACOX3', 'ACOXL', 'ACP1', 'ACP2', 'ACP4', 'ACP5', 'ACP 6', 'ACP7', 'ACP8', 'ACRP1', 'ACRV1', 'ACSBG1', 'ACSF2', 'ACSF3', 'ACSL1', 'ACSL3', 'ACSL4', 'ACSL5', 'ACSL6', 'ACSM1', 'ACSM 3', 'ACSM4', 'ACSM5', 'ACSS1', 'ACSS2', 'ACSS3', 'ACTA1', 'ACTA2', 'ACTA2-AS1', 'ACTB', 'ACTBL2', 'ACTBP1', 'ACTBP11', 'ACT BP12', 'ACTBP13', 'ACTBP7', 'ACTBP8', 'ACTC1', 'ACTG1', 'ACTG1P1', 'ACTG1P10', 'ACTG1P14', 'ACTG1P20', 'ACTG1P22', 'ACTG1P 3', 'ACTG2', 'ACTL10', 'ACTL6A', 'ACTL6B', 'ACTL8', 'ACTN1', 'ACTN2', 'ACTN3', 'ACTN4', 'ACTN4P2', 'ACTR10', 'ACTR1A', 'ACTR 1B', 'ACTR2', 'ACTR3', 'ACTR3B', 'ACTR3C', 'ACTR5', 'ACTR6', 'ACTR8', 'ACTRT3', 'ACVR1', 'ACVR1B', 'ACVR1C', 'ACVR2A', 'ACVR 2B', 'ACVR2B-AS1', 'ACVR1L', 'ACY1', 'ACY3', 'ACYP1', 'ACYP2', 'ADA', 'ADA2', 'ADAL', 'ADAM10', 'ADAM11', 'ADAM12', 'ADAM1 5', 'ADAM17', 'ADAM19', 'ADAM20P1', 'ADAM21', 'ADAM22', 'ADAM23', 'ADAM28', 'ADAM33', 'ADAM7', 'ADAM8', 'ADAM9', 'ADAMTS1', 'ADAMTS10', 'ADAMTS12', 'ADAMTS13', 'ADAMTS14', 'ADAMTS15', 'ADAMTS16', 'ADAMTS17', 'ADAMTS18', 'ADAMTS19', 'ADAMTS19-AS1', 'ADAMTS2', 'ADAMTS20', 'ADAMTS3', 'ADAMTS4', 'ADAMTS5', 'ADAMTS6', 'ADAMTS7', 'ADAMTS8', 'ADAMTS9', 'ADAMTS9-AS2', 'ADAMTS1 1', 'ADAMTS12', 'ADAMTS13', 'ADAMTS14', 'ADAMTS14-AS1', 'ADAMTS5', 'ADAP1', 'ADAP2', 'ADAR', 'ADAR1', 'ADAR2B-AS1', 'ADAT 1', 'ADAT2', 'ADAT3', 'ADCK1', 'ADCK2', 'ADCK5', 'ADCY1', 'ADCY10', 'ADCY2', 'ADCY3', 'ADCY4', 'ADCY5', 'ADCY6', 'ADCY7', 'ADCY8', 'ADCY9', 'ADCYAP1', 'ADD1', 'ADD2', 'ADD3', 'ADGB', 'ADGRA2', 'ADGRA3', 'ADGRB1', 'ADGRB2', 'ADGRB3', 'ADG RD1', 'ADGRD1-AS1', 'ADGRD2', 'ADGRE1', 'ADGRE2', 'ADGRE3', 'ADGRE4P', 'ADGRE5', 'ADGRF1', 'ADGRF3', 'ADGRF4', 'ADGRF5', 'AD GRF5P1', 'ADGRG1', 'ADGRG2', 'ADGRG3', 'ADGRG5', 'ADGRG6', 'ADGRG7', 'ADGR1L', 'ADGR2L', 'ADGR3L', 'ADGR4L', 'ADGRV1', 'ADH1 C', 'ADH5', 'ADH6', 'ADHFE1', 'ADI1', 'ADIPOR1', 'ADIPOR2', 'ADIRF', 'ADIRF-AS1', 'ADK', 'ADM', 'ADM2', 'ADM5', 'ADNP', 'ADN P-AS1', 'ADNP2', 'ADO', 'ADORA1', 'ADORA2A', 'ADORA2B', 'ADORA3', 'ADPGK', 'ADPRH', 'ADPRHL1', 'ADPRHL2', 'ADPRM', 'ADRA1B', 'ADRA1D', 'ADRA2A', 'ADRA2B', 'ADRA2C', 'ADRB1', 'ADRB2', 'ADRM1', 'ADSL', 'ADSS', 'ADSSL1', 'ADTRP', 'AEBP1', 'AEBP2', 'AE N', 'AES', 'AFAP1', 'AFAP1-AS1', 'AFAP1L1', 'AFAP1L2', 'AFDN', 'AFDN-DT', 'AFF1', 'AFF2', 'AFF3', 'AFF4', 'AFG1L', 'AFG3L1 P', 'AFG3L2', 'AFMID', 'AFP', 'AFTPH', 'AGA', 'AGAP1', 'AGAP1-IT1', 'AGAP12P', 'AGAP13P', 'AGAP2', 'AGAP2-AS1', 'AGAP3', 'AG

```


, 'BA94', 'BA95', 'BA96', 'BA9E2', 'BA9U1', 'BA1AP2-U1', 'BA1AP2L1', 'BA1AP2L2', 'BA1AP3', 'BA1', 'BA1P2', 'BAMBI', 'BANCR', 'BANF1', 'BANF1P2', 'BANK1', 'BANP', 'BAP1', 'BARD1', 'BARHL2', 'BARX1', 'BARX1-DT', 'BARX2', 'BASP 1', 'BASP1-AS1', 'BASP1P1', 'BATF', 'BATF2', 'BATF3', 'BAX', 'BAZ1A', 'BAZ1B', 'BAZ2A', 'BAZ2B', 'BBC3', 'BBIP1', 'BBIP1P1', 'BBOF1', 'BBOX1', 'BBOX1-AS1', 'BBS1', 'BBS10', 'BBS12', 'BBS2', 'BBS4', 'BBS5', 'BBS7', 'BBS9', 'BBX', 'BCAM', 'BCAN', 'BCA P29', 'BCAP31', 'BCAP31P1', 'BCAP31P2', 'BCAR1', 'BCAR3', 'BCAR4', 'BCAS1', 'BCAS2', 'BCAS2P1', 'BCAS3', 'BCAS4', 'BCAT1', 'BCAT2', 'BCCIP', 'BCDIN3D', 'BCDIN3D-AS1', 'BCEHE', 'BCKDHA', 'BCKDK', 'BCL10', 'BCL11A', 'BCL11B', 'BCL 2', 'BCL2A1', 'BCL2L1', 'BCL2L10', 'BCL2L11', 'BCL2L12', 'BCL2L13', 'BCL2L14', 'BCL2L15', 'BCL2L2', 'BCL2L2-PABPN1', 'BCL3', 'BCL6', 'BCL6-AS1', 'BCL6B', 'BCL7A', 'BCL7B', 'BCL7C', 'BCL9', 'BCL9L', 'BCLAF1', 'BCLAF3', 'BCO1', 'BCO2', 'BCOR', 'BCORL 1', 'BCORP1', 'BCR', 'BCRP2', 'BCRP3', 'BCRP4', 'BCRP9', 'BCS1L', 'BCYRN1', 'BDH1', 'BDH2', 'BDKR81', 'BDKR82', 'BDNF', 'BDN F-AS', 'BDP1', 'BEAN1', 'BECN1', 'BEGAIN', 'BEND3', 'BEND3P1', 'BEND4', 'BEND5', 'BEND6', 'BEND7', 'BEST 1', 'BEST3', 'BEST4', 'BET1', 'BET1L', 'BEX1', 'BEX2', 'BEX3', 'BEX4', 'BEX5', 'BFAR', 'BFSP1', 'BFSP2', 'BFSP2-AS1', 'BGLA P', 'BGLT3', 'BGN', 'BHLHA15', 'BHLHB9', 'BHLHE22', 'BHLHE23', 'BHLHE40', 'BHLHE40-AS1', 'BHLHE41', 'BHMG1', 'BHMT', 'BHMT 2', 'BICC1', 'BICD1', 'BICD2', 'BICDL1', 'BICDL2', 'BICRA', 'BICRAL', 'BID', 'BIK', 'BIN1', 'BIN2', 'BIN3', 'BIRC 3', 'BIRC5', 'BIRC6', 'BIRC6-AS2', 'BIRC7', 'BISPR', 'BIVM', 'BLACAT1', 'BLACE', 'BLCAP', 'BLID', 'BLK', 'BLM', 'BLMH', 'BLN K', 'BLOC1S1', 'BLOC1S2', 'BLOC1S3', 'BLOC1S4', 'BLOC1S5', 'BLOC1S6', 'BLVRA', 'BLZF1', 'BME', 'BMI1', 'BMP1', 'BMP 10', 'BMP2', 'BMP2K', 'BMP4', 'BMP5', 'BMP6', 'BMP7', 'BMP8A', 'BMP8B', 'BMPER', 'BMPR1A', 'BMPR1B', 'BMPR1B-DT', 'BMPR2', 'BMS1', 'BMS1P1', 'BMS1P10', 'BMS1P14', 'BMS1P2', 'BMS1P20', 'BMS1P22', 'BMS1P4', 'BMS1P7', 'BMS1P8', 'BMS1P9', 'BMT2', 'BM X', 'BNC1', 'BNC2', 'BNC2-AS1', 'BNIP1', 'BNIP2', 'BNIP3', 'BNIP3L1', 'BNIP3P11', 'BNIP3P15', 'BNIP3P17', 'BNIP3P2', 'BNIP3P2 4', 'BNIP3P27', 'BNIP3P30', 'BNIP3P37', 'BNIP3P39', 'BNIP3P4', 'BNIP3P41', 'BNIP3P9', 'BNIP1', 'BOC', 'BOD1', 'BOD1L1', 'BO K', 'BOLA1', 'BOLA2-SMG1P6', 'BOLA2B', 'BOLA2P2', 'BOLA2P3', 'BOLA3', 'BOLA3-AS1', 'BOLA3P2', 'BOP1', 'BORA', 'BORCS5', 'BOR CS6', 'BORCS7', 'BORCS8', 'BORCS8-MEF2B', 'BPGM', 'BPHL', 'BPIFA2', 'BPIFA4P', 'BPIFB1', 'BPIFB4', 'BPNT1', 'BPTF', 'BPTFP 1', 'BRAF', 'BRAP', 'BRAT1', 'BRC41', 'BRC42', 'BRCC3', 'BRD1', 'BRD2', 'BRD3', 'BRD3OS', 'BRD4', 'BRD7', 'BRD7P2', 'BRD8', 'BRD9', 'BRD9P2', 'BRDT', 'BRF1', 'BRF2', 'BRI3', 'BRI3P', 'BRICD5', 'BRINP1', 'BRINP2', 'BRINP3', 'BRIP1', 'BRIX1', 'BRK 1', 'BRMS1', 'BRMS1L', 'BROX', 'BRPF1', 'BRPF3', 'BRSK1', 'BRSK2', 'BRWD1', 'BRWD1-AS1', 'BRWD1-IT1', 'BRWD3', 'BSCL2', 'BSDC1', 'BSG', 'BSN', 'BSN-DT', 'BSND', 'BSPRY', 'BST1', 'BST2', 'BTAF1', 'BTBD1', 'BTBD10', 'BTBD11', 'BTBD16', 'BTBD19', 'BTBD2', 'BTBD3', 'BTBD6', 'BTBD7', 'BTBD7P1', 'BTBD8', 'BTBD9', 'BTC', 'BTD', 'BTF3', 'BTF3L4', 'BTF3L4P2', 'BTF3 P6', 'BTG1', 'BTG2', 'BTG3', 'BTG3-AS1', 'BTG4', 'BTK', 'BTLA', 'BTN1A1', 'BTN2A1', 'BTN2A2', 'BTN2A3P', 'BTN3A1', 'BTN3A2', 'BTN3A3', 'BTNL10', 'BTNL2', 'BTNL3', 'BTNL9', 'BTTRC', 'BUB1', 'BUB3', 'BUD13', 'BUD1P1', 'BUD23', 'BUD31', 'BVE S', 'BVES-AS1', 'BYSL', 'BZW1', 'BZW1P2', 'BZW2', 'C10orf111', 'C10orf126', 'C10orf142', 'C10orf25', 'C10orf55', 'C10orf67', 'C10orf71', 'C10orf782', 'C10orf88', 'C10orf90', 'C10orf91', 'C10orf95', 'C10orf99', 'C11orf1', 'C11orf21', 'C11orf24', 'C11o rf42', 'C11orf44', 'C11orf45', 'C11orf49', 'C11orf52', 'C11orf53', 'C11orf54', 'C11orf58', 'C11orf65', 'C11orf68', 'C11orf7 1', 'C11orf72', 'C11orf74', 'C11orf80', 'C11orf86', 'C11orf87', 'C11orf91', 'C11orf94', 'C11orf95', 'C11orf96', 'C11orf98', 'C12orf10', 'C12orf29', 'C12orf4', 'C12orf40', 'C12orf43', 'C12orf45', 'C12orf49', 'C12orf50', 'C12orf56', 'C12orf57', 'C12o rf60', 'C12orf65', 'C12orf66', 'C12orf71', 'C12orf73', 'C12orf75', 'C12orf76', 'C13orf42', 'C14orf119', 'C14orf132', 'C14orf 28', 'C14orf39', 'C14orf93', 'C15orf39', 'C15orf40', 'C15orf41', 'C15orf48', 'C15orf53', 'C15orf54', 'C15orf59', 'C15orf59-A S1', 'C15orf61', 'C15orf62', 'C15orf65', 'C16orf45', 'C16orf46', 'C16orf47', 'C16orf54', 'C16orf58', 'C16orf70', 'C16orf72', 'C16orf74', 'C16orf751', 'C17orf53', 'C17orf58', 'C17orf67', 'C17orf75', 'C17orf77', 'C17orf80', 'C17orf82', 'C17orf97', 'C17 orf99', 'C18orf21', 'C18orf25', 'C18orf32', 'C18orf54', 'C19orf12', 'C19orf18', 'C19orf24', 'C19orf25', 'C19orf33', 'C19orf3 8', 'C19orf44', 'C19orf47', 'C19orf48', 'C19orf53', 'C19orf54', 'C19orf57', 'C19orf66', 'C19orf70', 'C19orf71', 'C19orf73', 'C19orf81', 'C19orf84', 'C1D', 'C1DP1', 'C1GALT1', 'C1GALT1C1', 'C1GALT1C1L', 'C1QBP', 'C1QL1', 'C1QL3', 'C1QL 4', 'C1QTNF1', 'C1QTNF1-AS1', 'C1QTNF2', 'C1QTNF3', 'C1QTNF4', 'C1QTNF6', 'C1QTNF98', 'C1R', 'C1RL', 'C1RL-AS1', 'C1S', 'C1orf100', 'C1orf109', 'C1orf112', 'C1orf115', 'C1orf116', 'C1orf122', 'C1orf123', 'C1orf127', 'C1orf131', 'C1orf13 7', 'C1orf143', 'C1orf159', 'C1orf162', 'C1orf174', 'C1orf195', 'C1orf198', 'C1orf21', 'C1orf210', 'C1orf216', 'C1orf220', 'C1orf226', 'C1orf229', 'C1orf35', 'C1orf43', 'C1orf50', 'C1orf52', 'C1orf53', 'C1orf54', 'C1orf56', 'C1orf61', 'C1orf74', 'C1orf94', 'C2', 'C20orf141', 'C20orf144', 'C20orf194', 'C20orf196', 'C20orf197', 'C20orf204', 'C20orf207', 'C20orf96', 'C21o rf2', 'C21orf58', 'C21orf91', 'C21orf91-0T1', 'C22orf15', 'C22orf23', 'C22orf34', 'C22orf39', 'C22orf46', 'C2CD2', 'C2CD2L', 'C2CD3', 'C2CD4A', 'C2CD4C', 'C2CD4D', 'C2CD5', 'C2CD6', 'C2orf15', 'C2orf16', 'C2orf27A', 'C2orf27AP1', 'C2orf27AP3', 'C2or f40', 'C2orf42', 'C2orf48', 'C2orf49', 'C2orf54', 'C2orf66', 'C2orf68', 'C2orf69', 'C2orf69P1', 'C2orf70', 'C2orf72', 'C2orf 74', 'C2orf76', 'C2orf81', 'C2orf88', 'C2orf91', 'C2orf92', 'C3', 'C3AR1', 'C3P1', 'C3orf14', 'C3orf18', 'C3orf33', 'C3orf3 5', 'C3orf38', 'C3orf49', 'C3orf52', 'C3orf58', 'C3orf62', 'C3orf67', 'C3orf70', 'C3orf80', 'C3orf85', 'C4A', 'C4B', 'C4BP A', 'C4BPAP1', 'C4BPB', 'C4orf19', 'C4orf3', 'C4orf33', 'C4orf36', 'C4orf45', 'C4orf46', 'C4orf47', 'C4orf48', 'C4orf50', 'C4 orf54', 'C5', 'C5AR1', 'C5AR2', 'C5orf15', 'C5orf17', 'C5orf22', 'C5orf24', 'C5orf30', 'C5orf34', 'C5orf38', 'C5orf46', 'C5 orf49', 'C5orf51', 'C5orf56', 'C5orf63', 'C5orf66', 'C5orf66-AS1', 'C6orf106', 'C6orf120', 'C6orf132', 'C6orf136', 'C6orf14 1', 'C6orf15', 'C6orf163', 'C6orf203', 'C6orf222', 'C6orf223', 'C6orf226', 'C6orf47', 'C6orf48', 'C6orf52', 'C6orf58', 'C6or f62', 'C6orf89', 'C6orf99', 'C7orf26', 'C7orf31', 'C7orf43', 'C7orf50', 'C7orf55-LUC7L2', 'C7orf61', 'C7orf65', 'C7orf69', 'C7orf71', 'C8A', 'C8G', 'C8orf31', 'C8orf33', 'C8orf34', 'C8orf34-AS1', 'C8orf37', 'C8orf44', 'C8orf46', 'C8orf48', 'C8orf4 9', 'C8orf58', 'C8orf59', 'C8orf59P', 'C8orf74', 'C8orf76', 'C8orf82', 'C8orf86', 'C8orf88', 'C9orf106', 'C9orf116', 'C9orf 129', 'C9orf135', 'C9orf139', 'C9orf147', 'C9orf152', 'C9orf16', 'C9orf163', 'C9orf24', 'C9orf3', 'C9orf40', 'C9orf41-AS1', 'C9orf43', 'C9orf64', 'C9orf66', 'C9orf72', 'C9orf78', 'C9orf84', 'C9orf85', 'C9orf92', 'CA10', 'CA11', 'CA12', 'CA13', 'CA1 4', 'CA15P1', 'CA2', 'CA3', 'CA3-AS1', 'CA4', 'CA5A', 'CA5B', 'CA5BP1', 'CA6', 'CA8', 'CA9', 'CAAP1', 'CAB39', 'CAB39L', 'CA B39P1', 'CABCOCO1', 'CABIN1', 'CABLES1', 'CABLES2', 'CABP1', 'CABP4', 'CABP7', 'CABYR', 'CACFD1', 'CACHD1', 'CACNA1A', 'CAC NA1B', 'CACNA1C', 'CACNA1C-AS1', 'CACNA1C-AS2', 'CACNA1D', 'CACNA1E', 'CACNA1G-AS1', 'CACNA1H', 'CACNA1I', 'CACNA1 S', 'CACNA2D1', 'CACNA2D2', 'CACNA2D3', 'CACNA2D4', 'CACNB1', 'CACNB2', 'CACNB3', 'CACNB4', 'CACNG4', 'CACNG6', 'CACNG7', 'C ACNG8', 'CACTIN', 'CACTIN-AS1', 'CACUL1', 'CACYBP', 'CACYBP2', 'CAD', 'CADM1', 'CADM2', 'CADM3', 'CADM4', 'CADPS 2', 'CAHM', 'CALB1', 'CALB2', 'CALC', 'CALCOCO1', 'CALCM2', 'CALCOC2', 'CALCR', 'CALCRL', 'CALD1', 'CALHM1', 'CALHM2', 'CAL HM3', 'CALHM4', 'CALHM5', 'CALHM6', 'CALM1', 'CALM2', 'CALM2P', 'CALM3', 'CALML3', 'CALML3-AS1', 'CALML4', 'CALML5', 'CALML 6', 'CALN1', 'CALR', 'CALR4P', 'CALU', 'CALY', 'CAMK1', 'CAMK1D', 'CAMK1G', 'CAMK2', 'CAMK2B', 'CAMK2D', 'CAMK2G', 'CAMK2N 1', 'CAMK2N2', 'CAMK4', 'CAMKK1', 'CAMKK2', 'CAMKMT', 'CAMKV', 'CAMLG', 'CAMP', 'CAMSAP1', 'CAMSAP2', 'CAMSAP3', 'CAMTA1', 'CAMTA1-DT', 'CAMTA1-IT1', 'CAMTA2', 'CAND1', 'CAND2', 'CANT4', 'CANX', 'CAP1', 'CAP2', 'CAPG', 'CAPN1', 'CAPN10', 'CAPN10-D T', 'CAPN11', 'CAPN12', 'CAPN13', 'CAPN14', 'CAPN15', 'CAPN2', 'CAPN3', 'CAPN5', 'CAPN6', 'CAPN7', 'CAPN8', 'CAPN9', 'CAPNS 1', 'CAPRIN1', 'CAPRIN2', 'CAPS', 'CAPS2', 'CAPS1', 'CAPS2', 'CAPZA1', 'CAPZA2', 'CAPZB', 'CAR10', 'CARD11', 'CARD14', 'CARD16', 'CA RD17', 'CARD18', 'CARD19', 'CARD46', 'CARD8', 'CARD8-AS1', 'CARD9', 'CARF', 'CARHSP1', 'CARM1', 'CARMILL1', 'CARMIL2', 'CARMIL 3', 'CARMN', 'CARNMT1', 'CARN51', 'CARS', 'CARS2', 'CARTPT', 'CASC1', 'CASC10', 'CASC15', 'CASC18', 'CASC19', 'CASC20', 'CAS C21', 'CASC3', 'CASC4', 'CASC6', 'CASC8', 'CASC9', 'CASC1', 'CASK', 'CASKIN1', 'CASKIN2', 'CASP1', 'CASP10', 'CASP14', 'CASP 1P2', 'CASP2', 'CASP3', 'CASP4', 'CASP5', 'CASP6', 'CASP7', 'CASP8', 'CASP8P2', 'CASP9', 'CASPQ1', 'CASS4', 'CAST', 'CASTOR 1', 'CASTOR2', 'CASTOR3', 'CASZ1', 'CAT', 'CATIP-AS1', 'CATSPER1', 'CATSPER2', 'CATSPER3', 'CATSPER8', 'CATSPERG', 'CAV1', 'CAV2', 'CAVIN1', 'CAVIN2', 'CAVIN3', 'CAVIN4', 'CBARP', 'CBFA2T2', 'CBFA2T3', 'CBFB', 'CBLB', 'CBLC', 'CBLL1', 'CBLN1', 'CBLN2', 'CBLN3', 'CBLR1', 'CBLR3-AS1', 'CBR4', 'CBS', 'CBL', 'CBLW1', 'CBWD2', 'CBWD3', 'CBWD4P', 'CBWD5', 'CBWD6', 'CBWD7', 'CBX1', 'CBX2', 'CBX3', 'CBX3P2', 'CBX3P7', 'CBX4', 'CBX5', 'CBX5P1', 'CBX6', 'CBX7', 'CBX8', 'CB Y1', 'CBY3', 'CC2D1A', 'CC2D1B', 'CC2D2A', 'CCAR1', 'CCAR2', 'CCAT1', 'CCBE1', 'CCCD102A', 'CCCD102B', 'CCCD106', 'CCCD10 7', 'CCCD110', 'CCCD112', 'CCCD113', 'CCCD114', 'CCCD115', 'CCCD116', 'CCCD117', 'CCCD12', 'CCCD120', 'CCCD121', 'CCCD122', 'CCCD124', 'CCCD125', 'CCCD126', 'CCCD127', 'CCCD130', 'CCCD134', 'CCCD136', 'CCCD137', 'CCCD138', 'CCCD14', 'CCCD142', 'CC CD144A', 'CCCD144B', 'CCCD144CP', 'CCCD144CN', 'CCCD144NL', 'CCCD144NL-AS1', 'CCCD146', 'CCCD148', 'CCCD149', 'CCCD150', 'CCCD15 0B1', 'CCCD151', 'CCCD152', 'CCCD153', 'CCCD154', 'CCCD155', 'CCCD156', 'CCCD157', 'CCCD158', 'CCCD159', 'CCCD160', 'CCCD161', 'CCCD162', 'CCCD163', 'CCCD164'

'1', 'COX7B', 'COX7B2', 'COX7BP1', 'COX7C', 'COX8A', 'CP', 'CPA2', 'CPA3', 'CPA4', 'CPA5', 'CPA6', 'CPAMD8', 'CPB2', 'CPB2-AS 1', 'CPD', 'CPE', 'CPEB1', 'CPEB2', 'CPEB3', 'CPEB4', 'CPED1', 'CPHL1P', 'CPLANE1', 'CPLANE2', 'CPLX1', 'CPLX2', 'CPM', 'CPN 1', 'CPNE1', 'CPNE2', 'CPNE3', 'CPNE4', 'CPNE5', 'CPNE7', 'CPNE8', 'CPNE9', 'CP0', 'CPOX', 'CPPED1', 'CPQ', 'CPS1', 'CPSF1', 'CPSF1P1', 'CPSF2', 'CPSF3', 'CPSF4', 'CPSF4L', 'CPSF6', 'CPSF7', 'CP1A1', 'CP1T1', 'CP1T1C', 'CP2', 'CPTP', 'CPVL', 'CRB 1', 'CPXM 2', 'CPZ', 'CR1L', 'CR2', 'CRABP1', 'CRABP2', 'CRACR2A', 'CRACR2B', 'CRADD', 'CRAMP1', 'CRAT', 'CRB1', 'CRB2', 'CRB 3', 'CRB3P1', 'CRBN', 'CRCP', 'CREB1', 'CREB3', 'CREB3L1', 'CREB3L2', 'CREB3L3', 'CREB3L4', 'CREB5', 'CREBBP', 'CREBL2', 'CR EBRF', 'CREBFZ', 'CREG1', 'CREG2', 'CRELD1', 'CRELD2', 'CREM', 'CRH', 'CRHBP', 'CRHR1', 'CRM1', 'CRIP1', 'CRIP2', 'CRIP3', 'CRIPAK', 'CRIP1', 'CRISP2', 'CRISP3', 'CRISPLD1', 'CRISPLD2', 'CRK', 'CRKL', 'CRLF1', 'CRLF2', 'CRLF3', 'CRLS1', 'CRMP1', 'CRNDE', 'CRNL1', 'CROCC', 'CROCC2', 'CROCCP2', 'CROCCP3', 'CROCCP5', 'CROT', 'CRTAC1', 'CRTAM', 'CRTAP', 'CRTC1', 'CRTC2', 'CRTC3', 'CRTC3-AS1', 'CRY1', 'CRY2', 'CRYAB', 'CRYBA2', 'CRYBA4', 'CRYBB1', 'CRYBB2P1', 'CRYBB3', 'CRYBG1', 'CRYBG2', 'CRYB G3', 'CRYGN', 'CRYGS', 'CRYL1', 'CRYM', 'CRYM-AS1', 'CRYZ', 'CRYZL1', 'CRYZL2P', 'CRYZP1', 'CS', 'CSAD', 'CSAG1', 'CSAG2', 'CSAG3', 'CSAG4', 'CSDC2', 'CSDE1', 'CSE1L', 'CSE1L-AS1', 'CSF1', 'CSF1R', 'CSF2', 'CSF2RA', 'CSF2RB', 'CSF2RB1', 'CSF3', 'CSF3R', 'CSGALNACT1', 'CSGALNACT2', 'CSH2', 'CSK', 'CSKMT', 'CSMD1', 'CSMD2', 'CSMD3', 'CSN3', 'CSNK1A1', 'CSNK1A1P1', 'CSN K1D', 'CSNK1E', 'CSNK1G1', 'CSNK1G2', 'CSNK1G3', 'CSNK2A1', 'CSNK2A2', 'CSNK2A3', 'CSNK2B', 'CSPG4', 'CSPG4P1 0', 'CSPG4P11', 'CSPG4P12', 'CSPG5', 'CSPP1', 'CSRNP1', 'CSRNP2', 'CSRNP3', 'CSRNP1', 'CSRNP2', 'CSRNP3', 'CST1', 'CST11', 'CST 2', 'CST3', 'CST4', 'CST6', 'CST7', 'CSTA', 'CSTB', 'CSTF1', 'CSTF2', 'CSTF2T', 'CSTF3', 'CSTF3-DT', 'CT45A1', 'CT45A10', 'CT45A2', 'CT45A3', 'CT45A5', 'CT55', 'CT62', 'CT83', 'CTAG1B', 'CTAG2', 'CTAGE15', 'CTAGE4', 'CTAGE5', 'CTAGE7P', 'CTB1P1', 'CTB1P1-AS1', 'CTB1P1-DT', 'CTB2P', 'CTB2P1', 'CTB2P2P', 'CTBS', 'CTC1', 'CTCF1', 'CTDNEP1', 'CTDP1', 'CTDSP 2', 'CTDSP1', 'CTDSP2', 'CTF1', 'CTGF', 'CTGLF8P', 'CTH', 'CTHRC1', 'CTIF', 'CTLA4', 'CTNNA1', 'CTNNA2', 'CTNNA3', 'CTNNAL 1', 'CTNNB1', 'CTNNB1P1', 'CTNNBL1', 'CTNND1', 'CTNND2', 'CTNS', 'CTPS1', 'CTPS2', 'CTR9', 'CTS4', 'CTS5', 'CTS6', 'CTSD', 'CTSE', 'CTSG', 'CTSH', 'CTS4', 'CTS5', 'CTS5P1', 'CTS5P2', 'CTS5P8', 'CTS0', 'CTSS', 'CTS4', 'CTS5', 'CTS6', 'CTT N', 'CTTNBP2', 'CTTNBP2N1', 'CTU1', 'CTU2', 'CTXN1', 'CTXND1', 'CUBN', 'CUECD1', 'CUL1', 'CUL2', 'CUL3', 'CUL4A', 'CUL4B', 'CUL5', 'CUL7', 'CUL9', 'CUTA', 'CUTALP', 'CUTC', 'CUX1', 'CUX2', 'CUZD1', 'CWC15', 'CWC22', 'CWC25', 'CWC27', 'CWF 19L1', 'CWF19L2', 'CWH43', 'CX3CL1', 'CX3CR1', 'CXADR', 'CXADR1', 'CXADR2', 'CXADR3', 'CXCL1', 'CXCL10', 'CXCL11', 'CXCL1 2', 'CXCL13', 'CXCL14', 'CXCL17', 'CXCL2', 'CXCL3', 'CXCL5', 'CXCL6', 'CXCL8', 'CXKL9', 'CXKR3', 'CXKR4', 'CXKR6', 'CXKC1', 'CXKC4', 'CXKC5', 'CXKC5-AS1', 'CXKorf21', 'CXKorf38', 'CXorf40A', 'CXorf40B', 'CXorf56', 'CXorf57', 'CXorf65', 'CYB5 61', 'CYB561A3', 'CYB561D1', 'CYB561D2', 'CYB5A', 'CYB5B', 'CYB5D1', 'CYB5D2', 'CYB5R1', 'CYB5R2', 'CYB5R3', 'CYB5R4', 'CYB5 RL', 'CYBA', 'CYBB', 'CYBC1', 'CYBRD1', 'CYC1', 'CYCS', 'CYCSP10', 'CYCSP24', 'CYCSP29', 'CYCSP34', 'CYCSP41', 'CYCSP49', 'CYCSP52', 'CYCSP6', 'CYCPT', 'CYFIP1', 'CYFIP2', 'CYGB', 'CYHR1', 'CYLD', 'CYP11A1', 'CYP17A1', 'CYP19A1', 'CYP1A1', 'CYP1A 2', 'CYP1B1', 'CYP1B1-AS1', 'CYP1B1P', 'CYP20A1', 'CYP21A1P', 'CYP21A2', 'CYP24A1', 'CYP26A1', 'CYP26B1', 'CYP27A1', 'CYP27B 1', 'CYP27C1', 'CYP2A6', 'CYP2A7', 'CYP2A7P2', 'CYP2AB1P', 'CYP2B6P', 'CYP2B7P', 'CYP2C18', 'CYP2C9', 'CYP2D6', 'CYP2D7', 'CY P2D8P', 'CYP2E1', 'CYP2F1', 'CYP2F2P', 'CYP2G1P', 'CYP2J2', 'CYP2R1', 'CYP2S1', 'CYP2T1P', 'CYP2U1', 'CYP2W1', 'CYP39A1', 'CY P3A5', 'CYP3A7', 'CYP46A1', 'CYP46A4P', 'CYP4A22-AS1', 'CYP4B1', 'CYP4F11', 'CYP4F12', 'CYP4F2', 'CYP4F22', 'CYP4F23P', 'CY P4F26P', 'CYP4F27P', 'CYP4F29P', 'CYP4F3', 'CYP4F30P', 'CYP4F32P', 'CYP4F34P', 'CYP4F35P', 'CYP4F62P', 'CYP4F8', 'CYP4V2', 'CYP4X1', 'CYP4Z1', 'CYP4Z2P', 'CYP51A1', 'CYP7B1', 'CYR61', 'CYREN', 'CYS1', 'CYSLTR1', 'CYSLTR2', 'CYSR1', 'CYSTM1', 'CYT H1', 'CYTH2', 'CYTH3', 'CYTH4', 'CYTIP', 'CYTL1', 'CYTOR', 'CYYR1', 'CYYR1-AS1', 'D2HGDH', 'DAAM1', 'DAAM2', 'DAB1', 'DAB2', 'DAB2IP', 'DACH1', 'DACH2', 'DACT1', 'DACT2', 'DACT3', 'DAD1', 'DAG1', 'DAGLA', 'DAGLB', 'DALRD3', 'DANCR', 'DANT2', 'DAP', 'DAP3', 'DAP3P1', 'DAP3P2', 'DAPK1', 'DAPK2', 'DAPK3', 'DAPL1', 'DAPP1', 'DARS', 'DARS-AS1', 'DARS2', 'DAW1', 'DAXX', 'DAZAP 1', 'DAZAP2', 'DAZL1', 'DBF4', 'DBF4P1', 'DBH', 'DBH-AS1', 'DBI', 'DBN1', 'DBND1', 'DBND2', 'DBNL', 'DBP', 'DRB1', 'DBT', 'DBX1', 'DCAF1', 'DCAF10', 'DCAF11', 'DCAF12', 'DCAF12L1', 'DCAF13', 'DCAF15', 'DCAF16', 'DCAF17', 'DCAF 4', 'DCAF4L2', 'DCAF5', 'DCAF6', 'DCAF7', 'DCAF8', 'DCAF8L2', 'DCAKD', 'DCBLD1', 'DCBLD2', 'DCD', 'DCDC2', 'DCDC2B', 'DCHS 1', 'DCHS2', 'DCK', 'DCLK1', 'DCLK2', 'DCLRE1A', 'DCLRE1B', 'DCLRE1C', 'DCLRE1C1P', 'DCN', 'DCP1A', 'DCP1B', 'DCP2', 'DCPS', 'DCST1', 'DCST1-AS1', 'DCST2', 'DCSTAMP', 'DCT', 'DCTD', 'DCTN1', 'DCTN2', 'DCTN3', 'DCTN4', 'DCTN5', 'DCTN6', 'DCTPP1', 'DC UN1D1', 'DCUN1D2', 'DCUN1D2-AS', 'DCUN1D3', 'DCUN1D4', 'DCUN1D5', 'DCXR', 'DDA1', 'DDAH1', 'DDAH2', 'DDB1', 'DDB2', 'DCC', 'DDHD1', 'DDHD2', 'DDI2', 'DDIAS', 'DDIT3', 'DDIT4', 'DDIT4L', 'DDN', 'DDN-AS1', 'DDO', 'DDOST', 'DDR1', 'DDR2', 'DDRKG1', 'DDT', 'DDTL', 'DDTP1', 'DDX1', 'DDX10', 'DDX10P1', 'DDX11', 'DDX11-AS1', 'DDX11L2', 'DDX11L8', 'DDX12P', 'DDX17', 'DDX18', 'DDX19A', 'DDX19B', 'DDX20', 'DDX21', 'DDX23', 'DDX24', 'DDX27', 'DDX28', 'DDX31', 'DDX39A', 'DDX39B', 'DDX39BP2', 'DDX3X', 'DDX3Y', 'DDX41', 'DDX42', 'DDX43', 'DDX43P', 'DDX46', 'DDX47', 'DDX49', 'DDX5', 'DDX50', 'DDX50P1', 'DDX51', 'DDX52', 'DDX 53', 'DDX54', 'DDX55', 'DDX56', 'DDX58', 'DDX59', 'DDX6', 'DDX60', 'DDX60L', 'DEAF1', 'DECRI', 'DECRI2', 'DEDD', 'DEDD2', 'DE F6', 'DEF8', 'DEF45', 'DEFB1', 'DEFB109D', 'DEFB124', 'DEFB126', 'DEFB131B', 'DEFB131E', 'DEFB132', 'DEFB133', 'DEFB136', 'DE EG51', 'DEGS2', 'DEK', 'DELE1', 'DENND1A', 'DENND1B', 'DENND1C', 'DENND2A', 'DENND2C', 'DENND2D', 'DENND3', 'DENND4A', 'DENN D4B', 'DENND4C', 'DENND5A', 'DENND5B', 'DENND5B-AS1', 'DENND6A', 'DENR', 'DEPD1C', 'DEPD1C1', 'DEPD1C2', 'DEPD1C4', 'DEPD1C5', 'DEPD7', 'DEPPI', 'DEPTOR', 'DERA', 'DERL1', 'DERL2', 'DERL3', 'DES', 'DESI1', 'DESI2', 'DET1', 'DEUP1', 'DEXT1', 'DFFA', 'D FFB', 'DFFBP1', 'DGAT1', 'DGAT2', 'DGCR10', 'DGCR11', 'DGCR12', 'DGCR2', 'DGCR5', 'DGCR6', 'DGCR6L', 'DGCR8', 'DGCR9', 'DGK A', 'DGKB', 'DGKD', 'DGKE', 'DGKG', 'DGKH', 'DGKT', 'DGKQ', 'DGLUCY', 'DGUOK', 'DGUOK-AS1', 'DHCRC4', 'DHCRC7', 'DHDD S', 'DHDH', 'DHFR', 'DHFR2', 'DHFRP1', 'DHM', 'DHODH', 'DHP5', 'DHR51', 'DHR512', 'DHR513', 'DHR52', 'DHR53', 'DHR S4', 'DHR54-AS1', 'DHR54L1', 'DHR54L2', 'DHR57', 'DHR57B', 'DHR59', 'DHRSX', 'DHTKD1', 'DHX15', 'DHX16', 'DHX29', 'DHX30', 'DHX32', 'DHX33', 'DHX34', 'DHX35', 'DHX36', 'DHX37', 'DHX38', 'DHX40', 'DHX57', 'DHX58', 'DHX8', 'DHX9', 'DIABLO', 'DIAPH 1', 'DIAPH2', 'DIAPH2-AS1', 'DIAPH3', 'DICER1', 'DICER1-AS1', 'DID01', 'DIEXF', 'DIMT1', 'DIO1', 'DIO2', 'DIO3', 'DIO3OS', 'DIP2A', 'DIP2A-IT1', 'DIP2B', 'DIP2C', 'DIRAS1', 'DIRAS2', 'DIRAS3', 'DIRC2', 'DIRC3', 'DIRC3-AS1', 'DIS3', 'DIS3L', 'DIS3L 2', 'DISC1', 'DISC1-IT1', 'DISC1P1', 'DISP1', 'DISP2', 'DISP3', 'DIXDC1', 'DKC1', 'DKK1', 'DKK2', 'DKK3', 'DKK4', 'DKKL1', 'DLAT', 'DLC1', 'DLD', 'DLEC1', 'DLEU1', 'DLEU2', 'DLEU2L', 'DLEU7', 'DLEU7-AS1', 'DLG1', 'DLG2', 'DLG3', 'DLG3-AS1', 'D LG4', 'DLG5', 'DLG5-AS1', 'DLGAP1', 'DLGAP1-AS1', 'DLGAP1-AS2', 'DLGAP1-AS3', 'DLGAP1-AS4', 'DLGAP3', 'DLGAP4', 'DLGAP4-AS1', 'DLGAP5', 'DLK1', 'DLK2', 'DLL1', 'DLL3', 'DLL4', 'DLST1', 'DLX1', 'DLX2', 'DLX2-DT', 'DLX3', 'DLX4', 'DLX5', 'DLX 6', 'DLX6-AS1', 'DM1-AS', 'DMAC1', 'DMAC2', 'DMAP1', 'DMBT1', 'DMBX1', 'DMC1', 'DMD', 'DMGDH', 'DMKN', 'DMP1', 'DMPK', 'DMRT 1', 'DMRT2', 'DMRT3', 'DMRTA1', 'DMRTA2', 'DMTF1', 'DMTN', 'DMWD', 'DMXL1', 'DMXL2', 'DNA2', 'DNAA1', 'DNAAF2', 'DNAAF3', 'DNAAF4', 'DNAAF5', 'DNAH1', 'DNAH10S', 'DNAH11', 'DNAH14', 'DNAH17', 'DNAH17-AS1', 'DNAH2', 'DNAH3', 'DNAH5', 'DNAH6', 'DN AH7', 'DNAH8', 'DNAH9', 'DNAI1', 'DNAJA1', 'DNAJA1P5', 'DNAJA2', 'DNAJA3', 'DNAJA4', 'DNAJB1', 'DNAJB11', 'DNAJB12', 'DNAJB1 3', 'DNAJB14', 'DNAJB2', 'DNAJB4', 'DNAJB5', 'DNAJB6', 'DNAJB7', 'DNAJB9', 'DNAJC1', 'DNAJC11', 'DNAJC12', 'DNAJC 13', 'DNAJC14', 'DNAJC15', 'DNAJC16', 'DNAJC17', 'DNAJC18', 'DNAJC19', 'DNAJC19P5', 'DNAJC19P9', 'DNAJC21', 'DNAJC 22', 'DNAJC24', 'DNAJC25', 'DNAJC27', 'DNAJC28', 'DNAJC3', 'DNAJC3-AS1', 'DNAJC30', 'DNAJC4', 'DNAJC5', 'DNAJC5B', 'DNAJC5G', 'DNAJC6', 'DNAJC7', 'DNAJC8', 'DNAJC9', 'DNA1L', 'DNA14', 'DNA1L1', 'DNAE1', 'DNAE11', 'DNAE12', 'DNAE2', 'DNAE2B', 'DND1', 'DND1P1', 'DNER', 'DNH1', 'DNLZ', 'DNM1', 'DNM1L', 'DNM1P1', 'DNM1P35', 'DNM1P41', 'DNM1P46', 'DNM2', 'D NM3', 'DNM3OS', 'DNMMPB', 'DNNMBP-AS1', 'DNNMT1', 'DNNMT3', 'DNNMT3B', 'DNNPEP', 'DNNPH1', 'DNTT', 'DNTTIP1', 'DOC2A', 'DOC2B', 'DOC2GP', 'DOCK1', 'DOCK10', 'DOCK11', 'DOCK2', 'DOCK3', 'DOCK4', 'DOCK5', 'DOCK6', 'DOCK7', 'DOCK8', 'DOCK9-DT', 'DOHH', 'DOK1', 'DOK2', 'DOK3', 'DOK4', 'DOKS', 'DOK6', 'DOK7', 'DOLPK1', 'DONSON', 'DOPEY1', 'DOPEY2', 'DOT1L', 'DPA GT1', 'DPCD', 'DPEP1', 'DPF1', 'DPF2', 'DPF3', 'DPH1', 'DPH2', 'DPH3', 'DPH5', 'DPH6', 'DPH6-DT', 'DPH7', 'DPM1', 'DPM2', 'DPM3', 'DPP10', 'DPP10-AS1', 'DPP3', 'DPP4', 'DPP5', 'DPP7', 'DPP8', 'DPP9', 'DPPA2', 'DPPA3', 'DPPA4', 'DPRXP4', 'DPT', 'D PY19L1', 'DPY19L1P1', 'DPY19L1P2', 'DPY19L2', 'DPY19L2P1', 'DPY19L2P2', 'DPY19L2P3', 'DPY19L3', 'DPY19L4', 'DPY30', 'D PYD', 'DPYD-IT1', 'DPYSL2', 'DPYSL3', 'DPYSL4', 'DPYSL5', 'DQX1', 'DR1', 'DRAIC', 'DRAM1', 'DRAM2', 'DRAP1', 'DRAXIN', 'DRC3', 'DR D1', 'DRD2', 'DRD4', 'DRG1', 'DRG2', 'DRICH1', 'DROSHA', 'DRP2', 'DSC1', 'DSC2', 'DSC3', 'DSCAM', 'DSCAM-AS1', 'DSCAM1', 'D SCAS', 'DSCC1', 'DSCR10', 'DSCR4', 'DSCR4-IT1', 'DSCR8', 'DSE', 'DSEL', 'DSG1', 'DSG2', 'DSG2-AS1', 'DSG3', 'D SG4', 'DSN1', 'DSP', 'DST', 'DSTN', 'DSTNP1', 'DSTNP2', 'DSTNP4', 'DSTYK', 'DTD1', 'DTD2', 'DTHD1', 'DTL', 'DTNA', 'DTNB1', 'DTNB1P', 'DTW 1', 'DTWD2', 'DTX1', 'DTX2', 'DTX2P1', 'DTX2P1-UPK3BP1-PMS2P11', 'DTX3', 'DTX4', 'DTYMK', 'DUOB', 'DUOX1', 'DUOX2',

'DUOXA2', 'DUS1L', 'DUS2', 'DUS3L', 'DUS4L', 'DUSP1', 'DUSP10', 'DUSP11', 'DUSP12', 'DUSP13', 'DUSP14', 'DUSP15', 'DUSP16', 'DUSP18', 'DUSP19', 'DUSP2', 'DUSP22', 'DUSP23', 'DUSP26', 'DUSP27', 'DUSP28', 'DUSP3', 'DUSP4', 'DUSP5', 'DUSP5P1', 'DUSP6', 'DUSP7', 'DUSP8', 'DUSP8P3', 'DUSP8P5', 'DUSP9', 'DUT', 'DUTP2', 'DUTP4', 'DUTP6', 'DUX4L26', 'DUX4L27', 'DUX4L50', 'DUX4L9', 'DUXAP2', 'DUXAP9', 'DVL1', 'DVL2', 'DVL3', 'DXO', 'DYDC2', 'DYM', 'DYNAP', 'DYNCL1H1', 'DYNCL1I1', 'DYNCL1I2', 'DYNCL1I2P1', 'DYNCL1I1P1', 'DYNCL1I2', 'DYNCL2H1', 'DYNCL2I1', 'DYNLL1', 'DYNLL1P4', 'DYNLL2', 'DYNLRB1', 'DYNLRB2', 'DYNLT1', 'DYNLT3', 'DYRK1A', 'DYRK1B', 'DYRK2', 'DYRK3', 'DYRK4', 'DYSF', 'DZANK1', 'DZIP1', 'DZIP1L', 'DZIP3', 'E2F1', 'E2F2', 'E2F3', 'E2F3P1', 'E2F3P2', 'E2F4', 'E2F5', 'E2F6', 'E2F6P4', 'E2F7', 'E2F8', 'E4F1', 'EAF1', 'EAF2', 'EAPP', 'EARS2', 'EBAG9', 'EBAG9P1', 'EBF1', 'EBF2', 'EBF3', 'EBF4', 'EBI3', 'EBLN2', 'EBLN3P', 'EBNA1B2P', 'EBP', 'EBPL', 'ECD', 'ECE1', 'ECE2', 'ECEL1', 'ECEL1P2', 'ECH1', 'ECHDC1', 'ECHDC2', 'ECHS1', 'ECI1', 'ECI2', 'ECM1', 'ECM2', 'ECPAS', 'ECSR', 'ECSIT', 'ECT2', 'ED A', 'EDA2R', 'EDAR', 'EDARADD', 'EDC3', 'EDC4', 'EDDM13', 'EDDM3A', 'EDDM3B', 'EDEM1', 'EDEM2', 'EDEM3', 'EDF1', 'EDIL3', 'EDN1', 'EDN2', 'EDN3', 'EDNRB', 'EDRF1', 'EDRF1-AS1', 'EEA1', 'EED', 'EEF1A1', 'EEF1A1P1', 'EEF1A1P10', 'EEF1A1P11', 'EEF1A1P13', 'EEF1A1P19', 'EEF1A1P5', 'EEF1A1P6', 'EEF1A1P9', 'EEF1A2', 'EEF1AKMT1', 'EEF1AKMT2', 'EEF1B2', 'EEF1B2P3', 'EEF1D1', 'EEF1D2P1', 'EEF1D3P1', 'EEF1DP4', 'EEF1D5P1', 'EEF1E1', 'EEF1E1P1', 'EEF1G1P1', 'EEF1G1P4', 'EEF1G5P1', 'EEF2', 'EEF2KMT', 'EEFFSEC', 'EEP1D1', 'EFCAB10', 'EFCAB11', 'EFCAB13', 'EFCAB14', 'EFCAB14-AS1', 'EFCAB2', 'EFCAB7', 'EFCC1', 'EFEMP1', 'EFEMP2', 'EFHB', 'EFHC1', 'EFHF1', 'EFHD2', 'EFL1', 'EFNA1', 'EFNA2', 'EFNA3', 'EFNA4', 'EFNA5', 'EFNB1', 'EFNB2', 'EFNB3', 'EFR3A', 'EFR3B', 'EFS', 'EFTUD2', 'EGF', 'EGFEM1P', 'EGFL6', 'EGFL8', 'EGFLAM', 'EGFLAM-AS4', 'EGRF', 'EGRF-AS1', 'EGLN1', 'EGLN2', 'EGOT', 'EGR1', 'EGR2', 'EGR3', 'EGR4', 'EHPB1', 'EHPB1L1', 'EHD1', 'EHD2', 'EHD3', 'EHD4', 'EHF', 'EHHADH', 'EHMT1', 'EHMT2', 'EHMT2-AS1', 'EI24', 'EID1', 'EID2', 'EID2B', 'EID3', 'EIF1', 'EIF1A', 'EIF1AX', 'EIF1AX-AS1', 'EIF1AXP1', 'EIF1AY', 'EIF1B', 'EIF1B-AS1', 'EIF1P4', 'EIF1P6', 'EIF2', 'EIF2AK1', 'EIF2AK2', 'EIF2AK3', 'EIF2AK4', 'EIF2B1', 'EIF2B2', 'EIF2B3', 'EIF2B4', 'EIF2B5', 'EIF2D', 'EIF2S1', 'EIF2S2', 'EIF2S2P3', 'EIF2S2P4', 'EIF2S3', 'EIF2S3B', 'EIF3A', 'EIF3B', 'EIF3D', 'EIF3E', 'EIF3EP1', 'EIF3F1', 'EIF3F3P3', 'EIF3G', 'EIF3H', 'EIF3I', 'EIF3J', 'EIF3J-DT', 'EIF3K', 'EIF3L', 'EIF3M', 'EIF4A1', 'EIF4A1P12', 'EIF4A2', 'EIF4A3', 'EIF4B4P3', 'EIF4BP6', 'EIF4BP7', 'EIF4E', 'EIF4E2', 'EIF4E3', 'EIF4EBP1', 'EIF4EBP1P1', 'EIF4EBP2', 'EIF4EBP3', 'EIF4ENIF1', 'EIF4EP1', 'EIF4EP2', 'EIF4EP4', 'EIF4G1', 'EIF4G2', 'EIF4H', 'EIF4HP1', 'EIF4HP2', 'EIF5', 'EIF5A', 'EIF5A2', 'EIF5A5L1', 'EIF5A5P3', 'EIF5B', 'EIF6', 'EIPR1', 'ELAC1', 'ELAC2', 'ELAVL1', 'ELAVL2', 'ELDR', 'ELF1', 'ELF2', 'ELF3', 'ELF3-AS1', 'ELF4', 'ELF5', 'ELF6', 'ELF7', 'ELF8', 'ELF9', 'ELF10', 'ELF11', 'ELF12', 'ELF13', 'ELM01', 'ELM02', 'ELM03', 'ELMOD1', 'ELMOD2', 'ELMOD3', 'ELMSAN1', 'ELN-AS1', 'ELOA', 'ELOA-AS1', 'ELOB', 'ELOBP4', 'ELOC', 'ELOCP18', 'ELOCP19', 'ELOCP2', 'ELOCP21', 'ELOCP31', 'ELOF1', 'ELOVL1', 'ELOVL2', 'ELOVL2-AS1', 'ELOVL3', 'ELOVL4', 'ELOVL5', 'ELOVL6', 'ELOVL7', 'ELP1', 'ELP2', 'ELP3', 'ELP4', 'ELP5', 'ELP6', 'EMB', 'EMBP1', 'EMC1', 'EMC10', 'EMC2', 'EMC3', 'EMC3-AS1', 'EMC4', 'EMC6', 'EMC7', 'EMC8', 'EMC9', 'EMD', 'EME1', 'EME2', 'EMG1', 'EMID1', 'EMILIN1', 'EMILIN2', 'EMILIN3', 'EML1', 'EML2', 'EML2-AS1', 'EML3', 'EML4', 'EML5', 'EML6', 'EMP1', 'EMP2', 'EMP3', 'EMSY', 'EMX1', 'EMX2', 'EMX20S', 'EN1', 'EN2', 'ENAH', 'ENAM', 'ENC1', 'ENDOD1', 'ENDOG', 'ENDOU', 'ENDOV', 'ENG', 'ENGASE', 'ENHO', 'ENKD1', 'ENKUR', 'ENO1', 'ENO1-AS1', 'ENO1-IT1', 'ENO2', 'ENO3', 'ENOPH1', 'ENOSF1', 'ENOX1', 'ENOX1-AS2', 'ENOX2', 'ENPEP', 'ENPP1', 'ENPP2', 'ENPP3', 'ENPP4', 'ENPP5', 'ENPP6', 'ENPP7P11', 'ENPP7P2', 'ENSA', 'ENTHD1', 'ENTPD1', 'ENTPD1-AS1', 'ENTPD2', 'ENTPD3', 'ENTPD3-AS1', 'ENTPD4', 'ENTPD5', 'ENTPD6', 'ENTPD7', 'ENTPD8', 'ENY2', 'EOGT', 'EOMES', 'EP300', 'EP300-AS1', 'EP400', 'EP400P1', 'EPAS1', 'EPB41', 'EPB41L1', 'EPB41L2', 'EPB41L3', 'EPB41L4A', 'EPB41L4A-AS1', 'EPB41L4A-DT', 'EPB41L4B', 'EPB41L5', 'EPC1', 'EPC2', 'EPCAM', 'EPDR1', 'EPGS', 'EPGN', 'EPAH1', 'EPAH1-AS1', 'EPAH10', 'EPAH2', 'EPAH3', 'EPAH4', 'EPAH5', 'EPAH5-AS1', 'EPAH6', 'EPAH7', 'EPAH8', 'EPHB1', 'EPHB2', 'EPHB3', 'EPHB4', 'EPHB6', 'EPHX1', 'EPHX2', 'EPHX3', 'EPHX4', 'EPM2A', 'EPM2AIP1', 'EPN1', 'EPN2', 'EPN3', 'EPOP', 'EPOR', 'EPPK1', 'EPRS', 'EPS15L1', 'EPS15L1P1', 'EPS8L1', 'EPS8L2', 'EPS8L3', 'EPST1', 'EPYC', 'ERAL1', 'ERAP1', 'ERAP2', 'ERBB2', 'ERBB3', 'ERBB4', 'ERBIN', 'ERC1', 'ERC2', 'ERCC1', 'ERCC2', 'ERCC3', 'ERCC4', 'ERCC5', 'ERCC6', 'ERCC6L', 'ERCC6L2', 'ERCC8', 'EREG', 'ERF', 'ERG', 'ERG28', 'ERGIC1', 'ERGIC2', 'ERGIC3', 'ERH', 'ERI1', 'ERI2', 'ERI3', 'ERI3-IT1', 'ERICD', 'ERIC1', 'ERIC2', 'ERIC3', 'ERICHS', 'ERICHS-AS1', 'ERLEC1', 'ERLIN1', 'ERMAP', 'ERMRD', 'ERMN', 'ERMP1', 'ERN1', 'ERN2', 'ERO1A', 'ERO1B', 'ERP27', 'ERP29', 'ERP44', 'ERRFI1', 'ERV3-1', 'ERVE-1', 'ERVFRD-1', 'ERVH-1', 'ERVH48-1', 'ERVK-28', 'ERVK13-1', 'ERVK3-1', 'ERVK9-11', 'ERVMER34-1', 'ERVMER61-1', 'ERVV-2', 'ERWV-1', 'ESA M', 'ESCO1', 'ESCO2', 'ESD', 'ESF1', 'ESM1', 'ESPL1', 'ESPN', 'ESPNL', 'ESPNP', 'ESR1', 'ESRG', 'ESRP1', 'ESRP2', 'ESRRA', 'ESRRB', 'ESRRG', 'ESS2', 'ESX1', 'ESYT1', 'ESYT2', 'ESYT3', 'ETAA1', 'ETF1', 'ETFB', 'ETFBKMT', 'ETFDH', 'ETFRF1', 'ETHE1', 'ETNK1', 'ETNP1', 'ETS1', 'ETS2', 'ETV1', 'ETV2', 'ETV3', 'ETV4', 'ETV5', 'ETV5-AS1', 'ETV6', 'ETV7', 'EVA1A', 'EVA1B', 'EVA1C', 'EVC', 'EVC2', 'EVI2A', 'EVI2B', 'EVI15', 'EVISL', 'EVL', 'EVPL', 'EVPLL', 'EVX1', 'EVX1-AS', 'EWSAT1', 'EWSR1', 'EXD2', 'EXD3', 'EX01', 'EX05', 'EXOC1', 'EXOC1L', 'EXOC2', 'EXOC3', 'EXOC3-AS1', 'EXOC3L1', 'EXOC3L2', 'EXOC3L4', 'EXOC4', 'EXOC5', 'EXOC6', 'EXOC6B', 'EXOC7', 'EXOC8', 'EXOG', 'EXOS1', 'EXOSC10', 'EXOSC2', 'EXOSC3', 'EXOSC4', 'EXOSC5', 'EXOSC6', 'EXOSC7', 'EXOSC8', 'EXPH', 'EXT1', 'EXT2', 'EXTL1', 'EXTL2', 'EXTL3', 'EXTL3-A S1', 'EYA1', 'EYA2', 'EYA3', 'EYA4', 'EZH1', 'EZH2', 'EZR', 'EZR-AS1', 'F10', 'F11-AS1', 'F11R', 'F12', 'F13A1', 'F2', 'F2 R', 'F2RL1', 'F2RL2', 'F2RL3', 'F3', 'F5', 'F7', 'F8', 'F8A1', 'F8A3', 'FA2H', 'FAAH', 'FAAH2', 'FAAHP1', 'FAAP100', 'FAAP2 0', 'FAAP24', 'FABP1', 'FABP2', 'FABP3', 'FABP4', 'FABP5', 'FABP5P7', 'FABP6', 'FABP7', 'FABP9', 'FADD', 'FADS1', 'FADS2', 'FADS3', 'FADS6', 'FAF1', 'FAF2', 'FAF4', 'FAF1D', 'FAHD2A', 'FAHD2B', 'FAHD2C', 'FAHD2P1', 'FAIM2', 'FAIM2', 'FAM10 2A', 'FAM102B', 'FAM104A', 'FAM104B', 'FAM106A', 'FAM106CP', 'FAM106D', 'FAM107A', 'FAM107B', 'FAM108A', 'FAM110 B', 'FAM110D', 'FAM111A', 'FAM111A-DT', 'FAM111B', 'FAM114A1', 'FAM114A2', 'FAM117A', 'FAM117B', 'FAM118A', 'FAM118B', 'FAM120A', 'FAM120AOS', 'FAM120B', 'FAM120C', 'FAM122A', 'FAM122B', 'FAM122C', 'FAM124A', 'FAM124B', 'FAM126A', 'FAM126B', 'FAM12 9A', 'FAM129B', 'FAM129C', 'FAM131A', 'FAM131B', 'FAM131C', 'FAM133A', 'FAM133B', 'FAM133C', 'FAM133DP', 'FAM135A', 'FAM136 A', 'FAM13A', 'FAM13A-AS1', 'FAM13B', 'FAM13C', 'FAM149A1', 'FAM149B1', 'FAM151A', 'FAM151B', 'FAM153B', 'FAM155A', 'FAM155 B', 'FAM156A', 'FAM157A', 'FAM157C', 'FAM160A1', 'FAM160A1-DT', 'FAM160A2', 'FAM160B1', 'FAM160B2', 'FAM161A', 'FAM161B', 'FAM162A', 'FAM163A', 'FAM163B', 'FAM166B', 'FAM167A', 'FAM167B', 'FAM168A', 'FAM168B', 'FAM169A', 'FAM169B', 'FAM171A1', 'FAM171A2', 'FAM171B', 'FAM172A', 'FAM173A', 'FAM173B', 'FAM174A', 'FAM174B', 'FAM177A1', 'FAM177B', 'FAM177B', 'FAM180A', 'FAM180B', 'FAM181A', 'FAM181B', 'FAM183A', 'FAM183B', 'FAM184A', 'FAM185A', 'FAM185B', 'FAM186B', 'FAM186B', 'FAM189A1', 'FAM189A2', 'FAM189B', 'FAM192A', 'FAM193A', 'FAM193B', 'FAM196A', 'FAM196B', 'FAM198A', 'FAM198B', 'FAM198B-AS1', 'FAM199X', 'FAM199Z', 'FAM19A2', 'FAM19A3', 'FAM19A4', 'FAM19A5', 'FAM200A', 'FAM200B', 'FAM201A', 'FAM201B', 'FAM204A', 'FAM206A', 'FAM207A', 'FAM208A', 'FAM20 8B', 'FAM209B', 'FAM20A', 'FAM20B', 'FAM20C', 'FAM210A', 'FAM210B', 'FAM212A', 'FAM212B', 'FAM212B-AS1', 'FAM213A', 'FAM213B', 'FAM214A', 'FAM214B', 'FAM215A', 'FAM215B', 'FAM216A', 'FAM217B', 'FAM218A', 'FAM219A', 'FAM220A', 'FAM221A', 'FAM222A', 'FAM222A-AS1', 'FAM222B', 'FAM226B', 'FAM227A', 'FAM227B', 'FAM228A', 'FAM228B', 'FAM229A', 'FAM229B', 'FAM230C', 'FAM234A', 'FAM234B', 'FAM238B', 'FAM238C', 'FAM239A', 'FAM239B', 'FAM239C', 'FAM241A', 'FAM241B', 'FAM24B', 'FAM25A', 'FAM25C', 'FAM27AP1', 'FAM27B', 'FAM27C', 'FAM27D1', 'FAM27E2', 'FAM27E3', 'FAM32A', 'FAM35A', 'FAM35B', 'FAM35DP', 'FAM33A', 'FAM33B', 'FAM3C2', 'FAM43A', 'FAM43B', 'FAM45A', 'FAM45B', 'FAM46A', 'FAM46B', 'FAM46C', 'FAM47E', 'FAM47E-S TBD1', 'FAM49A', 'FAM49B', 'FAM50A', 'FAM50B', 'FAM53A', 'FAM53B', 'FAM53C', 'FAM57A', 'FAM57B', 'FAM66A', 'FAM66B', 'FAM66C', 'FAM69A', 'FAM69B', 'FAM69C', 'FAM71D', 'FAM71E1', 'FAM71F1', 'FAM71F2', 'FAM72A', 'FAM72B', 'FAM72C', 'FAM72D', 'FAM74A 3', 'FAM76A', 'FAM76B', 'FAM78A', 'FAM78B', 'FAM81A', 'FAM83A', 'FAM83A-AS1', 'FAM83B', 'FAM83C-AS1', 'FAM83D', 'FAM83E', 'FAM83F', 'FAM83G', 'FAM83H', 'FAM83H-AS1', 'FAM84A', 'FAM84B', 'FAM85B', 'FAM86B1', 'FAM86B2', 'FAM86B3P', 'FAM86C1', 'FAM86C2P', 'FAM86D', 'FAM86FP', 'FAM86GP', 'FAM86HP', 'FAM86JP', 'FAM86LP', 'FAM87B', 'FAM89A', 'FAM89B', 'FAM8A1', 'FAM90A1', 'FAM91A1', 'FAM92A1', 'FAM92A1P1', 'FAM95C', 'FAM96A', 'FAM96AP2', 'FAM96B', 'FAM98A', 'FAM98B', 'FAM98C', 'FAM98D', 'FAN1', 'FANCA', 'FANCB', 'FANCC', 'FANCD2', 'FANCE', 'FANCF', 'FANCI', 'FANCL', 'FANCM', 'FANK1', 'FAP', 'FAR1', 'FAR1-IT1', 'FAR2', 'FAR2P1', 'FAR2P2', 'FAR2P3', 'FAR2P4', 'FAR1P', 'FAR1P-AS1', 'FAR2P', 'FAR52', 'FARSA', 'FARSA-AS1', 'FARSB', 'FAS', 'FASLG', 'FASN', 'FASTK', 'FASTKD1', 'FASTKD2', 'FASTKD3', 'FASTKD5', 'FAT1', 'FAT2', 'FAT3', 'FAT4', 'FAT4', 'FAU', 'FAWC', 'FAXDC2', 'FBF1', 'FBF1', 'FBL1', 'FBL1M', 'FBL1N', 'FBLN1', 'FBLN2', 'FBLN7', 'FBN1', 'FBN2', 'FBN3', 'FBP1', 'FBP2', 'FBRS', 'FBRS1', 'FBRL12', 'FBXL12', 'FBXL13', 'FBXL14', 'FBXL15', 'FBXL16', 'FBXL17', 'FBXL18', 'FBXL19', 'FBXL19-AS1', 'FBXL2', 'FBXL20', 'FBXL21', 'FBXL22', 'FBXL3', 'FBXL4', 'FBXL5', 'FBXL6', 'FBXL7', 'FBXL8', 'FBX010', 'FBX011', 'FBX016',

1GMVII-55-1 , 1GMVII-40-1 , 1GMVII-51-2 , 1GMVII-60-1 , 1GMVII-75-1 , 1GMVII-77-2 , 1GHVIII-38-1' , 'IGHVIII-51-1' , 'IGHVIII-82' , 'IGIP' , 'IGKC' , 'IGKJ5' , 'IGKV1D-42' , 'IGKV1D-43' , 'IGKV1D-8' , 'IGKV10R-2' , 'IGKV10R2-108' , 'IGKV10R9-2' , 'IGKV2-29' , 'IGKV2D-40' , 'IGKV3-11' , 'IGKV3D-7' , 'IGKV30R2-268' , 'IGKV5-2' , 'IGLC2' , 'IGLC3' , 'IGLC6' , 'IGLC7' , 'IGLCOR22-1' , 'IGLJCOR18' , 'IGLL1' , 'IGLL3P' , 'IGLL4P' , 'IGLON5' , 'IGLV1-44' , 'IGLV1-51' , 'IGLV2-14' , 'IGLV2-23' , 'IGLV2-33' , 'IGLV2-8' , 'IGLV3-10' , 'IGLV5-52' , 'IGLV6-57' , 'IGLVIVOR22-1' , 'IGSF1' , 'IGSF10' , 'IGSF11' , 'IGSF21' , 'IGSF22' , 'IGSF23' , 'IGSF5' , 'IGSF8' , 'IGSF9' , 'IGSF9B' , 'IHH' , 'IKBIP' , 'IKBKB' , 'IKBKG' , 'IKZP1' , 'IKZF2' , 'IKZF3' , 'IKZF4' , 'IL10' , 'IL10RA' , 'IL10RB' , 'IL11' , 'IL11RA' , 'IL12A' , 'IL12B' , 'IL12RB1' , 'IL12RB2' , 'IL13' , 'IL13RA1' , 'IL13RA2' , 'IL15' , 'IL15RA' , 'IL16' , 'IL17B' , 'IL17D' , 'IL17F' , 'IL17RA' , 'IL17RB' , 'IL17RC' , 'IL17RD' , 'IL17RE' , 'IL18' , 'IL18BP' , 'IL18R1' , 'IL18RAP' , 'IL1A' , 'IL1B' , 'IL1R1' , 'IL1R2' , 'IL1RAP' , 'IL1RAPL1' , 'IL1RAPL2' , 'IL1R1L' , 'IL1RL2' , 'IL1RN' , 'IL20' , 'IL20RA' , 'IL20RB' , 'IL20RB-AS1' , 'IL21-AS1' , 'IL21R' , 'IL22RA1' , 'IL23A' , 'IL23R' , 'IL24' , 'IL26' , 'IL27RA' , 'IL2RA' , 'IL2RB' , 'IL2RG' , 'IL31' , 'IL31RA' , 'IL32' , 'IL33' , 'IL34' , 'IL36A' , 'IL36B' , 'IL36RN' , 'IL37' , 'IL3RA' , 'IL4' , 'IL4R' , 'IL5' , 'IL5RA' , 'IL6' , 'IL6R' , 'IL6ST' , 'IL7' , 'IL7R' , 'IL9' , 'IL9R' , 'IL9RP3' , 'ILDR1' , 'ILDR2' , 'ILF2' , 'ILF3' , 'ILF3-DT' , 'ILK' , 'ILKAP' , 'ILVBL' , 'IMMP1L' , 'IMMP2L' , 'IMMT' , 'IMP3' , 'IMP4' , 'IMPA1' , 'IMPA2' , 'IMPACT' , 'IMPAD1' , 'IMPDH1' , 'IMPDH1P4' , 'IMPDH1P5' , 'IMPDH1P8' , 'IMPDH2' , 'INA' , 'INAFM1' , 'INAFM2' , 'INAVA' , 'INCA1' , 'INCENP' , 'INE1' , 'INF2' , 'ING1' , 'ING2' , 'ING3' , 'ING4' , 'ING5' , 'INHA' , 'INHBA' , 'INHBB' , 'INHBC' , 'INHBE' , 'INIP1' , 'INIPM' , 'IN080' , 'IN080B' , 'IN080C' , 'IN080D' , 'IN080E' , 'INPP1' , 'INPP4A' , 'INPP4B' , 'INPP5A' , 'INPP5B' , 'INPP5D' , 'INPP5E' , 'INPP5F' , 'INPP5J' , 'INPP5K' , 'INPPL1' , 'INSIG1' , 'INSIG2' , 'INSL3' , 'INSL4' , 'INSL6' , 'INSM1' , 'INSR' , 'INSR' , 'INTS1' , 'INTS10' , 'INTS11' , 'INTS12' , 'INTS13' , 'INTS14' , 'INTS2' , 'INTS3' , 'INTS4' , 'INTS4P2' , 'INTSS' , 'INTS6' , 'INTS6-AS1' , 'INTS6L' , 'INTS7' , 'INTS8' , 'INTS9' , 'INTS9-AS1' , 'INTU' , 'INVS' , 'IP6K1' , 'IP6K2' , 'IP6K3' , 'IPCEF1' , 'IPMK' , 'IP011' , 'IPO13' , 'IPO4' , 'IPO5' , 'IPO5P1' , 'IPO7' , 'IPO8' , 'IPO9' , 'IPP' , 'IPPKP1' , 'IQANK1' , 'IQCAC1' , 'IQCBC1' , 'IQC' , 'IQCQD' , 'IQCCE' , 'IQCFC1' , 'IQCGB' , 'IQCCH-AS1' , 'IQCJ' , 'IQCJ-SCHIP1' , 'IQCJ-SCHIP1-AS1' , 'IQCCK' , 'IQCN' , 'IQCAGP1' , 'IQCAGP2' , 'IQCAGP3' , 'IQCSEC1' , 'IQCSEC2' , 'IQCUB' , 'IРАK1' , 'IРАK1BР1' , 'IРАK2' , 'IРАK3' , 'IРАK4' , 'IРЕB2' , 'IRF1' , 'IRF2' , 'IRF2BР1' , 'IRF2BР2' , 'IRF3' , 'IRF4' , 'IRF5' , 'IRF6' , 'IRF7' , 'IRF8' , 'IRF9' , 'IRGM' , 'IRGQ' , 'IRS1' , 'IRS2' , 'IR53P' , 'IRX1' , 'IRX2' , 'IRX3' , 'IRX4' , 'IRX5' , 'IRX6' , 'ISCA1' , 'ISCA1P1' , 'ISCA1P4' , 'ISCA2' , 'ISCA2P1' , 'ISCU' , 'ISG 15' , 'ISG20' , 'ISG20L2' , 'ISL1' , 'ISLR' , 'ISLR2' , 'ISM1' , 'ISM2' , 'ISOC1' , 'ISOC2' , 'ISPD' , 'ISPD-AS1' , 'IST1' , 'IS X' , 'ISX-AS1' , 'ISY1' , 'ISYNA1' , 'ITCH' , 'ITCH-IT1' , 'ITFG1' , 'ITFG1-AS1' , 'ITFG2' , 'ITFG2-AS1' , 'ITGA1' , 'ITGA10' , 'ITGA1 1' , 'ITGA2' , 'ITGA2B' , 'ITGA3' , 'ITGA4' , 'ITGA5' , 'ITGA6' , 'ITGA7' , 'ITGA8' , 'ITGA9' , 'ITGA9-AS1' , 'ITGAЕ' , 'ITGAЕ' , 'ITGA M' , 'ITGAВ' , 'ITGAВ1' , 'ITGB1-DT' , 'ITGB1BР1' , 'ITGB1BР2' , 'ITGB2' , 'ITGB2-AS1' , 'ITGB3' , 'ITGB3B' , 'ITGB4' , 'ITGB5' , 'ITGB6' , 'ITGB7' , 'ITGB8' , 'ITGBL1' , 'ITIH2' , 'ITIH5' , 'ITIH6' , 'ITK' , 'ITLN2' , 'ITM2A' , 'ITM2B' , 'ITM2C' , 'IT PA' , 'ITPK1' , 'ITPKA' , 'ITPKB' , 'ITPKB-AS1' , 'ITPKB-IT1' , 'ITPKC' , 'ITPR1' , 'ITPR1-DT' , 'ITPR2' , 'ITPR3' , 'ITPRIP' , 'ITPRIPL 1' , 'ITPRIPL2' , 'ITSN1' , 'ITSN2' , 'IVD' , 'IVL' , 'IVNS1ABP' , 'IWS1' , 'IYD' , 'IZUMO1' , 'IZUMO3' , 'IZUMO4' , 'JADE1' , 'JADE2' , 'JADE3' , 'JAG1' , 'JAG2' , 'JAGN1' , 'JAK1' , 'JAK2' , 'JAK3' , 'JAKMIP1' , 'JAKMIP2' , 'JAKMIP2-AS1' , 'JAKMIP3' , 'JAM2' , 'JAM3' , 'JAML' , 'JARID2' , 'JAZF1' , 'JCAD' , 'JCHAIN' , 'JD2P' , 'JHY' , 'JKAMP' , 'JMJD1C' , 'JMJD1C-AS1' , 'JMJD4' , 'JMJD6' , 'JMJD7' , 'JMJD7 -PLA2G4B' , 'JMJD8' , 'JMY' , 'JOSD1' , 'JOSD2' , 'JPH1' , 'JPH2' , 'JPH3' , 'JPH4' , 'JPT1' , 'JPT2' , 'JPX' , 'JRK' , 'JRK1' , 'JRK1-AS 1' , 'JSRP1' , 'JTB' , 'JUN' , 'JUNB' , 'JUND' , 'JUP' , 'KAAG1' , 'KALRN' , 'KAN1' , 'KAN2' , 'KAN3' , 'KAN4' , 'KANSL1' , 'KANSL1-AS 1' , 'KANSL1L' , 'KANSL2' , 'KANTR' , 'KARS' , 'KAT14' , 'KAT2A' , 'KAT2B' , 'KAT5' , 'KAT6A' , 'KAT6B' , 'KAT7' , 'KAT8' , 'KAT8' , 'KATNA1' , 'KATNAL1' , 'KATNAL2' , 'KATNB1' , 'KATNB1L' , 'KAZALD1' , 'KAZN' , 'KAZN-AS1' , 'KBTBD11' , 'KBTBD12' , 'KBTBD2' , 'KBTBD3' , 'KBTBD4' , 'KBTBD6' , 'KBTBD7' , 'KBTBD8' , 'KCMF1' , 'KCNA2' , 'KCNA3' , 'KCNA7' , 'KCNAB1' , 'KCNA2B' , 'KCNA3B' , 'KCNB1' , 'KCNB2' , 'KCNC3' , 'KCNC4' , 'KCND1' , 'KCND2' , 'KCND3' , 'KCNE3' , 'KCNE4' , 'KCNF1' , 'KCNG1' , 'KCNG3' , 'KCNG4' , 'KCNH1' , 'KCNH1-IT1' , 'KC NH2' , 'KCNH3' , 'KCNH4' , 'KCNH5' , 'KCNH7' , 'KCNH8' , 'KCNI1' , 'KCNI2' , 'KCNI3' , 'KCNT1' , 'KCNT2' , 'KCNU1' , 'KCNV2' , 'KCP' , 'KCTD1' , 'KCTD10' , 'KCTD11' , 'KCTD12' , 'KCTD13' , 'KCTD14' , 'KCTD15' , 'KCTD16' , 'KCTD17' , 'KCTD18' , 'KCTD19' , 'KCTD2' , 'KCTD2 0' , 'KCTD21' , 'KCTD21-AS1' , 'KCTD3' , 'KCTD4' , 'KCTD5' , 'KCTD6' , 'KCTD7' , 'KCTD9' , 'KDELC1' , 'KDEL2C' , 'KDELR1' , 'KDELR2' , 'K DELR3' , 'KDF1' , 'KDM1A' , 'KDM2A' , 'KDM3A' , 'KDM3B' , 'KDM4A' , 'KDM4A-AS1' , 'KDM4B' , 'KDM4C' , 'KDM4D' , 'KDM5 A' , 'KDM5B' , 'KDMSC' , 'KDMSC-IT1' , 'KDM5D' , 'KDM6A' , 'KDM6B' , 'KDM7A' , 'KDM7A-DT' , 'KDM8' , 'KDR' , 'KDR' , 'KEAP1' , 'KEL' , 'K ERA' , 'KHDC1' , 'KHDC1P1' , 'KHDC4' , 'KHDRBS1' , 'KHDKRS3' , 'KHK' , 'KHNNY' , 'KHSRP' , 'KHSRPP1' , 'KIAAA040' , 'KIAAA0100' , 'KIAAA02 32' , 'KIAAA0319' , 'KIAAA0319L' , 'KIAAA0355' , 'KIAAA0391' , 'KIAAA0513' , 'KIAAA0556' , 'KIAAA0586' , 'KIAAA0753' , 'KIAAA0825' , 'KIAAA089 5' , 'KIAAA0895L' , 'KIAAA0930' , 'KIAAA1024' , 'KIAAA1024L' , 'KIAAA1107' , 'KIAAA1109' , 'KIAAA1143' , 'KIAAA1147' , 'KIAAA1191' , 'KIAAA121 1' , 'KIAAA1211L' , 'KIAAA1217' , 'KIAAA1257' , 'KIAAA1324' , 'KIAAA1324L' , 'KIAAA1328' , 'KIAAA1468' , 'KIAAA1522' , 'KIAAA1549' , 'KIAAA1549 L' , 'KIAAA1551' , 'KIAAA1586' , 'KIAAA1614' , 'KIAAA1614-AS1' , 'KIAAA1671' , 'KIAAA1755' , 'KIAAA1841' , 'KIAAA1958' , 'KIAAA2013' , 'KIAAA202 6' , 'KIDINS220' , 'KIF11' , 'KIF12' , 'KIF13A' , 'KIF13B' , 'KIF14' , 'KIF15' , 'KIF16B' , 'KIF17' , 'KIF18A' , 'KIF18B' , 'KIF1A' , 'K F1B' , 'KIF1B' , 'KIF1C' , 'KIF20A' , 'KIF20B' , 'KIF21A' , 'KIF21B' , 'KIF22' , 'KIF23' , 'KIF24' , 'KIF25' , 'KIF25-AS1' , 'KIF26A' , 'KIF26B' , 'KIF26B-AS1' , 'KIF27' , 'KIF28P' , 'KIF2A' , 'KIF2C' , 'KIF3A' , 'KIF4A' , 'KIF5A' , 'KIF5C' , 'KIF6' , 'KIF7' , 'KIF9' , 'KIF9-AS1' , 'KIFAP3' , 'KIFC1' , 'KIFC2' , 'KIFC3' , 'KIN' , 'KTR2DЛ' , 'KIR3DЛ1' , 'KIR3DЛ2' , 'KIR3DХ1' , 'KIRREЛ1' , 'KIRREЛ1-IT1' , 'KIRREЛ2' , 'KIRREЛ3' , 'KISS1' , 'KISS1R' , 'KIT' , 'KITLG' , 'KIZ' , 'KL' , 'KLB' , 'KLC1' , 'KLC2' , 'KLC 3' , 'KLC4' , 'KLF1' , 'KLF10' , 'KLF11' , 'KLF12' , 'KLF13' , 'KLF14' , 'KLF15' , 'KLF16' , 'KLF17' , 'KLF2' , 'KLF2P1' , 'KLF2P3' , 'KLF 2P4' , 'KLF3' , 'KLF3-AS1' , 'KLF4' , 'KLF5' , 'KLF6' , 'KLF7' , 'KLF7-IT1' , 'KLF8' , 'KLF9' , 'KLHDC1' , 'KLHDC2' , 'KLHDC 3' , 'KLHDC4' , 'KLHDC7A' , 'KLHDC7B' , 'KLHDC8A' , 'KLHDC8B' , 'KLHDC9' , 'KLHL1' , 'KLHL11' , 'KLHL12' , 'KLHL13' , 'KLHL14' , 'KLHL1 5' , 'KLHL17' , 'KLHL18' , 'KLHL20' , 'KLHL21' , 'KLHL22' , 'KLHL23' , 'KLHL24' , 'KLHL25' , 'KLHL26' , 'KLHL28' , 'KLHL29' , 'KLHL2P1' , 'KLHL3' , 'KLHL30' , 'KLHL31' , 'KLHL32' , 'KLHL33' , 'KLHL35' , 'KLHL36' , 'KLHL38' , 'KLHL4' , 'KLHL41' , 'KLHL42' , 'KL HL5' , 'KLHL6' , 'KLHL6-AS1' , 'KLHL7' , 'KLHL7-DT' , 'KLHL8' , 'KLHL9' , 'KLK1' , 'KLK10' , 'KLK11' , 'KLK12' , 'KLK13' , 'KLK14' , 'KLK1 5' , 'KLK2' , 'KLK3' , 'KLK4' , 'KLK5' , 'KLK6' , 'KLK7' , 'KLK8' , 'KLK9' , 'KLKB1' , 'KLKP1' , 'KLKN1' , 'KLLA1P' , 'KLRC2' , 'KLRC3' , 'KL RLC4' , 'KLRC4-KLRK1' , 'KLRF1' , 'KLRG1' , 'KLRG2' , 'KLRK1' , 'KMO' , 'KMT2A' , 'KMT2B' , 'KMT2D' , 'KMT2E' , 'KMT2E-AS1' , 'KMT5A' , 'KMT5B' , 'KMT5C' , 'KNCN' , 'KND1C' , 'KNL1' , 'KNOP1' , 'KNOP1P4' , 'KNOP1P5' , 'KNSTRN' , 'KNTC1' , 'KPNA1' , 'KPNA2' , 'KPNA2' , 'KPNA3' , 'KPNA4' , 'KPNA5' , 'KPNA6' , 'KPNA7' , 'KPRP' , 'KPTN' , 'KRAS' , 'KRB1A' , 'KRB2A' , 'KRBOX1' , 'KRBOX4' , 'KRCC1' , 'KREMEN2' , 'KRI1' , 'KRIT1' , 'KRR1' , 'KRR1P1' , 'KRT1' , 'KRT10' , 'KRT12' , 'KRT12P7' , 'KRT13' , 'KRT14' , 'KRT15' , 'KRT1 6' , 'KRT16P2' , 'KRT16P6' , 'KRT17' , 'KRT17P2' , 'KRT18' , 'KRT18P34' , 'KRT18P5' , 'KRT18P63' , 'KRT18P65' , 'KRT18P7' , 'KRT19' , 'K RT2' , 'KRT20' , 'KRT22' , 'KRT22P' , 'KRT23' , 'KRT24' , 'KRT27' , 'KRT32' , 'KRT31' , 'KRT32' , 'KRT33B' , 'KRT34' , 'KRT35' , 'KRT3 6' , 'KRT38' , 'KRT39' , 'KRT4' , 'KRT40' , 'KRT43P' , 'KRT5' , 'KRT6A' , 'KRT6B' , 'KRT6C' , 'KRT7' , 'KRT7-AS' , 'KRT72' , 'KRT73' , 'K RT75' , 'KRT79' , 'KRT8' , 'KRT80' , 'KRT81' , 'KRT83' , 'KRT86' , 'KRT87P' , 'KRT89P' , 'KRT8P11' , 'KRT8P12' , 'KRT8P26' , 'KRT8P3' , 'K RT8P30' , 'KRT8P33' , 'KRT8P38' , 'KRT8P39' , 'KRT8P46' , 'KRT8P48' , 'KRT9' , 'KRTAP1-1' , 'KRTAP1-5' , 'KRTAP17-1' , 'KRTAP19-1' , 'K RTAP19-8' , 'KRTAP2-3' , 'KRTAP2-5P' , 'KRTAP3-1' , 'KRTAP3-2' , 'KRTAP3-3' , 'KRTAP3-4P' , 'KRTAP4-1' , 'KRTAP4-16' , 'KRTAP4-7' , 'K RTAP4-8' , 'KRTAP4-9' , 'KRTAP5-1' , 'KRTAP5-AS1' , 'KRTCAP2' , 'KRTCAP3' , 'KRTDAP' , 'KSR1' , 'KSR2' , 'KTI12' , 'KTN1' , 'KTN1-AS1' , 'KXD1' , 'KYAT1' , 'KYAT3' , 'KYNU' , 'L1CAM' , 'L1TD1' , 'L2HGDH' , 'L3MBTL1' , 'L3MBTL2' , 'L3MBTL3' , 'L3MBTL4' , 'L3MB TL4-AS1' , 'LACC1' , 'LACTB' , 'LACTB2' , 'LAD1' , 'LAG3' , 'LAGE3' , 'LAGE3P1' , 'LAIR1' , 'LAIR2' , 'LAMA1' , 'LAMA2' , 'LAMA3' , 'LAMA 4' , 'LAMA5' , 'LAMA5-AS1' , 'LAMB1' , 'LAMB2' , 'LAMB3' , 'LAMB4' , 'LAMC1' , 'LAMC1-AS1' , 'LAMC2' , 'LAMC3' , 'LAMP1' , 'LAMP2' , 'LAM P3' , 'LAMP5' , 'LAMTOR1' , 'LAMTOR2' , 'LAMTOR3' , 'LAMTOR4' , 'LAMTOR5' , 'LAMTOR5-AS1' , 'LANCL1' , 'LANCL1-AS1' , 'LANCL2' , 'LANCL 3' , 'LAP3' , 'LAPTM4A' , 'LAPTM4B' , 'LAPTM5' , 'LARGE1' , 'LARGE2' , 'LARP1' , 'LARP1B' , 'LARP1P1' , 'LARP4' , 'LARP4B' , 'LARP6' , 'L ARP7' , 'LARS' , 'LARS2' , 'LAS1L' , 'LASP1' , 'LAT' , 'LAT2' , 'LATS1' , 'LATS2' , 'LAX1' , 'LAYN' , 'LBH' , 'LBHD1' , 'LBP' , 'LBR' , 'LB X1-AS1' , 'LBX2' , 'LBX2-AS1' , 'LC45' , 'LC45L' , 'LCAT' , 'LCE1F' , 'LCE2A' , 'LCK' , 'LCLAT1' , 'LCMT1' , 'LCMT1-AS1' , 'LCMT2' , 'LCN 1' , 'LCN12' , 'LCN15' , 'LCN1P1' , 'LCN2' , 'LCOR' , 'LCP1' , 'LCP2' , 'LCT' , 'LCTL' , 'LDAH' , 'LDB1' , 'LDB2' , 'LDB3' , 'LDH A' , 'LDHAP1' , 'LDHAP4' , 'LDHAP' , 'LDHR' , 'LDHD' , 'LDHDI' , 'LDIRD1' , 'LDIRD3' , 'LDIRD4' , 'LDIRD1P' , 'LDIRD2P' , 'L ECT

~, ~L001~, ~L002~, ~L003~, ~L004~, ~L005~, ~L006~, ~L007~, ~L008~, ~L009~, ~L0010~, ~L0011~, ~L0012~, ~L0013~, ~L0014~, ~L0015~, ~L0016~, ~L0017~, ~L0018~, ~L0019~, ~L0020~, ~L0021~, ~L0022~, ~L0023~, ~L0024~, ~L0025~, ~L0026~, ~L0027~, ~L0028~, ~L0029~, ~L0030~, ~L0031~, ~L0032~, ~L0033~, ~L0034~, ~L0035~, ~L0036~, ~L0037~, ~L0038~, ~L0039~, ~L0040~, ~L0041~, ~L0042~, ~L0043~, ~L0044~, ~L0045~, ~L0046~, ~L0047~, ~L0048~, ~L0049~, ~L0050~, ~L0051~, ~L0052~, ~L0053~, ~L0054~, ~L0055~, ~L0056~, ~L0057~, ~L0058~, ~L0059~, ~L0060~, ~L0061~, ~L0062~, ~L0063~, ~L0064~, ~L0065~, ~L0066~, ~L0067~, ~L0068~, ~L0069~, ~L0070~, ~L0071~, ~L0072~, ~L0073~, ~L0074~, ~L0075~, ~L0076~, ~L0077~, ~L0078~, ~L0079~, ~L0080~, ~L0081~, ~L0082~, ~L0083~, ~L0084~, ~L0085~, ~L0086~, ~L0087~, ~L0088~, ~L0089~, ~L0090~, ~L0091~, ~L0092~, ~L0093~, ~L0094~, ~L0095~, ~L0096~, ~L0097~, ~L0098~, ~L0099~, ~L00100~, ~L00101~, ~L00102~, ~L00103~, ~L00104~, ~L00105~, ~L00106~, ~L00107~, ~L00108~, ~L00109~, ~L00110~, ~L00111~, ~L00112~, ~L00113~, ~L00114~, ~L00115~, ~L00116~, ~L00117~, ~L00118~, ~L00119~, ~L00120~, ~L00121~, ~L00122~, ~L00123~, ~L00124~, ~L00125~, ~L00126~, ~L00127~, ~L00128~, ~L00129~, ~L00130~, ~L00131~, ~L00132~, ~L00133~, ~L00134~, ~L00135~, ~L00136~, ~L00137~, ~L00138~, ~L00139~, ~L00140~, ~L00141~, ~L00142~, ~L00143~, ~L00144~, ~L00145~, ~L00146~, ~L00147~, ~L00148~, ~L00149~, ~L00150~, ~L00151~, ~L00152~, ~L00153~, ~L00154~, ~L00155~, ~L00156~, ~L00157~, ~L00158~, ~L00159~, ~L00160~, ~L00161~, ~L00162~, ~L00163~, ~L00164~, ~L00165~, ~L00166~, ~L00167~, ~L00168~, ~L00169~, ~L00170~, ~L00171~, ~L00172~, ~L00173~, ~L00174~, ~L00175~, ~L00176~, ~L00177~, ~L00178~, ~L00179~, ~L00180~, ~L00181~, ~L00182~, ~L00183~, ~L00184~, ~L00185~, ~L00186~, ~L00187~, ~L00188~, ~L00189~, ~L00190~, ~L00191~, ~L00192~, ~L00193~, ~L00194~, ~L00195~, ~L00196~, ~L00197~, ~L00198~, ~L00199~, ~L00200~, ~L00201~, ~L00202~, ~L00203~, ~L00204~, ~L00205~, ~L00206~, ~L00207~, ~L00208~, ~L00209~, ~L00210~, ~L00211~, ~L00212~, ~L00213~, ~L00214~, ~L00215~, ~L00216~, ~L00217~, ~L00218~, ~L00219~, ~L00220~, ~L00221~, ~L00222~, ~L00223~, ~L00224~, ~L00225~, ~L00226~, ~L00227~, ~L00228~, ~L00229~, ~L00230~, ~L00231~, ~L00232~, ~L00233~, ~L00234~, ~L00235~, ~L00236~, ~L00237~, ~L00238~, ~L00239~, ~L00240~, ~L00241~, ~L00242~, ~L00243~, ~L00244~, ~L00245~, ~L00246~, ~L00247~, ~L00248~, ~L00249~, ~L00250~, ~L00251~, ~L00252~, ~L00253~, ~L00254~, ~L00255~, ~L00256~, ~L00257~, ~L00258~, ~L00259~, ~L00260~, ~L00261~, ~L00262~, ~L00263~, ~L00264~, ~L00265~, ~L00266~, ~L00267~, ~L00268~, ~L00269~, ~L00270~, ~L00271~, ~L00272~, ~L00273~, ~L00274~, ~L00275~, ~L00276~, ~L00277~, ~L00278~, ~L00279~, ~L00280~, ~L00281~, ~L00282~, ~L00283~, ~L00284~, ~L00285~, ~L00286~, ~L00287~, ~L00288~, ~L00289~, ~L00290~, ~L00291~, ~L00292~, ~L00293~, ~L00294~, ~L00295~, ~L00296~, ~L00297~, ~L00298~, ~L00299~, ~L00200~, ~L00201~, ~L00202~, ~L00203~, ~L00204~, ~L00205~, ~L00206~, ~L00207~, ~L00208~, ~L00209~, ~L002010~, ~L002011~, ~L002012~, ~L002013~, ~L002014~, ~L002015~, ~L002016~, ~L002017~, ~L002018~, ~L002019~, ~L002020~, ~L002021~, ~L002022~, ~L002023~, ~L002024~, ~L002025~, ~L002026~, ~L002027~, ~L002028~, ~L002029~, ~L002030~, ~L002031~, ~L002032~, ~L002033~, ~L002034~, ~L002035~, ~L002036~, ~L002037~, ~L002038~, ~L002039~, ~L002040~, ~L002041~, ~L002042~, ~L002043~, ~L002044~, ~L002045~, ~L002046~, ~L002047~, ~L002048~, ~L002049~, ~L002050~, ~L002051~, ~L002052~, ~L002053~, ~L002054~, ~L002055~, ~L002056~, ~L002057~, ~L002058~, ~L002059~, ~L002060~, ~L002061~, ~L002062~, ~L002063~, ~L002064~, ~L002065~, ~L002066~, ~L002067~, ~L002068~, ~L002069~, ~L002070~, ~L002071~, ~L002072~, ~L002073~, ~L002074~, ~L002075~, ~L002076~, ~L002077~, ~L002078~, ~L002079~, ~L002080~, ~L002081~, ~L002082~, ~L002083~, ~L002084~, ~L002085~, ~L002086~, ~L002087~, ~L002088~, ~L002089~, ~L002090~, ~L002091~, ~L002092~, ~L002093~, ~L002094~, ~L002095~, ~L002096~, ~L002097~, ~L002098~, ~L002099~, ~L0020100~, ~L0020101~, ~L0020102~, ~L0020103~, ~L0020104~, ~L0020105~, ~L0020106~, ~L0020107~, ~L0020108~, ~L0020109~, ~L0020110~, ~L0020111~, ~L0020112~, ~L0020113~, ~L0020114~, ~L0020115~, ~L0020116~, ~L0020117~, ~L0020118~, ~L0020119~, ~L0020120~, ~L0020121~, ~L0020122~, ~L0020123~, ~L0020124~, ~L0020125~, ~L0020126~, ~L0020127~, ~L0020128~, ~L0020129~, ~L0020130~, ~L0020131~, ~L0020132~, ~L0020133~, ~L0020134~, ~L0020135~, ~L0020136~, ~L0020137~, ~L0020138~, ~L0020139~, ~L0020140~, ~L0020141~, ~L0020142~, ~L0020143~, ~L0020144~, ~L0020145~, ~L0020146~, ~L0020147~, ~L0020148~, ~L0020149~, ~L0020150~, ~L0020151~, ~L0020152~, ~L0020153~, ~L0020154~, ~L0020155~, ~L0020156~, ~L0020157~, ~L0020158~, ~L0020159~, ~L0020160~, ~L0020161~, ~L0020162~, ~L0020163~, ~L0020164~, ~L0020165~, ~L0020166~, ~L0020167~, ~L0020168~, ~L0020169~, ~L0020170~, ~L0020171~, ~L0020172~, ~L0020173~, ~L0020174~, ~L0020175~, ~L0020176~, ~L0020177~, ~L0020178~, ~L0020179~, ~L0020180~, ~L0020181~, ~L0020182~, ~L0020183~, ~L0020184~, ~L0020185~, ~L0020186~, ~L0020187~, ~L0020188~, ~L0020189~, ~L0020190~, ~L0020191~, ~L0020192~, ~L0020193~, ~L0020194~, ~L0020195~, ~L0020196~, ~L0020197~, ~L0020198~, ~L0020199~, ~L00201100~, ~L00201110~, ~L00201120~, ~L00201130~, ~L00201140~, ~L00201150~, ~L00201160~, ~L00201170~, ~L00201180~, ~L00201190~, ~L00201200~, ~L00201210~, ~L00201220~, ~L00201230~, ~L00201240~, ~L00201250~, ~L00201260~, ~L00201270~, ~L00201280~, ~L00201290~, ~L00201300~, ~L00201310~, ~L00201320~, ~L00201330~, ~L00201340~, ~L00201350~, ~L00201360~, ~L00201370~, ~L00201380~, ~L00201390~, ~L00201400~, ~L00201410~, ~L00201420~, ~L00201430~, ~L00201440~, ~L00201450~, ~L00201460~, ~L00201470~, ~L00201480~, ~L00201490~, ~L00201500~, ~L00201510~, ~L00201520~, ~L00201530~, ~L00201540~, ~L00201550~, ~L00201560~, ~L00201570~, ~L00201580~, ~L00201590~, ~L00201600~, ~L00201610~, ~L00201620~, ~L00201630~, ~L00201640~, ~L00201650~, ~L00201660~, ~L00201670~, ~L00201680~, ~L00201690~, ~L00201700~, ~L00201710~, ~L00201720~, ~L00201730~, ~L00201740~, ~L00201750~, ~L00201760~, ~L00201770~, ~L00201780~, ~L00201790~, ~L00201800~, ~L00201810~, ~L00201820~, ~L00201830~, ~L00201840~, ~L00201850~, ~L00201860~, ~L00201870~, ~L00201880~, ~L00201890~, ~L00201900~, ~L00201910~, ~L00201920~, ~L00201930~, ~L00201940~, ~L00201950~, ~L00201960~, ~L00201970~, ~L00201980~, ~L00201990~, ~L00202000~, ~L00202010~, ~L00202020~, ~L00202030~, ~L00202040~, ~L00202050~, ~L00202060~, ~L00202070~, ~L00202080~, ~L00202090~, ~L00202100~, ~L00202104~, ~L00202105~, ~L00202106~, ~L00202108~, ~L00202109~, ~L00202110~, ~L00202119~, ~L00202121~, ~L00202128~, ~L00202130~, ~L00202132~, ~L00202152~, ~L00202154~, ~L00202156~, ~L00202158~, ~L00202160~, ~L00202163~, ~L00202166~, ~L00202167~, ~L00202170~, ~L00202175~, ~L00202180~, ~L00202188~, ~L00202193~, ~L00202195~, ~L00202197~, ~L00202198~, ~L00202199~, ~L00202201~, ~L00202206~, ~L00202210~, ~L00202211~, ~L00202212~, ~L00202220~, ~L00202223~, ~L00202225~, ~L00202228~, ~L00202241~, ~L00202243~, ~L00202246~, ~L00202251~, ~L00202253~, ~L00202255~, ~L00202257~, ~L00202262~, ~L00202263~, ~L00202265~

0', 'MIR141', 'MIR143', 'MIR146B', 'MIR148A', 'MIR148B', 'MIR149', 'MIR151A', 'MIR151B', 'MIR153-1', 'MIR153-2', 'MIR1538', 'MIR155HG', 'MIR16-2', 'MIR17HG', 'MIR181A2', 'MIR181A2', 'MIR181B1', 'MIR181B2', 'MIR181D', 'MIR182T', 'MIR186', 'MIR1913', 'MIR1915', 'MIR193A', 'MIR193B', 'MIR193BHG', 'MIR194-1', 'MIR194-2HG', 'MIR196A1', 'MIR196A2', 'MIR1972-1', 'MIR1972-2', 'MIR199A1', 'MIR199A2', 'MIR200A', 'MIR200B', 'MIR200C', 'MIR200CHG', 'MIR203A', 'MIR204', 'MIR205', 'MIR2052HG', 'MIR205HG', 'MIR206', 'MIR210', 'MIR210HG', 'MIR211', 'MIR2116', 'MIR215', 'MIR217HG', 'MIR218-2', 'MIR219A1', 'MIR221', 'MIR222', 'MIR222HG', 'MIR227', 'MIR227B', 'MIR235', 'MIR23A', 'MIR23B', 'MIR23C', 'MIR24-2', 'MIR2467', 'MIR25', 'MIR26A2', 'MIR26B', 'MIR27A', 'MIR27B', 'MIR2909', 'MIR298', 'MIR29A', 'MIR29B2CHG', 'MIR302A', 'MIR302B', 'MIR302C', 'MIR302D', 'MIR302C', 'MIR312', 'MIR3121', 'MIR3122', 'MIR3124', 'MIR3125', 'MIR3126', 'MIR3127', 'MIR3128', 'MIR3129', 'MIR3131', 'MIR3132', 'MIR3133', 'MIR3134', 'MIR3135A', 'MIR3136', 'MIR3138', 'MIR3142', 'MIR3142HG', 'MIR3143', 'MIR3144', 'MIR3145', 'MIR3146', 'MIR3147', 'MIR3149', 'MIR3153', 'MIR3155A', 'MIR3157', 'MIR3161', 'MIR3164', 'MIR3173', 'MIR3174', 'MIR3176', 'MIR3177', 'MIR3178', 'MIR3181', 'MIR3183', 'MIR3185', 'MIR3186', 'MIR3187', 'MIR3188', 'MIR3189', 'MIR3192', 'MIR3193', 'MIR3195', 'MIR3196', 'MIR3197', 'MIR3198-1', 'MIR319H1', 'MIR32', 'MIR3202-1', 'MIR320A', 'MIR320B2', 'MIR320C1', 'MIR320D1', 'MIR320E', 'MIR324', 'MIR325HG', 'MIR326', 'MIR328', 'MIR331', 'MIR339', 'MIR33B', 'MIR342', 'MIR345', 'MIR34A', 'MIR34AHG', 'MIR34C', 'MIR3609', 'MIR361', 'MIR3613', 'MIR3616', 'MIR3619', 'MIR363', 'MIR3646', 'MIR3649', 'MIR3651', 'MIR3653', 'MIR3657', 'MIR365A', 'MIR3661', 'MIR3664', 'MIR3667', 'MIR3668', 'MIR3671', 'MIR3677', 'MIR3679', 'MIR3680-1', 'MIR3680-2', 'MIR3681HG', 'MIR3682', 'MIR3684', 'MIR3685', 'MIR3689E', 'MIR3689F', 'MIR3690', 'MIR3691', 'MIR3692', 'MIR374B', 'MIR378A', 'MIR378G', 'MIR378H', 'MIR378J', 'MIR3907', 'MIR3918', 'MIR3919', 'MIR3920', 'MIR3929', 'MIR3936', 'MIR3936HG', 'MIR3939', 'MIR3941', 'MIR3942', 'MIR3944', 'MIR3945', 'MIR3945HG', 'MIR3972', 'MIR421', 'MIR422A', 'MIR425', 'MIR4252', 'MIR4253', 'MIR4254', 'MIR4257', 'MIR4258', 'MIR4260', 'MIR4261', 'MIR4263', 'MIR4271', 'MIR4284', 'MIR4285', 'MIR429', 'MIR4292', 'MIR4293', 'MIR4296', 'MIR4297', 'MIR4298', 'MIR4299', 'MIR4301', 'MIR4304', 'MIR4312', 'MIR4316', 'MIR4322', 'MIR4324', 'MIR4326', 'MIR4419A', 'MIR4420', 'MIR4423', 'MIR4425', 'MIR4427', 'MIR4428', 'MIR4431', 'MIR4435-1', 'MIR4435-2', 'MIR4435-2HG', 'MIR4441', 'MIR4442', 'MIR4446', 'MIR4449', 'MIR4450', 'MIR4451', 'MIR4452', 'MIR4453HG', 'MIR4456', 'MIR4457', 'MIR4458HG', 'MIR4461', 'MIR4465', 'MIR4469', 'MIR4470', 'MIR4477A', 'MIR4477B', 'MIR4478', 'MIR4479', 'MIR4480', 'MIR4482', 'MIR4484', 'MIR4489', 'MIR4497', 'MIR4500', 'MIR4500HG', 'MIR4502', 'MIR4506', 'MIR4512', 'MIR4513', 'MIR4514', 'MIR4518', 'MIR4519', 'MIR4520-1', 'MIR4526', 'MIR4527HG', 'MIR4530', 'MIR4537', 'MIR4538', 'MIR4539', 'MIR4545', 'MIR4633', 'MIR4634', 'MIR4645', 'MIR4646', 'MIR4649', 'MIR4653', 'MIR4655', 'MIR4659A', 'MIR4664', 'MIR4665', 'MIR4666A', 'MIR4666B', 'MIR4668', 'MIR4669', 'MIR4671', 'MIR4672', 'MIR4673', 'MIR4677', 'MIR4682', 'MIR4685', 'MIR4686', 'MIR4689', 'MIR4693', 'MIR4697HG', 'MIR4698', 'MIR4701', 'MIR4706', 'MIR4710', 'MIR4713HG', 'MIR4714', 'MIR4716', 'MIR4721', 'MIR4728', 'MIR4730', 'MIR4731', 'MIR4736', 'MIR4737', 'MIR4740', 'MIR4744', 'MIR4746', 'MIR4747', 'MIR4748', 'MIR4753', 'MIR4754', 'MIR4755', 'MIR4757', 'MIR4759', 'MIR4760', 'MIR4761', 'MIR4762', 'MIR4766', 'MIR4767', 'MIR4768', 'MIR4771-2', 'MIR4777', 'MIR4779', 'MIR4782', 'MIR4783', 'MIR4786', 'MIR4787', 'MIR4789', 'MIR4793', 'MIR4797', 'MIR4802', 'MIR486-1', 'MIR491', 'MIR4999', 'MIR499A', 'MIR5008', 'MIR5009', 'MIR5010', 'MIR503', 'MIR503HG', 'MIR505', 'MIR5091', 'MIR5092', 'MIR5093', 'MIR5094', 'MIR5187', 'MIR5188', 'MIR5190', 'MIR5191', 'MIR5192', 'MIR5194', 'MIR5197', 'MIR544B', 'MIR545', 'MIR548A1', 'MIR548A1', 'MIR548AC', 'MIR548AM', 'MIR548AN', 'MIR548AQ', 'MIR548AR', 'MIR548AT', 'MIR548AV', 'MIR548E', 'MIR548H3', 'MIR548I2', 'MIR548K', 'MIR548L', 'MIR548N', 'MIR548P', 'MIR548Q', 'MIR548T', 'MIR548U', 'MIR548V', 'MIR548XHG', 'MIR550A2', 'MIR550A3', 'MIR551A', 'MIR553', 'MIR554', 'MIR556', 'MIR557', 'MIR5579', 'MIR5581', 'MIR5582', 'MIR5585', 'MIR5587', 'MIR559', 'MIR561', 'MIR563', 'MIR566', 'MIR567', 'MIR568', 'MIR5684', 'MIR5685', 'MIR5689HG', 'MIR569', 'MIR5690', 'MIR5692A1', 'MIR5692B', 'MIR5692C1', 'MIR5692C2', 'MIR5693', 'MIR5697', 'MIR5699', 'MIR570', 'MIR5702', 'MIR570H', 'MIR573', 'MIR574', 'MIR575', 'MIR576', 'MIR577', 'MIR579', 'MIR580', 'MIR583', 'MIR583H', 'MIR588', 'MIR589', 'MIR590', 'MIR593', 'MIR595', 'MIR596', 'MIR599', 'MIR600H', 'MIR601', 'MIR604', 'MIR6069', 'MIR6071', 'MIR6075', 'MIR6076', 'MIR6077', 'MIR6078', 'MIR608', 'MIR6080', 'MIR6081', 'MIR6087', 'MIR6088', 'MIR6090', 'MIR6124', 'MIR613', 'MIR615', 'MIR616', 'MIR619', 'MIR620', 'MIR621', 'MIR623', 'MIR624', 'MIR627', 'MIR629', 'MIR631', 'MIR635', 'MIR640', 'MIR641', 'MIR642A', 'MIR643', 'MIR644A', 'MIR645', 'MIR646', 'MIR646HG', 'MIR647', 'MIR648', 'MIR649', 'MIR6504', 'MIR6505', 'MIR6509', 'MIR6510', 'MIR6512', 'MIR6514', 'MIR656', 'MIR657', 'MIR659', 'MIR661', 'MIR662', 'MIR664A', 'MIR670', 'MIR670HG', 'MIR6718', 'MIR6719', 'MIR6721', 'MIR6723', 'MIR6728', 'MIR6730', 'MIR6731', 'MIR6736', 'MIR6738', 'MIR6739', 'MIR6740', 'MIR6746', 'MIR6747', 'MIR6750', 'MIR6753', 'MIR6754', 'MIR6757', 'MIR6762', 'MIR6763', 'MIR6766', 'MIR6768', 'MIR6772', 'MIR6775', 'MIR6776', 'MIR6779', 'MIR6780B', 'MIR6781', 'MIR6782', 'MIR6784', 'MIR6784', 'MIR6793', 'MIR6796', 'MIR6797', 'MIR6801', 'MIR6804', 'MIR6806', 'MIR6810', 'MIR6812', 'MIR6813', 'MIR6814', 'MIR6817', 'MIR6819', 'MIR6826', 'MIR6827', 'MIR6829', 'MIR6832', 'MIR6835', 'MIR6838', 'MIR6839', 'MIR6842', 'MIR6851', 'MIR6855', 'MIR6856', 'MIR6859-1', 'MIR6864', 'MIR6867', 'MIR6870', 'MIR6877', 'MIR6880', 'MIR6881', 'MIR6888', 'MIR6889', 'MIR6895', 'MIR7-3HG', 'MIR708', 'MIR7106', 'MIR711', 'MIR7111', 'MIR7152', 'MIR7156', 'MIR7158', 'MIR7161', 'MIR718', 'MIR760', 'MIR762', 'MIR762HG', 'MIR765', 'MIR766', 'MIR767', 'MIR7845', 'MIR7847', 'MIR7848', 'MIR7851', 'MIR7854', 'MIR7856', 'MIR7857', 'MIR7974', 'MIR7976', 'MIR8055', 'MIR8058', 'MIR8070', 'MIR8071-2', 'MIR8078', 'MIR8081', 'MIR8083', 'MIR8087', 'MIR8089', 'MIR887', 'MIR9-3HG', 'MIR921', 'MIR92B', 'MIR93', 'MIR933', 'MIR936', 'MIR938', 'MIR941-1', 'MIR941-4', 'MIR941-5', 'MIR942', 'MIR95', 'MIR98', 'MIR99AHG', 'MIR99B', 'MIRLET7A1', 'MIRLET7A2', 'MIRLET7BHG', 'MIRLET7C', 'MIRLET7D', 'MIRLET7F1', 'MIRLET7G', 'MIRLET7I', 'MIS12', 'MIS18A', 'MIS18A-AS1', 'MIS18B1', 'MISP1', 'MISP3', 'MITD1', 'MITF', 'MIXL1', 'MKI67', 'MKKS', 'MKL1', 'MKL2', 'MKLN1', 'MKLN1-AS', 'MKNK1', 'MKNK1', 'MKNK2', 'MKNR1', 'MKNR2', 'MKNR2OS', 'MKNR3', 'MKNR3-AS1', 'MKNR4P', 'MKNR9P', 'MKS1', 'MKX', 'MLANA', 'MLC1', 'MLEC', 'MLF1', 'MLF2', 'MLH1', 'MLH3', 'MLIP', 'MLLT1', 'MLLT10', 'MLLT10P1', 'MLLT11', 'MLLT3', 'MLLT6', 'MLN', 'MLPH', 'MLST8', 'MLX', 'MLXIP', 'MLXIP1', 'MLYCD', 'MMAA', 'MMAB', 'MMACHC', 'MMADHC', 'MMD', 'MME', 'MME-AS1', 'MMEL1', 'MMGT1', 'MMP1', 'MMP10', 'MMP11', 'MMP12', 'MMP13', 'MMP14', 'MMP15', 'MMP16', 'MMP17', 'MMP19', 'MMP2', 'MMP20', 'MMP23B', 'MMP24', 'MMP240S', 'MMP25', 'MMP25-AS1', 'MMP28', 'MMP3', 'MMP7', 'MMP8', 'MMP9', 'MMPRN2', 'MMS19', 'MMS22L', 'MN1', 'MNA1', 'MND1', 'MNS1', 'MNT', 'MNX1', 'MNX1-AS1', 'MNX1-AS2', 'MOAP1', 'MOB1A', 'MOB1B', 'MOB2', 'MOB3A', 'MOB3C', 'MOB4', 'MOCOS', 'MOC51', 'MOC52', 'MOC53', 'MOG', 'MOGAT2', 'MOGAT3', 'MOGS', 'MOK', 'MON1A', 'MON1B', 'MON2', 'MORC1', 'MORC2', 'MORC3', 'MORC4', 'MORF4L1', 'MORF4L2', 'MORF4L2-AS1', 'MORN1', 'MORN2', 'MORN3', 'MORN4', 'MOSMO', 'MOSPD1', 'MOSPD2', 'MOSPD3', 'MOV10', 'MOV10L1', 'MOXD1', 'MPC1', 'MP2', 'MPDU1', 'MPDZ1', 'MPG', 'MPHOSPH10', 'MPHOSPH6', 'MPHOSPH8', 'MPHOSPH9', 'MPI', 'MPLKIP', 'MPND', 'MPO', 'PP1', 'MPP2', 'MPP3', 'MPP4', 'MPP5', 'MPP6', 'MPP7', 'MPPE1', 'MPPED2', 'MPRIP', 'MPRIP1', 'MPST', 'MPTX1', 'MPV17', 'MPV17L', 'MPV17L2', 'MPZ', 'MPZL1', 'MPZL2', 'MPZL3', 'MR1', 'MRAP2', 'MRAS', 'MRC1', 'MRC2', 'MRE11', 'MREG', 'MREGP1', 'MRFAP1', 'MRFAP1L1', 'MRGBP', 'MRGPRF', 'MRGPRX3', 'MRGPRX4', 'MRI1', 'MRLN1', 'MRM1', 'MRM2', 'MRM3', 'MRNIP', 'MRO', 'MROH3P', 'MR0H6', 'MRHOB8', 'MRPL11', 'MRPL10', 'MRPL11', 'MRPL12', 'MRPL13', 'MRPL14', 'MRPL15', 'MRPL15', 'MRPL16', 'MRPL17', 'MRPL18', 'MRPL19', 'MRPL2', 'MRPL20', 'MRPL20P1', 'MRPL21', 'MRPL22', 'MRPL23', 'MRPL23-AS1', 'MRPL24', 'MRPL27', 'MRPL28', 'MRPL3', 'MRPL30', 'MRPL32', 'MRPL33', 'MRPL34', 'MRPL35', 'MRPL35P2', 'MRPL35P3', 'MRPL36', 'MRPL37', 'MRPL38', 'MRPL39', 'MRPL4', 'MRPL40', 'MRPL41', 'MRPL42', 'MRPL42P6', 'MRPL43', 'MRPL44', 'MRPL45', 'MRPL45P1', 'MRPL46', 'MRPL47', 'MRPL48', 'MRPL48P1', 'MRPL49', 'MRPL50', 'MRPL51', 'MRPL52', 'MRPL53', 'MRPL53P1', 'MRPL54', 'MRPL55', 'MRPL57', 'MRPL57P3', 'MRPL58', 'MRPL9', 'MRPS10', 'MRPS11', 'MRPS12', 'MRPS14', 'MRPS15', 'MRPS16', 'MRPS17', 'MRPS18A', 'MRPS18AP1', 'MRPS18B', 'MRPS18C', 'MRPS2', 'MRPS21', 'MRPS22', 'MRPS23', 'MRPS24', 'MRPS25', 'MRPS26', 'MRPS27', 'MRPS28', 'MRPS30', 'MRPS30-DT', 'MRPS31', 'MRPS31P4', 'MRPS31P5', 'MRPS33', 'MRPS33P4', 'MRPS34', 'MRPS35', 'MRPS36', 'MRPS36P5', 'MRPS5', 'MRPS6', 'MRPS7', 'MRPS9', 'MRRF', 'MRSS2', 'MRT04', 'MRV11', 'MSAA1', 'MSAA3', 'MS4AA4', 'MS4AA7', 'MS4AB', 'MSANTD2', 'MSANTD3', 'MSANTD4', 'MSANTD5', 'MSC-AS1', 'MSH2', 'MSH3', 'MSH4', 'MSHS', 'MSHS-SAPCD1', 'MSH6', 'MSI1', 'MSI2', 'MSL1', 'MSL2', 'MSL3', 'MSL3P1', 'MSLN', 'MSMB', 'MSMO1', 'MSMP', 'MSN', 'MSR1', 'MSRA', 'MSRB1', 'MSRB2', 'MSRB3', 'MSS51', 'MST1', 'MST1L', 'MST1P2', 'MST1R', 'MSTO1', 'MSTO2P', 'MSX1', 'MSX2', 'MSX2P1', 'MT-ATP6', 'MT-C01', 'MT-C02', 'MT-C03', 'MT-CYB', 'MT-ND1', 'MT-ND2', 'MT-ND3', 'MT-ND4', 'MT-ND4L', 'MT-ND5', 'MT-ND6', 'MT-RNR1', 'MT-TA', 'MT-TA', 'MT-TD', 'MT-TE', 'MT-TE', 'MT-TF', 'MT-TI', 'MT-TM', 'MT-TN', 'MT-TP', 'MT-TQ', 'MT-TS1', 'MT-TT', 'MT-TW', 'MT-TY', 'MT1A', 'MT1DP', 'MT1E', 'MT1F', 'MT1G', 'MT1H', 'MT1JP', 'MT1L', 'MT1M', 'MT1P1', 'MT1X', 'MT1XP1', 'MT2A', 'MT3', 'MT4', 'MTA1', 'MTA2', 'MTA3', 'MTAP', 'MTATP6P1', 'MTATP6P16', 'MTATP6P27', 'MTATP8P1', 'MTBD', 'MTCH1', 'MTCH2', 'MTCL1', 'MTC01P1', 'MTC01P12', 'MTC

ARA', 'RARA-AS1', 'RARB', 'RARG', 'RARRES1', 'RARRES2', 'RARRES2P10', 'RARRES2P6', 'RARRES2P8', 'RARRES3', 'RARS', 'RARS2', 'RASA1', 'RASA2', 'RASA3', 'RASA3-IT1', 'RASA4', 'RASA4CP', 'RASAL1', 'RASAL2', 'RASAL2-AS1', 'RASAL3', 'RASD1', 'RASD2', 'RASEF', 'RASGEF1A', 'RASGEF1B', 'RASGEF1C', 'RASGRF1', 'RASGRF2', 'RASGRF2-AS1', 'RASGRP1', 'RASGRP2', 'RASGRP3', 'RASGRP4', 'RASIP1', 'RASL10A', 'RASL10B', 'RASL11A', 'RASL11B', 'RASL12', 'RASSF1', 'RASSF1-AS1', 'RASSF10', 'RASSF2', 'RASSF3', 'RASSF4', 'RASSFS5', 'RASSF6', 'RASSF7', 'RASSF8', 'RASSFF8-AS1', 'RASSF9', 'RAVER1', 'RAVER2', 'RB1', 'RB1CC1', 'RBAK', 'RBAKDN', 'RBBDP4', 'RBBDP5', 'RBBDP6', 'RBBDP7', 'RBBDP8', 'RBBDP8NL', 'RBBDP9', 'RBCK1', 'RBFA', 'RBFADN', 'RBFOX2', 'RBKS', 'RBL1', 'RBL2', 'RBML0', 'RBML1', 'RBML2', 'RBML2B', 'RBML2B-AS1', 'RBML4', 'RBML4-BM4', 'RBML5', 'RBML5B', 'RBML7', 'RBML7P4', 'RBML8', 'RBML9', 'RBML10', 'RBML20', 'RBML22', 'RBML22P3', 'RBML23', 'RBML24', 'RBML25', 'RBML26', 'RBML26-AS1', 'RBML27', 'RBML28', 'RBML3', 'RBML34', 'RBML38', 'RBML39', 'RBML4', 'RBML41', 'RBML42', 'RBML43', 'RBML45', 'RBML47', 'RBML48', 'RBML48P1', 'RBML49', 'RBML5', 'RBML5-AS1', 'RBML6', 'RBML7', 'RBML8A', 'RBMS1', 'RBMS1P1', 'RBMS2', 'RBMS3', 'RBMS3-AS2', 'RBMS3-AS3', 'RBMX', 'RBMX2', 'RBMLX1', 'RBMY2QP', 'RBP1', 'RBP4', 'RBPS5', 'RBPS7', 'RBPSJ', 'RBPM5', 'RBPM5-AS1', 'RBPM52', 'RBNS', 'RBX1', 'RC3H1', 'RC3H1-IT1', 'RC3H2', 'RCAN1', 'RCAN2', 'RCAN3', 'RCBTB1', 'RCBTB2', 'RCC1', 'RCC1L', 'RCC2', 'RCC2P6', 'RCCD1', 'RCE1', 'RCHY1', 'RCL1', 'RCN1', 'RCN1P2', 'RCN2', 'RCN3', 'RCOR1', 'RCOR2', 'RCOR3', 'RCDS1', 'RDH10', 'RDH10-AS1', 'RDH11', 'RDH12', 'RDH13', 'RDH14', 'RDH16', 'RDH5', 'RDH8', 'RDM1', 'RDX', 'REC114', 'RECK', 'RECQL', 'RECQL4', 'RECQL5', 'REEP1', 'REEP2', 'REEP3', 'REEP4', 'REEP5', 'REEP6', 'REG1A', 'REG4', 'REL', 'RELA', 'RELB', 'RELL1', 'RELL2', 'RELN', 'RELT', 'REM1', 'REM2', 'REN', 'RENBP', 'REP15', 'REPIN1', 'REPS1', 'REPS2', 'RER1', 'RERE', 'RERG', 'RERG-AS1', 'RERG-IT1', 'RERGL', 'REST', 'RET', 'R ETNLB', 'RETREG1', 'RETREG2', 'RETREG3', 'RETSAT', 'REV1', 'REV3L', 'REX1BD', 'REX01', 'REX02', 'REX04', 'REX05', 'RFC1', 'RFC2', 'RFC3', 'RFC4', 'RFC5', 'RFCSD', 'RFFL', 'RFK', 'RFLNA', 'RFLNB', 'RFNG', 'RFPL1S', 'RFPL2', 'RFPL3S', 'RFPL4A', 'RFPL4B', 'RFT1', 'RFTN1', 'RFTN2', 'RFWD3', 'RFX1', 'RFX2', 'RFX3', 'RFX3-AS1', 'RFX5', 'RFX6', 'RFX7', 'RFX8', 'RFXANK', 'RFXAP', 'RGCC', 'RGL1', 'RGL2', 'RGL3', 'RGL4', 'RGMA', 'RGMB', 'RGMB-AS1', 'RGN', 'RGP1', 'RGPD8', 'RGSI', 'RGS10', 'RGS11', 'R GS12', 'RGS13', 'RGS14', 'RGS15', 'RGS17', 'RGS17P1', 'RGS18', 'RGS19', 'RGS20', 'RGS22', 'RGS3', 'RGS4', 'RGS5', 'R GS6', 'RGS7', 'RGS7BP', 'RGS9', 'RGS9BP', 'RHAG', 'RBHD2', 'RHBD2', 'RHBD3', 'RHBDF1', 'RHBDF2', 'RHBDL1', 'RHBDL2', 'RH DL3', 'RHBG', 'RHCE', 'RHCG', 'RHD', 'RHEB', 'RHEBL1', 'RHEBP1', 'RHEBP2', 'RHEX', 'RHNO1', 'RHOA', 'RHOA-IT1', 'RHOB', 'RHO BTB1', 'RHOBTB2', 'RHOBTB3', 'RHOC', 'RHOD', 'RHOF', 'RHOG', 'RHOH', 'RH01', 'RHQO1', 'RHQOP1', 'RHQOP3', 'RHOT1', 'RHOT1P1', 'RHOT1P2', 'RHOT1P3', 'RHOT2', 'RHOU', 'RHOX1P1', 'RHOXF2', 'RHPN1', 'RHPN1-AS1', 'RHPN2', 'RIBC1', 'RIB C2', 'RIC1', 'RIC3', 'RIC8A', 'RIC8B', 'RICTOR', 'RIDA', 'RIF1', 'RILP1', 'RILP1L', 'RILP2', 'RIMKLA', 'RIMKLB', 'RIMKLB2P', 'RIMS1', 'RIMS2', 'RIMS3', 'RIMS4', 'RIN1', 'RIN2', 'RIN3', 'RING1', 'RINL', 'RINT1', 'RIOK1', 'RIOK2', 'RIOK3', 'RIOX1', 'RIOX2', 'RIPK1', 'RIPK2', 'RIPK3', 'RIPK4', 'RIPOR1', 'RIPOR2', 'RIPOR3', 'RIPPLY1', 'RIPPLY2', 'RIPPLY3', 'RIT1', 'RITA1', 'RLBP1', 'RLF', 'RLIM', 'RLN1', 'RLN2', 'RMC1', 'RMDN1', 'RMDN2', 'RMDN2-AS1', 'RMDN3', 'RM11', 'RM12', 'RMND1', 'RM ND5A', 'RMND5B', 'RMST', 'RN7SKP1', 'RN7SKP106', 'RN7SKP11', 'RN7SKP116', 'RN7SKP118', 'RN7SKP123', 'RN7SKP137', 'RN7SKP149', 'RN7SKP150', 'RN7SKP154', 'RN7SKP16', 'RN7SKP173', 'RN7SKP179', 'RN7SKP198', 'RN7SKP208', 'RN7SKP239', 'RN7SKP24', 'RN7SKP240', 'RN7SKP243', 'RN7SKP26', 'RN7SKP269', 'RN7SKP271', 'RN7SKP275', 'RN7SKP276', 'RN7SKP281', 'RN7SKP287', 'RN7SKP292', 'RN7 SKP296', 'RN7SKP30', 'RN7SKP38', 'RN7SKP67', 'RN7SKP69', 'RN7SKP70', 'RN7SKP72', 'RN7SKP72', 'RN7SKP78', 'RN7SKP80', 'RN7SKP 87', 'RN7SKP97', 'RN7SL113P', 'RN7SL118P', 'RN7SL121P', 'RN7SL125P', 'RN7SL128P', 'RN7SL12P', 'RN7SL130P', 'RN7SL134P', 'RN7 SL138P', 'RN7SL145P', 'RN7SL146P', 'RN7SL15P', 'RN7SL153P', 'RN7SL15P', 'RN7SL172P', 'RN7SL181P', 'RN7SL192P', 'RN7SL2', 'RN7SL202P', 'RN7SL208P', 'RN7SL219P', 'RN7SL220P', 'RN7SL221P', 'RN7SL236P', 'RN7SL239P', 'RN7SL23P', 'RN7SL246P', 'RN7SL258 P', 'RN7SL262P', 'RN7SL268P', 'RN7SL269P', 'RN7SL270P', 'RN7SL274P', 'RN7SL280P', 'RN7SL285P', 'RN7SL288P', 'RN7SL3', 'RN7SL 314P', 'RN7SL328P', 'RN7SL329P', 'RN7SL32P', 'RN7SL330P', 'RN7SL333P', 'RN7SL338P', 'RN7SL34P', 'RN7SL356P', 'RN7SL359P', 'RN7SL368P', 'RN7SL381P', 'RN7SL38P', 'RN7SL391P', 'RN7SL399P', 'RN7SL403P', 'RN7SL404P', 'RN7SL405P', 'RN7SL417P', 'RN7SL425 P', 'RN7SL431P', 'RN7SL440P', 'RN7SL441P', 'RN7SL444P', 'RN7SL449P', 'RN7SL455P', 'RN7SL471P', 'RN7SL473P', 'RN7SL477P', 'RN7SL481P', 'RN7SL487P', 'RN7SL491P', 'RN7SL494P', 'RN7SL49P', 'RN7SL50P', 'RN7SL505P', 'RN7SL510P', 'RN7SL521P', 'RN7SL523P', 'RN7SL524P', 'RN7SL525P', 'RN7SL526P', 'RN7SL531P', 'RN7SL535P', 'RN7SL541P', 'RN7SL558P', 'RN7SL559P', 'RN7SL566P', 'RN7SL574P', 'RN7SL583P', 'RN7SL585P', 'RN7SL589P', 'RN7SL594P', 'RN7SL600P', 'RN7SL602P', 'RN7SL605P', 'RN7SL608P', 'RN7SL619P', 'RN7SL634P', 'RN7SL635P', 'RN7SL648P', 'RN7SL650P', 'RN7SL659P', 'RN7SL663P', 'RN7SL670P', 'RN7SL671P', 'RN7SL6 74P', 'RN7SL67P', 'RN7SL688P', 'RN7SL689P', 'RN7SL68P', 'RN7SL726P', 'RN7SL736P', 'RN7SL738P', 'RN7SL744P', 'RN7SL745P', 'RN7SL749P', 'RN7SL74P', 'RN7SL751P', 'RN7SL75P', 'RN7SL775P', 'RN7SL784P', 'RN7SL786P', 'RN7SL801P', 'RN7SL809P', 'RN7SL811P', 'RN7SL812P', 'RN7SL817P', 'RN7SL81P', 'RN7SL823P', 'RN7SL832P', 'RN7SL834P', 'RN7SL8P', 'RN7SL96P', 'RNA5-85', 'RNA59', 'RNA5SP101', 'RNA5SP108', 'RNA5SP109', 'RNA5SP111', 'RNA5SP118', 'RNA5SP122', 'RNA5SP123', 'RNA5SP123', 'RNA5SP131', 'RNA5SP132', 'RNA5SP13 9', 'RNA5SP151', 'RNA5SP154', 'RNA5SP155', 'RNA5SP156', 'RNA5SP159', 'RNA5SP161', 'RNA5SP162', 'RNA5SP165', 'RNA5SP174', 'RN A5SP175', 'RNA5SP179', 'RNA5SP18', 'RNA5SP187', 'RNA5SP194', 'RNA5SP195', 'RNA5SP20', 'RNA5SP201', 'RNA5SP203', 'RNA5SP21', 'RNA5SP212', 'RNA5SP217', 'RNA5SP219', 'RNA5SP221', 'RNA5SP233', 'RNA5SP241', 'RNA5SP243', 'RNA5SP247', 'RNA5SP248', 'RNA5SP260', 'RNA5SP264', 'RNA5SP268', 'RNA5SP277', 'RNA5SP278', 'RNA5SP283', 'RNA5SP284', 'RNA5SP29', 'RNA5SP295', 'RNA5SP297', 'RN A5SP298', 'RNA5SP299', 'RNA5SP300', 'RNA5SP301', 'RNA5SP310', 'RNA5SP311', 'RNA5SP317', 'RNA5SP319', 'RNA5SP323', 'RNA5SP3 3', 'RNA5SP334', 'RNA5SP340', 'RNA5SP345', 'RNA5SP352', 'RNA5SP364', 'RNA5SP37', 'RNA5SP379', 'RNA5SP383', 'RNA5SP385', 'RNA5SP39', 'RNA5SP390', 'RNA5SP393', 'RNA5SP394', 'RNA5SP395', 'RNA5SP40', 'RNA5SP405', 'RNA5SP406', 'RNA5SP409', 'RNA5SP410', 'RNA5SP411', 'RNA5SP412', 'RNA5SP413', 'RNA5SP414', 'RNA5SP415', 'RNA5SP416', 'RNA5SP417', 'RNA5SP418', 'RNA5SP419', 'RNA5SP420', 'RNA5SP421', 'RNA5SP422', 'RNA5SP423', 'RNA5SP434', 'RNA5SP435', 'RNA5SP437', 'RNA5SP440', 'RNA5SP441', 'RNA5SP449', 'RNA5SP450', 'RNA5SP452', 'RNA5SP46', 'RNA5SP460', 'RNA5SP462', 'RNA5SP464', 'RNA5SP465', 'RNA5SP469', 'RNA5SP473', 'RNA5SP474', 'RNA5SP477', 'RNA5SP478', 'RNA5SP479', 'RNA5SP48', 'RNA5SP488', 'RNA5SP490', 'RNA5SP491', 'RNA5SP493', 'RNA5SP494', 'RN A5SP496', 'RNA5SP498', 'RNA5SP502', 'RNA5SP507', 'RNA5SP527', 'RNA5SP530', 'RNA5SP534', 'RNA5SP535', 'RNA5SP78', 'RNA5SP82', 'RNA5SP85', 'RNA5SP88', 'RNA5SP92', 'RNA5SP97', 'RNASE1', 'RNASE10', 'RNASE13', 'RNASE61', 'RNASEH1', 'RNASEH1-AS1', 'RNASEH1 P1', 'RNASEH2A', 'RNASEH2B', 'RNASEH2C', 'RNASEK', 'RNASEK-C17orf49', 'RNASEL', 'RNASET2', 'RND1', 'RND2', 'RND3', 'RNF10', 'RNF103', 'RNF11', 'RNF111', 'RNF112', 'RNF113A', 'RNF114', 'RNF115', 'RNF121', 'RNF122', 'RNF123', 'RNF125', 'RNF126', 'RNF128', 'RNF13', 'RNF130', 'RNF135', 'RNF138', 'RNF139', 'RNF139-AS1', 'RNF14', 'RNF141', 'RNF144A', 'RNF144A-AS1', 'RNF144B', 'RNF145', 'RNF146', 'RNF149', 'RNF150', 'RNF152', 'RNF157', 'RNF157-AS1', 'RNF165', 'RNF166', 'RNF167', 'RNF168', 'RNF169', 'RNF17', 'RNF170', 'RNF175', 'RNF180', 'RNF181', 'RNF182', 'RNF183', 'RNF185', 'RNF186', 'RNF187', 'RNF19A', 'RNF19B', 'RNF2', 'RNF20', 'RNF207', 'RNF208', 'RNF212', 'RNF213', 'RNF214', 'RNF215', 'RNF216', 'RNF216P1', 'RNF217', 'RNF219', 'RNF220', 'RNF222', 'RNF223', 'RNF224', 'RNF225', 'RNF227', 'RNF24', 'RNF25', 'RNF26', 'RNF2P1', 'RNF31', 'RNF32', 'RNF34', 'RNF38', 'RNF39', 'RNF4', 'RNF40', 'RNF41', 'RNF43', 'RNF44', 'RNF5', 'RNF5P1', 'RNF6', 'RNF7', 'RNF8', 'RNF1T1', 'RNF1T-DT', 'RNF2T', 'RNGTT', 'RNH1', 'RNLS1', 'RNMT1', 'RNPC3', 'RNPEP1', 'RNPEP1L1', 'RNPS1', 'RNUI1-1', 'RNUI1-103P', 'RNUI1-106P', 'RNUI1-114P', 'RNUI1-120P', 'RNUI1-122P', 'RNUI1-129P', 'RNUI1-130P', 'RNUI1-134P', 'RNUI1-138P', 'RNUI1-150P', 'RNUI1-16P', 'RNUI1-2', 'RNUI1-36P', 'RNUI1-47P', 'RNUI1-51P', 'RNUI1-59P', 'RNUI1-67P', 'RNUI1-70P', 'RNUI1-72P', 'RNUI1-73P', 'RNUI1-75P', 'RNUI1-85P', 'RNUI1-91P', 'RNUI1-98P', 'RNUI11', 'RNUI11-3P', 'RNUI2-11P', 'RNUI2-11P', 'RNUI2-13P', 'RNUI2-17P', 'RNUI2-22P', 'RNUI2-27P', 'RNUI2-28P', 'RNUI2-29P', 'RNUI2-32P', 'RNUI2-34P', 'RNUI2-35P', 'RNUI2-38P', 'RNUI2-3P', 'RNUI2-42P', 'RNUI2-46P', 'RNUI2-49P', 'RNUI2-51P', 'RNUI2-52P', 'RNUI2-58P', 'RNUI2-62P', 'RNUI2-63P', 'RNUI2-68P', 'RNUI2-69P', 'RNUI2-6P', 'RNUI2-7P', 'RNUI4-1', 'RNUI4-14P', 'RNUI4-2', 'RNUI4-21P', 'RNUI4-23P', 'RNUI4-24P', 'RNUI4-25P', 'RNUI4-29P', 'RNUI4-31P', 'RNUI4-36P', 'RNUI4-38P', 'RNUI4-39P', 'RNUI4-47P', 'RNUI4-49P', 'RNUI4-52P', 'RNUI4-50P', 'RNUI4-54P', 'RNUI4-5P', 'RNUI4-62P', 'RNUI4-73P', 'RNUI4-77P', 'RNUI4-78P', 'RNUI4-80P', 'RNUI4-82P', 'RNUI4-85P', 'RNUI4-89P', 'RNUI4-ATAC', 'RNUI4ATAC1P', 'RNUI4ATAC1P', 'RNUI5A-1', 'RNUI5A-2P', 'RNUI5A-8P', 'RNUI5B-1', 'RNUI5B-2P', 'RNUI5D-1', 'RNUI5E-1', 'RNUI5E-10P', 'RNUI5E-4P', 'RNUI5E-8P', 'RNUI5F-1', 'RNUI6-1', 'RNUI6-1003P', 'RN U6-1004P', 'RNUI6-1005P', 'RNUI6-100P', 'RNUI6-1010P', 'RNUI6-1011P', 'RNUI6-1013P', 'RNUI6-1016P', 'RNUI6-101P', 'RNUI6-1024P', 'RN U6-1025P', 'RNUI6-1039P', 'RNUI6-103P', 'RNUI6-1042P', 'RNUI6-1043P', 'RNUI6-1045P', 'RNUI6-1048P', 'RNUI6-1053P', 'RNUI6-1054P', 'RN UI6-1055P', 'RNUI6-1056P', 'RNUI6-1057P', 'RNUI6-1061P', 'RNUI6-1062P', 'RNUI6-1069P', 'RNUI6-106P', 'RNUI6-1079P', 'RNUI6-107P', 'RN UI6-1082P', 'RNUI6-1091P', 'RNUI6-1093P', 'RNUI6-1095P', 'RNUI6-1098P', 'RNUI6-1099P', 'RNUI6-10P', 'RNUI6-1109P', 'RNUI6-1111P', 'RN UI6-1112P', 'RNUI6-1114P', 'RNUI6-1121P', 'RNUI6-1125P', 'RNUI6-1136P', 'RNUI6-1141P', 'RNUI6-1152P', 'RNUI6-1153P', 'RNUI6-1154P',

'RNU6-1157P', 'RNU6-1160P', 'RNU6-1161P', 'RNU6-1164P', 'RNU6-1165P', 'RNU6-1170P', 'RNU6-1178P', 'RNU6-1188P', 'RNU6-1189P', 'RNU6-118P', 'RNU6-1194P', 'RNU6-1198P', 'RNU6-1203P', 'RNU6-1206P', 'RNU6-1209P', 'RNU6-1212P', 'RNU6-1223P', 'RNU6-1225P', 'RNU6-122P', 'RNU6-1231P', 'RNU6-1237P', 'RNU6-1238P', 'RNU6-123P', 'RNU6-1242P', 'RNU6-1249P', 'RNU6-1256P', 'RNU6-1257P', 'RNU6-125P', 'RNU6-1262P', 'RNU6-1263P', 'RNU6-1266P', 'RNU6-126P', 'RNU6-1272P', 'RNU6-1278P', 'RNU6-1283P', 'RNU6-1284P', 'RNU6-1285P', 'RNU6-1294P', 'RNU6-12P', 'RNU6-1304P', 'RNU6-1306P', 'RNU6-1308P', 'RNU6-130P', 'RNU6-1315P', 'RNU6-1316P', 'RNU6-1321P', 'RNU6-1324P', 'RNU6-1327P', 'RNU6-132P', 'RNU6-1331P', 'RNU6-1333P', 'RNU6-1337P', 'RNU6-135P', 'RNU6-136P', 'RNU6-137P', 'RNU6-140P', 'RNU6-141P', 'RNU6-154P', 'RNU6-163P', 'RNU6-167P', 'RNU6-171P', 'RNU6-172P', 'RN
U6-173P', 'RNU6-178P', 'RNU6-181P', 'RNU6-182P', 'RNU6-188P', 'RNU6-189P', 'RNU6-190P', 'RNU6-194P', 'RNU6-195P', 'RNU6-196P', 'RNU6-198P', 'RNU6-199P', 'RNU6-19P', 'RNU6-2', 'RNU6-202P', 'RNU6-204P', 'RNU6-208P', 'RNU6-20P', 'RNU6-213P', 'RNU6-216P', 'RNU6-218P', 'RNU6-226P', 'RNU6-227P', 'RNU6-228P', 'RNU6-229P', 'RNU6-230P', 'RNU6-237P', 'RNU6-242P', 'RNU6-247P', 'RNU6-254P', 'RNU6-260P', 'RNU6-262P', 'RNU6-268P', 'RNU6-26P', 'RNU6-277P', 'RNU6-282P', 'RNU6-288P', 'RNU6-291P', 'RNU6-298P', 'RNU6-301P', 'RNU6-302P', 'RNU6-307P', 'RNU6-30P', 'RNU6-312P', 'RNU6-315P', 'RNU6-316P', 'RNU6-322P', 'RNU6-323P', 'RNU6-324P', 'RNU6-32P', 'RNU6-334P', 'RNU6-33P', 'RNU6-341P', 'RNU6-343P', 'RNU6-352P', 'RNU6-353P', 'RNU6-354P', 'RNU6-367P', 'RNU6-36P', 'RNU6-370P', 'RNU6-375P', 'RNU6-378P', 'RNU6-379P', 'RNU6-37P', 'RNU6-380P', 'RNU6-387P', 'RNU6-388P', 'RNU6-3P', 'RNU6-402P', 'RNU6-403P', 'RNU6-406P', 'RNU6-407P', 'RNU6-415P', 'RNU6-418P', 'RNU6-419P', 'RNU6-423P', 'RNU6-427P', 'RNU6-430P', 'RNU6-431P', 'RNU6-437P', 'RNU6-438P', 'RNU6-444P', 'RNU6-446P', 'RNU6-450P', 'RNU6-45P', 'RNU6-464P', 'RNU6-469P', 'RNU6-475P', 'RNU6-476P', 'RNU6-479P', 'RNU6-481P', 'RNU6-483P', 'RNU6-484P', 'RNU6-485P', 'RNU6-48P', 'RNU6-490P', 'RNU6-497P', 'RNU6-4P', 'RNU6-504P', 'RNU6-505P', 'RNU6-50P', 'RNU6-510P', 'RNU6-514P', 'RNU6-516P', 'RNU6-517P', 'RNU6-519P', 'RNU6-524P', 'RNU6-529P', 'RNU6-531P', 'RNU6-539P', 'RNU6-540P', 'RNU6-541P', 'RNU6-548P', 'RNU6-549P', 'RNU6-552P', 'RNU6-553P', 'RNU6-564P', 'RNU6-574P', 'RNU6-577P', 'RNU6-583P', 'RNU6-584P', 'RNU6-588P', 'RNU6-58P', 'RNU6-595P', 'RNU6-597P', 'RNU6-5P', 'RNU6-605P', 'RNU6-606P', 'RNU6-60P', 'RNU6-610P', 'RNU6-611P', 'RNU6-616P', 'RNU6-617P', 'RNU6-61P', 'RNU6-621P', 'RNU6-623P', 'RNU6-628P', 'RNU6-62P', 'RNU6-633P', 'RNU6-638P', 'RNU6-642P', 'RNU6-644P', 'RNU6-646P', 'RNU6-647P', 'RNU6-652P', 'RNU6-658P', 'RNU6-669P', 'RNU6-677P', 'RNU6-678P', 'RNU6-680P', 'RNU6-689P', 'RNU6-695P', 'RNU6-696P', 'RNU6-6P', 'RNU6-7', 'RNU6-702P', 'RNU6-703P', 'RNU6-705P', 'RNU6-718P', 'RNU6-720P', 'RNU6-722P', 'RNU6-725P', 'RNU6-731P', 'RNU6-737P', 'RNU6-744P', 'RNU6-748P', 'RNU6-757P', 'RNU6-758P', 'RNU6-759P', 'RNU6-761P', 'RNU6-762P', 'RNU6-766P', 'RNU6-767P', 'RNU6-769P', 'RNU6-77P', 'RNU6-78P', 'RNU6-781P', 'RNU6-786P', 'RNU6-792P', 'RNU6-795P', 'RNU6-8P', 'RNU6-808P', 'RNU6-80P', 'RNU6-813P', 'RNU6-817P', 'RNU6-821P', 'RNU6-824P', 'RNU6-826P', 'RNU6-828P', 'RNU6-833P', 'RNU6-834P', 'RNU6-836P', 'RNU6-840P', 'RNU6-844P', 'RNU6-850P', 'RNU6-853P', 'RNU6-856P', 'RNU6-863P', 'RNU6-864P', 'RNU6-875P', 'RNU6-876P', 'RNU6-878P', 'RNU6-879P', 'RNU6-87P', 'RNU6-880P', 'RNU6-881P', 'RNU6-882P', 'RNU6-883P', 'RNU6-88P', 'RNU6-890P', 'RNU6-892P', 'RNU6-893P', 'RNU6-9', 'RNU6-902P', 'RNU6-906P', 'RNU6-907P', 'RNU6-909P', 'RNU6-90P', 'RNU6-911P', 'RNU6-915P', 'RNU6-91P', 'RNU6-920P', 'RNU6-921P', 'RNU6-925P', 'RNU6-928P', 'RNU6-929P', 'RNU6-930P', 'RNU6-937P', 'RNU6-942P', 'RNU6-944P', 'RNU6-945P', 'RNU6-948P', 'RNU6-949P', 'RNU6-94P', 'RNU6-951P', 'RNU6-955P', 'RNU6-958P', 'RNU6-964P', 'RNU6-968P', 'RNU6-969P', 'RNU6-96P', 'RNU6-95P', 'RNU6-97P', 'RNU6-988P', 'RNU6-998P', 'RNU6ATAAC14P', 'RNU6ATAAC16P', 'RNU6ATAAC18P', 'RNU6ATAAC24P', 'RNU6ATAAC27P', 'RNU6ATAAC29P', 'RNU6ATAAC36P', 'RNU6ATAAC39P', 'RNU6ATAAC40P', 'RNU6ATAAC8P', 'RNU6ATAAC9P', 'RNU6V', 'RNU7-1', 'RNU7-107P', 'RNU7-10P', 'RNU7-110P', 'RNU7-113P', 'RNU7-123P', 'RNU7-124P', 'RNU7-128P', 'RNU7-129P', 'RNU7-13P', 'RNU7-140P', 'RNU7-141P', 'RNU7-143P', 'RNU7-151P', 'RNU7-154P', 'RNU7-159P', 'RNU7-160P', 'RNU7-165P', 'RNU7-169P', 'RNU7-170P', 'RNU7-171P', 'RNU7-173P', 'RNU7-175P', 'RNU7-181P', 'RNU7-186P', 'RNU7-18P', 'RNU7-195P', 'RNU7-196P', 'RNU7-20P', 'RNU7-26P', 'RNU7-30P', 'RNU7-38P', 'RNU7-3P', 'RNU7-40P', 'RNU7-43P', 'RNU7-45P', 'RNU7-47P', 'RNU7-48P', 'RNU7-49P', 'RNU7-59P', 'RNU7-70P', 'RNU7-71P', 'RNU7-74P', 'RNU7-75P', 'RNU7-77P', 'RNU7-79P', 'RNU7-84P', 'RNU7-85P', 'RNU7-8P', 'RNU7-90P', 'RNU7-96P', 'RNU7-97P', 'RNUV1-1', 'RNUV1-15', 'RNUV1-19', 'RNUV1-2', 'RNUV1-3', 'RNUV1-4', 'RNUV1-6', 'RNUV1-7', 'RNY1', 'RNY1P10', 'RNY1P11', 'RNY1P12', 'RNY1P13', 'RNY1P15', 'RNY1P16', 'RNY1P19', 'RNY3P1', 'RNY3P16', 'RNY3P4', 'RNY3P7', 'RNY3P8', 'RNY3P9', 'RNY4P10', 'RNY4P17', 'RNY4P19', 'RNY4P24', 'RNY4P25', 'RNY4P36', 'RNY4P37', 'ROB01', 'ROB02', 'ROB03', 'ROB04', 'ROCK1', 'ROCK1P1', 'ROCK2', 'ROCR', 'ROGDI', 'ROM1', 'ROM1', 'ROPN1', 'ROPN1L', 'ROR1', 'ROR2', 'RORA', 'RORA-AS1', 'RORB', 'RORC', 'ROS1', 'RP1', 'RP2', 'RP9', 'RP9P', 'RPA1', 'RPA2', 'RPA3', 'RPA4', 'RPAIN', 'RPAP1', 'RPAP2', 'RPAP3', 'RPARP-AS1', 'RPE', 'RPE65', 'RPEP4', 'RPF1', 'RPF2', 'RPGR', 'RPGRIP1L', 'RPH3AL', 'RPIA', 'RPIAP1', 'RPIAP1', 'RPL10A', 'RPL10AP6', 'RPL10L', 'RPL10P12', 'RPL10P13', 'RPL10P16', 'RPL10P6', 'RPL10P8', 'RPL10P9', 'RPL11', 'RPL11P3', 'RPL12', 'RPL12P1', 'RPL12P14', 'RPL12P25', 'RPL12P37', 'RPL12P38', 'RPL12P4', 'RPL12P44', 'RPL12P47', 'RPL12P9', 'RPL13', 'RPL13A', 'RPL13AP20', 'RPL13AP25', 'RPL13A5P', 'RPL13A5P', 'RPL13P1', 'RPL13P5', 'RPL14', 'RPL14P1', 'RPL15', 'RPL15P3', 'RPL17', 'RPL17P50', 'RPL17P51', 'RPL18', 'RPL18A', 'RPL18AP2', 'RPL18AP3', 'RPL18AP7', 'RPL18P13', 'RPL19', 'RPL19P11', 'RPL21', 'RPL21P10', 'RPL21P121', 'RPL21P123', 'RPL21P16', 'RPL21P3', 'RPL21P40', 'RPL21P75', 'RPL21P93', 'RPL22', 'RPL22L1', 'RPL22P1', 'RPL22P19', 'RPL22P2', 'RPL22P24', 'RPL22P3', 'RPL23', 'RPL23A', 'RPL23AP11', 'RPL23AP14', 'RPL23AP2', 'RPL23AP20', 'RPL23AP32', 'RPL23AP34', 'RPL23AP42', 'RPL23AP49', 'RPL23AP53', 'RPL23AP55', 'RPL23AP63', 'RPL23AP64', 'RPL23AP67', 'RPL23AP77', 'RPL23AP81', 'RPL23AP82', 'RPL23AP85', 'RPL23AP87', 'RPL23AP95', 'RPL23AP97', 'RPL23P2', 'RPL23P8', 'RPL24', 'RPL24P2', 'RPL24P4', 'RPL24P8', 'RPL26', 'RPL26L1', 'RPL26P19', 'RPL26P27', 'RPL26P29', 'RPL26P30', 'RPL26P6', 'RPL27', 'RPL27A', 'RPL28', 'RPL29', 'RPL29P11', 'RPL29P14', 'RPL29P19', 'RPL29P24', 'RPL31', 'RPL31P11', 'RPL31P52', 'RPL32', 'RPL32P1', 'RPL32P3', 'RPL32P33', 'RPL32P33', 'RPL34P22', 'RPL35', 'RPL35A', 'RPL35AP15', 'RPL35AP32', 'RPL35AP4', 'RPL35P1', 'RPL35P2', 'RPL35P3', 'RPL35P5', 'RPL36', 'RPL36A', 'RPL36AL', 'RPL36AP15', 'RPL36AP33', 'RPL36AP39', 'RPL36AP40', 'RPL36AP43', 'RPL36AP51', 'RPL36P19', 'RPL37', 'RPL37A', 'RPL37P1', 'RPL37P15', 'RPL37P2', 'RPL37P3', 'RPL37P6', 'RPL38', 'RPL38P4', 'RPL39', 'RPL39L', 'RPL39P18', 'RPL39P3', 'RPL39P6', 'RPL39P38', 'RPL39P40', 'RPL39P5', 'RPL3P2', 'RPL3P4', 'RPL41', 'RPL41P2', 'RPL4P1', 'RPL4P4', 'RPL4P6', 'RPL5', 'RPL5P30', 'RPL5P4', 'RPL6', 'RPL6P27', 'RPL7', 'RPL7AP10', 'RPL7AP2', 'RPL7AP28', 'RPL7AP3', 'RPL7AP30', 'RPL7AP6', 'RPL7L1', 'RPL7P1', 'RPL7P16', 'RPL7P23', 'RPL7P38', 'RPL7P50', 'RPL7P57', 'RPL7P9', 'RPL8', 'RPL9', 'RPL9P28', 'RPL9P29', 'RPL9P9', 'RPLP0P2', 'RPLP0P6', 'RPLP1', 'RPLP1P13', 'RPLP1P6', 'RPLP2', 'RPN1', 'RPN2', 'RPP14', 'RPP21', 'RPP25', 'RPP25L', 'RPP30', 'RPP38', 'RPP40', 'RPRD1A', 'RPRD1B', 'RPRD2', 'RPRM', 'RPRML', 'RPS10', 'RPS10-NUDT3', 'RPS10P28', 'RPS10P5', 'RPS10P7', 'RPS11', 'RPS11P5', 'RPS11P7', 'RPS12', 'RPS12P16', 'RPS12P18', 'RPS12P2', 'RPS12P20', 'RPS12P26', 'RPS12P27', 'RPS12P31', 'RPS12P5', 'RPS13', 'RPS13P2', 'RPS14', 'RPS14P8', 'RPS15', 'RPS15A', 'RPS15AP1', 'RPS15AP01', 'RPS15AP12', 'RPS15AP16', 'RPS15AP36', 'RPS15AP9', 'RPS16', 'RPS16P5', 'RPS16P9', 'RPS17', 'RPS18', 'RPS18P9', 'RPS19', 'RPS19P1', 'RPS19P3', 'RPS2', 'RPS20', 'RPS20P14', 'RPS20P15', 'RPS20P22', 'RPS20P31', 'RPS20P33', 'RPS20P35', 'RPS20P4', 'RPS21', 'RPS23', 'RPS23P1', 'RPS23P2', 'RPS23P8', 'RPS24', 'RPS24P17', 'RPS24P8', 'RPS25', 'RPS25P9', 'RPS26', 'RPS26P21', 'RPS26P49', 'RPS26P52', 'RPS26P54', 'RPS27', 'RPS27A', 'RPS27AP10', 'RPS27AP12', 'RPS27AP13', 'RPS27AP16', 'RPS27AP2', 'RPS27P14', 'RPS27P16', 'RPS27P29', 'RPS28', 'RPS28P7', 'RPS29', 'RPS29P11', 'RPS29P14', 'RPS29P16', 'RPS29P17', 'RPS29P19', 'RPS29P22', 'RPS29P21', 'RPS29P5', 'RPS2P2', 'RPS2P4', 'RPS2P6', 'RPS2P55', 'RPS2P7P2', 'RPS3', 'RPS3A', 'RPS3AP24', 'RPS3AP26', 'RPS3AP3', 'RPS3AP38', 'RPS3AP46', 'RPS3AP5', 'RPS3AP54', 'RPS3AP6', 'RPS3P2', 'RPS4X', 'RPS4XP16', 'RPS4XP19', 'RPS4Y1', 'RPS5', 'RPS5P2', 'RPS5', 'RPS6KA1', 'RPS6KA2', 'RPS6KA2-IT1', 'RPS6KA3', 'RPS6KA4', 'RPS6KA5', 'RPS6KA6', 'RPS6KB1', 'RPS6KB2', 'RPS6KC1', 'RPS6KL1', 'RPS6P12', 'RPS6P25', 'RPS6P8', 'RPS7', 'RPS7P1', 'RPS7P10', 'RPS7P11', 'RPS7P3', 'RPS8', 'RPS9', 'RPSA', 'RPSAP12', 'RPSAP14', 'RPSAP15', 'RPSAP17', 'RPSAP19', 'RPSAP47', 'RPSAP50', 'RPSAP52', 'RPSAP54', 'RPSAP58', 'RPSAP69', 'RPSAP70', 'RPSAP71', 'RPSAP72', 'RPSAP9', 'RPTOR', 'RPU5D1', 'RPU5D2', 'RPU5D3', 'RPU5D4', 'RRAD', 'RRAGA', 'RRAGB', 'RRAGC', 'RRAS2', 'RRB2', 'RRB1', 'RRH', 'RRM1', 'RRM1-AS1', 'RRM2', 'RRM2B', 'RRM2P3', 'RRN3', 'RRN3P1', 'RRN3P2', 'RRN3P3', 'RRNAD1', 'RRP1', 'RRP12', 'RRP15', 'RRP1B', 'RRP36', 'RRP7A', 'RRP7B', 'RRP8', 'RRP9', 'RRS1', 'RRSAD1', 'RRSAD2', 'RRSB1', 'RRSB11', 'RRSF1', 'RRSF1-IT1', 'RSF1-IT2', 'RSL1D1', 'RSL24D1', 'RSL24D1P6', 'RSPH1', 'RSPH3', 'RSPH4A', 'RSPH9', 'RSP02', 'RSP03', 'RSP04', 'RSP05', 'RSPRY1', 'RSRC1', 'RSRC2', 'RTS1', 'RTS1', 'RTBDN', 'RTCA', 'RTCA-AS1', 'RTCB', 'RTEL1', 'RTEL1-TNFRSF6B', 'RTF1', 'RTF2', 'RTKN', 'RTKN2', 'RTL10', 'RTL3', 'RTL5', 'RTL6', 'RTL8A', 'RTL8B', 'RTL8C', 'RTL9', 'RTN1', 'RTN2', 'RTN3', 'RTN4', 'RTN4IP1', 'RTN4R', 'RTN4RL1', 'RTN4R2', 'RTP1', 'RTP4', 'RTP5', 'RTRAF', 'RTTN', 'RUBCN', 'RUBCNL', 'RUFY1', 'RUFY2', 'RUFY3', 'RUNDC1', 'RUNDC3A', 'RUNDC3A-AS

1', 'RUNDC3B', 'RUNX1', 'RUNX1T1', 'RUNX2', 'RUNX3', 'RUSC1', 'RUSC1', 'RUVBL1', 'RUVBL2', 'RWDD1', 'RWDD2A', 'RWDD2B', 'RWDD3', 'RWDD4', 'RWDD4P1', 'RXFP1', 'RXFP2', 'RXFP4', 'RXRA', 'RXRB', 'RXRG', 'RXYL1', 'RYBP', 'RYK', 'RYR1', 'RYR2', 'RYR3', 'S100A1', 'S100A10', 'S100A11', 'S100A12', 'S100A13', 'S100A14', 'S100A16', 'S100A2', 'S100A3', 'S100A4', 'S100A5', 'S100A6', 'S100A7', 'S100A7A', 'S100A8', 'S100A9', 'S100B', 'S100P', 'S100PBP', 'S100Z', 'S1PR1', 'S1PR2', 'S1PR3', 'S1PR4', 'S1PR5', 'SAA1', 'SAA2', 'SAA2-SAA4', 'SAA4', 'SAAL1', 'SAC3D1', 'SACM1L', 'SACS', 'SAE1', 'SAFB', 'SAFB2', 'SAGE1', 'SALL1', 'SALL2', 'SALL3', 'SALL4', 'SALL4P1', 'SALRNA1', 'SAMD1', 'SAMD8', 'SAMD9', 'SAMD9L', 'SAMHD1', 'SAMM50', 'SAMMSO', 'SAMSN1', 'SAMSN1-AS1', 'SAP130', 'SAP18', 'SAP30', 'SAP30BP', 'SAP30L', 'SAP30L-AS1', 'SAPCD1', 'SAPCD1-AS1', 'SAPCD2', 'SAR1A', 'SAR1B', 'SARAF', 'SARDH', 'SARM1', 'SARNP', 'SARS', 'SARS2', 'SART1', 'SART3', 'SASH1', 'SASH3', 'SASS6', 'SAT1', 'SAT2', 'SATB1', 'SATB1-AS1', 'SATB2', 'SATB2-AS1', 'SAV1', 'SAXO2', 'SAYS1', 'SBDS', 'SBDS1', 'SBF1', 'SBF2', 'SBF2-AS1', 'SBK1', 'SBK2', 'SBK3', 'SBN01', 'SBN02', 'SBSN', 'SBSN', 'SCD5', 'SCAANT1', 'SCAF1', 'SCAF4', 'SCAF8', 'SCAI', 'SCAMP1', 'SCAMP1-AS1', 'SCAMP2', 'SCAMP3', 'SCAMP4', 'SCAMPS', 'SCAND1', 'SCAND2P', 'SCAP1', 'SCAPER', 'SCARA3', 'SCARAS', 'SCARB1', 'SCARB2', 'SCARF1', 'SCARF2', 'SCARNA1', 'SCARNA10', 'SCARNA12', 'SCARNA13', 'SCARNA14', 'SCARNA15', 'SCARNA18', 'SCARNA18B', 'SCARNA20', 'SCARNA21', 'SCARNA22', 'SCARNA3', 'SCARNA4', 'SCARNA5', 'SCARNA6', 'SCARNA7', 'SCARNA8', 'SCARNA9', 'SCART1', 'SCCPDH', 'SCD5', 'SCDF1', 'SCFD2', 'SCG2', 'SCG3', 'SCGB1B2P', 'SCGB1D2', 'SCGB2A1', 'SCGB2A2', 'SCGB3A2', 'SCGN', 'SCHIP1', 'SCHLAP1', 'SCIMP1', 'SCIN1', 'SCLT1', 'SCLY1', 'SCMH1', 'SCML1', 'SCML2', 'SCML2P2', 'SCML4', 'SCNN1A', 'SCNN1B', 'SCNN2A', 'SCNN2B', 'SCNN3A', 'SCNN3B', 'SCNN4A', 'SCNN4B', 'SCNN5A', 'SCNN7A', 'SCNN8A', 'SCNN9A', 'SCNN11A', 'SCNN1B', 'SCNN1D', 'SCNN1G', 'SC01', 'SC02', 'SCOC', 'SCOC-AS1', 'SCOCP1', 'SCP2', 'SCPEP1', 'SCRG1', 'SCRI', 'SCR1B', 'SCRN1', 'SCRN2', 'SCRN3', 'SCT1', 'SCUBE1', 'SCUBE2', 'SCUBE3', 'SCX1', 'SCYL1', 'SCYL2', 'SCYL3', 'SDAD1', 'SDAD1P1', 'SDC1', 'SDC2', 'SDC3', 'SDC4', 'SDCBP1', 'SDCBP2', 'SDCBP2-AS1', 'SDCCAG3', 'SDCCAG8', 'SDE21', 'SDF2', 'SDF2L1', 'SDF4', 'SDH', 'SDHAF1', 'SDHAF2', 'SDHAF3', 'SDHAF4', 'SDHPA1', 'SDHPA3', 'SDHDB', 'SDHDC', 'SDHDP1', 'SDHPD6', 'SDK1', 'SDK2', 'SDR16C5', 'SDR39U1', 'SDR42E1', 'SDS', 'SDSL', 'SEC11A', 'SEC11C', 'SEC13', 'SEC14L1', 'SEC14L1P1', 'SEC14L2', 'SEC14L4', 'SEC14LS', 'SEC14L6', 'SEC16A', 'SEC22A', 'SEC22B', 'SEC23A', 'SEC23B', 'SEC23IP', 'SEC24A', 'SEC24B-AS1', 'SEC24C', 'SEC24D', 'SEC31A', 'SEC31B', 'SEC61A1', 'SEC61A2', 'SEC61B', 'SEC61G', 'SEC62', 'SEC62-AS1', 'SEC63', 'SEC63P', 'SEC7SBP2', 'SECTS2P2', 'SECTM1', 'SEH11', 'SEL11', 'SELENBP1', 'SELENOF', 'SELENOH', 'SELENOI', 'SELENOK', 'SELENOP1', 'SELENON', 'SELENON', 'SELENOOLP', 'SELENOPL', 'SELENOT', 'SELENOV', 'SELENOW', 'SELL', 'SEL', 'SELP', 'SEPLG', 'SEM1', 'SEM1P1', 'SEMA3A', 'SEMA3B', 'SEMA3B-AS1', 'SEMA3C', 'SEMA3D', 'SEMA3E', 'SEMA3F', 'SEMA3F-AS1', 'SEMA3G', 'SEMA4A', 'SEMA4B', 'SEMA4C', 'SEMA4D', 'SEMA4F', 'SEMA4G', 'SEMASA', 'SEMA5B', 'SEMA6A', 'SEMA6A-AS1', 'SEMA6B', 'SEMA6C', 'SEMA6D', 'SEMA7A', 'SENCR', 'SENPI1', 'SENPI2', 'SENPI3', 'SENPI5', 'SENPI6', 'SENPI7', 'SENPI8', 'SENPH1', 'SENPH2', 'SEP SECS', 'SEPSECS-AS1', 'SEPT1', 'SEPT10', 'SEPT11', 'SEPT14P12', 'SEPT14P4', 'SEPT2', 'SEPT2P1', 'SEPT3', 'SEPT4', 'SEPT4-AS1', 'SEPT5', 'SEPT6', 'SEPT7', 'SEPT7-AS1', 'SEPT7P2', 'SEPT7P3', 'SEPT8', 'SEPT9', 'SERAC1', 'SERBP1', 'SERBP1P1', 'SERBP1P5', 'SERBP1P6', 'SERF2', 'SERGEF', 'SERHL', 'SERHL2', 'SERINC1', 'SERINC2', 'SERINC3', 'SERINC5', 'SERP1', 'SERP2', 'SERPINA', 'SERPINA10', 'SERPINA3', 'SERPINA4', 'SERPINA5', 'SERPINA6', 'SERPINA7', 'SERPINA9', 'SERPINB1', 'SERPINB10', 'SERPINB11', 'SERPINB2', 'SERPINB3', 'SERPINB4', 'SERPINB5', 'SERPINB6', 'SERPINB7', 'SERPINB8', 'SERPINB9', 'SERPINB9P1', 'SERPINC', 'SERPIND1', 'SERPINE1', 'SERPINE2', 'SERPINE4P', 'SERPINF1', 'SERPINF2', 'SERPING1', 'SERPINH1', 'SERPINI1', 'SERTAD1', 'SERTAD2', 'SERTAD3', 'SERTAD4', 'SERTAD4-AS1', 'SERTM1', 'SERTM2', 'SESN1', 'SESN2', 'SESN3', 'SESTD1', 'SET', 'SETPB1', 'SETD1A', 'SETD1B', 'SETD2', 'SETD3', 'SETD4', 'SETD5', 'SETD6', 'SETD7', 'SETD9', 'SETDB1', 'SETDB2', 'SETMAR', 'SETP14', 'SETTP5', 'SETP9', 'SETX', 'SEZ6', 'SEZ6L', 'SEZ6L2', 'SF1', 'SF3A1', 'SF3A3', 'SF3B1', 'SF3B2', 'SF3B3', 'SF3B4', 'SF3B5', 'SF3B6', 'SF11', 'SFMBT1', 'SFMBT2', 'SFN', 'SFPPQ', 'SFR1', 'SFRP1', 'SFRP2', 'SFRP4', 'SFRPS', 'SFswap', 'SFT2D1', 'SF T2D2', 'SFT2D3', 'SFTA1P', 'SFTA2', 'SFTA3', 'SFTA1P1', 'SFTA2P', 'SFTFP1', 'SFTFP2', 'SFTFX1', 'SFTXN2', 'SFTXN3', 'SFTXN4', 'SFTXN5', 'SGCA', 'SGCB', 'SGCD', 'SGCE', 'SGC29', 'SGIP1', 'SGK1', 'SGK2', 'SGK3', 'SGMS1', 'SGMS1-AS1', 'SGMS2', 'SGO1', 'SGO2', 'SGP1', 'SGP2', 'SGSH', 'SGS1', 'SGS2', 'SGTA', 'SGTB1', 'SH2B1', 'SH2B2', 'SH2B3', 'SH2D1A', 'SH2D1B', 'SH2D2A', 'SH2D3A', 'SH2D3C', 'SH2D4A', 'SH2D5', 'SH3BGR', 'SH3BGR1L', 'SH3BGR2L', 'SH3BGR3L', 'SH3B1P1', 'SH3B2P2', 'SH3B2P4', 'SH3B2P5', 'SH3B5P-AS1', 'SH3B5P1', 'SH3D19', 'SH3D21', 'SH3GL1', 'SH3GL1P1', 'SH3GL1P2', 'SH3GL2', 'SH3GL3', 'SH3GLB1', 'SH3GLB2', 'SH3KB1P', 'SH3XP2D2A-AS1', 'SH3XP2D2B', 'SH3RF1', 'SH3RF2', 'SH3RF3', 'SH3RF3-AS1', 'SH3TC1', 'SH3TC2', 'SH3TC2-DT', 'SH3Y1L', 'SHANK1', 'SHANK2', 'SHANK2-AS2', 'SHANK3', 'SHARPIN', 'SHB', 'SHC1', 'SHC2', 'SHC3', 'SHC4', 'SHCB1P', 'SHD', 'SHE', 'SHF', 'SHHH', 'SHISA2', 'SHISA3', 'SHISA4', 'SHISA5', 'SHISA8', 'SHISA9', 'SHISAL1', 'SHISAL2A', 'SHISAL2B', 'SHKB1P', 'SHMT1', 'SHMT2', 'SHOC2', 'SHOX2', 'SHPK', 'SHPRH', 'SHQ1', 'SHROOM1', 'SHROOM2', 'SHROOM2P1', 'SHROOM3', 'SHROOM4', 'SHTN1', 'SI', 'SIAE', 'SIAH1', 'SIAH1P1', 'SIAH2', 'SIAH2-AS1', 'SIAH3', 'SIDT1', 'SIDT1-AS1', 'SIDT2', 'SIGIRR', 'SIGLEC10', 'SIGLEC12', 'SIGLEC14', 'SIGLEC15', 'SIGLEC16', 'SIGLEC16P', 'SIGLEC22P', 'SIGLEC26P', 'SIGLEC27P', 'SIGLEC6', 'SIGLEC7', 'SIGLEC8', 'SIGLEC9', 'SIGLEC11', 'SIGMAR1', 'SIIK1', 'SIIK1B', 'SIIK2', 'SIIK3', 'SIIKE1', 'SIILE1', 'SIM1', 'SIM2', 'SIMC1', 'SIN3A', 'SIN3B', 'SINHCAF', 'SINHCAF2P', 'SIPA1', 'SIPA1L1', 'SIPA1L2', 'SIPA1L3', 'SIRPA', 'SIRPB1', 'SIRPG', 'SIRT1', 'SIRT2', 'SIRT3', 'SIRT4', 'SIRT5', 'SIRT6', 'SIRT7', 'SIT1', 'SIVA1', 'SIX1', 'SIX2', 'SIX3', 'SIX3-AS1', 'SIX4', 'SIX5', 'SIX6', 'SKA1', 'SKA2', 'SKA3', 'SKAP1', 'SKAP2', 'SKI', 'SKIDA1', 'SKIL', 'SKINT1L', 'SKIV2L', 'SKP1', 'SKP1P1', 'SKP2', 'SLA2', 'SLAIN1', 'SLAIN2', 'SLAMF1', 'SLAMF6', 'SLAMF6P1', 'SLAMF7', 'SLAMF8', 'SLAMF9', 'SLBP', 'SLC10A3', 'SLC10A4', 'SLC10A5', 'SLC10A7', 'SLC11A2', 'SLC12A2', 'SLC12A3', 'SLC12A4', 'SLC12A5', 'SLC12A5-AS1', 'SLC12A6', 'SLC12A7', 'SLC12A8', 'SLC12A9', 'SLC12A9-AS1', 'SLC13A3', 'SLC13A4', 'SLC14A1', 'SLC15A1', 'SLC15A2', 'SLC15A3', 'SLC15A4', 'SLC16A1', 'SLC16A1-AS1', 'SLC16A10', 'SLC16A12', 'SLC16A13', 'SLC16A14', 'SLC16A2', 'SLC16A3', 'SLC16A4', 'SLC16A5', 'SLC16A6', 'SLC16A6P1', 'SLC16A7', 'SLC16A8', 'SLC16A9', 'SLC17A1', 'SLC17A3', 'SLC17A4', 'SLC17A5', 'SLC17A6', 'SLC17A7', 'SLC17A8', 'SLC17A9', 'SLC18A1', 'SLC18A2', 'SLC18A3', 'SLC18B1', 'SLC19A1', 'SLC19A2', 'SLC19A3', 'SLC1A1', 'SLC1A2', 'SLC1A3', 'SLC1A4', 'SLC1A5', 'SLC1A6', 'SLC1A7', 'SLC20A1', 'SLC20A2', 'SLC22A1', 'SLC22A15', 'SLC22A17', 'SLC22A18', 'SLC22A18AS', 'SLC22A22', 'SLC22A22P', 'SLC22A23', 'SLC22A31', 'SLC22A4', 'SLC22A5', 'SLC23A1', 'SLC23A2', 'SLC23A3', 'SLC24A1', 'SLC24A3', 'SLC24A5', 'SLC25A1', 'SLC25A10', 'SLC25A11', 'SLC25A12', 'SLC25A13', 'SLC25A14', 'SLC25A15', 'SLC25A15P4', 'SLC25A15P5', 'SLC25A16', 'SLC25A17', 'SLC25A18', 'SLC25A19', 'SLC25A1P5', 'SLC25A20', 'SLC25A21', 'SLC25A21-AS1', 'SLC25A22', 'SLC25A23', 'SLC25A24', 'SLC25A24P1', 'SLC25A24P2', 'SLC25A25', 'SLC25A25-AS1', 'SLC25A26', 'SLC25A27', 'SLC25A28', 'SLC25A29', 'SLC25A3', 'SLC25A30', 'SLC25A32', 'SLC25A33', 'SLC25A34', 'SLC25A34-AS1', 'SLC25A35', 'SLC25A36', 'SLC25A37', 'SLC25A38', 'SLC25A39', 'SLC25A4', 'SLC25A40', 'SLC25A41', 'SLC25A42', 'SLC25A43', 'SLC25A44', 'SLC25A45', 'SLC25A46', 'SLC25A47P1', 'SLC25A48', 'SLC25A5', 'SLC25A5-AS1', 'SLC25A51', 'SLC25A51', 'SLC25A52', 'SLC25A53', 'SLC25A5P5', 'SLC25A6', 'SLC26A1', 'SLC26A1P1', 'SLC26A10', 'SLC26A11', 'SLC26A2', 'SLC26A4', 'SLC26A4-AS1', 'SLC26A6', 'SLC26A7', 'SLC26A9', 'SLC27A1', 'SLC27A2', 'SLC27A3', 'SLC27A4', 'SLC27A5', 'SLC27A6', 'SLC28A1', 'SLC28A3', 'SLC29A1', 'SLC29A2', 'SLC29A3', 'SLC29A4', 'SLC2A1', 'SLC2A1-AS1', 'SLC2A10', 'SLC2A11', 'SLC2A12', 'SLC2A13', 'SLC2A14', 'SLC2A2', 'SLC2A3', 'SLC2A4', 'SLC2A4R', 'SLC2A5', 'SLC2A6', 'SLC2A8', 'SLC2A9', 'SLC30A1', 'SLC30A10', 'SLC30A2', 'SLC30A3', 'SLC30A4', 'SLC30A5', 'SLC30A7', 'SLC30A8', 'SLC30A9', 'SLC31A1', 'SLC31A1P1', 'SLC31A2', 'SLC32A1', 'SLC33A1', 'SLC34A2', 'SLC34A3', 'SLC35A1', 'SLC35A2', 'SLC35A3', 'SLC35A4', 'SLC35A5', 'SLC35B1', 'SLC35B2', 'SLC35B3', 'SLC35B4', 'SLC35C1', 'SLC35C2', 'SLC35D1', 'SLC35D2', 'SLC35D3', 'SLC35E1', 'SLC35E1P1', 'SLC35E2A', 'SLC35E2B', 'SLC35E3', 'SLC35E4', 'SLC35F1', 'SLC35F2', 'SLC35F3', 'SLC35F4', 'SLC35F5', 'SLC35F6', 'SLC35G1', 'SLC35G2', 'SLC36A1', 'SLC36A2', 'SLC36A3', 'SLC36A4', 'SLC37A1', 'SLC37A2', 'SLC37A3', 'SLC37A4', 'SLC38A1', 'SLC38A10', 'SLC38A11', 'SLC38A2', 'SLC38A3', 'SLC38A4', 'SLC38A5', 'SLC38A6', 'SLC38A7', 'SLC38A9', 'SLC39A1', 'SLC39A10', 'SLC39A11', 'SLC39A12', 'SLC39A13', 'SLC39A14', 'SLC39A15', 'SLC39A16', 'SLC39A17', 'SLC39A18', 'SLC39A19', 'SLC39A20', 'SLC39A21', 'SLC39A22', 'SLC39A23', 'SLC39A24', 'SLC39A25', 'SLC39A26', 'SLC39A27', 'SLC39A28', 'SLC39A29', 'SLC39A30', 'SLC39A31', 'SLC39A32', 'SLC39A33', 'SLC39A34', 'SLC39A35', 'SLC39A36', 'SLC39A37', 'SLC39A38', 'SLC39A39', 'SLC39A40', 'SLC39A41', 'SLC39A42', 'SLC39A43', 'SLC39A44', 'SLC39A45', 'SLC4A1', 'SLC4A2', 'SLC4A3', 'SLC4A4', 'SLC4A5', 'SLC4A6', 'SLC4A7', 'SLC4A8', 'SLC4A9', 'SLC4A10', 'SLC4A11', 'SLC4A12', 'SLC4A13', 'SLC4A14', 'SLC4A15', 'SLC4A16', 'SLC4A17', 'SLC4A18', 'SLC4A19', 'SLC4A20', 'SLC4A21', 'SLC4A22', 'SLC4A23', 'SLC4A24', 'SLC4A25', 'SLC4A26', 'SLC4A27', 'SLC4A28', 'SLC4A29', 'SLC4A30', 'SLC4A31', 'SLC4A32', 'SLC4A33', 'SLC4A34', 'SLC4A35', 'SLC4A36', 'SLC4A37', 'SLC4A38', 'SLC4A39', 'SLC4A40', 'SLC4A41', 'SLC4A42', 'SLC4A43', 'SLC4A44', 'SLC4A45', 'SLC4A46', 'SLC4A47', 'SLC4A48', 'SLC4A49', 'SLC4A50', 'SLC4A51', 'SLC4A52', 'SLC4A53', 'SLC4A54', 'SLC4A55', 'SLC4A56', 'SLC4A57', 'SLC4A58', 'SLC4A59', 'SLC4A60', 'SLC4A61', 'SLC4A62', 'SLC4A63', 'SLC4A64', 'SLC4A65', 'SLC4A66', 'SLC4A67', 'SLC4A68', 'SLC4A69', 'SLC4A70', 'SLC4A71', 'SLC4A72', 'SLC4A73', 'SLC4A74', 'SLC4A75', 'SLC4A76', 'SLC4A77', 'SLC4A78', 'SLC4A79', 'SLC4A80', 'SLC4A81', 'SLC4A82', 'SLC4A83', 'SLC4A84', 'SLC4A85', 'SLC4A86', 'SLC4A87', 'SLC4A88', 'SLC4A89', 'SLC4A90', 'SLC4A91', 'SLC4A92', 'SLC4A93', 'SLC4A94', 'SLC4A95', 'SLC4A96', 'SLC4A97', 'SLC4A98', 'SLC4A99', 'SLC4A100', 'SLC4A101', 'SLC4A102', 'SLC4A103', 'SLC4A104', 'SLC4A105', 'SLC4A106', 'SLC4A107', 'SLC4A108', 'SLC4A109', 'SLC4A110', 'SLC4A111', 'SLC4A112', 'SLC4A113', 'SLC4A114', 'SLC4A115', 'SLC4A116', 'SLC4A117', 'SLC4A118', 'SLC4A119', 'SLC4A120', 'SLC4A121', 'SLC4A122', 'SLC4A123', 'SLC4A124', 'SLC4A125', 'SLC4A126', 'SLC4A127', 'SLC4A128', 'SLC4A129', 'SLC4A130', 'SLC4A131', 'SLC4A132', 'SLC4A133', 'SLC4A134', 'SLC4A135', 'SLC4A136', 'SLC4A137', 'SLC4A138', 'SLC4A139', 'SLC4A140', 'SLC4A141', 'SLC4A142', 'SLC4A143', 'SLC4A144', 'SLC4A145', 'SLC4A146', 'SLC4A147', 'SLC4A148', 'SLC4A149', 'SLC4A150', 'SLC4A151', 'SLC4A152', 'SLC4A153', 'SLC4A154', 'SLC4A155', 'SLC4A156', 'SLC4A157', 'SLC4A158', 'SLC4A159', 'SLC4A160', 'SLC4A161', 'SLC4A162', 'SLC4A163', 'SLC4A164', 'SLC4A165', 'SLC4A166', 'SLC4A167', 'SLC4A168', 'SLC4A169', 'SLC4A170', 'SLC4A171', 'SLC4A172', 'SLC4A173', 'SLC4A174', 'SLC4A175', 'SLC4A176', 'SLC4A177', 'SLC4A178', 'SLC4A179', 'SLC4A180', 'SLC4A181', 'SLC4A182', 'SLC4A183', 'SLC4A184', 'SLC4A185', 'SLC4A186', 'SLC4A187', 'SLC4A188', 'SLC4A189', 'SLC4A190', 'SLC4A191', 'SLC4A192', 'SLC4A193', 'SLC4A194', 'SLC4A195', 'SLC4A196', 'SLC4A197', 'SLC4A198', 'SLC4A199', 'SLC4A200', 'SLC4A201', 'SLC4A202', 'SLC4A203', 'SLC4A204', 'SLC4A205', 'SLC4A206', 'SLC4A207', 'SLC4A208', 'SLC4A209', 'SLC4A210', 'SLC4A211', 'SLC4A212', 'SLC4A213', 'SLC4A214', 'SLC4A215', 'SLC4A216', 'SLC4A217', 'SLC4A218', 'SLC4A219', 'SLC4A220', 'SLC4A221', 'SLC4A222', 'SLC4A223', 'SLC4A224', 'SLC4A225', 'SLC4A226', 'SLC4A227', 'SLC4A228', 'SLC4A229', 'SLC4A230', 'SLC4A231', 'SLC4A232', 'SLC4A233', 'SLC4A234', 'SLC4A235', 'SLC4A236', 'SLC4A237', 'SLC4A238', 'SLC4A239', 'SLC4A240', 'SLC4A241', 'SLC4A242', 'SLC4A243', 'SLC4A244', 'SLC4A245', 'SLC4A246', 'SLC4A247', 'SLC4A248', 'SLC4A249', 'SLC4A250', 'SLC4A251', 'SLC4A252', 'SLC4A253', 'SLC4A254', 'SLC4A255', 'SLC4A256', 'SLC4A257', 'SLC4A258', 'SLC4A259', 'SLC4A260', 'SLC4A261', 'SLC4A262', 'SLC4A263', 'SLC4A264', 'SLC4A265', 'SLC4A266', 'SLC4A267', 'SLC4A268', 'SLC4A269', 'SLC4A270', 'SLC4A271', 'SLC4A272', 'SLC4A273', 'SLC4A274', 'SLC4A275', 'SLC4A276', 'SLC4A277', 'SLC4A278', 'SLC4A279', 'SLC4A280', 'SLC4A281', 'SLC4A282', 'SLC4A283', 'SLC4A284', 'SLC4A285', 'SLC4A286', 'SLC4A287', 'SLC4A288', 'SLC4A289', 'SLC4A290', 'SLC4A291', 'SLC4A292', 'SLC4A293', 'SLC4A294', 'SLC4A295', 'SLC4A296', 'SLC4A297', 'SLC4A298', 'SLC4A299', 'SLC4A300', 'SLC4A301', 'SLC4A302', 'SLC4A303', 'SLC4A304', 'SLC4A305', 'SLC4A306', 'SLC4A307', 'SLC4A308', 'SLC4A309', 'SLC4A310', 'SLC4A311', 'SLC4A312', 'SLC4A313', 'SLC4A314', 'SLC4A315', 'SLC4A316', 'SLC4A317', 'SLC4A318', 'SLC4A319', 'SLC4A320', 'SLC4A321', 'SLC4A322', 'SLC4A323', 'SLC4A324', 'SLC4A325', 'SLC4A326', 'SLC4A327', 'SLC4A328', 'SLC4A329', 'SLC4A330', 'SLC4A331', 'SLC4A332', 'SLC4A333', 'SLC4A334', 'SLC4A335', 'SLC4A336', 'SLC4A337', 'SLC4A338', 'SLC4A339', 'SLC4A340', 'SLC4A341', 'SLC4A342', 'SLC4A343', 'SLC4A344', 'SLC4A345', 'SLC4A346', 'SLC4A347', 'SLC4A348', 'SLC4A349', 'SLC4A350', 'SLC4A351', 'SLC4A352', 'SLC4A353', 'SLC4A354', 'SLC4A355', 'SLC4A356', 'SLC4A357', 'SLC4A358', 'SLC4A359', 'SLC4A360', 'SLC4A361', 'SLC4A362', 'SLC4A363', 'SLC4A364', 'SLC4A365', 'SLC4A366', 'SLC4A367', 'SLC4A368', 'SLC4A369', 'SLC4A370', 'SLC4A371', 'SLC4A372', 'SLC4A373', 'SLC4A374', 'SLC4A375', 'SLC4A376', 'SLC4A377', 'SLC4A378', 'SLC4A379', 'SLC4A380', 'SLC4A381', 'SLC4A382', 'SLC4A383', 'SLC4A384', 'SLC4A385', 'SLC4A386', 'SLC4A387', 'SLC4A388', 'SLC4A389', 'SLC4A390', 'SLC4A391', 'SLC4A392', 'SLC4A393', 'SLC4A394', 'SLC4A395', 'SLC4A396', 'SLC4A397', 'SLC4A398', 'SLC4A399', 'SLC4A400', 'SLC4A401', 'SLC4A402', 'SLC4A403', 'SLC4A404', 'SLC4A405', 'SLC4A406', 'SLC4A407', 'SLC4A408', 'SLC4A409', 'SLC4A410', 'SLC4A411', 'SLC4A412', 'SLC4A413', 'SLC4A414', 'SLC4A415', 'SLC4A416', 'SLC4A417', 'SLC4A418', 'SLC4A419', 'SLC4A420', 'SLC4A421', 'SLC4A422', 'SLC4A423', 'SLC4A424', 'SLC4A425', 'SLC4A426', 'SLC4A427', 'SLC4A428', 'SLC4A429', 'SLC4A430', 'SLC4A431', 'SLC4A432', 'SLC4A433', 'SLC4A434', 'SLC4A435', 'SLC4A436', 'SLC4A437', 'SLC4A438', 'SLC4A439', 'SLC4A440', 'SLC4A441', 'SLC4A442', 'SLC4A443', 'SLC4A444', 'SLC4A445', 'SLC4A446', 'SLC4A447', 'SLC4A448', 'SLC4A449', 'SLC4A450', 'SLC4A451', 'SLC4A452', 'SLC4A453', 'SLC4A454', 'SLC4A455', 'SLC4A456', 'SLC4A457', 'SLC4A458', 'SLC4A459', 'SLC4A460', 'SLC4A461', 'SLC4A462', 'SLC4A463', 'SLC4A464', 'SLC4A465', 'SLC4A466', 'SLC4A467', 'SLC4A468', 'SLC4A469', 'SLC4A470', 'SLC4A471', 'SLC4A472', 'SLC4A473', 'SLC4A474', 'SLC4A475', 'SLC4A476', 'SLC4A477', 'SLC4A478', 'SLC4A479', 'SLC4A480', 'SLC4A481', 'SLC4A482', 'SLC4A483', 'SLC4A484', 'SLC4A485', 'SLC4A486', 'SLC4A487', 'SLC4A488', 'SLC4A489', 'SLC4A490', 'SLC4A491', 'SLC4A492', 'SLC4A493', 'SLC4A494', 'SLC4A495', 'SLC4A496', 'SLC4A497', 'SLC4A498', 'SLC4A499', 'SLC4A500', 'SLC4A501', 'SLC4A502', 'SLC4A503', 'SLC4A504', 'SLC4A505', 'SLC4A506', 'SLC4A507', 'SLC4A508', 'SLC4A509', 'SLC4A510', 'SLC4A511', 'SLC4A512', 'SLC4A513', 'SLC4A514', 'SLC4A515', 'SLC4A516', 'SLC4A517', 'SLC4A518', 'SLC4A519', 'SLC4A520', 'SLC4A521', 'SLC4A522', 'SLC4A523', 'SLC4A524', 'SLC4A525', 'SLC4A526', 'SLC4A527', 'SLC4A528', 'SLC4A529', 'SLC4A530', 'SLC4A531', 'SLC4A532', 'SLC4A533', 'SLC4A534', 'SLC4A535', 'SLC4A536', 'SLC4A537', 'SLC4A538', 'SLC4A539', 'SLC4A540', 'SLC4A541', 'SLC4A542', 'SLC4A543', 'SLC4A544', 'SLC4A545', 'SLC4A546', 'SLC4A547', 'SLC4A548', 'SLC4A549', 'SLC4A550', 'SLC4A551', 'SLC4A552', 'SLC4A553', 'SLC4A554', 'SLC4A555', 'SLC4A556', 'SLC4A557', 'SLC4A558', 'SLC4A559', 'SLC4A560', 'SLC4A561', 'SLC4A562', 'SLC4A563', 'SLC4A564', 'SLC4A565', 'SLC4A566', 'SLC4A567', 'SLC4A568', 'SLC4A569', 'SLC4A570', 'SLC4A571', 'SLC4A572', 'SLC4A573', 'SLC4A574', 'SLC4A575', 'SLC4A576', 'SLC4A577', 'SLC4A578', 'SL

'SPL', 'SLC/A6', 'SLC/A605', 'SLC/A/1', 'SLC/A8', 'SLC8A1', 'SLC8B1', 'SLC9A1', 'SLC9A2', 'SLC9A3', 'SLC9A3-AS1', 'SLC9A3P2', 'SLC9A3R1', 'SLC9A3R2', 'SLC9A4', 'SLC9A5', 'SLC9A7', 'SLC9A7P1', 'SLC9A8', 'SLC9A9', 'SLC9B2', 'SLC01A2', 'SLC01B1', 'SLC01B3', 'SLC02A1', 'SLC02B1', 'SLC03A1', 'SLC04A1', 'SLC04A1-AS1', 'SLC04C1', 'SLC05A1', 'SLF1', 'SLF2', 'SLFN11', 'SLFN12', 'SLFN12L', 'SLFN13', 'SLFN14', 'SLFN5', 'SLFN11', 'SLFN11-AS1', 'SLIRP', 'SLIT1', 'SLIT2', 'SLIT2-IT1', 'SLIT3', 'SLITRK1', 'SLITRK2', 'SLITRK3', 'SLITRK4', 'SLITRK5', 'SLITRK6', 'SLK', 'SLMAP', 'SLN', 'SLPI', 'SLTM', 'SLU7', 'SLURP1', 'SLX4', 'SLX4IP', 'SMAD1', 'SMAD2', 'SMAD3', 'SMAD4', 'SMAD5', 'SMAD6', 'SMAD7', 'SMAD9', 'SMAD9-IT1', 'SMAGP', 'SMAP1', 'SMAP2', 'SMARCA1', 'SMARCA2', 'SMARCA4', 'SMARCA5', 'SMARCA6', 'SMARCAL1', 'SMARCB1', 'SMARCC1', 'SMARCC2', 'SMARCD1', 'SMARCD2', 'SMARCD3', 'SMARCE1', 'SMC1A', 'SMC1B', 'SMC2', 'SMC3', 'SMC4', 'SMC5', 'SMC5-AS1', 'SMC6', 'SMCHD1', 'SMCO2', 'SMC03', 'SMC04', 'SMCR2', 'SMCR8', 'SMDT1', 'SMG1', 'SMG1P1', 'SMG1P2', 'SMG1P3', 'SMG1P5', 'SMG1P7', 'SMG5', 'SMG6', 'SMG7', 'SMG8', 'SMG9', 'SMILR', 'SMIM1', 'SMIM10', 'SMIM10L1', 'SMIM10L2A', 'SMIM10L2B', 'SMIM12', 'SMIM13', 'SMIM14', 'SMIM15', 'SMIM15-AS1', 'SMIM17', 'SMIM18', 'SMIM19', 'SMIM2-AS1', 'SMIM20', 'SMIM22', 'SMIM23', 'SMIM24', 'SMIM25', 'SMIM26', 'SMIM27', 'SMIM28', 'SMIM29', 'SMIM3', 'SMIM30', 'SMIM31', 'SMIM37', 'SMIM4', 'SMIM5', 'SMIM6', 'SMIM7', 'SMIM8', 'SMKRL1', 'SMN1', 'SMN2', 'SMNDC1', 'SMO', 'SMOC1', 'SMOC2', 'SMOX', 'SMPD1', 'SMPD2', 'SMPD3', 'SMPD4', 'SMPD4P1', 'SMPD5', 'SMPLD3A', 'SMPLD3B', 'SMPX', 'SMS', 'SMTN', 'SMTNL1', 'SMU1', 'SMUG1', 'SMURF1', 'SMURF2', 'SMURF2P1', 'SMYD2', 'SMYD3', 'SMYD3-IT1', 'SMYD4', 'SMYD5', 'SNA11', 'SNA12', 'SNA13', 'SNA13-AS1', 'SNAP23P1', 'SNAP25', 'SNAP25-AS1', 'SNAP29', 'SNAP47', 'SNAP91', 'SNAPC1', 'SNAPC2', 'SNAPC3', 'SNAPC4', 'SNAPC5', 'SNAPIN', 'SNCA', 'SNCA-AS1', 'SNCAIP', 'SNCB', 'SNCG', 'SND1', 'SNED1', 'SNF8', 'SNHG1', 'SNHG10', 'SNHG11', 'SNHG12', 'SNHG14', 'SNHG15', 'SNHG16', 'SNHG17', 'SNHG18', 'SNH G19', 'SNHG20', 'SNHG21', 'SNHG22', 'SNHG25', 'SNHG26', 'SNHG28', 'SNHG3', 'SNHG4', 'SNHG5', 'SNHG6', 'SNHG7', 'SNHG8', 'SNH G9', 'SNIP1', 'SNN', 'SNORA11', 'SNORA11B', 'SNORA11F', 'SNORA12', 'SNORA13', 'SNORA14A', 'SNORA14B', 'SNORA15', 'SNORA16B', 'SNORA19', 'SNORA1B', 'SNORA20', 'SNORA22', 'SNORA22B', 'SNORA22C', 'SNORA23', 'SNORA24', 'SNORA24B', 'SNORA2A', 'SNORA2B', 'SNORA2C', 'SNORA30', 'SNORA31', 'SNORA33', 'SNORA35B', 'SNORA36A', 'SNORA37', 'SNORA38', 'SNORA38B', 'SNORA3B', 'SNORA46', 'SNORA47', 'SNORA49', 'SNORA51', 'SNORA53', 'SNORA54', 'SNORA55', 'SNORA58', 'SNORA58B', 'SNORA59A', 'SNORA59B', 'SNORA5A', 'SNORA5C', 'SNORA60A', 'SNORA63C', 'SNORA65', 'SNORA66', 'SNORA68B', 'SNORA69', 'SNORA70D', 'SNORA70F', 'SNORA70G', 'SNORA70J', 'SNORA71A', 'SNORA71B', 'SNORA71C', 'SNORA71D', 'SNORA72', 'SNORA73B', 'SNORA74A', 'SNORA74B', 'SNORA74C-2', 'SNORA74D', 'SNORA75', 'SNORA77', 'SNORA79B', 'SNORA80A', 'SNORA80B', 'SNORA80C', 'SNORA80D', 'SNORA80E', 'SNORA84', 'SNORA9B', 'SNORC', 'SNORD100', 'SNORD101', 'SNORD104', 'SNORD11', 'SNORD111', 'SNORD11B', 'SNORD113-1', 'SNORD113-2', 'SNORD113-3', 'SNORD113-4', 'SNORD113-9', 'SNORD114-1', 'SNORD114-10', 'SNORD114-11', 'SNORD114-13', 'SNORD114-2', 'SNORD114-20', 'SNORD114-3', 'SNORD114-4', 'SNORD114-7', 'SNORD116-1', 'SNORD116-12', 'SNORD116-13', 'SNORD116-14', 'SNORD116-15', 'SNORD116-16', 'SNORD116-17', 'SNORD116-18', 'SNORD116-19', 'SNORD116-2', 'SNORD116-22', 'SNORD116-23', 'SNORD116-24', 'SNORD116-26', 'SNORD116-27', 'SNORD116-29', 'SNORD116-30', 'SNORD116-4', 'SNORD116-5', 'SNORD116-6', 'SNORD116-7', 'SNORD116-8', 'SNORD116-9', 'SNORD117', 'SNORD11B', 'SNORD121A', 'SNORD121B', 'SNORD123', 'SNORD124', 'SNORD126', 'SNORD127', 'SNORD12B', 'SNORD12C', 'SNORD13E', 'SNORD13P1', 'SNORD14A', 'SNORD14E', 'SNORD15A', 'SNORD15B', 'SNORD17', 'SNORD19', 'SNORD1B', 'SNORD30', 'SNORD31B', 'SNORD36C', 'SNORD37', 'SNORD38A', 'SNORD38C', 'SNORD3A', 'SNORD3B-1', 'SNORD3B-2', 'SNORD41', 'SNORD42B', 'SNORD46', 'SNORD48', 'SNORD4A', 'SNORD51', 'SNORD53', 'SNORD53B', 'SNORD56B', 'SNORD59A', 'SNORD6', 'SNORD60', 'SNORD62A', 'SNORD62B', 'SNORD63', 'SNORD63B', 'SNORD65C', 'SNORD66', 'SNORD67', 'SNORD69', 'SNORD7', 'SNORD70', 'SNORD71', 'SNORD72', 'SNORD73B', 'SNORD8', 'SNORD82', 'SNORD83A', 'SNORD88', 'SNORD9', 'SNORD90', 'SNORD91A', 'SNORD91B', 'SNORD92', 'SNORD93', 'SNORD94', 'SNORD96B', 'SNORD97', 'SNORD97', 'SNPH', 'SNRK', 'SNRK-AS1', 'SNRN200', 'SNRN25', 'SNRN27', 'SNRN35', 'SNRN40', 'SNRN48P', 'SNRN70', 'SNRPA', 'SNRPA1', 'SNRPB', 'SNRPB2', 'SNRPC', 'SNRPCP1', 'SNRPCP13', 'SNRPCP16', 'SNRPCP19', 'SNRPCP3', 'SNRPCP5', 'SNRPCP9', 'SNRPD1', 'SNRPD2', 'SNRPD3', 'SNRPE', 'SNRPEP4', 'SNRPF', 'SNRPF4', 'SNRPG', 'SNRPGP10', 'SNRPGP13', 'SNRPGP14', 'SNRGP15', 'SNRGP3', 'SNRGP4', 'SNRPN', 'SNTA1', 'SNTB1', 'SNTB2', 'SNTG2', 'SNU13', 'SNUPN', 'SNURF', 'SNW1', 'SNX1', 'SNX10', 'SNX12', 'SNX13', 'SNX14', 'SNX15', 'SNX16', 'SNX17', 'SNX18', 'SNX18P10', 'SNX18P12', 'SNX18P13', 'SNX18P1Y', 'SNX18P2', 'SNX18P26', 'SNX18P3', 'SNX18P7', 'SNX18P9', 'SNX19', 'SNX2', 'SNX20', 'SNX21', 'SNX22', 'SNX24', 'SNX25', 'SNX25P1', 'SNX27', 'SNX29', 'SNX3', 'SNX30', 'SNX32', 'SNX33', 'SNX4', 'SNX5', 'SNX6', 'SNX6P1', 'SNX7', 'SNX8', 'SNX9', 'SOAT1', 'SOAT2', 'SOBP', 'SOC51', 'SOC52', 'SOC52-AS1', 'SOC52P1', 'SOC53', 'SOC54', 'SOC55', 'SOC55P4', 'SOC56', 'SOC57', 'SOD1', 'SOD1P', 'SOD2', 'SOD3', 'SOGA1', 'SOHLH2', 'SON', 'SORBS1', 'SORBS2', 'SORBS3', 'SORCS1', 'SORCS2', 'SORCS3', 'SORD', 'SORD2P', 'SORL1', 'SORT1', 'SOS1', 'SOS1-IT1', 'SOS2', 'SOST', 'SOSTDC1', 'SOWAHA', 'SOWAHB', 'SOWAHC', 'SOWAHD', 'SOX1', 'SOX10', 'SOX11', 'SOX12', 'SOX13', 'SOX14', 'SOX15', 'SOX17', 'SOX18', 'SOX2', 'SOX2-OT', 'SOX21', 'SOX21-AS1', 'SOX3', 'SOX30', 'SOX4', 'SOX5', 'SOX5-AS1', 'SOX6', 'SOX7', 'SOX8', 'SOX9', 'SP1', 'SP100', 'SP110', 'SP140', 'SP140L', 'SP2', 'SP2-AS1', 'SP3', 'SP3P', 'SP4', 'SP5', 'SP6', 'SP8', 'SP9', 'SPA17', 'SPAAR', 'SPACA3', 'SPACA4', 'SPAC6', 'SPACAGP-AS', 'SPACA9', 'SPAG1', 'SPAG16', 'SPAG17', 'SPAG4', 'SPAG5', 'SPAG5-AS1', 'SPAG6', 'SPAG7', 'SPAG8', 'SPAG9', 'SPANXA1', 'SPANXB1', 'SPANXC', 'SPANXD', 'SPANXN1', 'SPANXN4', 'SPARC', 'SPARCL1', 'SPART', 'SPART-AS1', 'SPAST', 'SPATA12', 'SPATA13', 'SPATA16', 'SPATA17', 'SPATA18', 'SPATA2', 'SPATA20', 'SPATA20P1', 'SPATA22', 'SPATA24', 'SPATA25', 'SPATA2L', 'SPATA3', 'SPATA3-AS1', 'SPATA31C2', 'SPATA31D1', 'SPATA32', 'SPATA33', 'SPATA41', 'SPATA46', 'SPATA48', 'SPATAS5', 'SPATAS51', 'SPATA6L', 'SPATA7', 'SPATC1', 'SPATC1L', 'SPATS1', 'SPATS2', 'SPATS2L', 'SPC24', 'SPC25', 'SPCS1', 'SPCS2', 'SPCS2P4', 'SPCS3', 'SPDEF', 'SPDL1', 'SPDYC', 'SPDYE1', 'SPDYE1P', 'SPDYE12P', 'SPDYE12P', 'SPDYE19P', 'SPDYE21P', 'SPDYE21P', 'SPDYE3', 'SPDYE5', 'SPDYE6', 'SPEC1C', 'SPEC1L', 'SPEC1F', 'SPEC2', 'SPEC', 'SPEN', 'SPESP1', 'SPG11', 'SPG21', 'SPG7', 'SPHK1', 'SPHK2', 'SPII1', 'SPIB', 'SPICE1', 'SPIDR', 'SPIN1', 'SPIN2A', 'SPIN2B', 'SPIN3', 'SPIN4', 'SPINDOC', 'SPINK1', 'SPINK13', 'SPINK2', 'SPINK4', 'SPINK5', 'SPINK6', 'SPINK7', 'SPINK8', 'SPINK9', 'SPINT1', 'SPINT1-AS1', 'SPINT2', 'SPIRE1', 'SPIRE2', 'SPN', 'SPNS1', 'SPNS2', 'SPNS3', 'SPOC1D', 'SPOCK1', 'SPOCK2', 'SPOCK3', 'SPON1', 'SPON1-AS1', 'SPON2', 'SPOP', 'SPOPL', 'SPOUT1', 'SPPP1', 'SPPL2A', 'SPPL2B', 'SPPL3', 'SPR', 'SPRED1', 'SPRED2', 'SPRED3', 'SPRDN', 'SPRR1A', 'SPRR1B', 'SPRR2A', 'SPRR2D', 'SPR2E', 'SPRR2F', 'SPRR3', 'SPRTN', 'SPRY1', 'SPRY2', 'SPRY3', 'SPRY4', 'SPRY4-AS1', 'SPRY4-IT1', 'SPRYD3', 'SPRYD4', 'SPRYD7', 'SPSB1', 'SPSB2', 'SPSB3', 'SPSB4', 'SPTA1', 'SPTA1', 'SPTB1', 'SPTB1', 'SPTB2', 'SPTB4', 'SPTB5', 'SPTB5N', 'SPTL1', 'SPTL1P1', 'SPTL2C2', 'SPTL3C3', 'SPTSSB', 'SPTY20D1', 'SPTY2D10S', 'SPX', 'SQLE', 'SQOR', 'SQSTM1', 'SRA1', 'SRARP', 'SRBD1', 'SRC', 'SRCAP', 'SRCIN1', 'SRD5A1', 'SRD5A3', 'SREBF1', 'SREBF2', 'SREK1', 'SREK1IP1', 'SRE', 'SRFBP1', 'SRGAP1', 'SRGAP2', 'SRGAP2B', 'SRGAP2C', 'SRGAP2D', 'SRGAP3', 'SRGAP3-AS4', 'SRGN', 'SRI', 'SRM', 'SRMS', 'SRP14', 'SRP14-AS1', 'SRP19', 'SRP54', 'SRP54-AS1', 'SRP68', 'SRP68P1', 'SRP68P2', 'SRP68P3', 'SRP72', 'SRP9P1', 'SRPK1', 'SRPK2', 'SRPK3', 'SRPRA', 'SRPRB', 'SRPX', 'SRPX2', 'SRR', 'SRRD', 'SRM1', 'SRM1P2', 'SRRM2', 'SRRM2-AS1', 'SRRM3', 'SRRM3', 'SRRT', 'SRSF1', 'SRSF10', 'SRSF11', 'SRSF12', 'SRSF2', 'SRSF3', 'SRSF4', 'SRSF5', 'SRSF6', 'SRSF7', 'SRSF8', 'SRSF9', 'SRSF9P1', 'SRXN1', 'SS18', 'SS18L1', 'SS18L2', 'SSB', 'SSBP1', 'SSBP2', 'SSBP3', 'SSBP3-AS1', 'SSBP4', 'SSC4D', 'SSC5D', 'SSFA2', 'SSH1', 'SSH2', 'SSH3', 'SSNA1', 'SSNP', 'SSPO', 'SSR1', 'SSR2', 'SSR3', 'SSRA', 'SSR4P1', 'SSR1P1', 'SSSCA1', 'SSCA1-AS1', 'SST', 'SSTR1', 'SST2R2', 'SSTR3', 'SSTR5', 'SSTR5-AS1', 'SSU2', 'SSX1', 'SSX2', 'SSX2B', 'SSX2IP', 'SSX3', 'SSX5', 'SSX6', 'SSX8', 'ST13', 'ST13P19', 'ST13P4', 'ST13P6', 'ST14', 'ST18', 'ST20', 'ST20-AS1', 'ST20-MTHFS', 'ST3GAL1', 'ST3GAL1P1', 'ST3GAL2', 'ST3GAL3', 'ST3GAL4', 'ST3GAL5', 'ST3GAL5-AS1', 'ST3GAL6', 'ST3GAL6-AS1', 'ST5', 'ST6GAL1', 'ST6GAL2', 'ST6GAL2-IT1', 'ST6GALNA1C1', 'ST6GALNAC2', 'ST6GALNAC3', 'ST6GALNAC4', 'ST6GALNAC5', 'ST6GALNAC6', 'ST7', 'ST7-AS1', 'ST7L', 'ST8SIA1', 'ST8SIA2', 'ST8SIA4', 'ST8SIA5', 'ST8SIA6-AS1', 'STAC', 'STAC2', 'STAC3', 'STAG1', 'STAG2', 'STAG3', 'STAG3L1', 'STAG3L2', 'STAG3L3', 'STAG3L4', 'STAG3L5P', 'STAG3L5P-VPRIG2P-PILRB', 'STAM', 'STAM-AS1', 'STAM2', 'STAMB', 'STAMBPL1', 'STAP1', 'STAP2', 'STAR', 'STAR10', 'STAR1D13', 'STAR13-IT1', 'STARDB', 'STARDB3', 'STARDB4', 'STARDB5', 'STARDB5-AS1', 'STARDB5', 'STARDB7', 'STARDB8', 'STARDB9', 'STAT1', 'STAT2', 'STAT3', 'STAT4', 'STAT5A', 'STAT5B', 'STAT6', 'STAU1', 'STAU2', 'STBD1', 'STC1', 'STC2', 'STEAP1', 'STEAP1', 'STEAP2', 'STEAP3', 'STEAP3-AS1', 'STEAP4', 'STIL', 'STIM1', 'STIM2', 'STIMATE', 'STIP1', 'STK10', 'STK11', 'STK11IP', 'STK16', 'STK17A', 'STK17B', 'STK19', 'STK19B', 'STK24', 'STK25', 'STK26', 'STK3', 'STK31', 'STK32A', 'STK32A-AS1', 'STK32B', 'STK32C', 'STK33', 'STK35', 'STK36', 'STK38', 'STK38L', 'STK39', 'STK4', 'STK40', 'STKLD1', 'STMN1', 'STMN1P1', 'STMN2', 'STMN3', 'STMNND1', 'STMP1', 'STN1', 'STOM', 'STOML1', 'STOML2', 'STON1', 'STON2', 'STOX1', 'STO X2', 'STPG1', 'STPG3', 'STPG3-AS1', 'STPG4', 'STRA6', 'STRADA', 'STRADB', 'STRAP', 'STRBP', 'STRIP1', 'STRIP2', 'STRN', 'STRN3', 'STRN4', 'STS', 'STT3A', 'STT3A-AS1', 'STT3B', 'STUB1', 'STX10', 'STX11', 'STX12', 'STX16', 'STX16-NPEPL1', 'STX17', 'STV17 AC1', 'CTV10', 'CTV10 AC1', 'CTV10', 'CTV1A', 'CTV1D', 'CTV2', 'CTV3', 'CTV4', 'CTV5', 'CTV6', 'CTV7', 'CTV8', 'CTV9', 'CTV10

STXBP1', 'STXBP2', 'STXBP3', 'STXBP4', 'STXBP5', 'STXBP5-AS1', 'STXBP5L', 'STXBP6', 'STYK1', 'STYX', 'STYXL1', 'SUB1', 'SUCLA2', 'SUCLA2-AS1', 'SUCLA2P1', 'SUCLG1', 'SUCLG2', 'SUCLG2-AS1', 'SUCLG2P4', 'SUCNR1', 'SUCO', 'SDS3', 'SDS3P1', 'SUFU', 'SUGC T', 'SUGP1', 'SUGP2', 'SUGT1', 'SUGT1P1', 'SUGT1P3', 'SUGT1P4-STRAG6LP', 'SULF1', 'SULF2', 'SULT1A1', 'SULT1A2', 'SULT1A3', 'SULT1B1', 'SULT1C2', 'SULT1C3', 'SULT1C4', 'SULT1E1', 'SULT2A1', 'SULT2B1', 'SULT4A1', 'SUMF1', 'SUMF2', 'SUMO1', 'SUMO1P 1', 'SUMO1P3', 'SUMO2', 'SUMO2P12', 'SUMO2P17', 'SUMO2P19', 'SUMO3', 'SUMO4', 'SUN1', 'SUN2', 'SUN3', 'SUOX', 'SUPT16H', 'SU PT20H', 'SUPT20HL1', 'SUPT20HL2', 'SUPT3H', 'SUPT4H1', 'SUPT5H', 'SUPT6H', 'SUPT7L', 'SUPV3L1', 'SURF1', 'SURF2', 'SURF4', 'SURF6', 'SUUSD1', 'SUUSD2', 'SUUSD3', 'SUUSD4', 'SUUSD5', 'SUUSD6', 'SUV39H1', 'SUV39H2', 'SUZ12', 'SUZ12P1', 'SV2A', 'SVEP1', 'SV1L', 'SVIL-AS1', 'SVILP1', 'SVIP', 'SWAP70', 'SWIS', 'SWISAP1', 'SWIT1', 'SYAP1', 'SYBU', 'SYCE1', 'SYCE1L', 'SYCE2', 'SYCE3', 'SYCP1', 'SYCP2', 'SYCP2L', 'SYDE1', 'SYDE2', 'SYF2', 'SYK', 'SYMPK', 'SYN1', 'SYN2', 'SYNC', 'SYNCRIP', 'SYNDIG1', 'SYNE1', 'SYNE2', 'SYNE3', 'SYNE4', 'SYNGAP1', 'SYNGAP1-AS1', 'SYNGR1', 'SYNGR2', 'SYNGR3', 'SYNGR4', 'SYNJ1', 'SYNJ2', 'SY NJ2BP', 'SYNJ2BP-COX16', 'SYNM', 'SYNP0', 'SYNP02', 'SYNP02L', 'SYNPR', 'SYNPR-AS1', 'SYNRG', 'SYP', 'SYPL1', 'SYPL1P2', 'SYPL2', 'SY51', 'SYT1', 'SYT10', 'SYT11', 'SYT12', 'SYT13', 'SYT14', 'SYT15', 'SYT16', 'SYT17', 'SYT3', 'SYT4', 'SYT5', 'SYT6', 'SYT7', 'SYT8', 'SYT9', 'SYTL1', 'SYTL2', 'SYTL3', 'SYTL4', 'SYTL5', 'SYVN1', 'SZRD1', 'SZT2', 'SZT2-AS1', 'TAAR1', 'TAB1', 'TAB2', 'TAB3', 'TAB3-AS1', 'TAB3-AS2', 'TAC1', 'TAC4', 'TACC1', 'TACC2', 'TACCO1', 'TACR1', 'TACR3', 'TACSTD2', 'TADA1', 'TADA2A', 'TADA2B', 'TADA3', 'TAF1', 'TAF10', 'TAF11', 'TAF12', 'TAF13', 'TAF15', 'TAF1A', 'TAF1A-AS1', 'TAF1B', 'TAF1C', 'TAF1D', 'TAF2', 'TAF3', 'TAF4', 'TAF4B', 'TAF5', 'TAF5L', 'TAF6', 'TAF6L', 'TAF7', 'TAF7L', 'TAF8', 'TAF9', 'T AF9B', 'TAF9BP2', 'TAF9P3', 'TAGAP', 'TAGLN1', 'TAGLN2', 'TAGLN3', 'TAL1', 'TAL2', 'TALD01', 'TAMM41', 'TANC1', 'TANC2', 'TAN G02', 'TANGO6', 'TANK', 'TAOK1', 'TAOK2', 'TAOK3', 'TAP1', 'TAP2', 'TAPBP', 'TAPBP1', 'TAPT1', 'TAPT1-AS1', 'TARB1', 'TARB2', 'TARDBP', 'TARI', 'TARS', 'TARSL2', 'TASR1', 'TASR13', 'TASR210', 'TASR214', 'TASR215P', 'TASR219', 'TASR220', 'TASR23', 'TASR230', 'TASR231', 'TASR238', 'TASR24', 'TASR25', 'TASR26P', 'TASP1', 'TAT', 'TATDN1', 'TATDN2', 'TATDN2P 2', 'TATDN2P3', 'TATDN3', 'TAX1B1', 'TAX1B3', 'TAZ', 'TBATA', 'TBC1D1', 'TBC1D10A', 'TBC1D10B', 'TBC1D10C', 'TBC1D12', 'TB C1D13', 'TBC1D14', 'TBC1D15', 'TBC1D16', 'TBC1D17', 'TBC1D17', 'TBC1D2', 'TBC1D20', 'TBC1D22A', 'TBC1D22B', 'TBC1D23', 'TBC1 D24', 'TBC1D25', 'TBC1D27P', 'TBC1D28', 'TBC1D30', 'TBC1D31', 'TBC1D32', 'TBC1D33', 'TBC1D34', 'TBC1D45', 'TBC1D7', 'TBC1D8', 'TBC1D8-AS1', 'TBC1D8B', 'TBC1D9', 'TBC1D9B', 'TBCA', 'TBCAP1', 'TBCB', 'TBCC', 'TBCCD1', 'TBCD', 'TBCE', 'TBCEL', 'TBCK', 'TBK1', 'TBKBP1', 'TBL1X1', 'TBL1XR1', 'TBL1Y', 'TBL2', 'TBL3', 'TBP', 'TBP1', 'TBRG1', 'TBRG4', 'T BX1', 'TBRG4', 'TBRG4', 'TBX10', 'TBX15', 'TBX18', 'TBX19', 'TBX2', 'TBX2-AS1', 'TBX20', 'TBX21', 'TBX3', 'TBX4', 'TBX5', 'TBX5-AS1', 'TBX6', 'TBX2A R', 'TBXAS1', 'TBXT', 'TC2N', 'TCAF1', 'TCAF1P1', 'TCAF2', 'TCAF2P1', 'TCAIM', 'TCAM1P', 'TCAP', 'TCEA1', 'TCEA1P2', 'TCEA1P3', 'TCEA1 P4', 'TCEA2', 'TCEA3', 'TCEAL1', 'TCEAL2', 'TCEAL3', 'TCEAL3-AS1', 'TCEAL4', 'TCEAL7', 'TCEAL8', 'TCEAL9', 'TCEANC', 'TCEANC2', 'TCERG1', 'TCERG1L', 'TCF12', 'TCF15', 'TCF19', 'TCF20', 'TCF21', 'TCF24', 'TCF3', 'TCF4', 'TCF4-AS2', 'TCF7', 'TCF7L1', 'TCF7L2', 'TCFL5', 'TCHH', 'TCHH1', 'TCHP', 'TCIM', 'TCIRG1', 'TCL1A', 'TCL6', 'TCN1', 'TCN2', 'TCN2', 'TCOF1', 'TCP1', 'TCP11', 'TCP11L1', 'TCP11L2', 'TCTA', 'TCTE1', 'TCTE3', 'TCTEX1D2', 'TCTEX1D4', 'TCTN1', 'TCTN2', 'TCTN3', 'TDG', 'TDGF1', 'TDGF1P6', 'TDGF1P7', 'TDH', 'TD02', 'TDP1', 'TDP2', 'TDRD1', 'TDRD12', 'TDRD15', 'TDRD3', 'TDRD5', 'TDRD6', 'TDRD7', 'TDRD 9', 'TDRG1', 'TDRKH', 'TDRKH-AS1', 'TDRP', 'TEAD1', 'TEAD2', 'TEAD3', 'TEAD4', 'TEC', 'TECP1', 'TECP2', 'TECR', 'TEDC1', 'TE DCD2', 'TEF', 'TEFM', 'TEK', 'TEKT2', 'TEKT4', 'TEKT4P2', 'TEKT5', 'TELO2', 'TEMN3-AS1', 'TEM1-CDK3', 'TEM1', 'TEMN2', 'TEMN3', 'TEM4', 'TEM4', 'TEP1', 'TEPSIN', 'TERF1', 'TERF2', 'TERF2P', 'TERT', 'TES', 'TESC', 'TESK1', 'TESK2', 'TESMIN', 'TESPA1', 'TET1', 'TET2', 'TET3', 'TEX10', 'TEX11', 'TEX1B', 'TEX14', 'TEX15', 'TEX19', 'TEX2', 'TEX21P', 'TEX22', 'TEX261', 'TEX264', 'TEX29', 'TEX30', 'TEX37', 'TEX38', 'TEX41', 'TEX45', 'TEX48', 'TEX49', 'TEX53', 'TEX9', 'TF', 'TFAM', 'TFAP2A', 'TFAP2A -AS1', 'TFAP2B', 'TFAP2C', 'TFAP2E', 'TFAP4', 'TFB1M', 'TFB2M', 'TFCP2', 'TFCP2L1', 'TFDP1', 'TFDP2', 'TFDP3', 'TFE3', 'T FFB', 'TFFC', 'TFF1', 'TFF2', 'TFF3', 'TFFG', 'TFIP11', 'TFPI', 'TFPI2', 'TFPT', 'TFR2', 'TFRC', 'TGDS', 'TGFA', 'TGFB1 11', 'TGFB2', 'TGFB2-AS1', 'TGFB3', 'TGFB3-AS1', 'TGFB1', 'TGFB1R', 'TGFB2', 'TGFB2R', 'TGFB3', 'TGFB3L', 'TGFBRAP1', 'TGIF1', 'TG IF2', 'TGIF2LX', 'TGIF2P1', 'TGM1', 'TGM2', 'TGM3', 'TGM4', 'TGM5', 'TGM6', 'TGM7', 'TGLN2', 'TGS1', 'TH', 'TH2LCRR', 'THAD A', 'THAP1', 'THAP10', 'THAP11', 'THAP12', 'THAP12P3', 'THAP2', 'THAP3', 'THAP4', 'THAP5', 'THAP5P1', 'THAP6', 'THAP7', 'THA P7-AS1', 'THAP8', 'THAP9', 'THAP9-AS1', 'THBD', 'THBS1', 'THBS2', 'THBS3', 'THBS4', 'THEG', 'THEM4', 'THEMS', 'THEM6', 'THEM IS', 'THEMIS2', 'THG1L', 'THNSL1', 'THNSL2', 'THOC1', 'THOC2', 'THOC3', 'THOC5', 'THOC6', 'THOC7', 'THOC7-AS1', 'THOP1', 'TH PO', 'THRA', 'THRAP3', 'THRB', 'THRSP', 'THSD1', 'THSD4', 'THSD7A', 'THSD7B', 'THTPA', 'THUMP1', 'THUMPD2', 'THUMPD3', 'THU MPD3-AS1', 'THY1', 'THYN1', 'TIA1', 'TIAF1', 'TIAL1', 'TIAM1', 'TICAM2', 'TICRR', 'TIE1', 'TİFA', 'TİGA', 'TİGD1', 'TİGD2', 'TİGD3', 'TİGD4', 'TİGD5', 'TİGD6', 'TİGD7', 'TİGIT', 'TİMD4', 'TİMELESS', 'TİMM10', 'TİMM10B', 'Tİ MM13', 'TİMM17A', 'TİMM17B', 'TİMM17B1P', 'TİMM21', 'TİMM22', 'TİMM23', 'TİMM29', 'TİMM44', 'TİMM50', 'TİMM8A', 'TİM M8AP1', 'TİMM8B', 'TİMM9', 'TİMM9P2', 'TİMMDC1', 'TİMP1', 'TİMP2', 'TİMP3', 'ТИMP4', 'ТИAG1', 'TİNCR', 'TİINF2', 'TİPARP', 'TİPARP-AS1', 'TİPIN', 'TİPRL', 'TİRAP', 'TİJAP1', 'TİJP1', 'TİJP2', 'TİJP3', 'TK1', 'TK2', 'TKFC', 'TKT', 'TKTL1', 'Tİ LCD1', 'TİLCD2', 'TİLCD1', 'TİLCD2', 'TİLE1', 'TİLE2', 'TİLE3', 'TİLE4', 'TİLE6', 'TİLK1', 'TİLK2', 'TİLL2', 'TİLN1', 'TİLN2', 'TİLNRD1', 'TİLRL1', 'TİLR10', 'TİLR2', 'TİLR3', 'TİLR4', 'TİLR5', 'TİLR6', 'TİLR8-AS1', 'TİLX1', 'TİLX1NB', 'TİLX2', 'TİLX3', 'TİMD 201', 'TİMD22', 'TİMD23', 'TİMF51', 'TİMF51-AS1', 'TİMF45', 'TİMF45F1', 'TİMF45F19', 'TİMF45F19-TCTEX1D2', 'TİMF45F20', 'TİMF45 4', 'TİMF45F5', 'TİMF6SF1', 'TİMF6SF2', 'TİMF7SF2', 'TİMF7SF3', 'TİMF9SF1', 'TİMF9SF2', 'TİMF9SF3', 'TİMA16', 'TİMA7', 'TMBIM1', 'T MBIM4', 'TMBIM6', 'TMC1', 'TMC3', 'TMC3-AS1', 'TMC4', 'TMC5', 'TMC6', 'TMC7', 'TMC8', 'TMCC1', 'TMCC1-AS1', 'TMCC2', 'TMCC3', 'T MC01', 'TMC03', 'TMC04', 'TMC06', 'TMC06', 'TMC06', 'TME1D10P2', 'TME1D11P', 'TME2D2', 'TME3D', 'TME4D', 'TME5D', 'TME6D', 'T MED7', 'TİMED8', 'TİMED9', 'TİEFF1', 'TİEFF2', 'TİEM100', 'TİEM101', 'TİEM102', 'TİEM104', 'TİEM105', 'TİEM106A', 'TİEM106 B', 'TİEM106C', 'TİEM107', 'TİEM108', 'TİEM108-AS1', 'TİEM109', 'TİEM110', 'TİEM111', 'TİEM115', 'TİEM116', 'TİEM117', 'TİEM119', 'TİEM120A', 'TİEM120B', 'TİEM121', 'TİEM121B', 'TİEM123', 'TİEM125', 'TİEM126A', 'TİEM126B', 'TİEM127', 'TİEM12 8', 'TİEM129', 'TİEM130', 'TİEM131', 'TİEM131L', 'TİEM132A', 'TİEM132B', 'TİEM132D', 'TİEM132D-AS1', 'TİEM132E', 'TİEM134', 'TİEM135', 'TİEM136', 'TİEM138', 'TİEM139', 'TİEM140', 'TİEM141', 'TİEM143', 'TİEM144', 'TİEM145', 'TİEM147', 'TİEM147-AS1', 'TİEM148', 'TİEM149', 'TİEM14B', 'TİEM14C', 'TİEM150A', 'TİEM150B', 'TİEM150C', 'TİEM151A', 'TİEM154', 'TİEM155', 'TİEM156', 'TİEM158', 'TİEM159', 'TİEM160', 'TİEM161A', 'TİEM161B', 'TİEM161B-AS1', 'TİEM161BP1', 'TİEM163', 'TİEM164', 'TİEM165', 'TİEM167A', 'Tİ EM167B', 'TİEM168', 'TİEM169', 'TİEM171', 'TİEM170B', 'TİEM171', 'TİEM173', 'TİEM175', 'TİEM176A', 'TİEM176B', 'Tİ MEM177', 'TİEM178A', 'TİEM178B', 'TİEM179', 'TİEM179B', 'TİEM18', 'TİEM181', 'TİEM182', 'TİEM183A', 'TİEM183B', 'TİEM184A', 'TİEM184B', 'TİEM184C', 'TİEM185A', 'TİEM185B', 'TİEM186', 'TİEM187', 'TİEM188', 'TİEM189', 'TİEM190', 'TİEM191A', 'TİEM191B', 'TİEM191C', 'TİEM192', 'TİEM198', 'TİEM198B', 'TİEM199', 'TİEM200A', 'TİEM200B', 'TİEM200C', 'TİEM201', 'TİEM202-AS1', 'TİEM203', 'TİEM204', 'TİEM205', 'TİEM206', 'TİEM208', 'TİEM209', 'TİEM210', 'TİEM211', 'TİEM212', 'TİEM214', 'TİEM215', 'TİEM 216', 'TİEM217', 'TİEM218', 'TİEM219', 'TİEM220', 'TİEM221', 'TİEM222', 'TİEM223', 'TİEM225B', 'TİEM229A', 'TİEM229B', 'Tİ EM230', 'TİEM231', 'TİEM232', 'TİEM233', 'TİEM234', 'TİEM236', 'TİEM237', 'TİEM238', 'TİEM238L', 'TİEM239', 'TİEM240', 'TİEM 241', 'TİEM242', 'TİEM243', 'TİEM244', 'TİEM245', 'TİEM246', 'TİEM248', 'TİEM249', 'TİEM25', 'TİEM250', 'TİEM251', 'TİEM 252', 'TİEM253', 'TİEM254', 'TİEM254-AS1', 'TİEM255A', 'TİEM255B', 'TİEM256', 'TİEM256-PLSCR3', 'TİEM258', 'TİEM259', 'TİEM 26', 'TİEM26-AS1', 'TİEM260', 'TİEM262', 'TİEM263', 'TİEM266', 'TİEM267', 'TİEM268', 'TİEM269', 'TİEM270', 'TİEM272', 'TİEM27 3', 'TİEM30A', 'TİEM30B', 'TİEM33', 'TİEM35A', 'TİEM35B', 'TİEM37', 'TİEM38A', 'TİEM38B', 'TİEM39A', 'TİEM39B', 'TİEM 40', 'TİEM41A', 'TİEM41B', 'TİEM42', 'TİEM43', 'TİEM44', 'TİEM44-AS1', 'TİEM45', 'TİEM45A', 'TİEM45B', 'TİEM45F', 'TİEM50A', 'TİEM50B', 'TİEM 51', 'TİEM51-AS1', 'TİEM52', 'TİEM52B', 'TİEM53', 'TİEM54', 'TİEM56', 'TİEM56-RWDD3', 'TİEM59', 'TİEM59L', 'TİEM60', 'TİEM6 1', 'TİEM62', 'TİEM63A', 'TİEM63B', 'TİEM63C', 'TİEM64', 'TİEM65', 'TİEM67', 'TİEM68', 'TİEM69', 'TİEM70', 'TİEM71', 'TİEM 74', 'TİEM74B', 'TİEM75', 'TİEM79', 'TİEM80', 'TİEM81', 'TİEM86A', 'TİEM86B', 'TİEM87A', 'TİEM87B', 'TİEM88B', 'TİEM 88', 'TİEM89A', 'TİEM89B', 'TİEM99', 'TİEM99L', 'TİEM99', 'TİEM99', 'TİEM99B', 'TİEM99B', 'TİEM99C', 'TİEM99D', 'TİEM99E', 'TİEM 99F', 'TİEM99G', 'TİEM99H', 'TİEM99I', 'TİEM99J', 'TİEM99K', 'TİEM99L', 'TİEM99M', 'TİEM99N', 'TİEM99P', 'TİEM99Q', 'TİEM99R', 'TİEM99T', 'TİEM99U', 'TİEM99V', 'TİEM99W', 'TİEM99X', 'TİEM99Y', 'TİEM99Z', 'TİEM99AA', 'TİEM99AB', 'TİEM99AC', 'TİEM99AD', 'TİEM99AE', 'TİEM99AF', 'TİEM99AG', 'TİEM99AH', 'TİEM99AJ', 'TİEM99AK', 'TİEM99AL', 'TİEM99AM', 'TİEM99AN', 'TİEM99AO', 'TİEM99AP', 'TİEM99AQ', 'TİEM99AR', 'TİEM99AS', 'TİEM99AT', 'TİEM99AU', 'TİEM99AV', 'TİEM99AW', 'TİEM99AX', 'TİEM99AY', 'TİEM99AZ', 'TİEM99BA', 'TİEM99BB', 'TİEM99BC', 'TİEM99BD', 'TİEM99BE', 'TİEM99BF', 'TİEM99CG', 'TİEM99CH', 'TİEM99DH', 'TİEM99EH', 'TİEM99FH', 'TİEM99GH', 'TİEM99IJ', 'TİEM99KL', 'TİEM99MN', 'TİEM99OQ', 'TİEM99PR', 'TİEM99QV', 'TİEM99RS', 'TİEM99TU', 'TİEM99VF', 'TİEM99WZ', 'TİEM99X', 'TİEM99Y', 'TİEM99Z', 'TİEM99AA', 'TİEM99AB', 'TİEM99AC', 'TİEM99AD', 'TİEM99AE', 'TİEM99AF', 'TİEM99AG', 'TİEM99AH', 'TİEM99AJ', 'TİEM99AK', 'TİEM99AL', 'TİEM99AM', 'TİEM99AN', 'TİEM99AO', 'TİEM99AP', 'TİEM99AQ', 'TİEM99AR', 'TİEM99AS', 'TİEM99AT', 'TİEM99AU', 'TİEM99AV', 'TİEM99AW', 'TİEM99AX', 'TİEM99AY', 'TİEM99AZ', 'TİEM99BA', 'TİEM99BB', 'TİEM99BC', 'TİEM99BD', 'TİEM99BE', 'TİEM99CF', 'TİEM99DG', 'TİEM99FH', 'TİEM99GH', 'TİEM99IJ', 'TİEM99KL', 'TİEM99MN', 'TİEM99OQ', 'TİEM99PR', 'TİEM99QV', 'TİEM99RS', 'TİEM99TU', 'TİEM99VF', 'TİEM99WZ', 'TİEM99X', 'TİEM99Y', 'TİEM99Z', 'TİEM99AA', 'TİEM99AB', 'TİEM99AC', 'TİEM99AD', 'TİEM99AE', 'TİEM99AF', 'TİEM99AG', 'TİEM99AH', 'TİEM99AJ', 'TİEM99AK', 'TİEM99AL', 'TİEM99AM', 'TİEM99AN', 'TİEM99AO', 'TİEM99AP', 'TİEM99AQ', 'TİEM99AR', 'TİEM99AS', 'TİEM99AT', 'TİEM99AU', 'TİEM99AV', 'TİEM99AW', 'TİEM99AX', 'TİEM99AY', 'TİEM99AZ', 'TİEM99BA', 'TİEM99BB', 'TİEM99BC', 'TİEM99BD', 'TİEM99BE', 'TİEM99CF', 'TİEM99DG', 'TİEM99FH', 'TİEM99GH', 'TİEM99IJ', 'TİEM99KL', 'TİEM99MN', 'TİEM99OQ', 'TİEM99PR', 'TİEM99QV', 'TİEM99RS', 'TİEM99TU', 'TİEM99VF', 'TİEM99WZ', 'TİEM99X', 'TİEM99Y', 'TİEM99Z', 'TİEM99AA', 'TİEM99AB', 'TİEM99AC', 'TİEM99AD', 'TİEM99AE', 'TİEM99AF', 'TİEM99AG', 'TİEM99AH', 'TİEM99AJ', 'TİEM99AK', 'TİEM99AL', 'TİEM99AM', 'TİEM99AN', 'TİEM99AO', 'TİEM99AP', 'TİEM99AQ', 'TİEM99AR', 'TİEM99AS', 'TİEM99AT', 'TİEM99AU', 'TİEM99AV', 'TİEM99AW', 'TİEM99AX', 'TİEM99AY', 'TİEM99AZ', 'TİEM99BA', 'TİEM99BB', 'TİEM99BC', 'TİEM99BD', 'TİEM99BE', 'TİEM99CF', 'TİEM99DG', 'TİEM99FH', 'TİEM99GH', 'TİEM99IJ', 'TİEM99KL', 'TİEM99MN', 'TİEM99OQ', 'TİEM99PR', 'TİEM99QV', 'TİEM99RS', 'TİEM99TU', 'TİEM99VF', 'TİEM99WZ', 'TİEM99X', 'TİEM99Y', 'TİEM99Z', 'TİEM99AA', 'TİEM99AB', 'TİEM99AC', 'TİEM99AD', 'TİEM99AE', 'TİEM99AF', 'TİEM99AG', 'TİEM99AH', 'TİEM99AJ', 'TİEM99AK', 'TİEM99AL', 'TİEM99AM', 'TİEM99AN', 'TİEM99AO', 'TİEM99AP', 'TİEM99AQ', 'TİEM99AR', 'TİEM99AS', 'TİEM99AT', 'TİEM99AU', 'TİEM99AV', 'TİEM99AW', 'TİEM99AX', 'TİEM99AY', 'TİEM99AZ', 'TİEM99BA', 'TİEM99BB', 'TİEM99BC', 'TİEM99BD', 'TİEM99BE', 'TİEM99CF', 'TİEM99DG', 'TİEM99FH', 'TİEM99GH', 'TİEM99IJ', 'TİEM99KL', 'TİEM99MN', 'TİEM99OQ', 'TİEM99PR', 'TİEM99QV', 'TİEM99RS', 'TİEM99TU', 'TİEM99VF', 'TİEM99WZ', 'TİEM99X', 'TİEM99Y', 'TİEM99Z', 'TİEM99AA', 'TİEM99AB', 'TİEM99AC', 'TİEM99AD', 'TİEM99AE', 'TİEM99AF', 'TİEM99AG', 'TİEM99AH', 'TİEM99AJ', 'TİEM99AK', 'TİEM99AL', 'TİEM99AM', 'TİEM99AN', 'TİEM99AO', 'TİEM99AP', 'TİEM99AQ', 'TİEM99AR', 'TİEM99AS', 'TİEM99AT', 'TİEM99AU', 'TİEM99AV', 'TİEM99AW', 'TİEM99AX', 'TİEM99AY', 'TİEM99AZ', 'TİEM99BA', 'TİEM99BB', 'TİEM99BC', 'TİEM99BD', 'TİEM99BE', 'TİEM99CF', 'TİEM99DG', 'TİEM99FH', 'TİEM99GH', 'TİEM99IJ', 'TİEM99KL', 'TİEM99MN', 'TİEM99OQ', 'TİEM99PR', 'TİEM99QV', 'TİEM99RS', 'TİEM99TU', 'TİEM99VF', 'TİEM99WZ', 'TİEM99X', 'TİEM99Y', 'TİEM99Z', 'TİEM99AA', 'TİEM99AB', 'TİEM99AC', 'TİEM99AD', 'TİEM99AE', 'TİEM99AF', 'TİEM99AG', 'TİEM99AH', 'TİEM99AJ', 'TİEM99AK', 'TİEM99AL', 'TİEM99AM', 'TİEM99AN', 'TİEM99AO', 'TİEM99AP', 'TİEM99AQ', 'TİEM99AR', 'TİEM99AS', 'TİEM99AT', 'TİEM99AU', 'TİEM99AV', 'TİEM99AW', 'TİEM99AX', 'TİEM99AY', 'TİEM99AZ', 'TİEM99BA', 'TİEM99BB', 'TİEM99BC', 'TİEM99BD', 'TİEM99BE', 'TİEM99CF', 'TİEM99DG', 'TİEM99FH', 'TİEM99GH', 'TİEM99IJ', 'TİEM99KL', 'TİEM99MN', 'TİEM99OQ', 'TİEM99PR', 'TİEM99QV', 'TİEM99RS', 'TİEM99TU', 'TİEM99VF', 'TİEM99WZ', 'TİEM99X', 'TİEM99Y', 'TİEM99Z', 'TİEM99AA', 'TİEM99AB', 'TİEM99AC', 'TİEM99AD', 'TİEM99AE', 'TİEM99AF', 'TİEM99AG', 'TİEM99AH', 'TİEM99AJ', 'TİEM99AK', 'TİEM99AL', 'TİEM99AM', 'TİEM99AN', 'TİEM99AO', 'TİEM99AP', 'TİEM99AQ', 'TİEM99AR', 'TİEM99AS', 'TİEM99AT', 'TİEM99AU', 'TİEM99AV', 'TİEM99AW', 'TİEM99AX', 'TİEM99AY', 'TİEM99AZ', 'TİEM99BA', 'TİEM99BB', 'TİEM99BC', 'TİEM99BD', 'TİEM99BE', 'TİEM99CF', 'TİEM99DG', 'TİEM99FH', 'TİEM99GH', 'TİEM99IJ', 'TİEM99KL', 'TİEM99MN', 'TİEM99OQ', 'TİEM99PR', 'TİEM99QV', 'TİEM99RS', 'TİEM99TU', 'TİEM99VF', 'TİEM99WZ', 'TİEM99X', 'TİEM99Y', 'TİEM99Z', 'TİEM99AA', 'TİEM99AB', 'TİEM99AC', 'TİEM99AD', 'TİEM99AE', 'TİEM99AF', 'TİEM99AG', 'TİEM99AH', 'TİEM99AJ', 'TİEM99AK', 'TİEM99AL', 'TİEM99AM', 'TİEM99AN', 'TİEM99AO', 'TİEM99AP', 'TİEM99AQ', 'TİEM99AR', 'TİEM99AS', 'TİEM99AT', 'TİEM99AU', 'TİEM99AV', 'TİEM99AW', 'TİEM99AX', 'TİEM99AY', 'TİEM99AZ', 'TİEM99BA', 'TİEM99BB', 'TİEM99BC', 'TİEM99BD', 'TİEM99BE', 'TİEM99CF', 'TİEM99DG', 'TİEM99FH', 'TİEM99GH', 'TİEM99IJ', 'TİEM99KL', 'TİEM99MN', 'TİEM99OQ', 'TİEM99PR', 'TİEM99QV', 'TİEM99RS', 'TİEM99TU', 'TİEM99VF', 'TİEM99WZ', 'TİEM99X', 'TİEM99Y', 'TİEM99Z', 'TİEM99AA', 'TİEM99AB', 'TİEM99AC', 'TİEM99AD', 'TİEM99AE', 'TİEM99AF', 'TİEM99AG', 'TİEM99AH', 'TİEM99AJ', 'TİEM99AK', 'TİEM99AL', 'TİEM99AM', 'TİEM99AN', 'TİEM99AO', 'TİEM99AP', 'TİEM99AQ', 'TİEM99AR', 'TİEM99AS', 'TİEM99AT', 'TİEM99AU', 'TİEM99AV', 'TİEM99AW', 'TİEM99AX', 'TİEM99AY', 'TİEM99AZ', 'TİEM99BA', 'TİEM99BB', 'TİEM99BC', 'TİEM99BD', 'TİEM99BE', 'TİEM99CF', 'TİEM99DG', 'TİEM99FH', 'TİEM99GH', 'TİEM99IJ', 'TİEM99KL', 'TİEM99MN', 'TİEM99OQ', 'TİEM99PR', 'TİEM99QV', 'TİEM99RS', 'TİEM99TU', 'TİEM99VF', 'TİEM99WZ', 'TİEM99X', 'TİEM99Y', 'TİEM99Z', 'TİEM99AA', 'TİEM99AB', 'TİEM99AC', 'TİEM99AD', 'TİEM99AE', 'TİEM99AF', 'TİEM99AG', 'TİEM99AH', 'TİEM99AJ', 'TİEM99AK', 'TİEM99AL', 'TİEM99AM', 'TİEM99AN', 'TİEM99AO', 'TİEM99AP', 'TİEM99AQ', 'TİEM99AR', 'TİEM99AS', 'TİEM99AT', 'TİEM99AU', 'TİEM99AV', 'TİEM99AW', 'TİEM99AX', 'TİEM99AY', 'TİEM99AZ', 'TİEM99BA', 'TİEM99BB', 'TİEM99BC', 'TİEM99BD', 'TİEM99BE', 'TİEM99CF', 'TİEM99DG', 'TİEM99FH', 'TİEM99GH', 'TİEM99IJ', 'TİEM99KL', 'TİEM99MN', 'TİEM99OQ', 'TİEM99PR', 'TİEM99QV', 'TİEM99RS', 'TİEM99TU', 'TİEM99VF', 'TİEM99WZ', 'TİEM99X', 'TİEM99Y', 'TİEM99Z', 'TİEM99AA', 'TİEM99AB', 'TİEM99AC', 'TİEM99AD', 'TİEM99AE', 'TİEM99AF', 'TİEM99AG', 'TİEM99AH', 'TİEM99AJ', 'TİEM99AK', 'TİEM99AL', 'TİEM99AM', 'TİEM99AN', 'TİEM99AO', 'TİEM99AP', 'TİEM99AQ', 'TİEM99AR', 'TİEM99AS', 'TİEM99AT', 'TİEM99AU', 'TİEM99AV', 'TİEM99AW', 'TİEM99AX', 'TİEM99AY', 'TİEM99AZ', 'TİEM99BA', 'TİEM99BB', 'TİEM99BC', 'TİEM99BD', 'TİEM99BE', 'TİEM99CF', 'TİEM99DG', 'TİEM99FH', 'TİEM99GH', 'TİEM99IJ', 'TİEM99KL', 'TİEM99MN', 'TİEM99OQ', 'TİEM99PR', 'TİEM99QV', 'TİEM99RS', 'TİEM99TU', 'TİEM99VF', 'TİEM99WZ', 'TİEM99X', 'TİEM99Y', 'TİEM99Z', 'TİEM99AA', 'TİEM99AB', 'TİEM99AC', 'TİEM99AD', 'TİEM99AE', 'TİEM99AF', 'TİEM99AG', 'TİEM99AH', 'TİEM99AJ', 'TİEM99AK', 'TİEM99AL', 'TİEM99AM', 'TİEM99AN', 'TİEM99AO', 'TİEM99AP', 'TİEM99AQ', 'TİEM99AR', 'TİEM99AS', 'TİEM99AT', 'TİEM99AU', 'TİEM99AV', 'TİEM99AW', 'TİEM99AX', 'TİEM99AY', 'TİEM99AZ', 'TİEM99BA', 'TİEM99BB', 'TİEM99BC', 'TİEM99BD', 'TİEM99BE', 'TİEM99CF', 'TİEM99DG', 'TİEM99FH', 'TİEM99GH', 'TİEM99IJ', 'TİEM99KL', 'TİEM99MN', 'TİEM99OQ', 'TİEM99PR', 'TİEM99QV', 'TİEM99RS', 'TİEM99TU', 'TİEM99VF', 'TİEM99WZ', 'TİEM99X', 'TİEM99Y', 'TİEM99Z', 'TİEM99AA', 'TİEM99AB', 'TİEM99AC', 'TİEM99AD', 'TİEM99AE', 'TİEM99AF', 'TİEM99AG', 'TİEM99AH', 'TİEM99AJ', 'TİEM99AK', 'TİEM99AL', 'TİEM99AM', 'TİEM99AN', 'TİEM99AO', 'TİEM99AP', 'TİEM99AQ', 'TİEM99AR', 'TİEM99AS', 'TİEM99AT', 'TİEM99AU', 'TİEM99AV', 'TİEM99AW', 'TİEM99AX', 'TİEM99AY', 'TİEM99AZ', 'TİEM99BA', 'TİEM99BB', 'TİEM99BC', 'TİEM99BD', 'TİEM99BE', 'TİEM99CF', 'TİEM99DG', 'TİEM99FH', 'TİEM99GH', 'TİEM99IJ', 'TİEM99KL', 'TİEM99MN', 'TİEM99OQ', 'TİEM99PR', 'TİEM99QV', 'TİEM99RS', 'TİEM99TU', 'TİEM99VF', 'TİEM99WZ', 'TİEM99X', 'TİEM99Y', 'TİEM99Z', 'TİEM99AA', 'TİEM99AB', 'TİEM99AC', 'TİEM99AD', 'TİEM99AE', 'TİEM99AF', 'TİEM99AG', 'TİEM99AH', 'TİEM99AJ', 'TİEM99AK', 'TİEM99AL', 'TİEM99AM', 'TİEM99AN', 'TİEM99AO', 'TİEM99AP', 'TİEM99AQ', 'TİEM99AR', 'TİEM99AS', 'TİEM99AT', 'TİEM99AU', 'TİEM99AV', 'TİEM99AW', 'TİEM99AX', 'TİEM99AY', 'TİEM99AZ', 'TİEM99BA', 'TİEM99BB', 'TİEM99BC', 'TİEM99BD', 'TİEM99BE', 'TİEM99CF', 'TİEM99DG', 'TİEM99FH', 'TİEM99GH', 'TİEM99IJ', 'TİEM99KL', 'TİEM99MN', 'TİEM99OQ', 'TİEM99PR', 'TİEM99QV', 'TİEM99RS', 'TİEM99TU', 'TİEM99VF', 'TİEM99WZ', 'TİEM99X', 'TİEM99Y', 'TİEM99Z', 'TİEM99AA', 'TİEM99AB', 'TİEM99AC', 'TİEM99AD', 'TİEM99AE', 'TİEM99AF', 'TİEM99AG', 'TİEM99AH', 'TİEM99AJ', 'TİEM99AK', 'TİEM99AL', 'TİEM99AM', 'TİEM99AN', 'TİEM99AO

'B', 'TNFRSF12A', 'TNFRSF13B', 'TNFRSF13C', 'TNFRSF14', 'TNFRSF14-AS1', 'TNFRSF17', 'TNFRSF18', 'TNFRSF19', 'TNFRSF1A', 'TNFRSF1B', 'TNFRSF21', 'TNFRSF25', 'TNFRSF4', 'TNFRSF8', 'TNFRSF9', 'TNFSF10', 'TNFSF11', 'TNFSF12', 'TNFSF13', 'TNFSF13B', 'TNFSF14', 'TNFSF15', 'TNFSF18', 'TNFSF4', 'TNFSF8', 'TNFSF9', 'TNK1', 'TNK2', 'TNK2-AS1', 'TNK5', 'TNKS1BP1', 'TNKS2', 'TNKS2-AS1', 'TNMD', 'TNN', 'TNNC1', 'TNNC2', 'TNNI2', 'TNNT1', 'TNNT2', 'TNNT3', 'TNP1', 'TNP01', 'TNP02', 'TNP03', 'TNRC18', 'TNRC6A', 'TNRC6B', 'TNRC6C', 'TNRC6C-AS1', 'TNNS1', 'TNNS2', 'TNNS3', 'TNS4', 'TNXB', 'TOB1', 'TOB1-AS1', 'TOB2', 'TOB2P1', 'TOE1', 'TOGARAM1', 'TOLLIP', 'TOLLIP-AS1', 'TOM1', 'TOM1L1', 'TOM1L2', 'TOMM20', 'TOMM22', 'TOMM24', 'TOMM40L', 'TOMMS', 'TOMM70', 'TONSL', 'TONSL-AS1', 'TOP1', 'TOP1MT', 'TOP2A', 'TOP2B', 'TOP3A', 'TOP3B', 'TOPBP1', 'TOPORS', 'TOR1A', 'TOR1AIP1', 'TOR1AIP2', 'TOR1B', 'TOR2A', 'TOR3A', 'TOR4A', 'TOX', 'TOX2', 'TOX3', 'TOX4', 'TP53AI1', 'TP53BP1', 'TP53BP2', 'TP53I11', 'TP53I13', 'TP53INP1', 'TP53INP2', 'TP53RK', 'TP53TG1', 'TP53TG3D', 'TP53TG3GP', 'TP53TG5', 'TP63', 'TP73', 'TP73-AS1', 'TPB', 'TPBGL', 'TPCN1', 'TPCN2', 'TPD52', 'TPD52L1', 'TPD52L2', 'TPGS1', 'TPGS2', 'TPI1', 'TPI1P2', 'TPK1', 'TPM1', 'TPM1-AS', 'TPM2', 'TPM3', 'TPM3P6', 'TPM3P9', 'TPM4', 'TPMT1', 'TPMT1P', 'TPP1', 'TPP2', 'TPPP', 'TPPP3', 'TPR', 'TPRA1', 'TPRG1', 'TPRG1-AS1', 'TPRG1L', 'TPRG1LP1', 'TPRK', 'TPRKBP2', 'TPRN', 'TPRX1', 'TPSG1', 'TPSP2', 'TPST1', 'TPS2T', 'TPT1', 'TPT1-AS1', 'TPT1P13', 'TPT1P4', 'TPT1P5', 'TPTE', 'TPTE2P5', 'TPTE2P6', 'TPTEP1', 'TPTEP2', 'TPX2', 'TRA2A', 'TRA2B', 'TRABD', 'TRABD2A', 'TRABD2B', 'TRAC', 'TRADD', 'TRAFF1', 'TRAFF2', 'TRAFF3', 'TRAFF3IP1', 'TRAFF3IP2', 'TRAFF3IP2-AS1', 'TRAFF3IP3', 'TRAFF4', 'TRAFF5', 'TRAFF6', 'TRAFF7', 'TRAFFD1', 'TRAFFI', 'TRAJ1', 'TRAJ10', 'TRAJ11', 'TRAJ12', 'TRAJ14', 'TRAJ16', 'TRAJ21', 'TRAJ23', 'TRAJ27', 'TRAJ3', 'TRAJ31', 'TRAJ33', 'TRAJ34', 'TRAJ35', 'TRAJ36', 'TRAJ37', 'TRAJ38', 'TRAJ39', 'TRAJ4', 'TRAJ40', 'TRAJ41', 'TRAJ42', 'TRAJ42', 'TRAJ43', 'TRAJ44', 'TRAJ45', 'TRAJ47', 'TRAJ49', 'TRAJ5', 'TRAJ52', 'TRAJ59', 'TRAJ6', 'TRAJ60', 'TRAJ7', 'TRAJ8', 'TRAJ9', 'TRAK1', 'TRAK2', 'TRAM1', 'TRAM2', 'TRAM2-AS1', 'TRANK1', 'TRAP1', 'TRAPPCC1', 'TRAPPCC10', 'TRAPPCC11', 'TRAPPCC12', 'TRAPPCC12-AS1', 'TRAPPCC13', 'TRAPPCC13P1', 'TRAPPCC2', 'TRAPPCC2B', 'TRAPPCC2L', 'TRAPPCC3', 'TRAPPCC3L', 'TRAPPCC4', 'TRAPPCC5', 'TRAPPCC6A', 'TRAPPCC6B', 'TRAPPCC8', 'TRAPPCC9', 'TRARG1', 'TRAT1', 'TRAV1-1', 'TRAV1-2', 'TRAV18', 'TRAV19', 'TRAV2', 'TRAV22', 'TRAV26-2', 'TRAV29DVS', 'TRAV3', 'TRAV30', 'TRAV34', 'TRAV38-1', 'TRAV39', 'TRAV4', 'TRAV41', 'TRAV5', 'TRAV6', 'TRAV8-2', 'TRAV8-3', 'TRAV8-5', 'TRBC2', 'TRB2-1', 'TRB2-2P', 'TRB2-3', 'TRB2-4', 'TRB2-7', 'TRBV1', 'TRBV10-3', 'TRBV12-3', 'TRBV12-4', 'TRBV13', 'TRBV18', 'TRBV19', 'TRBV2', 'TRBV20-1', 'TRBV20R9-2', 'TRBV24-1', 'TRBV25-1', 'TRBV26R9-2', 'TRBV28', 'TRBV29R9-2', 'TRBV3-1', 'TRBV30', 'TRBV4-2', 'TRBV5-3', 'TRBV5-4', 'TRBV6-5', 'TRBV6-6', 'TRBV6-7', 'TRBV7-3', 'TRBV7-4', 'TRBV7-7', 'TRBV7-9', 'TRBV8-2', 'TRBV9', 'TRDC', 'TRDJ2', 'TRDJ3', 'TRDJ4', 'TRDMT1', 'TRDN', 'TR EH', 'TREHPI', 'TREM1', 'TREM2', 'TREML2', 'TRERF1', 'TREX2', 'TRG15', 'TRGC1', 'TRGC2', 'TRGJ2', 'TRGJP2', 'TR GV10', 'TRGV2', 'TRGV3', 'TRGV4', 'TRGV5', 'TRGV7', 'TRGV9', 'TRGVA', 'TRGBV', 'TRHDE', 'TRHDE-AS1', 'TRI-TAT2-2', 'TRIAP1', 'TRIB1', 'TRIB2', 'TRIB3', 'TRIL', 'TRIM10', 'TRIM11', 'TRIM13', 'TRIM14', 'TRIM15', 'TRIM16', 'TRIM16L', 'TRIM17', 'TRIM2', 'TRIM21', 'TRIM22', 'TRIM23', 'TRIM24', 'TRIM25', 'TRIM26', 'TRIM27', 'TRIM28', 'TRIM29', 'TRIM3', 'TRIM31', 'TRIM31-AS1', 'TRIM32', 'TRIM33', 'TRIM35', 'TRIM36', 'TRIM37', 'TRIM38', 'TRIM39', 'TRIM4', 'TRIM40', 'TRIM41', 'TRIM44', 'TR IM45', 'TRIM46', 'TRIM47', 'TRIM48', 'TRIM49', 'TRIM49B', 'TRIM49C', 'TRIM5', 'TRIM50', 'TRIM51', 'TRIM51CP', 'TRIM51FP', 'TR IM51GP', 'TRIM52', 'TRIM52-AS1', 'TRIM53AP', 'TRIM53CP', 'TRIM54', 'TRIM55', 'TRIM56', 'TRIM58', 'TRIM59', 'TRIM6', 'TRIM60 P17', 'TRIM60P18', 'TRIM61', 'TRIM62', 'TRIM63', 'TRIM65', 'TRIM66', 'TRIM67', 'TRIM68', 'TRIM69', 'TRIM7', 'TRIM71', 'TRIM72', 'TRIM73', 'TRIM74', 'TRIM8', 'TRIM9', 'TRIML2', 'TRIO', 'TRIP10', 'TRIP11', 'TRIP12', 'TRIP13', 'TRIP4', 'TRIP 6', 'TRIQK', 'TRIR', 'TRIT1', 'TRMO', 'TRMT1', 'TRMT10A', 'TRMT10B', 'TRMT10C', 'TRMT11', 'TRMT112', 'TRMT12', 'TRMT13', 'TR MT1', 'TRMT2A', 'TRMT2B', 'TRMT2B-AS1', 'TRMT44', 'TRMT5', 'TRMT6', 'TRMT61A', 'TRMT61B', 'TRMT9B', 'TRMU', 'TRNAU1AP', 'TR NP1', 'TRNT1', 'TRO', 'TROAP', 'TROVE2', 'TRPA1', 'TRPC1', 'TRPC3', 'TRPC4', 'TRPC4AP', 'TRPC5OS', 'TRPC6', 'TRPM1', 'TRPM 2', 'TRPM2-AS', 'TRPM3', 'TRPM4', 'TRPM5', 'TRPM6', 'TRPM8', 'TRPS1', 'TRPV2', 'TRPV3', 'TRPV4', 'TRPV5', 'TRPV6', 'TRRAP', 'TRUB1', 'TRUB2', 'TSACC', 'TSC1', 'TSC2', 'TSC22D1', 'TSC22D1-AS1', 'TSC22D2', 'TSC22D3', 'TSC22D4', 'TSE N15', 'TSEN2', 'TSEN34', 'TSEN54', 'TSFM', 'TSG101', 'TSGA10', 'TSHR', 'TSHZ1', 'TSHZ2', 'TSHZ3', 'TSKS', 'TSKU', 'TSLP', 'TS N', 'TSENARE1', 'TSENAX', 'TSENAXP1', 'TSPAN1', 'TSPAN10', 'TSPAN11', 'TSPAN12', 'TSPAN13', 'TSPAN14', 'TSPAN15', 'TSPAN17', 'TSPAN18', 'TSPAN19', 'TSPAN2', 'TSPAN3', 'TSPAN31', 'TSPAN32', 'TSPAN33', 'TSPAN34', 'TSPAN5', 'TSPAN6', 'TSPAN7', 'TSPAN8', 'TSPAN9', 'TSP EAR', 'TSPPEAR-AS1', 'TSPPEAR-AS2', 'TSP0', 'TSP02', 'TSPQAOP1', 'TSPQAOP1-AS1', 'TSPY26P', 'TSPY11', 'TSPY12', 'TSPY14', 'TSPY15', 'TSPY16', 'TSPY17', 'TSPY18', 'TSPY19', 'TSPY20P', 'TSPY20P1', 'TSPY20P2', 'TSPY20P3', 'TSPY20P4', 'TSPY20P5', 'TSPY20P6', 'TSPY20P7', 'TSPY20P8', 'TSPY20P9', 'TSPY20P10', 'TSPY20P11', 'TSPY20P12', 'TSPY20P13', 'TSPY20P14', 'TSPY20P15', 'TSPY20P16', 'TSPY20P17', 'TSPY20P18', 'TSPY20P19', 'TSPY20P20', 'TSPY20P21', 'TSPY20P22', 'TSPY20P23', 'TSPY20P24', 'TSPY20P25', 'TSPY20P26', 'TSPY20P27', 'TSPY20P28', 'TSPY20P29', 'TSPY20P30', 'TSPY20P31', 'TSPY20P32', 'TSPY20P33', 'TSPY20P34', 'TSPY20P35', 'TSPY20P36', 'TSPY20P37', 'TSPY20P38', 'TSPY20P39', 'TSPY20P40', 'TSPY20P41', 'TSPY20P42', 'TSPY20P43', 'TSPY20P44', 'TSPY20P45', 'TSPY20P46', 'TSPY20P47', 'TSPY20P48', 'TSPY20P49', 'TSPY20P50', 'TSPY20P51', 'TSPY20P52', 'TSPY20P53', 'TSPY20P54', 'TSPY20P55', 'TSPY20P56', 'TSPY20P57', 'TSPY20P58', 'TSPY20P59', 'TSPY20P60', 'TSPY20P61', 'TSPY20P62', 'TSPY20P63', 'TSPY20P64', 'TSPY20P65', 'TSPY20P66', 'TSPY20P67', 'TSPY20P68', 'TSPY20P69', 'TSPY20P70', 'TSPY20P71', 'TSPY20P72', 'TSPY20P73', 'TSPY20P74', 'TSPY20P75', 'TSPY20P76', 'TSPY20P77', 'TSPY20P78', 'TSPY20P79', 'TSPY20P80', 'TSPY20P81', 'TSPY20P82', 'TSPY20P83', 'TSPY20P84', 'TSPY20P85', 'TSPY20P86', 'TSPY20P87', 'TSPY20P88', 'TSPY20P89', 'TSPY20P90', 'TSPY20P91', 'TSPY20P92', 'TSPY20P93', 'TSPY20P94', 'TSPY20P95', 'TSPY20P96', 'TSPY20P97', 'TSPY20P98', 'TSPY20P99', 'TSPY20P100', 'TSPY20P101', 'TSPY20P102', 'TSPY20P103', 'TSPY20P104', 'TSPY20P105', 'TSPY20P106', 'TSPY20P107', 'TSPY20P108', 'TSPY20P109', 'TSPY20P110', 'TSPY20P111', 'TSPY20P112', 'TSPY20P113', 'TSPY20P114', 'TSPY20P115', 'TSPY20P116', 'TSPY20P117', 'TSPY20P118', 'TSPY20P119', 'TSPY20P120', 'TSPY20P121', 'TSPY20P122', 'TSPY20P123', 'TSPY20P124', 'TSPY20P125', 'TSPY20P126', 'TSPY20P127', 'TSPY20P128', 'TSPY20P129', 'TSPY20P130', 'TSPY20P131', 'TSPY20P132', 'TSPY20P133', 'TSPY20P134', 'TSPY20P135', 'TSPY20P136', 'TSPY20P137', 'TSPY20P138', 'TSPY20P139', 'TSPY20P140', 'TSPY20P141', 'TSPY20P142', 'TSPY20P143', 'TSPY20P144', 'TSPY20P145', 'TSPY20P146', 'TSPY20P147', 'TSPY20P148', 'TSPY20P149', 'TSPY20P150', 'TSPY20P151', 'TSPY20P152', 'TSPY20P153', 'TSPY20P154', 'TSPY20P155', 'TSPY20P156', 'TSPY20P157', 'TSPY20P158', 'TSPY20P159', 'TSPY20P160', 'TSPY20P161', 'TSPY20P162', 'TSPY20P163', 'TSPY20P164', 'TSPY20P165', 'TSPY20P166', 'TSPY20P167', 'TSPY20P168', 'TSPY20P169', 'TSPY20P170', 'TSPY20P171', 'TSPY20P172', 'TSPY20P173', 'TSPY20P174', 'TSPY20P175', 'TSPY20P176', 'TSPY20P177', 'TSPY20P178', 'TSPY20P179', 'TSPY20P180', 'TSPY20P181', 'TSPY20P182', 'TSPY20P183', 'TSPY20P184', 'TSPY20P185', 'TSPY20P186', 'TSPY20P187', 'TSPY20P188', 'TSPY20P189', 'TSPY20P190', 'TSPY20P191', 'TSPY20P192', 'TSPY20P193', 'TSPY20P194', 'TSPY20P195', 'TSPY20P196', 'TSPY20P197', 'TSPY20P198', 'TSPY20P199', 'TSPY20P200', 'TSPY20P201', 'TSPY20P202', 'TSPY20P203', 'TSPY20P204', 'TSPY20P205', 'TSPY20P206', 'TSPY20P207', 'TSPY20P208', 'TSPY20P209', 'TSPY20P210', 'TSPY20P211', 'TSPY20P212', 'TSPY20P213', 'TSPY20P214', 'TSPY20P215', 'TSPY20P216', 'TSPY20P217', 'TSPY20P218', 'TSPY20P219', 'TSPY20P220', 'TSPY20P221', 'TSPY20P222', 'TSPY20P223', 'TSPY20P224', 'TSPY20P225', 'TSPY20P226', 'TSPY20P227', 'TSPY20P228', 'TSPY20P229', 'TSPY20P230', 'TSPY20P231', 'TSPY20P232', 'TSPY20P233', 'TSPY20P234', 'TSPY20P235', 'TSPY20P236', 'TSPY20P237', 'TSPY20P238', 'TSPY20P239', 'TSPY20P240', 'TSPY20P241', 'TSPY20P242', 'TSPY20P243', 'TSPY20P244', 'TSPY20P245', 'TSPY20P246', 'TSPY20P247', 'TSPY20P248', 'TSPY20P249', 'TSPY20P250', 'TSPY20P251', 'TSPY20P252', 'TSPY20P253', 'TSPY20P254', 'TSPY20P255', 'TSPY20P256', 'TSPY20P257', 'TSPY20P258', 'TSPY20P259', 'TSPY20P260', 'TSPY20P261', 'TSPY20P262', 'TSPY20P263', 'TSPY20P264', 'TSPY20P265', 'TSPY20P266', 'TSPY20P267', 'TSPY20P268', 'TSPY20P269', 'TSPY20P270', 'TSPY20P271', 'TSPY20P272', 'TSPY20P273', 'TSPY20P274', 'TSPY20P275', 'TSPY20P276', 'TSPY20P277', 'TSPY20P278', 'TSPY20P279', 'TSPY20P280', 'TSPY20P281', 'TSPY20P282', 'TSPY20P283', 'TSPY20P284', 'TSPY20P285', 'TSPY20P286', 'TSPY20P287', 'TSPY20P288', 'TSPY20P289', 'TSPY20P290', 'TSPY20P291', 'TSPY20P292', 'TSPY20P293', 'TSPY20P294', 'TSPY20P295', 'TSPY20P296', 'TSPY20P297', 'TSPY20P298', 'TSPY20P299', 'TSPY20P300', 'TSPY20P301', 'TSPY20P302', 'TSPY20P303', 'TSPY20P304', 'TSPY20P305', 'TSPY20P306', 'TSPY20P307', 'TSPY20P308', 'TSPY20P309', 'TSPY20P310', 'TSPY20P311', 'TSPY20P312', 'TSPY20P313', 'TSPY20P314', 'TSPY20P315', 'TSPY20P316', 'TSPY20P317', 'TSPY20P318', 'TSPY20P319', 'TSPY20P320', 'TSPY20P321', 'TSPY20P322', 'TSPY20P323', 'TSPY20P324', 'TSPY20P325', 'TSPY20P326', 'TSPY20P327', 'TSPY20P328', 'TSPY20P329', 'TSPY20P330', 'TSPY20P331', 'TSPY20P332', 'TSPY20P333', 'TSPY20P334', 'TSPY20P335', 'TSPY20P336', 'TSPY20P337', 'TSPY20P338', 'TSPY20P339', 'TSPY20P340', 'TSPY20P341', 'TSPY20P342', 'TSPY20P343', 'TSPY20P344', 'TSPY20P345', 'TSPY20P346', 'TSPY20P347', 'TSPY20P348', 'TSPY20P349', 'TSPY20P350', 'TSPY20P351', 'TSPY20P352', 'TSPY20P353', 'TSPY20P354', 'TSPY20P355', 'TSPY20P356', 'TSPY20P357', 'TSPY20P358', 'TSPY20P359', 'TSPY20P360', 'TSPY20P361', 'TSPY20P362', 'TSPY20P363', 'TSPY20P364', 'TSPY20P365', 'TSPY20P366', 'TSPY20P367', 'TSPY20P368', 'TSPY20P369', 'TSPY20P370', 'TSPY20P371', 'TSPY20P372', 'TSPY20P373', 'TSPY20P374', 'TSPY20P375', 'TSPY20P376', 'TSPY20P377', 'TSPY20P378', 'TSPY20P379', 'TSPY20P380', 'TSPY20P381', 'TSPY20P382', 'TSPY20P383', 'TSPY20P384', 'TSPY20P385', 'TSPY20P386', 'TSPY20P387', 'TSPY20P388', 'TSPY20P389', 'TSPY20P390', 'TSPY20P391', 'TSPY20P392', 'TSPY20P393', 'TSPY20P394', 'TSPY20P395', 'TSPY20P396', 'TSPY20P397', 'TSPY20P398', 'TSPY20P399', 'TSPY20P400', 'TSPY20P401', 'TSPY20P402', 'TSPY20P403', 'TSPY20P404', 'TSPY20P405', 'TSPY20P406', 'TSPY20P407', 'TSPY20P408', 'TSPY20P409', 'TSPY20P410', 'TSPY20P411', 'TSPY20P412', 'TSPY20P413', 'TSPY20P414', 'TSPY20P415', 'TSPY20P416', 'TSPY20P417', 'TSPY20P418', 'TSPY20P419', 'TSPY20P420', 'TSPY20P421', 'TSPY20P422', 'TSPY20P423', 'TSPY20P424', 'TSPY20P425', 'TSPY20P426', 'TSPY20P427', 'TSPY20P428', 'TSPY20P429', 'TSPY20P430', 'TSPY20P431', 'TSPY20P432', 'TSPY20P433', 'TSPY20P434', 'TSPY20P435', 'TSPY20P436', 'TSPY20P437', 'TSPY20P438', 'TSPY20P439', 'TSPY20P440', 'TSPY20P441', 'TSPY20P442', 'TSPY20P443', 'TSPY20P444', 'TSPY20P445', 'TSPY20P446', 'TSPY20P447', 'TSPY20P448', 'TSPY20P449', 'TSPY20P450', 'TSPY20P451', 'TSPY20P452', 'TSPY20P453', 'TSPY20P454', 'TSPY20P455', 'TSPY20P456', 'TSPY20P457', 'TSPY20P458', 'TSPY20P459', 'TSPY20P460', 'TSPY20P461', 'TSPY20P462', 'TSPY20P463', 'TSPY20P464', 'TSPY20P465', 'TSPY20P466', 'TSPY20P467', 'TSPY20P468', 'TSPY20P469', 'TSPY20P470', 'TSPY20P471', 'TSPY20P472', 'TSPY20P473', 'TSPY20P474', 'TSPY20P475', 'TSPY20P476', 'TSPY20P477', 'TSPY20P478', 'TSPY20P479', 'TSPY20P480', 'TSPY20P481', 'TSPY20P482', 'TSPY20P483', 'TSPY20P484', 'TSPY20P485', 'TSPY20P486', 'TSPY20P487', 'TSPY20P488', 'TSPY20P489', 'TSPY20P490', 'TSPY20P491', 'TSPY20P492', 'TSPY20P493', 'TSPY20P494', 'TSPY20P495', 'TSPY20P496', 'TSPY20P497', 'TSPY20P498', 'TSPY20P499', 'TSPY20P500', 'TSPY20P501', 'TSPY20P502', 'TSPY20P503', 'TSPY20P504', 'TSPY20P505', 'TSPY20P506', 'TSPY20P507', 'TSPY20P508', 'TSPY20P509', 'TSPY20P510', 'TSPY20P511', 'TSPY20P512', 'TSPY20P513', 'TSPY20P514', 'TSPY20P515', 'TSPY20P516', 'TSPY20P517', 'TSPY20P518', 'TSPY20P519', 'TSPY20P520', 'TSPY20P521', 'TSPY20P522', 'TSPY20P523', 'TSPY20P524', 'TSPY20P525', 'TSPY20P526', 'TSPY20P527', 'TSPY20P528', 'TSPY20P529', 'TSPY20P530', 'TSPY20P531', 'TSPY20P532', 'TSPY20P533', 'TSPY20P534', 'TSPY20P535', 'TSPY20P536', 'TSPY20P537', 'TSPY20P538', 'TSPY20P539', 'TSPY20P540', 'TSPY20P541', 'TSPY20P542', 'TSPY20P543', 'TSPY20P544', 'TSPY20P545', 'TSPY20P546', 'TSPY20P547', 'TSPY20P548', 'TSPY20P549', 'TSPY20P550', 'TSPY20P551', 'TSPY20P552', 'TSPY20P553', 'TSPY20P554', 'TSPY20P555', 'TSPY20P556', 'TSPY20P557', 'TSPY20P558', 'TSPY20P559', 'TSPY20P560', 'TSPY20P561', 'TSPY20P562', 'TSPY20P563', 'TSPY20P564', 'TSPY20P565', 'TSPY20P566', 'TSPY20P567', 'TSPY20P568', 'TSPY20P569', 'TSPY20P570', 'TSPY20P571', 'TSPY20P572', 'TSPY20P573', 'TSPY20P574', 'TSPY20P575', 'TSPY20P576', 'TSPY20P577', 'TSPY20P578', 'TSPY20P579', 'TSPY20P580', 'TSPY20P581', 'TSPY20P582', 'TSPY20P583', 'TSPY20P584', 'TSPY20P585', 'TSPY20P586', 'TSPY20P587', 'TSPY20P588', 'TSPY20P589', 'TSPY20P590', 'TSPY20P591', 'TSPY20P592', 'TSPY20P593', 'TSPY20P594', 'TSPY20P595', 'TSPY20P596', 'TSPY20P597', 'TSPY20P598', 'TSPY20P599', 'TSPY20P600', 'TSPY20P601', 'TSPY20P602', 'TSPY20P603', 'TSPY20P604', 'TSPY20P605', 'TSPY20P606', 'TSPY20P607', 'TSPY20P608', 'TSPY20P609', 'TSPY20P610', 'TSPY20P611', 'TSPY20P612', 'TSPY20P613', 'TSPY20P614', 'TSPY20P615', 'TSPY20P616', 'TSPY20P617', 'TSPY20P618', 'TSPY20P619', 'TSPY20P620', 'TSPY20P621', 'TSPY20P622', 'TSPY20P623', 'TSPY20P624', 'TSPY20P625', 'TSPY20P626', 'TSPY20P627', 'TSPY20P628', 'TSPY20P629', 'TSPY20P630', 'TSPY20P631', 'TSPY20P632', 'TSPY20P633', 'TSPY20P634', 'TSPY20P635', 'TSPY20P636', 'TSPY20P637', 'TSPY20P638', 'TSPY20P639', 'TSPY20P640', 'TSPY20P641', 'TSPY20P642', 'TSPY20P643', 'TSPY20P644', 'TSPY20P645', 'TSPY20P646', 'TSPY20P647', 'TSPY20P648', 'TSPY20P649', 'TSPY20P650', 'TSPY20P651', 'TSPY20P652', 'TSPY20P653', 'TSPY20P654', 'TSPY20P655', 'TSPY20P656', 'TSPY20P657', 'TSPY20P658', 'TSPY20P659', 'TSPY20P660', 'TSPY20P661', 'TSPY20P662', 'TSPY20P663', 'TSPY20P664', 'TSPY20P665', 'TSPY20P666', 'TSPY20P667', 'TSPY20P668', 'TSPY20P669', 'TSPY20P670', 'TSPY20P671', 'TSPY20P672', 'TSPY20P673', 'TSPY20P674', 'TSPY20P675', 'TSPY20P676', 'TSPY20P677', 'TSPY20P678', 'TSPY20P679', 'TSPY20P680', 'TSPY20P681', 'TSPY20P682', 'TSPY20P683', 'TSPY20P684', 'TSPY20P685', 'TSPY20P686', 'TSPY20P687', 'TSPY20P688', 'TSPY20P689', 'TSPY20P690', 'TSPY20P691', 'TSPY20P692', 'TSPY20P693', 'TSPY20P694', 'TSPY20P695', 'TSPY20P696', 'TSPY20P697', 'TSPY20P698', 'TSPY20P699', 'TSPY20P700', 'TSPY20P701', 'TSPY20P702', 'TSPY20P703', 'TSPY20P704', 'TSPY20P705', 'TSPY20P706', 'TSPY20P707', 'TSPY20P708', 'TSPY20P709', 'TSPY20P710', 'TSPY20P711', 'TSPY20P712', 'TSPY20P713', 'TSPY20P714', 'TSPY20P715', 'TSPY20P716', 'TSPY20P717', 'TSPY20P718', 'TSPY20P719', 'TSPY20P720', 'TSPY20P721', 'TSPY20P722', 'TSPY20P723', 'TSPY20P724', 'TSPY20P725', 'TSPY20P726', 'TSPY20P727', 'TSPY20P728', 'TSPY20P729', 'TSPY20P730', 'TSPY20P731', 'TSPY20P732', 'TSPY20P733', 'TSPY20P734', 'TSPY20P735', 'TSPY20P736', 'TSPY20P737', 'TSPY20P738', 'TSPY20P739', 'TSPY20P740', 'TSPY20P741', 'TSPY20P742', '

'UXS1', 'UXT', 'UXT-AS1', 'VAC14', 'VAC14-AS1', 'VAMP1', 'VAMP2', 'VAMP3', 'VAMP4', 'VAMP5', 'VAMP7', 'VAMP8', 'VANGL1', 'VANGL2', 'VAPA', 'VAPB', 'VARS', 'VARS2', 'VASH1', 'VASH1-AS1', 'VASH2', 'VASN', 'VASP', 'VAT1', 'VAT1', 'VAV1', 'VAV2', 'VAV3', 'VAX1', 'VAX2', 'VBP1', 'VCAM1', 'VCAN', 'VCAN-AS1', 'VCL', 'VCP', 'VCP1P1', 'VCPKMT', 'VCX', 'VCX2', 'VCX3A', 'VCX3B', 'VDAC1', 'VDAC1P1', 'VDAC1P10', 'VDAC2', 'VDAC2P3', 'VDAC3', 'VEGFA', 'VEGFB', 'VEGFC', 'VEGFD', 'VENTX', 'VENTXP3', 'VEPH1', 'VEZF1', 'VEZT', 'VGF', 'VGLL1', 'VGLL2', 'VGLL3', 'VGLL4', 'VHL', 'VILL', 'VILL', 'VIM', 'VIM-AS1', 'VIPI', 'VIPAS39', 'VIPR1', 'VIPR1-AS1', 'VIPR2', 'VIRMA', 'VIT', 'VKORC1', 'VKORC1L1', 'VLDLR', 'VLDLR-AS1', 'VMA21', 'VMAC', 'VM01', 'VMP1', 'VN1R1', 'VN1R07P', 'VN1R08P', 'VN1R42P', 'VN1R43P', 'VN1R51P', 'VN1R53P', 'VN1R68P', 'VN1R7P', 'VN1R80P', 'VN1R81P', 'VN1R83P', 'VN1R84P', 'VN1R85P', 'VN1R88P', 'VN2R19P', 'VNN1', 'VNN2', 'VOPP1', 'VPREB1', 'VPREB3', 'VPS11', 'VPS13A', 'VPS13B', 'VPS13C', 'VPS13D', 'VPS16', 'VPS18', 'VPS25', 'VPS26A', 'VPS26A1', 'VPS26B', 'VPS26C', 'VPS28', 'VPS29', 'VPS33A', 'VPS33B', 'VPS35', 'VPS35L', 'VPS36', 'VPS37A', 'VPS37B', 'VPS37D', 'VPS39', 'VPS41', 'VPS45', 'VPS4A', 'VPS4B', 'VPS50', 'VPS51', 'VPS52', 'VPS53', 'VPS54', 'VPS72', 'VPS8', 'VPS91', 'VPS91-AS1', 'VRK1', 'VRK2', 'VRK3', 'VRTN', 'VSI1G1', 'VSI1G10', 'VSI1G10L', 'VSI1G2', 'VSI1G8', 'VSI1R', 'VSNL1', 'VSTM1', 'VSTM2B', 'VSTM2L', 'VSTM4', 'VSTM5', 'VSX1', 'VTA1', 'VTA1P1', 'VTCN1', 'VTI1A', 'VTI1B', 'VTI1B2P', 'VTN1', 'VTRNA1-2', 'VTRNA1-3', 'VTRNA2-1', 'VWA1', 'VWA2', 'VWA5A', 'VWA5B2', 'VWA7', 'VWA8', 'VWCE', 'VWDE', 'VWFP1', 'WAC', 'WAC-AS1', 'WAPL', 'WARS', 'WARS2', 'WARS2-IT1', 'WARSP2', 'WASF1', 'WASF2', 'WASF3', 'WASF5P', 'WASH2P', 'WASH3P', 'WASH6P', 'WASH7P', 'WASHC1', 'WASHC2A', 'WASHC2C', 'WASHC3', 'WASHC4', 'WASHCS', 'WASHC5-AS1', 'WASIR1', 'WASIR2', 'WASL', 'WBP1', 'WBP1L', 'WBP2', 'WBP2N', 'WBP4', 'WDCP', 'WDFY1', 'WDFY2', 'WDFY4', 'WHDH1', 'WDPCP', 'WDR1', 'WDR11', 'WDR11-AS1', 'WDR12', 'WDR13', 'WDR17', 'WDR18', 'WDR19', 'WDR20', 'WDR24', 'WDR25', 'WDR26', 'WDR27', 'WDR3', 'WDR31', 'WDR33', 'WDR34', 'WDR35', 'WDR36', 'WDR37', 'WDR38', 'WDR4', 'WDR41', 'WDR43', 'WDR44', 'WDR45', 'WDR45B', 'WDR46', 'WDR47', 'WDR48', 'WDR5', 'WDR53', 'WDR54', 'WDR55', 'WDR59', 'WDR58', 'WDR6', 'WDR60', 'WDR61', 'WDR62', 'WDR63', 'WDR66', 'WDR7', 'WDR70', 'WDR72', 'WDR73', 'WDR74', 'WDR75', 'WDR76', 'WDR77', 'WDR78', 'WDR81', 'WDR82P', 'WDR83', 'WDR83OS', 'WDR86', 'WDR86-AS1', 'WDR87', 'WDR88', 'WDR89', 'WDR90', 'WDR91', 'WDR93', 'WDR97', 'WDSUB1', 'WDT1', 'WDYHV1', 'WEEL1', 'WEE2-AS1', 'WFDC1', 'WFDC10B', 'WFDC12', 'WFDC2', 'WFDC21P', 'WFDC3', 'WFIKKN1', 'WFIKKN2', 'WFS1', 'WHAMM', 'WHAMMP2', 'WHAMMP3', 'WHRN', 'WIFI1', 'WIPF1', 'WIPF2', 'WIPF3', 'WIP1I', 'WIP1I2', 'WISP1', 'WISP2', 'WISP3', 'WIZ', 'WLS', 'WNK1', 'WNK2', 'WNK3', 'WNK4', 'WNT1', 'WNT10A', 'WNT10B', 'WNT11', 'WNT16', 'WNT2', 'WNT2B', 'WNT3', 'WNT3A', 'WNT4', 'WNT5A', 'WNT5A-AS1', 'WNT5B', 'WNT6', 'WNT7A', 'WNT7B', 'WNT8A', 'WNT8B', 'WNT9A', 'WNT9B', 'WRAP53', 'WRAP73', 'WRB', 'WRN', 'WRNIP1', 'WSB1', 'WSB2', 'WSCD1', 'WSCD2', 'WT1', 'WT1-AS1', 'WTAP', 'WTAPP1', 'WTIP', 'WWC1', 'WWC2', 'WWC2-AS2', 'WWC3', 'WWOX', 'WWP1', 'WWP2', 'WWTR1', 'WWTR1-IT1', 'XAB2', 'XAF1', 'XAGE2', 'XAGE3', 'XAGE5', 'XBPI', 'XBPIP1', 'XCL1', 'XDH', 'XG', 'XYG1', 'XIAP', 'XIRP2', 'XIST', 'XK', 'XKR3', 'XKR4', 'XKR5', 'XKR6', 'XKR7', 'XKR8', 'XKR9', 'XPA', 'XPC', 'XPNEP1', 'XPNEP2', 'XPNEP3', 'XP01', 'XP04', 'XP05', 'XP06', 'XP07', 'XPOT', 'XPR1', 'XRC01', 'XRC2', 'XRC3', 'XRC4', 'XRC5', 'XRC6', 'XRN1', 'XRN2', 'XRR1', 'XXYL1', 'XXYL1-AS1', 'XXYL1-AS2', 'XYLB', 'XYLT1', 'XYLT2', 'YAE1D1', 'YAF2', 'YAP1', 'YARS', 'YARS2', 'YBEY', 'YBX1', 'YBX1P1', 'YBX1P0', 'YBX2', 'YBX3', 'YDJC', 'YEATS2', 'YEATS2-AS1', 'YEATS4', 'YES1', 'YES1P1', 'YIF1A', 'YIF1B', 'YIPF1', 'YIPF2', 'YIPF3', 'YIPF4', 'YIPF5', 'YIPF6', 'YJEFN3', 'YJU2', 'YKT6', 'YLPM1', 'YME11', 'YOD1', 'YPEL1', 'YPEL2', 'YPEL3', 'YPEL4', 'YPEL5', 'YPEL5P2', 'YRDC', 'YRDCP2', 'YTHDC1', 'YTHDF1', 'YTHDF2', 'YTHDF3', 'YTHDF3-AS1', 'YWHAB1', 'YWHABP1', 'YWHAE', 'YWHAE1', 'YWHAE2', 'YWHAG', 'YWHAH', 'YWHAO', 'YWHAZ', 'YWHAZP3', 'YWHAZP4', 'YY1', 'YY1AP1', 'YY1P1', 'YY2', 'ZADH2', 'ZAP70', 'ZBBX', 'ZBED1', 'ZBED2', 'ZBED3', 'ZBED3-AS1', 'ZBED4', 'ZBED5', 'ZBED5-AS1', 'ZBED6', 'ZBED6CL', 'ZBED8', 'ZBED9', 'ZBP1', 'ZBTB1', 'ZBTB10', 'ZBTB11', 'ZBTB11-AS1', 'ZBTB12', 'ZBTB12P', 'ZBTB14', 'ZBTB16', 'ZBTB17', 'ZBTB18', 'ZBTB2', 'ZBTB20', 'ZBTB20-AS4', 'ZBTB21', 'ZBTB22', 'ZBTB24', 'ZBTB25', 'ZBTB26', 'ZBTB3', 'ZBTB32', 'ZBTB34', 'ZBTB37', 'ZBTB38', 'ZBTB39', 'ZBTB4', 'ZBTB40', 'ZBTB40-IT1', 'ZBTB41', 'ZBTB42', 'ZBTB43', 'ZBTB44', 'ZBTB45', 'ZBTB45P2', 'ZTB46', 'ZBTB46-AS1', 'ZBTB47', 'ZBTB48', 'ZBTB49', 'ZBTB5', 'ZBTB6', 'ZBTB7A', 'ZBTB7B', 'ZBTB7C', 'ZBTB8A', 'ZBTB8B', 'ZBTB80S', 'ZC2H1A', 'ZC2H1C1', 'ZC3H10', 'ZC3H11A', 'ZC3H11B', 'ZC3H12A', 'ZC3H12B', 'ZC3H12C', 'ZC3H12D', 'ZC3H13', 'ZC3H14', 'ZC3H15', 'ZC3H18', 'ZC3H3', 'ZC3H4', 'ZC3H6', 'ZC3H7A', 'ZC3H7B', 'ZC3H8', 'ZC3H9A1', 'ZC3H9A1V1', 'ZC4H2', 'ZCCHC10', 'ZCCHC11', 'ZCCHC12', 'ZCCHC14', 'ZCCHC17', 'ZCCHC18', 'ZCCHC2', 'ZCCHC24', 'ZCCHC3', 'ZCCHC4', 'ZCCHC6', 'ZCCHC7', 'ZCCHC8', 'ZCCHC9', 'ZCRB1', 'ZCWPW1', 'ZDBF2', 'ZDHHC1', 'ZDHHC11', 'ZDHHC12', 'ZDHHC12', 'ZDHHC13', 'ZDHHC14', 'ZDHHC15', 'ZDHHC16', 'ZDHHC17', 'ZDHHC18', 'ZDHHC19', 'ZDHHC2', 'ZDHHC20', 'ZDHHC20-IT1', 'ZDHHC20P1', 'ZDHHC21', 'ZDHHC22', 'ZDHHC23', 'ZDHHC24', 'ZDHHC3', 'ZDHHC4', 'ZDHHC5', 'ZDHHC6', 'ZDHHC7', 'ZDHHC8', 'ZDHHC8P1', 'ZDHHC9', 'ZEB1', 'ZEB1-AS1', 'ZEB2', 'ZEB2-AS1', 'ZER1', 'ZFAND1', 'ZFAND2A', 'ZFAND2B', 'ZFAND3', 'ZFAND4', 'ZFAND5', 'ZFAND6', 'ZFAS1', 'ZFAT', 'ZFC3H1', 'ZFHX2', 'ZFHX3', 'ZFHX4', 'ZFHX4-AS1', 'ZFP1', 'ZFP14', 'ZFP2', 'ZFP28', 'ZFP3', 'ZFP30', 'ZFP36', 'ZFP36L1', 'ZFP36L2', 'ZFP37', 'ZFP41', 'ZFP42', 'ZFP57', 'ZFP62', 'ZFP64', 'ZFP69', 'ZFP82', 'ZFP90', 'ZFP91', 'ZFP92', 'ZFP1', 'ZFP2M', 'ZFP2M-AS1', 'ZFR1', 'ZFR2', 'ZFX', 'ZFY', 'ZFYVE1', 'ZFYVE16', 'ZFYVE19', 'ZFYVE21', 'ZFYVE26', 'ZFYVE27', 'ZFYVE28', 'ZFYVE9', 'ZG16B', 'ZGLP1', 'ZGPAT', 'ZGRF1', 'ZHX1', 'ZHX1-C8orf76', 'ZHX2', 'ZHX3', 'ZIC1', 'ZIC2', 'ZIC3', 'ZIC4', 'ZIC5', 'ZIK1', 'ZIM2-AS1', 'ZKSCAN1', 'ZKSCAN2', 'ZKSCAN3', 'ZKSCAN4', 'ZKSCAN5', 'ZKSCAN7', 'ZMAT1', 'ZMAT2', 'ZMAT3', 'ZMAT4', 'ZMAT5', 'ZMIZ1', 'ZMIZ1-AS1', 'ZMIZ2', 'ZMPSTE24', 'ZMMY1', 'ZMMY2', 'ZMMY3', 'ZMMY4', 'ZMMY5', 'ZMMY6', 'ZMMYND10', 'ZMYND10-AS1', 'ZMYND11', 'ZMYND12', 'ZMYND15', 'ZMYND19', 'ZMYND19P1', 'ZMYND8', 'ZNF10', 'ZNF100', 'ZNF101', 'ZNF106', 'ZNF107', 'ZNF112', 'ZNF114', 'ZNF117', 'ZNF12', 'ZNF121', 'ZNF124', 'ZNF131', 'ZNF132', 'ZNF133', 'ZNF134', 'ZNF135', 'ZNF136', 'ZNF137P', 'ZNF138', 'ZNF14', 'ZNF140', 'ZNF141', 'ZNF142', 'ZNF143', 'ZNF146', 'ZNF148', 'ZNF154', 'ZNF155', 'ZNF157', 'ZNF16', 'ZNF160', 'ZNF165', 'ZNF169', 'ZNF17', 'ZNF174', 'ZNF175', 'ZNF18', 'ZNF180', 'ZNF181', 'ZNF182', 'ZNF184', 'ZNF185', 'ZNF189', 'ZNF19', 'ZNF192P1', 'ZNF195', 'ZNF197', 'ZNF197-AS1', 'ZNF2', 'ZNF200', 'ZNF202', 'ZNF204P', 'ZNF205', 'ZNF205-AS1', 'ZNF207', 'ZNF208', 'ZNF211', 'ZNF212', 'ZNF213', 'ZNF213-AS1', 'ZNF214', 'ZNF215', 'ZNF217', 'ZNF219', 'ZNF22', 'ZNF221', 'ZNF222', 'ZNF223', 'ZNF224', 'ZNF225', 'ZNF226', 'ZNF227', 'ZNF229', 'ZNF23', 'ZNF230', 'ZNF232', 'ZNF233', 'ZNF234', 'ZNF235', 'ZNF236', 'ZNF236-DT', 'ZNF239', 'ZNF24', 'ZNF248', 'ZNF25', 'ZNF250', 'ZNF251', 'ZNF252P-AS1', 'ZNF253', 'ZNF254', 'ZNF256', 'ZNF257', 'ZNF26', 'ZNF263', 'ZNF264', 'ZNF266', 'ZNF267', 'ZNF268', 'ZNF271P', 'ZNF273', 'ZNF274', 'ZNF275', 'ZNF276', 'ZNF277', 'ZNF28', 'ZNF280A', 'ZNF280B', 'ZNF280C', 'ZNF280D', 'ZNF281', 'ZNF282', 'ZNF283', 'ZNF284', 'ZNF285', 'ZNF286A', 'ZNF286B', 'ZNF287', 'ZNF292', 'ZNF296', 'ZNF29P', 'ZNF3', 'ZNF30', 'ZNF30-AS1', 'ZNF30', 'ZNF300P1', 'ZNF302', 'ZNF304', 'ZNF311', 'ZNF316', 'ZNF317', 'ZNF318', 'ZNF319', 'ZNF32', 'ZNF32-AS1', 'ZNF32-AS2', 'ZNF320', 'ZNF321P', 'ZNF322', 'ZNF324', 'ZNF326', 'ZNF329', 'ZNF330', 'ZNF331', 'ZNF333', 'ZNF334', 'ZNF335', 'ZNF337', 'ZNF337-AS1', 'ZNF33A', 'ZNF33B', 'ZNF34', 'ZNF341', 'ZNF341-AS1', 'ZNF343', 'ZNF345', 'ZNF346', 'ZNF346-IT1', 'ZNF347', 'ZNF35', 'ZNF350', 'ZNF350-AS1', 'ZNF354A', 'ZNF354B', 'ZNF354C', 'ZNF358', 'ZNF362', 'ZNF365', 'ZNF366', 'ZNF367', 'ZNF37A', 'ZNF37BP', 'ZNF37CP', 'ZNF382', 'ZNF383', 'ZNF384', 'ZNF385A', 'ZNF385B', 'ZNF385C', 'ZNF385D', 'ZNF385D-AS1', 'ZNF391', 'ZNF394', 'ZNF395', 'ZNF396', 'ZNF397', 'ZNF398', 'ZNF404', 'ZNF407', 'ZNF408', 'ZNF41', 'ZNF410', 'ZNF414', 'ZNF415', 'ZNF416', 'ZNF417', 'ZNF418', 'ZNF419', 'ZNF420', 'ZNF423', 'ZNF425', 'ZNF426', 'ZNF426-DT', 'ZNF428', 'ZNF429', 'ZNF43', 'ZNF430', 'ZNF431', 'ZNF432', 'ZNF433', 'ZNF433-AS1', 'ZNF436', 'ZNF436-AS1', 'ZNF438', 'ZNF439', 'ZNF44', 'ZNF440', 'ZNF441', 'ZNF442', 'ZNF443', 'ZNF444', 'ZNF444P1', 'ZNF445', 'ZNF446', 'ZNF447', 'ZNF448', 'ZNF449', 'ZNF451', 'ZNF454', 'ZNF454', 'ZNF460', 'ZNF460-AS1', 'ZNF461', 'ZNF462', 'ZNF467', 'ZNF468', 'ZNF469', 'ZNF470', 'ZNF471', 'ZNF473', 'ZNF474', 'ZNF479', 'ZNF48', 'ZNF480', 'ZNF483', 'ZNF484', 'ZNF485', 'ZNF486', 'ZNF487', 'ZNF488', 'ZNF490', 'ZNF491', 'ZNF492', 'ZNF493', 'ZNF496', 'ZNF497', 'ZNF500', 'ZNF501', 'ZNF502', 'ZNF503', 'ZNF503-AS1', 'ZNF503-AS2', 'ZNF506', 'ZNF507', 'ZNF510', 'ZNF511', 'ZNF512', 'ZNF512B', 'ZNF513', 'ZNF514', 'ZNF516', 'ZNF517', 'ZNF518A', 'ZNF518B', 'ZNF519', 'ZNF521', 'ZNF524', 'ZNF525', 'ZNF526', 'ZNF527', 'ZNF528', 'ZNF528-AS1', 'ZNF529', 'ZNF529-AS1', 'ZNF530', 'ZNF532', 'ZNF536', 'ZNF536', 'ZNF541', 'ZNF542P', 'ZNF543', 'ZNF544', 'ZNF546', 'ZNF547', 'ZNF548', 'ZNF549', 'ZNF550', 'ZNF551', 'ZNF552', 'ZNF554', 'ZNF555', 'ZNF556', 'ZNF557', 'ZNF558', 'ZNF559', 'ZNF560', 'ZNF561', 'ZNF561-AS1', 'ZNF562', 'ZNF563', 'ZNF565', 'ZNF566', 'ZNF567', 'ZNF568', 'ZNF569', 'ZNF57', 'ZNF570', 'ZNF571', 'ZNF571-AS1', 'ZNF572', 'ZNF573', 'ZNF574', 'ZNF575', 'ZNF576', 'ZNF577', 'ZNF578', 'ZNF579', 'ZNF580', 'ZNF581', 'ZNF582', 'ZNF582-AS1', 'ZNF583', 'ZNF584', 'ZNF585A', 'ZNF585B', 'ZNF586', 'ZNF587', 'ZNF588', 'ZNF589', 'ZNF592', 'ZNF593', 'ZNF594', 'ZNF595', 'ZNF596', 'ZNF597', 'ZNF598', 'ZNF599', 'ZNF600', 'ZNF605', 'ZNF606', 'ZNF607', 'ZNF608', 'ZNF609', 'ZNF61', 'ZNF611', 'ZNF613', 'ZNF614', 'ZNF615', 'ZNF616', 'ZNF618', 'ZNF619', 'ZNF620', 'ZNF621', 'ZNF622', 'ZNF623', 'ZNF624',

```
'ZNF626', 'ZNF627', 'ZNF628', 'ZNF629', 'ZNF630', 'ZNF638', 'ZNF638-IT1', 'ZNF639', 'ZNF641', 'ZNF644', 'ZNF646', 'ZNF648', 'ZNF649', 'ZNF649-AS1', 'ZNF652', 'ZNF653', 'ZNF654', 'ZNF655', 'ZNF658', 'ZNF66', 'ZNF660', 'ZNF662', 'ZNF664', 'ZNF665', 'ZNF667', 'ZNF667-AS1', 'ZNF668', 'ZNF669', 'ZNF670', 'ZNF671', 'ZNF672', 'ZNF674', 'ZNF675', 'ZNF676', 'ZNF677', 'ZNF678', 'ZNF679', 'ZNF680', 'ZNF681', 'ZNF682', 'ZNF684', 'ZNF687', 'ZNF688', 'ZNF689', 'ZNF69', 'ZNF691', 'ZNF692', 'ZNF695', 'ZNF696', 'ZNF697', 'ZNF699', 'ZNF7', 'ZNF70', 'ZNF700', 'ZNF701', 'ZNF702P', 'ZNF703', 'ZNF704', 'ZNF706', 'ZNF707', 'ZNF708', 'ZNF71', 'ZNF710', 'ZNF710-AS1', 'ZNF711', 'ZNF713', 'ZNF714', 'ZNF716', 'ZNF717', 'ZNF718', 'ZNF720', 'ZNF721', 'ZNF723', 'ZNF724', 'ZNF725P', 'ZNF726', 'ZNF727', 'ZNF728', 'ZNF729', 'ZNF730', 'ZNF732', 'ZNF733P', 'ZNF735', 'ZNF76', 'ZNF736P9Y', 'ZNF737', 'ZNF738', 'ZNF74', 'ZNF740', 'ZNF746', 'ZNF747', 'ZNF749', 'ZNF750', 'ZNF75A', 'ZNF75D', 'ZNF76', 'ZNF761', 'ZNF763', 'ZNF764', 'ZNF765', 'ZNF766', 'ZNF767P', 'ZNF768', 'ZNF77', 'ZNF770', 'ZNF771', 'ZNF772', 'ZNF773', 'ZNF774', 'ZNF775', 'ZNF776', 'ZNF777', 'ZNF778', 'ZNF780A', 'ZNF780B', 'ZNF781', 'ZNF782', 'ZNF783', 'ZNF784', 'ZNF785', 'ZNF786', 'ZNF787', 'ZNF788P', 'ZNF789', 'ZNF79', 'ZNF790', 'ZNF790-AS1', 'ZNF791', 'ZNF792', 'ZNF793', 'ZNF793-AS1', 'ZNF799', 'ZNF8', 'ZNF80', 'ZNF800', 'ZNF804A', 'ZNF805', 'ZNF808', 'ZNF81', 'ZNF812P', 'ZNF813', 'ZNF814', 'ZNF815P', 'ZNF816', 'ZNF816-ZNF321P', 'ZNF818P', 'ZNF821', 'ZNF823', 'ZNF826P', 'ZNF827', 'ZNF829', 'ZNF83', 'ZNF830', 'ZNF831', 'ZNF833P', 'ZNF835', 'ZNF836', 'ZNF837', 'ZNF839', 'ZNF84', 'ZNF841', 'ZNF843', 'ZNF844', 'ZNF845', 'ZNF846', 'ZNF849P', 'ZNF85', 'ZNF850', 'ZNF852', 'ZNF853', 'ZNF860', 'ZNF862', 'ZNF865', 'ZNF876P', 'ZNF878', 'ZNF879', 'ZNF880', 'ZNF883', 'ZNF887P', 'ZNF888', 'ZNF891', 'ZNF90', 'ZNF91', 'ZNF92', 'ZNF92P3', 'ZNF93', 'ZNF98', 'ZNFX1', 'ZNHIT1', 'ZNHIT2', 'ZNHIT3', 'ZNHIT6', 'ZNRD1', 'ZNRD1-ASP', 'ZNRF1', 'ZNRF2', 'ZNRF2P1', 'ZNRF2P2', 'ZP1', 'ZP2', 'ZP3', 'ZPBP2', 'ZPLD1', 'ZPR1', 'ZRANB1', 'ZRANB2', 'ZRANB3', 'ZRSR2', 'ZSCAN1', 'ZSCAN10', 'ZSCAN12', 'ZSCAN12P1', 'ZSCAN16', 'ZSCAN16-AS1', 'ZSCAN18', 'ZSCAN2', 'ZSCAN20', 'ZSCAN21', 'ZSCAN22', 'ZSCAN23', 'ZSCAN25', 'ZSCAN26', 'ZSCAN29', 'ZSCAN30', 'ZSCAN31', 'ZSCAN32', 'ZSCAN4', 'ZSCAN5A', 'ZSCAN9', 'ZSWIM1', 'ZSWIM2', 'ZSWIM3', 'ZSWIM4', 'ZSWIM5', 'ZSWIM6', 'ZSWIM7', 'ZSWIM8', 'ZSWIM9', 'ZUP1', 'ZW10', 'ZWILCH', 'ZWINT', 'ZXD', 'ZXD', 'ZXDB', 'ZXDC', 'ZYG11A', 'ZYG11AP1', 'ZYG11B', 'ZYX', 'ZZEF1', 'ZZZ3']
```

Out[33]: 22393

Integrate all genes and save pickles

```
In [34]: codes['gene']=Mapping(set(unique_genes_cell).union(set(unique_genes_drug)))
#number of unique genes
len(codes['gene'].idx2item)
```

Out[34]: 24064

```
In [35]: #Update drug's genes
drug['gene_id']=drug['gene'].apply(lambda genes: [codes['gene'].item2idx[gene] for gene in genes])
#subset of drug features
drug[['id', 'smiles', 'fps', 'gene_id']].groupby('id').first().to_pickle(data_path+'drug_features.p')
#Merge cell line's tissue, disease type with gene_id
cell_gene = cell_gene_fpkm_norm.apply(lambda row: {codes['gene'].item2idx[gene]:row[gene] for gene in cell_gene_fpkm_norm.columns}, axis=1)
cell_gene.rename(columns={0:'gene_id'}, inplace=True)
cell_gene=cell_gene.groupby('cell_id').first()
cell_line = pd.merge(cell_line_tissue, cell_gene, how='left', on='cell_id')
cell_line['gene_id']=cell_line['gene_id'].apply(lambda x: {} if type(x) is float else x)
cell_line.to_pickle(data_path+'cell_features.p')
```

```
In [36]: pickle.dump(codes, open(data_path+'codes.p', 'wb'))
```

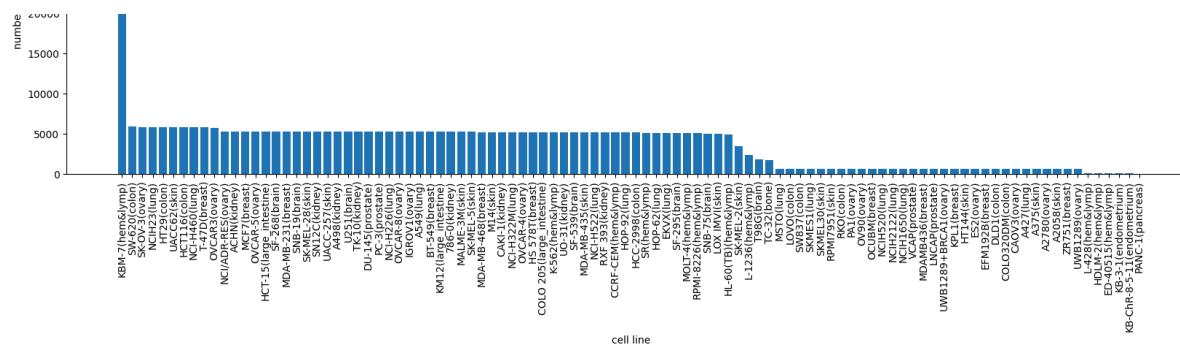
Data exploratory analysis for cell lines

```
In [37]: #Add cell line's tissue info into summary
summary_mean['tissue'] = summary_mean['cell_line_name'].apply(
    lambda x: cell_line_tissue.loc[x, 'tissue_id'] if x in cell_line_tissue.index else None
)
```

```
In [38]: cell_line_exp=summary_mean.groupby('cell_line_name').count()['drug_row'].sort_values(ascending=False)
plt.figure(figsize=(20,6))
plt.bar([codes['cell'].idx2item[cell_id] +('+'+codes['tissue'].idx2item[cell_line_tissue.loc[cell_id, 'tissue_id']] +'') for cell_id in summary_mean['cell_line_name'].unique()])
plt.xticks(rotation=90)
plt.xlabel('cell line')
plt.ylabel('number of blocks')
```

Out[38]: Text(0, 0.5, 'number of blocks')

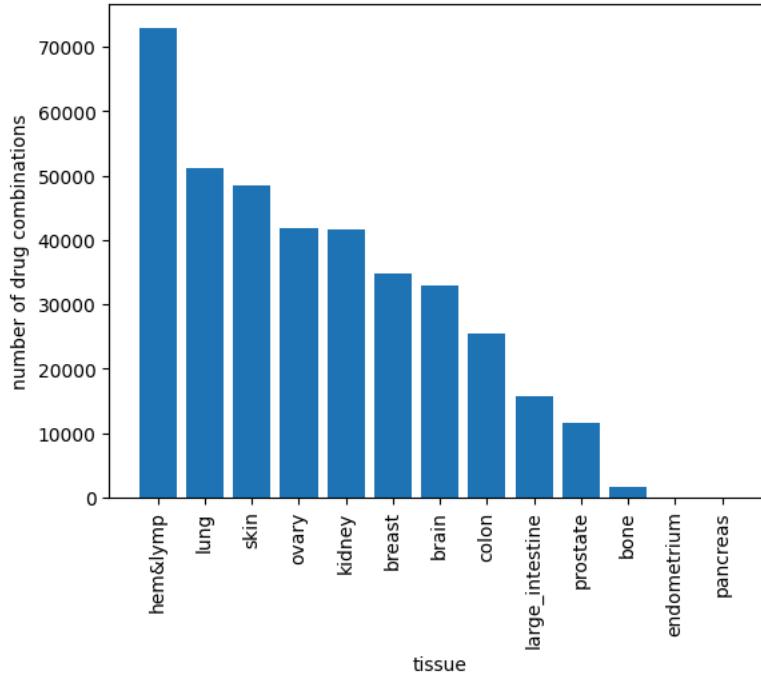




```
In [39]: tissue_exp=summary_mean.groupby('tissue').count()['drug_row'].sort_values(ascending=False)

plt.bar([codes['tissue'].idx2item[int(tissue_id)] for tissue_id in tissue_exp.index], tissue_exp)
plt.xticks(rotation=90)
plt.xlabel('tissue')
plt.ylabel('number of drug combinations')
# plt.savefig(data_path+'tissue_num_comb.png', bbox_inches='tight', dpi=200)
```

Out[39]: Text(0, 0.5, 'number of drug combinations')



```
In [40]: study_ids='', 'ONEIL', 'CLOUD', 'ALMANAC', 'FORCINA', 'NCATS_ATL', 'Mathews', 'NCATS_DIPG', 'NCATS_ES(FAKi/AURKi)', 'NCATS_ES(Nampt+PARP)', 'NCATS_MDR_CS', 'Phelan', 'NCATS_ATL', 'NCATS_DIPG', 'NCATS_ES(FAKi/AURKi)', 'NCATS_ES(Nampt+PARP)', 'ONEIL']
num_blocks=summary_mean.groupby(['study_id', 'tissue'], as_index=False)['drug_row'].count().rename(columns={'drug_row': 'num_blocks'})
num_blocks['tissue']=num_blocks['tissue'].apply(lambda x: codes['tissue'].idx2item[int(x)])
num_blocks['study_id']=num_blocks['study_id'].apply(lambda x: study_ids[x])

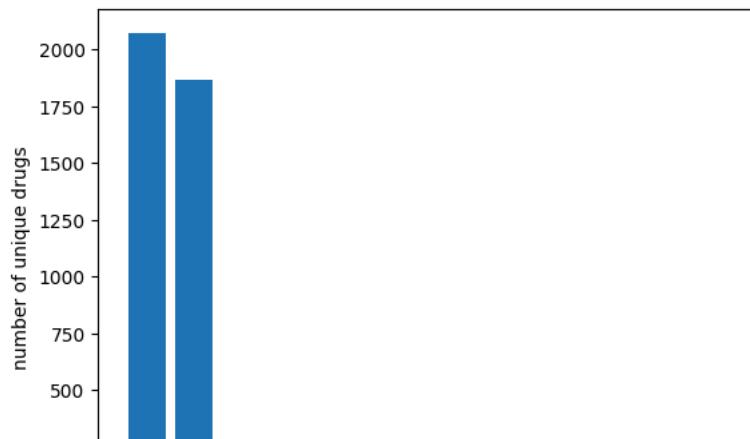
num_blocks_pivot = num_blocks.pivot_table(
    index='study_id',
    columns='tissue',
    values='num_blocks',
    aggfunc='sum'
).fillna(0)
study_ids = ['ALMANAC', 'CLOUD', 'NCATS_HL', 'FORCINA', 'Mathews', 'Wilson', 'Yohe', 'NCATS_MDR_CS', 'Phelan', 'NCATS_ATL', 'NCATS_DIPG', 'NCATS_ES(FAKi/AURKi)', 'NCATS_ES(Nampt+PARP)', 'ONEIL']
tissues = ['brain', 'breast', 'colon', 'endometrium', 'hem&lymph', 'kidney', 'lung', 'ovary', 'bone', 'prostate']
num_blocks_pivot = num_blocks_pivot.reindex(index=study_ids, columns=tissues).fillna(0)
```

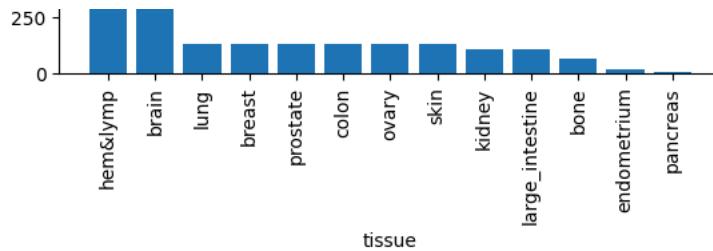
```
In [41]: log_norm=LogNorm(vmin=num_blocks_pivot.min().min()+1, vmax=num_blocks_pivot.max().max())
```



```
In [42]: #list unique drugs per tissue
tissue_drugs = pd.merge(summary_mean.groupby('tissue')[['drug_col']].apply(set).reset_index(name='drug_row'),
                       summary_mean.groupby('tissue')[['drug_col']].apply(set).reset_index(name='drug_row'),
                       on='tissue')
tissue_drugs['drugs']=tissue_drugs.apply(lambda row: row[1].union(row[2]), axis=1)
tissue_drugs.drop(['drug_row_x','drug_row_y'], axis=1, inplace=True)
tissue_drugs['num_drugs']=tissue_drugs['drugs'].apply(lambda x: len(x))
tissue_drugs.sort_values('num_drugs', ascending=False, inplace=True)
plt.bar([codes['tissue'].idx2item[int(tissue_id)] for tissue_id in tissue_drugs['tissue']], tissue_drugs['num_drugs'])
plt.xticks(rotation=90)
plt.xlabel('tissue')
plt.ylabel('number of unique drugs')
#plt.savefig(data_path+'tissue_num_drugs.png', bbox_inches='tight', dpi=200)
```

Out[42]: Text(0, 0.5, 'number of unique drugs')





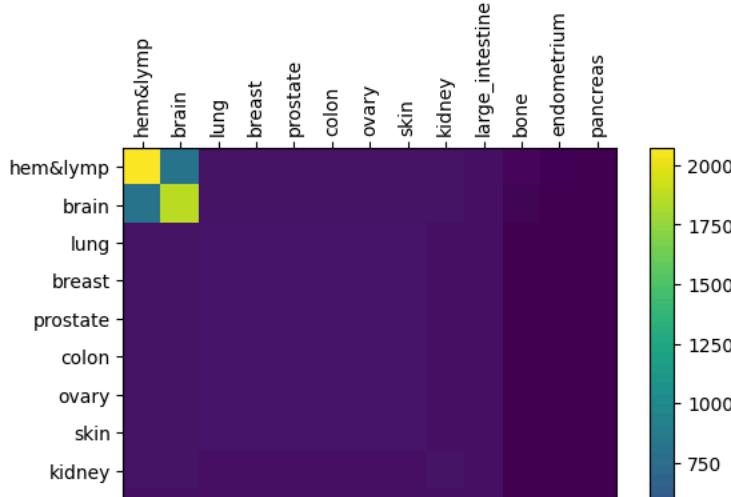
```
In [43]: num_intersects=np.zeros((len(tissue_drugs),len(tissue_drugs)))
for i, tissue_id in enumerate(tissue_drugs['tissue']):
    num_intersects[i,:]=tissue_drugs['drugs'].apply(lambda x: len(x.intersection(tissue_drugs.loc[tissue_id, 'drugs']))).v

fig,ax=plt.subplots()
im=ax.imshow(num_intersects)
ax.tick_params(top=True, bottom=False, labeltop=True, labelbottom=False)
ax.set_xticks(np.arange(len(tissue_drugs)))
ax.set_yticks(np.arange(len(tissue_drugs)))
ax.set_xticklabels([codes['tissue'].idx2item[int(tissue_id)] for tissue_id in tissue_drugs['tissue']])
ax.set_yticklabels([codes['tissue'].idx2item[int(tissue_id)] for tissue_id in tissue_drugs['tissue']])

fig.colorbar(im, ax=ax)

plt.setp(ax.get_xticklabels(), rotation=90)
#fig.tight_layout()
#plt.savefig(data_path+'heatmap.png', bb_inches='tight', dpi=200)
```

Out[43]: [None,
 None,
 None,
 None,
 None,
 None,
 None,
 None,
 None,
 None,
 None,
 None,
 None,
 None,
 None,
 None,
 None,
 None,
 None,
 None,
 None,
 None,
 None,
 None,
 None,
 None,
 None,
 None,
 None,
 None,
 None,
 None,
 None]





References

1. Jaeger, S., Duran-Frigola, M. & Aloy, P. Drug sensitivity in cancer cell lines is not tissue-specific. *Mol Cancer* 14, 40 (2015). <https://doi.org.proxy2.library.illinois.edu/10.1186/s12943-015-0312-6>
2. Kim, Y., Zheng, S., Tang, J., Jim Zheng, W., Li, Z., & Jiang, X. (2021). Anticancer drug synergy prediction in understudied tissues using transfer learning. *Journal of the American Medical Informatics Association : JAMIA*, 28(1), 42–51. <https://doi.org/10.1093/jamia/ocaa212>
3. Yao F, Madani Tonekaboni SA, Safikhani Z, Smirnov P, El-Hachem N, Freeman M, Manem VSK, Haibe-Kains B. Tissue specificity of in vitro drug sensitivity. *J Am Med Inform Assoc.* 2018 Feb 1;25(2):158-166. doi: 10.1093/jamia/ocx062. PMID: 29016819; PMCID: PMC6381764.
4. Zheng, S., Aldahdooh, J., Shadbahr, T., Wang, Y., Aldahdooh, D., Bao, J., Wang, W., & Tang, J. (2021). Drugcomb update: A more comprehensive drug sensitivity data repository and Analysis Portal. *Nucleic Acids Research*, 49(W1). <https://doi.org/10.1093/nar/gkab438>
5. Kim, Y. (2024). Genetic Features, https://drive.google.com/drive/folders/1ORFe0hw_oVCjnkJupji6ah2HwJGzSe3E?usp=sharing

 Open in Colab

```
In [1]: from google.colab import drive
drive.mount('/content/drive')

%cd /content/drive/My Drive/CS598_DLH/Dataset2

Mounted at /content/drive
/content/drive/My Drive/CS598_DLH/Dataset2
```

Model Building

This section will go over the implementation of the model

Our Citation for the original paper is in the References section at [2].

Original code is here: <https://github.com/yejinjkim/synergy-transfer>

```
In [2]: import os
import torch
import numpy as np
import pandas as pd
import pdb
import time
import pickle
import logging
import matplotlib.pyplot as plt
from utilities import Mapping

import torch.nn as nn
from torch.autograd import Variable
import torch.optim as optim
import torch.nn.functional as F
from torch.utils.data import Dataset, DataLoader

from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn import metrics
```

```
In [3]: data_path=''
save_path=''

bsz=128
cuda=True
device=3
#torch.cuda.set_device(device)

num_gene_compressed_drug=64
num_gene_compressed_cell=128

#isClassification=True #False for regression task
syn_threshold=30
ri_threshold=50

log_interval=100
epochs=10
```

Load data

```
In [4]: #drug pair, cell, scores
df=pickle.load(open(data_path+'summary_mean.p', 'rb'))
codes=pickle.load(open(data_path+'codes.p', 'rb'))
```

```
In [5]: df.head()
```

```
Out[5]: drug_row drug_col cell_line_name study_id ri_row ri_col synergy_loewe
0 2617 2002 0 1 21 070100 17 202580 1 126121
```

	Drug ID	Drug Name	Drug Type	Drug Feature 1	Drug Feature 2	Drug Feature 3	Drug Feature 4
1	1515	3003	0	4	-4.051616	17.392589	10.755529
2	904	1413	1	11	-71.949000	9.755000	-6.188270
3	904	3003	0	4	-9.231525	17.392589	0.739056
4	63	1413	1	11	-12.272000	15.133000	-18.680123

```
In [6]: # Drug's external features
drug_features=pickle.load(open(data_path+'drug_features.p', 'rb'))

In [7]: # Cell's external features
cell_features=pickle.load(open(data_path+'cell_features.p', 'rb'))

In [8]: num_celllines= len(codes['cell'].idx2item)
num_drugs=len(codes['drugs'].idx2item)

num_genes = len(codes['gene'].idx2item)
num_tissue = len(codes['tissue'].idx2item)
num_disease = len(codes['disease'].idx2item)

num_drug_fp=len(drug_features.loc[0,'fps'])
max_drug_sm_len = drug_features['smiles'].apply(lambda x: len(x)).max()
```

Gene compression

This section will go over the dataloaders that cover the cell lines compression for the model training and the dataloaders that cover the drug target compression.

Dataset and Dataloader

```
In [9]: class DrugTargetDataset(Dataset):
    def __init__(self, drug_features):
        self.drug_features = drug_features
    def __len__(self):
        return len(self.drug_features)
    def __getitem__(self,idx):
        gene_ids=self.drug_features.loc[idx, 'gene_id']
        genes=np.zeros(num_genes)
        genes[gene_ids]=1
        return genes

class CellGeneDataset(Dataset):
    def __init__(self, cell_features):
        self.cell_features = cell_features
    def __len__(self):
        return len(self.cell_features)
    def __getitem__(self,idx):
        gene_ids=self.cell_features.loc[idx, 'gene_id']
        genes = np.zeros(num_genes)
        for key,value in gene_ids.items():
            genes[key]=value
        return genes
```

```
In [10]: #Two layers of fully connected layers
class FC2(nn.Module):
    def __init__(self, in_features, out_features, dropout):
        super(FC2, self).__init__()

        self.bn = nn.BatchNorm1d(in_features)
        self.fc1 = nn.Linear(in_features, int(in_features/2))
        self.fc2 = nn.Linear(int(in_features/2),out_features)
        self.dropout= nn.Dropout(dropout)

    def forward(self, x):
        x = self.bn(x)
        x = self.dropout(x)
```

```

..  -----
x = self.dropout(F.relu(self.fc1(x)))
x = self.fc2(x)

return x

```

Compress gene features

Here is where the compression happens. For cell lines COSMIC and CCLE are combined.

For drug targets, DrugBank, NIH-LINC, and TTD are combined.

In [11]:

```

class GeneCompressor(nn.Module):
    def __init__(self, num_in, num_out, dropout=0.1):
        super(GeneCompressor, self).__init__()
        self.dropout=dropout
        self.encoder=nn.Linear(num_in, num_out)
        self.decoder=nn.Linear(num_out,num_in)

    def _encoder(self,x):
        return F.dropout(F.relu(self.encoder(x)), self.dropout, training=self.training)

    def _decoder(self,x):
        return F.dropout(self.decoder(x), self.dropout, training=self.training)

    def forward(self, x):
        x=self._encoder(x)
        x=self._decoder(x)
        return x

```

In [12]:

```

def geneCompressing(data_loader,num_gene_compressed, noise_weight=0.2, epochs=20, log_interval=10 ):
    #model
    geneCompressor=GeneCompressor(num_genes, num_out=num_gene_compressed, dropout=0.1)
    if cuda:
        geneCompressor=geneCompressor.cuda()
    criterion=nn.MSELoss()
    optimizer=optim.Adam(geneCompressor.parameters())
    for epoch in range(1,epochs+1):
        #train
        geneCompressor.train()
        total_loss=0
        start_time=time.time()
        for iteration, gene in enumerate(data_loader):
            gene=Variable(gene).float()
            noise=noise_weight*torch.randn(gene.shape)

            if cuda:
                gene=gene.cuda()
                noise=noise.cuda()
            optimizer.zero_grad()
            output=geneCompressor(gene+noise)
            loss=criterion(output,gene)
            loss.backward()
            optimizer.step()
            total_loss += loss.data
            if iteration % log_interval == 0 and iteration > 0:
                cur_loss = total_loss.item() / log_interval
                elapsed = time.time() - start_time
                print('| epoch {:3d} | {:5d}/{:5d} batches | ms/batch {:.5f} | loss {:.5f}'.format(epoch, iteration, int(log_interval), elapsed, cur_loss))
                total_loss = 0
                start_time = time.time()
        #test
        geneCompressor.eval()
        total_loss=0
        start_time=time.time()
        with torch.no_grad():
            for iteration, gene in enumerate(test_data_loader):
                gene=Variable(gene).float()
                if cuda:
                    gene=gene.cuda(device)
                output=geneCompressor(gene)
                loss=criterion(output,gene)
                total_loss += loss.data
            print(total_loss.item()/iteration)
    return geneCompressor

```

In []:

```
#drug's target gene data
drugGeneDataset=DrugTargetDataset(drug_features)
drugGeneDataset_loader = DataLoader(drugGeneDataset, batch_size=64, shuffle=True)
#Learn
drugGeneCompressor=geneCompressing(drugGeneDataset_loader, num_gene_compressed_drug)
#Save
drugGeneCompressor.eval()
drugGeneCompressed=np.array([drugGeneCompressor.cpu().encoder(torch.FloatTensor(drugGeneDataset[d])).data.numpy() for d in range(len(drugGeneDataset))])
torch.save(drugGeneCompressor.state_dict(), data_path+'drugGeneCompressor.p')
pickle.dump(drugGeneCompressed, open(data_path+'drugGeneCompressed.p', 'wb'))
```

epoch	1	10/	0 batches	ms/batch	99.24	loss	0.00786
epoch	1	20/	0 batches	ms/batch	24.56	loss	0.00518
epoch	1	30/	0 batches	ms/batch	26.08	loss	0.00521
epoch	1	40/	0 batches	ms/batch	24.08	loss	0.00482
epoch	2	10/	0 batches	ms/batch	24.55	loss	0.00473
epoch	2	20/	0 batches	ms/batch	24.11	loss	0.00414
epoch	2	30/	0 batches	ms/batch	23.87	loss	0.00365
epoch	2	40/	0 batches	ms/batch	23.93	loss	0.00325
epoch	3	10/	0 batches	ms/batch	26.85	loss	0.00249
epoch	3	20/	0 batches	ms/batch	22.73	loss	0.00200
epoch	3	30/	0 batches	ms/batch	24.10	loss	0.00173
epoch	3	40/	0 batches	ms/batch	22.76	loss	0.00149
epoch	4	10/	0 batches	ms/batch	26.11	loss	0.00116
epoch	4	20/	0 batches	ms/batch	23.67	loss	0.00091
epoch	4	30/	0 batches	ms/batch	22.52	loss	0.00081
epoch	4	40/	0 batches	ms/batch	22.72	loss	0.00072
epoch	5	10/	0 batches	ms/batch	26.20	loss	0.00061
epoch	5	20/	0 batches	ms/batch	21.74	loss	0.00049
epoch	5	30/	0 batches	ms/batch	23.51	loss	0.00042
epoch	5	40/	0 batches	ms/batch	23.13	loss	0.00042
epoch	6	10/	0 batches	ms/batch	43.16	loss	0.00034
epoch	6	20/	0 batches	ms/batch	49.40	loss	0.00034
epoch	6	30/	0 batches	ms/batch	70.86	loss	0.00025
epoch	6	40/	0 batches	ms/batch	70.09	loss	0.00026
epoch	7	10/	0 batches	ms/batch	73.79	loss	0.00024
epoch	7	20/	0 batches	ms/batch	72.46	loss	0.00020
epoch	7	30/	0 batches	ms/batch	50.39	loss	0.00018
epoch	7	40/	0 batches	ms/batch	60.25	loss	0.00025
epoch	8	10/	0 batches	ms/batch	53.68	loss	0.00020
epoch	8	20/	0 batches	ms/batch	22.95	loss	0.00016
epoch	8	30/	0 batches	ms/batch	24.32	loss	0.00021
epoch	8	40/	0 batches	ms/batch	22.88	loss	0.00017
epoch	9	10/	0 batches	ms/batch	27.57	loss	0.00017
epoch	9	20/	0 batches	ms/batch	25.64	loss	0.00015
epoch	9	30/	0 batches	ms/batch	23.15	loss	0.00014
epoch	9	40/	0 batches	ms/batch	23.98	loss	0.00015
epoch	10	10/	0 batches	ms/batch	26.26	loss	0.00014
epoch	10	20/	0 batches	ms/batch	23.63	loss	0.00015
epoch	10	30/	0 batches	ms/batch	22.28	loss	0.00014
epoch	10	40/	0 batches	ms/batch	22.51	loss	0.00013
epoch	11	10/	0 batches	ms/batch	25.22	loss	0.00014
epoch	11	20/	0 batches	ms/batch	36.72	loss	0.00014
epoch	11	30/	0 batches	ms/batch	43.79	loss	0.00016
epoch	11	40/	0 batches	ms/batch	52.88	loss	0.00012
epoch	12	10/	0 batches	ms/batch	45.77	loss	0.00015
epoch	12	20/	0 batches	ms/batch	43.97	loss	0.00012
epoch	12	30/	0 batches	ms/batch	23.14	loss	0.00018
epoch	12	40/	0 batches	ms/batch	23.66	loss	0.00015
epoch	13	10/	0 batches	ms/batch	25.66	loss	0.00020
epoch	13	20/	0 batches	ms/batch	23.09	loss	0.00013
epoch	13	30/	0 batches	ms/batch	22.94	loss	0.00013
epoch	13	40/	0 batches	ms/batch	24.41	loss	0.00013
epoch	14	10/	0 batches	ms/batch	37.14	loss	0.00021
epoch	14	20/	0 batches	ms/batch	43.13	loss	0.00012
epoch	14	30/	0 batches	ms/batch	57.98	loss	0.00013
epoch	14	40/	0 batches	ms/batch	106.06	loss	0.00012
epoch	15	10/	0 batches	ms/batch	56.58	loss	0.00016
epoch	15	20/	0 batches	ms/batch	43.45	loss	0.00011
epoch	15	30/	0 batches	ms/batch	39.23	loss	0.00014
epoch	15	40/	0 batches	ms/batch	37.20	loss	0.00012
epoch	16	10/	0 batches	ms/batch	27.39	loss	0.00015
epoch	16	20/	0 batches	ms/batch	25.82	loss	0.00012
epoch	16	30/	0 batches	ms/batch	23.11	loss	0.00013
epoch	16	40/	0 batches	ms/batch	24.31	loss	0.00014
epoch	17	10/	0 batches	ms/batch	26.26	loss	0.00016
epoch	17	20/	0 batches	ms/batch	24.80	loss	0.00011
epoch	17	30/	0 batches	ms/batch	22.07	loss	0.00013
epoch	17	40/	0 batches	ms/batch	23.16	loss	0.00013
epoch	18	10/	0 batches	ms/batch	24.49	loss	0.00015
epoch	18	20/	0 batches	ms/batch	24.41	loss	0.00013

```
| epoch 18 | 30/ 0 batches | ms/batch 22.82 | loss 0.00018
| epoch 18 | 40/ 0 batches | ms/batch 21.40 | loss 0.00012
| epoch 19 | 10/ 0 batches | ms/batch 27.31 | loss 0.00015
| epoch 19 | 20/ 0 batches | ms/batch 22.08 | loss 0.00014
| epoch 19 | 30/ 0 batches | ms/batch 23.24 | loss 0.00018
| epoch 19 | 40/ 0 batches | ms/batch 23.24 | loss 0.00013
| epoch 20 | 10/ 0 batches | ms/batch 25.41 | loss 0.00018
| epoch 20 | 20/ 0 batches | ms/batch 22.63 | loss 0.00012
| epoch 20 | 30/ 0 batches | ms/batch 22.74 | loss 0.00015
| epoch 20 | 40/ 0 batches | ms/batch 23.64 | loss 0.00013
```

In [13]:

```
drugGeneCompressed=pickle.load(open(data_path+'drugGeneCompressed.p', 'rb'))
drugGeneCompressed=torch.FloatTensor(drugGeneCompressed)
```

In []:

```
#cell Line's gene expression data
cellGeneDataset=CellGeneDataset(cell_features)
cellGeneDataset_loader = DataLoader(cellGeneDataset, batch_size=64, shuffle=True)
#Learn
cellGeneCompressor=geneCompressing(cellGeneDataset_loader,num_gene_compressed_cell, noise_weight=0.01, log_interval=1 )
#Save
cellGeneCompressor.eval()
cellGeneCompressed=np.array([cellGeneCompressor.cpu().encoder(torch.FloatTensor(cellGeneDataset[d])).data.numpy() for d in range(len(cellGeneDataset))])
torch.save(cellGeneCompressor.state_dict(), data_path:'cellGeneCompressor.p')
pickle.dump(cellGeneCompressed, open(data_path+'cellGeneCompressed.p', 'wb'))
```

```
| epoch 1 | 1/ 0 batches | ms/batch 414.69 | loss 2.74390
| epoch 2 | 1/ 0 batches | ms/batch 422.43 | loss 3.53132
| epoch 3 | 1/ 0 batches | ms/batch 416.57 | loss 2.99976
| epoch 4 | 1/ 0 batches | ms/batch 406.06 | loss 2.83008
| epoch 5 | 1/ 0 batches | ms/batch 201.79 | loss 2.40781
| epoch 6 | 1/ 0 batches | ms/batch 213.83 | loss 2.27313
| epoch 7 | 1/ 0 batches | ms/batch 237.67 | loss 2.26563
| epoch 8 | 1/ 0 batches | ms/batch 219.95 | loss 2.07515
| epoch 9 | 1/ 0 batches | ms/batch 216.21 | loss 2.00170
| epoch 10 | 1/ 0 batches | ms/batch 219.87 | loss 1.95035
| epoch 11 | 1/ 0 batches | ms/batch 218.58 | loss 1.74026
| epoch 12 | 1/ 0 batches | ms/batch 218.08 | loss 1.77221
| epoch 13 | 1/ 0 batches | ms/batch 208.76 | loss 1.78408
| epoch 14 | 1/ 0 batches | ms/batch 208.43 | loss 1.71462
| epoch 15 | 1/ 0 batches | ms/batch 203.65 | loss 1.66737
| epoch 16 | 1/ 0 batches | ms/batch 204.53 | loss 1.66733
| epoch 17 | 1/ 0 batches | ms/batch 244.21 | loss 1.67430
| epoch 18 | 1/ 0 batches | ms/batch 436.34 | loss 1.59274
| epoch 19 | 1/ 0 batches | ms/batch 260.93 | loss 1.54862
| epoch 20 | 1/ 0 batches | ms/batch 457.43 | loss 1.65043
```

In [14]:

```
cellGeneCompressed=pickle.load(open(data_path+'cellGeneCompressed.p', 'rb'))
cellGeneCompressed=torch.FloatTensor(cellGeneCompressed)
```

Synergy prediction

Here is where the train test splits are created for the model training and evaluation.

Train/test split in cross or external validation

In [15]:

```
def get_cell_of_interest(tissues):
    tissues_of_interests = [codes['tissue'].item2idx[minor_tissue] for minor_tissue in tissues]
    cell_of_interest = cell_features.index[cell_features['tissue_id'].isin(tissues_of_interests)].tolist()
    return cell_of_interest
```

Train/test for general model

In [16]:

```
minor_tissues=['bone', 'prostate' ]
cell_of_interest = get_cell_of_interest(minor_tissues)
df_tissue_of_interest = df.loc[df['cell_line_name'].isin(cell_of_interest),:]
df_all = df.drop(df_tissue_of_interest.index)
#specific database
#df_all = df_all[df_all['study_id'].isin([1,3])]
df_all = df_all.dropna()
#cross validation
df_train, df_test = train_test_split(df_all, test_size=0.2) #cross validation
#external validation
#df_all = df_all.sample(frac=1).reset_index(drop=True)
```

```
#af_train=df_all.loc[df_all['study_id']==3] : 'ALMANAC'
#df_test=df_all.loc[df_all['study_id']==1] 1: 'ONEIL'
```

Train/test for bone

```
In [17]: _df_bone = df.loc[df['cell_line_name'].isin(get_cell_of_interest(['bone'])),:]
# Cross validation
_df_train_bone, _df_test_bone = train_test_split(_df_bone, test_size=0.2, random_state=1)
# External validation
_df_train_bone=_df_bone.loc[_df_bone['study_id'].isin([7,8,9])]
_df_test_bone=_df_bone.loc[_df_bone['study_id'].isin([7,8,9])]
```

Train/test for prostate

```
In [18]: _df_prostate= df.loc[df['cell_line_name'].isin(get_cell_of_interest(['prostate'])),:]
#Cross validation
_df_train_prostate, _df_test_prostate = train_test_split(_df_prostate, test_size=0.2, random_state=1)
# External validation
_df_train_prostate=_df_prostate.loc[_df_prostate['study_id'].isin([1,3])]
_df_test_prostate=_df_prostate.loc[_df_prostate['study_id'].isin([1,3])]
```

Dataset and Dataloader

```
In [19]: class DrugCombDataset(Dataset):
    def __init__(self, df, drug_features, cell_features):
        self.df = df
        self.drug_features = drug_features
        self.cell_features = cell_features

    def __len__(self):
        return len(self.df)

    def __getitem__(self, idx):
        d1 = self.df.iloc[idx, 0]
        d2 = self.df.iloc[idx, 1]
        cell = self.df.iloc[idx,2]
        ri_d1 = 1.0 if float(self.df.iloc[idx,3]) >ri_threshold else 0
        ri_d2 = 1.0 if float(self.df.iloc[idx,4]) >ri_threshold else 0
        syn = 1.0 if self.df.iloc[idx, 5] >syn_threshold else 0

        #external features
        d1_fp = np.array(self.drug_features.loc[d1, 'fps'])
        d1_sm = self.drug_features.loc[d1, 'smiles']
        d1_sm = np.pad(d1_sm, pad_width=0, max_drug_sm_len-len(d1_sm)), mode='constant', constant_values=0)
        d1_gn=drugGeneCompressed[d1]

        d2_fp = np.array(self.drug_features.loc[d2, 'fps'])
        d2_sm = self.drug_features.loc[d2, 'smiles']
        d2_sm = np.pad(d2_sm, pad_width=0, max_drug_sm_len-len(d2_sm)), mode='constant', constant_values=0)
        d2_gn=drugGeneCompressed[d2]

        c_ts = self.cell_features.loc[cell, 'tissue_id']
        c_ds = self.cell_features.loc[cell, 'disease_id']

        c_gn= cellGeneCompressed[cell]

        sample = {
            'd1': d1,
            'd1_fp': d1_fp,
            'd1_sm': d1_sm,
            'd1_gn': d1_gn,

            'd2': d2,
            'd2_fp': d2_fp,
            'd2_sm': d2_sm,
            'd2_gn': d2_gn,

            'cell': cell,
            'c_ts': c_ts,
            'c_ds': c_ds, #missing -1
            'c_gn': c_gn,

            'ri_d1': ri_d1,
            'ri_d2': ri_d2,
```

```

        'syn': syn
    }

    return sample

```

Load dataset

General model

```
In [20]: train = DrugCombDataset(df_train, drug_features, cell_features)
train_loader = DataLoader(train, batch_size=bsz, shuffle=True)
test = DrugCombDataset(df_test, drug_features, cell_features)
test_loader = DataLoader(test, batch_size=bsz, shuffle=True)
```

Bone

```
In [21]: _train_bone = DrugCombDataset(_df_train_bone, drug_features, cell_features)
_train_loader_bone = DataLoader(_train_bone, batch_size=bsz, shuffle=True)
_test_bone = DrugCombDataset(_df_test_bone, drug_features, cell_features)
_test_loader_bone = DataLoader(_test_bone, batch_size=bsz, shuffle=False)
```

Prostate

```
In [22]: _train_prostate = DrugCombDataset(_df_train_prostate, drug_features, cell_features)
_train_loader_prostate = DataLoader(_train_prostate, batch_size=bsz, shuffle=True)
_test_prostate = DrugCombDataset(_df_test_prostate, drug_features, cell_features)
_test_loader_prostate = DataLoader(_test_prostate, batch_size=bsz, shuffle=False)
```

Prediction model

Our main training model will be called Comb(). For Comb() to work we need to include 2 other encoder models. These are DrugEncoder() and CellEncoder().

DrugEncoder() loads from the the drug combined dataloader. The comments in the model go over how the model goes through the different parts of the input data in order to prep for the forward pass.

CellEncoder() is much the same but uses the cell lines combined dataloader.

Comb() is the main prediction model. It is made from the drug encoder and the cell encoder models. The forward pass encodes the initial drug, then the starting drug.

The synergy is calculated by concatenation the two drug encoders and also the cell line encoder and passing them through a fully connected layer. The sensitivities for each drug encoding is also calculated by passing them through a fully connected layer after concatenation.

Original Paper Repo: <https://github.com/yejinjkim/synergy-transfer/tree/master>

```
In [23]: class DrugEncoder(nn.Module):
    def __init__(self,
                 num_drugs=num_drugs,
                 num_ID_emb=0,
                 num_drug_fp=num_drug_fp,
                 max_drug_sm_len=max_drug_sm_len,
                 num_gene = num_gene_compressed_drug,
                 num_comp_char=len(codes['mole'].idx2item),
                 fp_embed_sz = 32,
                 gene_embed_sz = int(num_gene_compressed_drug/2),
                 out_size=64,
                 dropout=0.3):
        super(DrugEncoder, self).__init__()

        self.dropout= dropout
        #DRUG
        #drug ID
        #self.embed_id = nn.Embedding(num_drugs, num_ID_emb)

        #compound ID
        self.embed_comp = nn.Embedding(num_comp_char, num_comp_char, padding_idx=0)#padding's idx=0
        #encoding compound
        self.encoderlayer = nn.TransformerEncoderLayer(d_model=num_comp_char, nhead=4)
```

```

        self.encoder = nn.TransformerEncoder(self.encoderlayer, num_layers=1)

    #fingerprint
    self.dense_fp = nn.Linear(num_drug_fp,fp_embed_sz)
    #gene
    self.dense_gene = nn.Linear(num_gene,gene_embed_sz)

    #depthwise for compound encoding
    self.conv = nn.Conv2d(1, 1, (1, num_comp_char), groups=1)

    #combined
    combined_sz = num_ID_emb+fp_embed_sz+max_drug_sm_len+gene_embed_sz
    self.FC2 = FC2(combined_sz, out_size, dropout)

    def forward(self, d_list):
        """
        id: bsz*1
        fp: bsz*num_drug_fp
        sm: bsz*max_drug_sm_len
        """
        id, fp, sm, gn = d_list

        sm = self.embed_comp(sm) #bsz*max_drug_sm_len*num_comp_char(embedding size)
        sm = self.encoder(sm)
        sm = self.conv(sm.unsqueeze(1)).squeeze()

        fp = F.relu(self.dense_fp(fp))
        gn = F.relu(self.dense_gene(gn))

        #combine
        x = torch.cat((fp, sm, gn),1) # bsz*[num_emb_id+num_drug_fp+max+drug_sm]
        x = self.FC2(x)

        return x

```

```

In [24]: class CellEncoder(nn.Module):
    def __init__(self,
                 num_cells=num_celllines,
                 num_tissue=0,
                 num_disease=num_disease,
                 num_ID_emb=0,
                 gene_embed_sz=int(num_gene_compressed_cell/2),
                 num_gene=num_gene_compressed_cell,
                 out_size=64,
                 dropout=0.3):
        super(CellEncoder, self).__init__()

        self.dropout= dropout
        #cell ID
        #self.embed_id = nn.Embedding(num_cells, num_ID_emb)
        #cell tissue
        #self.embed_ts = nn.Embedding(num_tissue, num_tissue)
        #cell disease
        self.embed_ds = nn.Embedding(num_disease, num_disease, padding_idx=3)
        #gene
        self.dense_gene = nn.Linear(num_gene,gene_embed_sz)

        #combined
        combined_sz = num_ID_emb+num_tissue+num_disease+gene_embed_sz
        self.FC2 = FC2(combined_sz, out_size, dropout)

    def forward(self, c_list):
        """
        id: bsz*1
        fp: bsz*num_drug_fp
        sm: bsz*max_drug_sm_len
        """
        id, ts, ds, gn = c_list
        ds = F.relu(self.embed_ds(ds)) #bsz*num_diseases

        gn = F.relu(self.dense_gene(gn)) #bsz*gene_embed_sz

        #combine
        x = torch.cat((ds, gn),1) # bsz*combined_sz
        x = self.FC2(x)

        return x

```

```
In [25]: class Comb(nn.Module):
    def __init__(self, num_celllines,
                 num_drugs=num_drugs,
                 num_drug_fp=num_drug_fp,
                 max_drug_sm_len=max_drug_sm_len,
                 num_comp_char=len(codes['mole'].idx2item),
                 num_ID_emb=0,
                 out_size=64,
                 dropout=0.3):

        super(Comb, self).__init__()

        self.dropout=dropout
        #drug
        self.drugEncoder = DrugEncoder()
        #cell
        self.cellEncoder = CellEncoder()
        #fc
        self.fc_syn = FC2(out_size*3, 1, dropout)
        self.fc_ri = FC2(out_size*2, 1, dropout)

    def forward(self, d1_list, d2_list, c_list):
        d1 = self.drugEncoder(d1_list)
        d2 = self.drugEncoder(d2_list)
        c = self.cellEncoder(c_list)

        syn = self.fc_syn(torch.cat((d1, d2, c),1))
        ri1 = self.fc_ri(torch.cat((d1,c),1))
        ri2 = self.fc_ri(torch.cat((d2,c),1))

        return syn, ri1, ri2
```

Training

The model training is as follows. We take our Comb() model and begin training with a Binary Cross Entropy Loss. We use the Adagrad optimizer.

For training we get all the training sample values from the training data loader and then we pass them into the model. The drug synergy is outputted by the model and we compute loss, compute the gradient, and iterate the model.

Hyperparams

We have a few hyperparameters that we've taken from the paper.

- Our learning rate is the default torch learning rate which is 0.001
- Our batch size is 100
- Our dropout is 0.1

Computational Requirements

To run this script there are computational requirements.

- We had CUDA set up using 3 devices on Google Colab Pro.
- We trained for 10 epochs
- The average runtime for each epoch was 66 ms per batch and we had 17 batches

```
In [58]: model = Comb()
if cuda:
    model = model.cuda()
#Regression
#criterion_mse = nn.MSELoss()
#Classification
criterion_bce = nn.BCEWithLogitsLoss()
optimizer = optim.Adagrad(model.parameters())

/usr/local/lib/python3.10/dist-packages/torch/nn/modules/transformer.py:286: UserWarning: enable_nested_tensor is True, but self.use_nested_tensor is False because encoder_layer.self_attn.batch_first was not True(use batch_first for better inference performance)
    warnings.warn(f"enable_nested_tensor is True, but self.use_nested_tensor is False because {why_not_sparsity_fast_path}")
```

In [27]:

```
#Training
def training(isAux, data_loader):
    model.train()
    total_loss = 0
    start_time = time.time()

    for iteration, sample in enumerate(data_loader):
        try:
            d1=Variable(sample['d1'])
            d1_fp = Variable(sample['d1_fp'].float())
            d1_sm = Variable(sample['d1_sm'])
            d1_gn = Variable(sample['d1_gn'].float())

            d2=Variable(sample['d2'])
            d2_fp = Variable(sample['d2_fp'].float())
            d2_sm = Variable(sample['d2_sm'])
            d2_gn = Variable(sample['d2_gn'].float())

            cell = Variable(sample['cell'])
            c_ts = Variable(sample['c_ts'])
            c_ds = Variable(sample['c_ds'])
            c_gn = Variable(sample['c_gn'].float())

            syn_true = Variable(sample['syn'].float())
            ri_d1=Variable(sample['ri_d1'].float())
            ri_d2=Variable(sample['ri_d2'].float())

            if cuda:
                d1=d1.cuda()
                d1_fp=d1_fp.cuda()
                d1_sm=d1_sm.cuda()
                d1_gn=d1_gn.cuda()

                d2=d2.cuda()
                d2_fp=d2_fp.cuda()
                d2_sm=d2_sm.cuda()
                d2_gn=d2_gn.cuda()

                cell=cell.cuda()
                c_ts=c_ts.cuda()
                c_ds=c_ds.cuda()
                c_gn=c_gn.cuda()

                syn_true=syn_true.cuda()
                ri_d1=ri_d1.cuda()
                ri_d2=ri_d2.cuda()

            optimizer.zero_grad()

            syn, ri1, ri2 = model((d1, d1_fp, d1_sm, d1_gn), (d2, d2_fp, d2_sm, d2_gn), (cell, c_ts, c_ds, c_gn) )

            if not isAux:
                loss = criterion_bce(syn, syn_true.view(-1,1))
            else:
                loss = criterion_bce(ri1, ri_d1.view(-1,1))+criterion_bce(ri2, ri_d2.view(-1,1))

            loss.backward()
            optimizer.step()
            total_loss += loss.data

            if iteration % log_interval == 0 and iteration > 0:
                cur_loss = total_loss.item() / log_interval
                elapsed = time.time() - start_time
                print('| epoch {:3d} | {:5d}/{:5d} batches | ms/batch {:.2f} | loss {:.5f}'.format(epoch, iteration, int(len(tr

                total_loss = 0
                start_time = time.time()
            except:
                continue

```

In [28]:

```
def evaluate(data_loader):
    model.eval()
    total_loss = 0
    total_loss_sen = 0
```

```

#Loss
with torch.no_grad():
    for iteration, sample in enumerate(data_loader):
        d1=Variable(sample['d1'])
        d1_fp = Variable(sample['d1_fp'].float())
        d1_sm = Variable(sample['d1_sm'])
        d1_gn = Variable(sample['d1_gn'].float())

        d2=Variable(sample['d2'])
        d2_fp = Variable(sample['d2_fp'].float())
        d2_sm = Variable(sample['d2_sm'])
        d2_gn = Variable(sample['d2_gn'].float())

        cell = Variable(sample['cell'])
        c_ts = Variable(sample['c_ts'])
        c_ds = Variable(sample['c_ds'])
        c_gn = Variable(sample['c_gn'].float())

        syn_true = Variable(sample['syn'].float())
        ri_d1=Variable(sample['ri_d1'].float())
        ri_d2=Variable(sample['ri_d2'].float())

        if cuda:
            d1=d1.cuda()
            d1_fp=d1_fp.cuda()
            d1_sm=d1_sm.cuda()
            d1_gn=d1_gn.cuda()

            d2=d2.cuda()
            d2_fp=d2_fp.cuda()
            d2_sm=d2_sm.cuda()
            d2_gn=d2_gn.cuda()

            cell=cell.cuda()
            c_ts=c_ts.cuda()
            c_ds=c_ds.cuda()
            c_gn=c_gn.cuda()

            syn_true=syn_true.cuda()
            ri_d1=ri_d1.cuda()
            ri_d2=ri_d2.cuda()

        syn,ri1,ri2 = model((d1, d1_fp, d1_sm, d1_gn), (d2, d2_fp, d2_sm, d2_gn), (cell, c_ts,c_ds,c_gn) )
        loss = criterion_bce(syn, syn_true.view(-1,1))
        total_loss +=loss.data
        loss_sen = (criterion_bce(ri1, ri_d1.view(-1,1))+criterion_bce(ri2, ri_d2.view(-1,1)))/2
        total_loss_sen += loss_sen.data

        print('syn mse', total_loss.item()/(iteration+1))
        print('sen_mse', total_loss_sen.item()/(iteration+1))

```

In []:

```

best_val_loss = None
try:
    for epoch in range(1, 11):
        epoch_start_time = time.time()
        training(False, train_loader)
        training(True, train_loader)
        evaluate(test_loader)
        print('-'*89)
except KeyboardInterrupt:
    print('-'*89)
    print('Existing from training early')

epoch 1 | 100/ 17 batches | ms/batch 90.82 | loss  0.37345
epoch 1 | 200/ 17 batches | ms/batch 90.16 | loss  0.32039
epoch 1 | 300/ 17 batches | ms/batch 84.83 | loss  0.30158
epoch 1 | 400/ 17 batches | ms/batch 89.44 | loss  0.29340
epoch 1 | 500/ 17 batches | ms/batch 89.39 | loss  0.28483
epoch 1 | 600/ 17 batches | ms/batch 86.29 | loss  0.29483
epoch 1 | 700/ 17 batches | ms/batch 89.03 | loss  0.27432
epoch 1 | 800/ 17 batches | ms/batch 88.85 | loss  0.27499
epoch 1 | 900/ 17 batches | ms/batch 84.80 | loss  0.27606
epoch 1 | 1000/ 17 batches | ms/batch 90.83 | loss  0.25913
epoch 1 | 1100/ 17 batches | ms/batch 106.84 | loss  0.27614
epoch 1 | 1200/ 17 batches | ms/batch 89.53 | loss  0.26944
epoch 1 | 1300/ 17 batches | ms/batch 88.52 | loss  0.26507
epoch 1 | 1400/ 17 batches | ms/batch 87.24 | loss  0.25955
epoch 1 | 1500/ 17 batches | ms/batch 95.21 | loss  0.26671

```

epoch	1	100/	17 batches	ms/batch	85.51	loss	0.240074
epoch	1	1600/	17 batches	ms/batch	86.15	loss	0.25302
epoch	1	1700/	17 batches	ms/batch	89.13	loss	0.25105
epoch	1	1800/	17 batches	ms/batch	85.88	loss	0.25979
epoch	1	1900/	17 batches	ms/batch	88.69	loss	0.26158
epoch	1	2000/	17 batches	ms/batch	88.73	loss	0.25020
epoch	1	2100/	17 batches	ms/batch	85.03	loss	0.25215
epoch	1	2200/	17 batches	ms/batch	88.87	loss	0.25493
epoch	1	100/	17 batches	ms/batch	89.23	loss	0.25521
epoch	1	200/	17 batches	ms/batch	88.87	loss	0.19739
epoch	1	300/	17 batches	ms/batch	87.29	loss	0.19561
epoch	1	400/	17 batches	ms/batch	89.25	loss	0.19333
epoch	1	500/	17 batches	ms/batch	88.99	loss	0.19156
epoch	1	600/	17 batches	ms/batch	85.56	loss	0.19176
epoch	1	700/	17 batches	ms/batch	90.17	loss	0.19115
epoch	1	800/	17 batches	ms/batch	92.52	loss	0.19478
epoch	1	900/	17 batches	ms/batch	85.34	loss	0.19162
epoch	1	1000/	17 batches	ms/batch	111.83	loss	0.18614
epoch	1	1100/	17 batches	ms/batch	91.03	loss	0.17870
epoch	1	1200/	17 batches	ms/batch	86.85	loss	0.19047
epoch	1	1300/	17 batches	ms/batch	88.06	loss	0.18680
epoch	1	1400/	17 batches	ms/batch	89.63	loss	0.18133
epoch	1	1500/	17 batches	ms/batch	87.01	loss	0.18589
epoch	1	1600/	17 batches	ms/batch	88.20	loss	0.17816
epoch	1	1700/	17 batches	ms/batch	89.07	loss	0.18036
epoch	1	1800/	17 batches	ms/batch	86.36	loss	0.18896
epoch	1	1900/	17 batches	ms/batch	91.01	loss	0.18275
epoch	1	2000/	17 batches	ms/batch	88.88	loss	0.18672
epoch	1	2100/	17 batches	ms/batch	85.36	loss	0.18494
epoch	1	2200/	17 batches	ms/batch	88.74	loss	0.19452

syn_mse 0.24762109866701784
sen_mse 0.0860751898610237

epoch	2	100/	17 batches	ms/batch	90.47	loss	0.27419
epoch	2	200/	17 batches	ms/batch	85.86	loss	0.25088
epoch	2	300/	17 batches	ms/batch	89.00	loss	0.24562
epoch	2	400/	17 batches	ms/batch	89.15	loss	0.24429
epoch	2	500/	17 batches	ms/batch	84.98	loss	0.25212
epoch	2	600/	17 batches	ms/batch	89.03	loss	0.24192
epoch	2	700/	17 batches	ms/batch	87.52	loss	0.24648
epoch	2	800/	17 batches	ms/batch	87.26	loss	0.23390
epoch	2	900/	17 batches	ms/batch	114.80	loss	0.24074
epoch	2	1000/	17 batches	ms/batch	88.69	loss	0.23923
epoch	2	1100/	17 batches	ms/batch	85.55	loss	0.23832
epoch	2	1200/	17 batches	ms/batch	89.15	loss	0.24623
epoch	2	1300/	17 batches	ms/batch	88.62	loss	0.24185
epoch	2	1400/	17 batches	ms/batch	84.98	loss	0.24383
epoch	2	1500/	17 batches	ms/batch	88.76	loss	0.24177
epoch	2	1600/	17 batches	ms/batch	87.74	loss	0.23688
epoch	2	1700/	17 batches	ms/batch	84.92	loss	0.23464
epoch	2	1800/	17 batches	ms/batch	102.03	loss	0.23436
epoch	2	1900/	17 batches	ms/batch	86.39	loss	0.23983
epoch	2	2000/	17 batches	ms/batch	87.29	loss	0.23810
epoch	2	2100/	17 batches	ms/batch	88.81	loss	0.23593
epoch	2	2200/	17 batches	ms/batch	86.11	loss	0.23512
epoch	2	100/	17 batches	ms/batch	91.43	loss	0.18588
epoch	2	200/	17 batches	ms/batch	85.81	loss	0.17566
epoch	2	300/	17 batches	ms/batch	89.56	loss	0.19378
epoch	2	400/	17 batches	ms/batch	89.42	loss	0.19098
epoch	2	500/	17 batches	ms/batch	84.84	loss	0.18725
epoch	2	600/	17 batches	ms/batch	88.98	loss	0.19255
epoch	2	700/	17 batches	ms/batch	87.96	loss	0.17754
epoch	2	800/	17 batches	ms/batch	86.83	loss	0.18681
epoch	2	900/	17 batches	ms/batch	89.19	loss	0.17715
epoch	2	1000/	17 batches	ms/batch	85.19	loss	0.18793
epoch	2	1100/	17 batches	ms/batch	85.98	loss	0.19478
epoch	2	1200/	17 batches	ms/batch	86.15	loss	0.18620
epoch	2	1300/	17 batches	ms/batch	93.71	loss	0.16920
epoch	2	1400/	17 batches	ms/batch	87.61	loss	0.17446
epoch	2	1500/	17 batches	ms/batch	88.82	loss	0.17335
epoch	2	1600/	17 batches	ms/batch	87.31	loss	0.18256
epoch	2	1700/	17 batches	ms/batch	88.55	loss	0.17498
epoch	2	1800/	17 batches	ms/batch	88.84	loss	0.18416
epoch	2	1900/	17 batches	ms/batch	120.57	loss	0.17417
epoch	2	2000/	17 batches	ms/batch	84.76	loss	0.18643
epoch	2	2100/	17 batches	ms/batch	88.77	loss	0.16985
epoch	2	2200/	17 batches	ms/batch	89.29	loss	0.17151

syn_mse 0.22192410871569204
sen_mse 0.08636473941301924

epoch	3	100/	17 batches	ms/batch	86.26	loss	0.24758
epoch	3	200/	17 batches	ms/batch	89.14	loss	0.23346
epoch	3	300/	17 batches	ms/batch	89.79	loss	0.23270
epoch	3	400/	17 batches	ms/batch	91.91	loss	0.22875
epoch	3	500/	17 batches	ms/batch	88.77	loss	0.23404

epoch	step	batch	ms/batch	loss
epoch 3	600/	17 batches	ms/batch 88.68	loss 0.22904
epoch 3	700/	17 batches	ms/batch 84.74	loss 0.24010
epoch 3	800/	17 batches	ms/batch 88.33	loss 0.23695
epoch 3	900/	17 batches	ms/batch 87.95	loss 0.22994
epoch 3	1000/	17 batches	ms/batch 85.40	loss 0.22902
epoch 3	1100/	17 batches	ms/batch 88.57	loss 0.23278
epoch 3	1200/	17 batches	ms/batch 88.44	loss 0.22900
epoch 3	1300/	17 batches	ms/batch 86.90	loss 0.22949
epoch 3	1400/	17 batches	ms/batch 89.04	loss 0.22361
epoch 3	1500/	17 batches	ms/batch 86.31	loss 0.23583
epoch 3	1600/	17 batches	ms/batch 87.67	loss 0.23086
epoch 3	1700/	17 batches	ms/batch 88.90	loss 0.22607
epoch 3	1800/	17 batches	ms/batch 84.93	loss 0.22763
epoch 3	1900/	17 batches	ms/batch 85.93	loss 0.22923
epoch 3	2000/	17 batches	ms/batch 91.91	loss 0.22784
epoch 3	2100/	17 batches	ms/batch 84.82	loss 0.22724
epoch 3	2200/	17 batches	ms/batch 88.16	loss 0.22513
epoch 3	100/	17 batches	ms/batch 92.22	loss 0.17491
epoch 3	200/	17 batches	ms/batch 88.66	loss 0.18266
epoch 3	300/	17 batches	ms/batch 86.88	loss 0.18339
epoch 3	400/	17 batches	ms/batch 87.51	loss 0.17298
epoch 3	500/	17 batches	ms/batch 91.41	loss 0.18649
epoch 3	600/	17 batches	ms/batch 85.57	loss 0.16899
epoch 3	700/	17 batches	ms/batch 88.72	loss 0.18898
epoch 3	800/	17 batches	ms/batch 89.10	loss 0.18108
epoch 3	900/	17 batches	ms/batch 124.27	loss 0.18111
epoch 3	1000/	17 batches	ms/batch 84.99	loss 0.18271
epoch 3	1100/	17 batches	ms/batch 89.13	loss 0.18572
epoch 3	1200/	17 batches	ms/batch 88.50	loss 0.17118
epoch 3	1300/	17 batches	ms/batch 86.86	loss 0.17241
epoch 3	1400/	17 batches	ms/batch 88.72	loss 0.16890
epoch 3	1500/	17 batches	ms/batch 86.36	loss 0.17620
epoch 3	1600/	17 batches	ms/batch 87.30	loss 0.17539
epoch 3	1700/	17 batches	ms/batch 89.43	loss 0.17477
epoch 3	1800/	17 batches	ms/batch 86.86	loss 0.17868
epoch 3	1900/	17 batches	ms/batch 96.74	loss 0.17409
epoch 3	2000/	17 batches	ms/batch 91.57	loss 0.17764
epoch 3	2100/	17 batches	ms/batch 84.97	loss 0.16828
epoch 3	2200/	17 batches	ms/batch 88.47	loss 0.16820

syn_mse 0.21511625700781933

sen_mse 0.08463449695273163

epoch	step	batch	ms/batch	loss
epoch 4	100/	17 batches	ms/batch 89.80	loss 0.22893
epoch 4	200/	17 batches	ms/batch 87.78	loss 0.22822
epoch 4	300/	17 batches	ms/batch 86.39	loss 0.22380
epoch 4	400/	17 batches	ms/batch 86.52	loss 0.22542
epoch 4	500/	17 batches	ms/batch 87.77	loss 0.22701
epoch 4	600/	17 batches	ms/batch 88.47	loss 0.23173
epoch 4	700/	17 batches	ms/batch 85.28	loss 0.22101
epoch 4	800/	17 batches	ms/batch 88.34	loss 0.21876
epoch 4	900/	17 batches	ms/batch 90.36	loss 0.23108
epoch 4	1000/	17 batches	ms/batch 89.79	loss 0.22141
epoch 4	1100/	17 batches	ms/batch 89.49	loss 0.21794
epoch 4	1200/	17 batches	ms/batch 88.60	loss 0.22576
epoch 4	1300/	17 batches	ms/batch 85.10	loss 0.22542
epoch 4	1400/	17 batches	ms/batch 88.45	loss 0.22843
epoch 4	1500/	17 batches	ms/batch 89.01	loss 0.21247
epoch 4	1600/	17 batches	ms/batch 85.10	loss 0.22265
epoch 4	1700/	17 batches	ms/batch 90.57	loss 0.22144
epoch 4	1800/	17 batches	ms/batch 87.90	loss 0.21101
epoch 4	1900/	17 batches	ms/batch 85.77	loss 0.21900
epoch 4	2000/	17 batches	ms/batch 86.63	loss 0.22178
epoch 4	2100/	17 batches	ms/batch 86.69	loss 0.22514
epoch 4	2200/	17 batches	ms/batch 87.34	loss 0.22214
epoch 4	100/	17 batches	ms/batch 86.12	loss 0.17531
epoch 4	200/	17 batches	ms/batch 92.29	loss 0.17434
epoch 4	300/	17 batches	ms/batch 88.94	loss 0.17664
epoch 4	400/	17 batches	ms/batch 85.02	loss 0.17885
epoch 4	500/	17 batches	ms/batch 95.64	loss 0.18009
epoch 4	600/	17 batches	ms/batch 88.52	loss 0.17440
epoch 4	700/	17 batches	ms/batch 158.34	loss 0.16528
epoch 4	800/	17 batches	ms/batch 87.43	loss 0.17713
epoch 4	900/	17 batches	ms/batch 87.74	loss 0.17340
epoch 4	1000/	17 batches	ms/batch 89.21	loss 0.16996
epoch 4	1100/	17 batches	ms/batch 87.44	loss 0.16924
epoch 4	1200/	17 batches	ms/batch 88.06	loss 0.17373
epoch 4	1300/	17 batches	ms/batch 86.50	loss 0.17844
epoch 4	1400/	17 batches	ms/batch 84.90	loss 0.16419
epoch 4	1500/	17 batches	ms/batch 87.50	loss 0.18112
epoch 4	1600/	17 batches	ms/batch 90.03	loss 0.17769
epoch 4	1700/	17 batches	ms/batch 85.34	loss 0.16519
epoch 4	1800/	17 batches	ms/batch 88.96	loss 0.16982
epoch 4	1900/	17 batches	ms/batch 89.04	loss 0.16417
epoch 4	2000/	17 batches	ms/batch 85.20	loss 0.17039

epoch	4	2100/	17 batches	ms/batch	89.24	loss	0.15739
epoch	4	2200/	17 batches	ms/batch	99.26	loss	0.16705
syn_mse	0.2059281143749658						
sen_mse	0.08394957507345598						
<hr/>							
epoch	5	100/	17 batches	ms/batch	86.48	loss	0.23197
epoch	5	200/	17 batches	ms/batch	89.68	loss	0.22511
epoch	5	300/	17 batches	ms/batch	88.61	loss	0.22449
epoch	5	400/	17 batches	ms/batch	86.21	loss	0.22578
epoch	5	500/	17 batches	ms/batch	88.86	loss	0.21709
epoch	5	600/	17 batches	ms/batch	87.36	loss	0.21564
epoch	5	700/	17 batches	ms/batch	85.92	loss	0.21465
epoch	5	800/	17 batches	ms/batch	88.48	loss	0.21103
epoch	5	900/	17 batches	ms/batch	86.16	loss	0.22269
epoch	5	1000/	17 batches	ms/batch	87.29	loss	0.22461
epoch	5	1100/	17 batches	ms/batch	88.57	loss	0.21009
epoch	5	1200/	17 batches	ms/batch	88.13	loss	0.21783
epoch	5	1300/	17 batches	ms/batch	92.37	loss	0.21370
epoch	5	1400/	17 batches	ms/batch	89.09	loss	0.21299
epoch	5	1500/	17 batches	ms/batch	86.09	loss	0.20976
epoch	5	1600/	17 batches	ms/batch	87.61	loss	0.21450
epoch	5	1700/	17 batches	ms/batch	88.56	loss	0.20617
epoch	5	1800/	17 batches	ms/batch	85.43	loss	0.22199
epoch	5	1900/	17 batches	ms/batch	90.03	loss	0.21811
epoch	5	2000/	17 batches	ms/batch	86.25	loss	0.21045
epoch	5	2100/	17 batches	ms/batch	84.52	loss	0.20493
epoch	5	2200/	17 batches	ms/batch	86.43	loss	0.20477
epoch	5	100/	17 batches	ms/batch	87.74	loss	0.17295
epoch	5	200/	17 batches	ms/batch	88.62	loss	0.17773
epoch	5	300/	17 batches	ms/batch	87.02	loss	0.16915
epoch	5	400/	17 batches	ms/batch	88.46	loss	0.17261
epoch	5	500/	17 batches	ms/batch	88.48	loss	0.17353
epoch	5	600/	17 batches	ms/batch	86.40	loss	0.17052
epoch	5	700/	17 batches	ms/batch	88.01	loss	0.17196
epoch	5	800/	17 batches	ms/batch	100.41	loss	0.16622
epoch	5	900/	17 batches	ms/batch	85.19	loss	0.17239
epoch	5	1000/	17 batches	ms/batch	89.48	loss	0.17228
epoch	5	1100/	17 batches	ms/batch	91.35	loss	0.16575
epoch	5	1200/	17 batches	ms/batch	85.07	loss	0.16501
epoch	5	1300/	17 batches	ms/batch	89.62	loss	0.17302
epoch	5	1400/	17 batches	ms/batch	88.46	loss	0.16168
epoch	5	1500/	17 batches	ms/batch	85.47	loss	0.17039
epoch	5	1600/	17 batches	ms/batch	89.08	loss	0.16898
epoch	5	1700/	17 batches	ms/batch	174.20	loss	0.16649
epoch	5	1800/	17 batches	ms/batch	90.49	loss	0.16146
epoch	5	1900/	17 batches	ms/batch	87.61	loss	0.16127
epoch	5	2000/	17 batches	ms/batch	87.99	loss	0.16093
epoch	5	2100/	17 batches	ms/batch	88.96	loss	0.16239
epoch	5	2200/	17 batches	ms/batch	86.73	loss	0.16720
syn_mse	0.2079108350122455						
sen_mse	0.08417936548877723						
<hr/>							
epoch	6	100/	17 batches	ms/batch	86.07	loss	0.23589
epoch	6	200/	17 batches	ms/batch	88.72	loss	0.21773
epoch	6	300/	17 batches	ms/batch	89.78	loss	0.21690
epoch	6	400/	17 batches	ms/batch	85.77	loss	0.21791
epoch	6	500/	17 batches	ms/batch	86.77	loss	0.20779
epoch	6	600/	17 batches	ms/batch	88.25	loss	0.21935
epoch	6	700/	17 batches	ms/batch	85.77	loss	0.21337
epoch	6	800/	17 batches	ms/batch	88.23	loss	0.21035
epoch	6	900/	17 batches	ms/batch	85.99	loss	0.20702
epoch	6	1000/	17 batches	ms/batch	86.83	loss	0.21494
epoch	6	1100/	17 batches	ms/batch	87.90	loss	0.21630
epoch	6	1200/	17 batches	ms/batch	86.20	loss	0.20955
epoch	6	1300/	17 batches	ms/batch	89.40	loss	0.21396
epoch	6	1400/	17 batches	ms/batch	88.03	loss	0.20370
epoch	6	1500/	17 batches	ms/batch	90.44	loss	0.20911
epoch	6	1600/	17 batches	ms/batch	87.93	loss	0.20918
epoch	6	1700/	17 batches	ms/batch	88.97	loss	0.21037
epoch	6	1800/	17 batches	ms/batch	85.11	loss	0.20367
epoch	6	1900/	17 batches	ms/batch	88.32	loss	0.20524
epoch	6	2000/	17 batches	ms/batch	88.01	loss	0.20445
epoch	6	2100/	17 batches	ms/batch	86.46	loss	0.20789
epoch	6	2200/	17 batches	ms/batch	88.66	loss	0.20930
epoch	6	100/	17 batches	ms/batch	88.17	loss	0.17667
epoch	6	200/	17 batches	ms/batch	88.92	loss	0.16380
epoch	6	300/	17 batches	ms/batch	86.42	loss	0.18059
epoch	6	400/	17 batches	ms/batch	87.26	loss	0.16649
epoch	6	500/	17 batches	ms/batch	89.59	loss	0.16756
epoch	6	600/	17 batches	ms/batch	85.98	loss	0.16998
epoch	6	700/	17 batches	ms/batch	87.85	loss	0.16845
epoch	6	800/	17 batches	ms/batch	88.75	loss	0.17205
epoch	6	900/	17 batches	ms/batch	85.57	loss	0.16229
epoch	6	1000/	17 batches	ms/batch	100.31	loss	0.17482

epoch	6	1100/	17 batches	ms/batch	89.00	loss	0.16390
epoch	6	1200/	17 batches	ms/batch	84.84	loss	0.15823
epoch	6	1300/	17 batches	ms/batch	91.49	loss	0.16368
epoch	6	1400/	17 batches	ms/batch	88.91	loss	0.16049
epoch	6	1500/	17 batches	ms/batch	84.80	loss	0.15994
epoch	6	1600/	17 batches	ms/batch	85.96	loss	0.16280
epoch	6	1700/	17 batches	ms/batch	85.66	loss	0.16323
epoch	6	1800/	17 batches	ms/batch	85.79	loss	0.16545
epoch	6	1900/	17 batches	ms/batch	88.91	loss	0.15469
epoch	6	2000/	17 batches	ms/batch	88.67	loss	0.15906
epoch	6	2100/	17 batches	ms/batch	86.68	loss	0.16670
epoch	6	2200/	17 batches	ms/batch	90.10	loss	0.15713

syn_mse 0.2047107783382703

sen_mse 0.08493331267829535

epoch	7	100/	17 batches	ms/batch	90.99	loss	0.21915
epoch	7	200/	17 batches	ms/batch	86.00	loss	0.22338
epoch	7	300/	17 batches	ms/batch	89.40	loss	0.20905
epoch	7	400/	17 batches	ms/batch	89.49	loss	0.21005
epoch	7	500/	17 batches	ms/batch	85.02	loss	0.20920
epoch	7	600/	17 batches	ms/batch	88.35	loss	0.20930
epoch	7	700/	17 batches	ms/batch	86.66	loss	0.20370
epoch	7	800/	17 batches	ms/batch	87.09	loss	0.19233
epoch	7	900/	17 batches	ms/batch	88.89	loss	0.21415
epoch	7	1000/	17 batches	ms/batch	86.70	loss	0.20003
epoch	7	1100/	17 batches	ms/batch	87.15	loss	0.20618
epoch	7	1200/	17 batches	ms/batch	185.70	loss	0.20738
epoch	7	1300/	17 batches	ms/batch	86.97	loss	0.21122
epoch	7	1400/	17 batches	ms/batch	91.59	loss	0.20435
epoch	7	1500/	17 batches	ms/batch	87.07	loss	0.21272
epoch	7	1600/	17 batches	ms/batch	89.89	loss	0.21273
epoch	7	1700/	17 batches	ms/batch	92.06	loss	0.21232
epoch	7	1800/	17 batches	ms/batch	86.06	loss	0.20890
epoch	7	1900/	17 batches	ms/batch	87.49	loss	0.21375
epoch	7	2000/	17 batches	ms/batch	88.83	loss	0.20580
epoch	7	2100/	17 batches	ms/batch	86.31	loss	0.20328
epoch	7	2200/	17 batches	ms/batch	88.43	loss	0.20216
epoch	7	100/	17 batches	ms/batch	85.98	loss	0.16976
epoch	7	200/	17 batches	ms/batch	88.68	loss	0.16277
epoch	7	300/	17 batches	ms/batch	88.07	loss	0.17407
epoch	7	400/	17 batches	ms/batch	86.60	loss	0.16108
epoch	7	500/	17 batches	ms/batch	88.43	loss	0.15259
epoch	7	600/	17 batches	ms/batch	88.66	loss	0.16216
epoch	7	700/	17 batches	ms/batch	87.43	loss	0.17377
epoch	7	800/	17 batches	ms/batch	88.84	loss	0.16462
epoch	7	900/	17 batches	ms/batch	86.55	loss	0.16210
epoch	7	1000/	17 batches	ms/batch	87.29	loss	0.15881
epoch	7	1100/	17 batches	ms/batch	99.72	loss	0.16792
epoch	7	1200/	17 batches	ms/batch	85.59	loss	0.15631
epoch	7	1300/	17 batches	ms/batch	89.96	loss	0.16230
epoch	7	1400/	17 batches	ms/batch	88.34	loss	0.16158
epoch	7	1500/	17 batches	ms/batch	85.09	loss	0.15565
epoch	7	1600/	17 batches	ms/batch	89.06	loss	0.16822
epoch	7	1700/	17 batches	ms/batch	87.07	loss	0.15735
epoch	7	1800/	17 batches	ms/batch	85.19	loss	0.16174
epoch	7	1900/	17 batches	ms/batch	88.42	loss	0.15687
epoch	7	2000/	17 batches	ms/batch	92.29	loss	0.16774
epoch	7	2100/	17 batches	ms/batch	85.14	loss	0.15070
epoch	7	2200/	17 batches	ms/batch	88.68	loss	0.15143

syn_mse 0.19395764780128064

sen_mse 0.084464144499900805

epoch	8	100/	17 batches	ms/batch	94.29	loss	0.22550
epoch	8	200/	17 batches	ms/batch	93.24	loss	0.21143
epoch	8	300/	17 batches	ms/batch	89.14	loss	0.20792
epoch	8	400/	17 batches	ms/batch	85.42	loss	0.21526
epoch	8	500/	17 batches	ms/batch	88.17	loss	0.20177
epoch	8	600/	17 batches	ms/batch	88.41	loss	0.20707
epoch	8	700/	17 batches	ms/batch	84.38	loss	0.20584
epoch	8	800/	17 batches	ms/batch	86.07	loss	0.21276
epoch	8	900/	17 batches	ms/batch	88.36	loss	0.20632
epoch	8	1000/	17 batches	ms/batch	84.91	loss	0.20236
epoch	8	1100/	17 batches	ms/batch	88.80	loss	0.20656
epoch	8	1200/	17 batches	ms/batch	87.48	loss	0.19932
epoch	8	1300/	17 batches	ms/batch	86.15	loss	0.20800
epoch	8	1400/	17 batches	ms/batch	89.13	loss	0.21005
epoch	8	1500/	17 batches	ms/batch	90.15	loss	0.20391
epoch	8	1600/	17 batches	ms/batch	87.41	loss	0.20172
epoch	8	1700/	17 batches	ms/batch	88.46	loss	0.20507
epoch	8	1800/	17 batches	ms/batch	86.98	loss	0.19650
epoch	8	1900/	17 batches	ms/batch	95.76	loss	0.20121
epoch	8	2000/	17 batches	ms/batch	87.85	loss	0.20382
epoch	8	2100/	17 batches	ms/batch	84.84	loss	0.19724
epoch	8	2200/	17 batches	ms/batch	89.29	loss	0.19956

epoch	8	100/	17 batches	ms/batch	88.95	loss	0.17329
epoch	8	200/	17 batches	ms/batch	87.32	loss	0.15844
epoch	8	300/	17 batches	ms/batch	85.96	loss	0.16529
epoch	8	400/	17 batches	ms/batch	88.00	loss	0.16821
epoch	8	500/	17 batches	ms/batch	89.71	loss	0.16362
epoch	8	600/	17 batches	ms/batch	85.71	loss	0.14998
epoch	8	700/	17 batches	ms/batch	89.36	loss	0.16062
epoch	8	800/	17 batches	ms/batch	89.82	loss	0.16259
epoch	8	900/	17 batches	ms/batch	85.21	loss	0.15908
epoch	8	1000/	17 batches	ms/batch	88.98	loss	0.15396
epoch	8	1100/	17 batches	ms/batch	88.47	loss	0.15822
epoch	8	1200/	17 batches	ms/batch	86.18	loss	0.16864
epoch	8	1300/	17 batches	ms/batch	94.50	loss	0.15576
epoch	8	1400/	17 batches	ms/batch	90.31	loss	0.15992
epoch	8	1500/	17 batches	ms/batch	84.97	loss	0.14849
epoch	8	1600/	17 batches	ms/batch	88.81	loss	0.15835
epoch	8	1700/	17 batches	ms/batch	88.21	loss	0.15257
epoch	8	1800/	17 batches	ms/batch	85.69	loss	0.16641
epoch	8	1900/	17 batches	ms/batch	86.69	loss	0.15042
epoch	8	2000/	17 batches	ms/batch	85.11	loss	0.15727
epoch	8	2100/	17 batches	ms/batch	87.23	loss	0.15850
epoch	8	2200/	17 batches	ms/batch	88.83	loss	0.16682

syn mse 0.2082602773155306

sen_mse 0.08656266274260138

epoch	9	100/	17 batches	ms/batch	199.01	loss	0.22109
epoch	9	200/	17 batches	ms/batch	92.92	loss	0.20910
epoch	9	300/	17 batches	ms/batch	89.16	loss	0.21194
epoch	9	400/	17 batches	ms/batch	86.15	loss	0.20465
epoch	9	500/	17 batches	ms/batch	88.36	loss	0.20572
epoch	9	600/	17 batches	ms/batch	88.04	loss	0.20716
epoch	9	700/	17 batches	ms/batch	85.64	loss	0.20374
epoch	9	800/	17 batches	ms/batch	89.09	loss	0.20079
epoch	9	900/	17 batches	ms/batch	86.40	loss	0.20441
epoch	9	1000/	17 batches	ms/batch	86.11	loss	0.20035
epoch	9	1100/	17 batches	ms/batch	88.01	loss	0.19077
epoch	9	1200/	17 batches	ms/batch	86.54	loss	0.20522
epoch	9	1300/	17 batches	ms/batch	87.69	loss	0.20769
epoch	9	1400/	17 batches	ms/batch	91.96	loss	0.20053
epoch	9	1500/	17 batches	ms/batch	85.43	loss	0.20130
epoch	9	1600/	17 batches	ms/batch	87.07	loss	0.19968
epoch	9	1700/	17 batches	ms/batch	87.69	loss	0.19870
epoch	9	1800/	17 batches	ms/batch	89.03	loss	0.20547
epoch	9	1900/	17 batches	ms/batch	87.27	loss	0.19679
epoch	9	2000/	17 batches	ms/batch	87.55	loss	0.19881
epoch	9	2100/	17 batches	ms/batch	86.30	loss	0.20023
epoch	9	2200/	17 batches	ms/batch	87.53	loss	0.19850
epoch	9	100/	17 batches	ms/batch	85.89	loss	0.16628
epoch	9	200/	17 batches	ms/batch	88.17	loss	0.16716
epoch	9	300/	17 batches	ms/batch	85.71	loss	0.16477
epoch	9	400/	17 batches	ms/batch	84.67	loss	0.15445
epoch	9	500/	17 batches	ms/batch	87.27	loss	0.16735
epoch	9	600/	17 batches	ms/batch	87.90	loss	0.15070
epoch	9	700/	17 batches	ms/batch	85.17	loss	0.16041
epoch	9	800/	17 batches	ms/batch	87.77	loss	0.15170
epoch	9	900/	17 batches	ms/batch	87.20	loss	0.16187
epoch	9	1000/	17 batches	ms/batch	86.09	loss	0.17312
epoch	9	1100/	17 batches	ms/batch	88.58	loss	0.16491
epoch	9	1200/	17 batches	ms/batch	92.03	loss	0.16646
epoch	9	1300/	17 batches	ms/batch	86.02	loss	0.15831
epoch	9	1400/	17 batches	ms/batch	87.75	loss	0.15201
epoch	9	1500/	17 batches	ms/batch	86.84	loss	0.16027
epoch	9	1600/	17 batches	ms/batch	86.08	loss	0.15829
epoch	9	1700/	17 batches	ms/batch	87.75	loss	0.14804
epoch	9	1800/	17 batches	ms/batch	86.63	loss	0.14767
epoch	9	1900/	17 batches	ms/batch	86.53	loss	0.15081
epoch	9	2000/	17 batches	ms/batch	89.10	loss	0.15256
epoch	9	2100/	17 batches	ms/batch	86.48	loss	0.15400
epoch	9	2200/	17 batches	ms/batch	86.34	loss	0.15238

syn mse 0.19272924847444176

sen_mse 0.08614719477718641

epoch	10	100/	17 batches	ms/batch	88.52	loss	0.21972
epoch	10	200/	17 batches	ms/batch	88.86	loss	0.20595
epoch	10	300/	17 batches	ms/batch	87.22	loss	0.20463
epoch	10	400/	17 batches	ms/batch	87.82	loss	0.20110
epoch	10	500/	17 batches	ms/batch	86.17	loss	0.19963
epoch	10	600/	17 batches	ms/batch	86.49	loss	0.20241
epoch	10	700/	17 batches	ms/batch	88.26	loss	0.20498
epoch	10	800/	17 batches	ms/batch	85.61	loss	0.20292
epoch	10	900/	17 batches	ms/batch	86.88	loss	0.19328
epoch	10	1000/	17 batches	ms/batch	87.84	loss	0.20297
epoch	10	1100/	17 batches	ms/batch	84.93	loss	0.19569
epoch	10	1200/	17 batches	ms/batch	85.93	loss	0.19453

..... 10 1000/ 17 batches ms/batch 85.40 loss 0.19202

```

| epoch  10 | 1500/   17 batches | ms/batchn 85.40 | loss  0.19682
| epoch  10 | 1400/   17 batches | ms/batch 86.66 | loss  0.20029
| epoch  10 | 1500/   17 batches | ms/batch 87.20 | loss  0.20568
| epoch  10 | 1600/   17 batches | ms/batch 87.49 | loss  0.19614
| epoch  10 | 1700/   17 batches | ms/batch 85.26 | loss  0.19466
| epoch  10 | 1800/   17 batches | ms/batch 90.10 | loss  0.19179
| epoch  10 | 1900/   17 batches | ms/batch 87.61 | loss  0.19984
| epoch  10 | 2000/   17 batches | ms/batch 85.07 | loss  0.20182
| epoch  10 | 2100/   17 batches | ms/batch 87.62 | loss  0.20299
| epoch  10 | 2200/   17 batches | ms/batch 87.36 | loss  0.19482
| epoch  10 | 100/    17 batches | ms/batch 88.66 | loss  0.16846
| epoch  10 | 200/    17 batches | ms/batch 87.75 | loss  0.15746
| epoch  10 | 300/    17 batches | ms/batch 85.02 | loss  0.15640
| epoch  10 | 400/    17 batches | ms/batch 87.06 | loss  0.15613
| epoch  10 | 500/    17 batches | ms/batch 87.61 | loss  0.16363
| epoch  10 | 600/    17 batches | ms/batch 87.47 | loss  0.15594
| epoch  10 | 700/    17 batches | ms/batch 86.38 | loss  0.15284
| epoch  10 | 800/    17 batches | ms/batch 86.54 | loss  0.15269
| epoch  10 | 900/    17 batches | ms/batch 86.33 | loss  0.15323
| epoch  10 | 1000/   17 batches | ms/batch 87.26 | loss  0.16105
| epoch  10 | 1100/   17 batches | ms/batch 86.42 | loss  0.15285
| epoch  10 | 1200/   17 batches | ms/batch 90.74 | loss  0.16485
| epoch  10 | 1300/   17 batches | ms/batch 88.21 | loss  0.15828
| epoch  10 | 1400/   17 batches | ms/batch 86.29 | loss  0.15676
| epoch  10 | 1500/   17 batches | ms/batch 86.13 | loss  0.15205
| epoch  10 | 1600/   17 batches | ms/batch 87.10 | loss  0.15843
| epoch  10 | 1700/   17 batches | ms/batch 85.68 | loss  0.16227
| epoch  10 | 1800/   17 batches | ms/batch 86.43 | loss  0.15633
| epoch  10 | 1900/   17 batches | ms/batch 87.53 | loss  0.14862
| epoch  10 | 2000/   17 batches | ms/batch 86.15 | loss  0.15551
| epoch  10 | 2100/   17 batches | ms/batch 86.24 | loss  0.15979
| epoch  10 | 2200/   17 batches | ms/batch 87.78 | loss  0.15591
syn mse 0.19367338270731857
sen_mse 0.0862170752209665
-----
```

```
In [71]: #model save
#torch.save(model.state_dict(), data_path+'15-Pooled.p')
#model Load
model.load_state_dict(torch.load(data_path+'15-Pooled.p'))
#model.eval()
```

Out[71]: <All keys matched successfully>

Evaluate the general model

The model is evaluated based on a few metrics the original paper uses the metrics as: AUROC, AUPRC

The original paper achieved these results:

{0.9577 AUROC, 0.8335 AUPRC} (General Model)

Our implementation for this project will aim to hit these metrics. Any ablations we will also aim to hit these metrics.

Achieved Results

```
In [30]: def dcg_score(y_score, y_true, k):
    """
        https://www.kaggle.com/davidgasquez/ndcg-scorer
        y_true: np.array, size= [n_samples]
        y_score: np.array, size=[n_samples]
        k: int, rank
    """
    order = np.argsort(y_score)[::-1]
    y_true = np.take(y_true, order[:k])

    #gain = 2 ** y_true -1
    gain = y_true

    discounts = np.log2(np.arange(len(y_true)) + 2)
    return np.sum(gain/discounts)

def evaluate_accuracy(data_loader):
    model.eval()

    syn_all=[]
    syn_true_all=[ ]
```

```

ri1_all=[]
ri1_true_all=[]
ri2_all=[]
ri2_true_all=[]

#loss
with torch.no_grad():
    for iteration, sample in enumerate(data_loader):
        d1=Variable(sample['d1'])
        d1_fp = Variable(sample['d1_fp']).float()
        d1_sm = Variable(sample['d1_sm'])
        d1_gn = Variable(sample['d1_gn']).float()

        d2=Variable(sample['d2'])
        d2_fp = Variable(sample['d2_fp']).float()
        d2_sm = Variable(sample['d2_sm'])
        d2_gn = Variable(sample['d2_gn']).float()

        cell = Variable(sample['cell'])
        c_ts = Variable(sample['c_ts'])
        c_ds = Variable(sample['c_ds'])
        c_gn = Variable(sample['c_gn']).float()

        syn_true = Variable(sample['syn']).float()
        ri_d1=Variable(sample['ri_d1'])
        ri_d2=Variable(sample['ri_d2'])

        if cuda:
            d1=d1.cuda()
            d1_fp=d1_fp.cuda()
            d1_sm=d1_sm.cuda()
            d1_gn=d1_gn.cuda()

            d2=d2.cuda()
            d2_fp=d2_fp.cuda()
            d2_sm=d2_sm.cuda()
            d2_gn=d2_gn.cuda()

            cell=cell.cuda()
            c_ts=c_ts.cuda()
            c_ds=c_ds.cuda()
            c_gn=c_gn.cuda()

        syn,ri1,ri2 = model((d1, d1_fp, d1_sm, d1_gn), (d2, d2_fp, d2_sm, d2_gn), (cell, c_ts,c_ds,c_gn) )

        syn_all.append(syn.data.cpu().numpy())
        syn_true_all.append(syn_true.numpy())

        ri1_all.append(ri1.data.cpu().numpy())
        ri1_true_all.append(ri_d1.numpy())

        ri2_all.append(ri2.data.cpu().numpy())
        ri2_true_all.append(ri_d2.numpy())

    return syn_all, syn_true_all, ri1_all, ri1_true_all, ri2_all, ri2_true_all

```

Evaluate synergy prediction (General Model)

```

In [73]: syn_all, syn_true_all, ri1_all, ri1_true_all, ri2_all, ri2_true_all= evaluate_accuracy(test_loader)

syn_all= [s.item() for syn in syn_all for s in syn]
syn_true_all = [s for syn in syn_true_all for s in syn]
#NDCG
print(dcg_score(syn_all,syn_true_all, k=20)/dcg_score(syn_true_all,syn_true_all, k=20))
#AUPRC
print(metrics.average_precision_score(syn_true_all, 1/(1 + np.exp(-np.array(syn_all)))))
#AUROC
print(metrics.roc_auc_score(syn_true_all, 1/(1 + np.exp(-np.array(syn_all)))))

1.0
0.7267833130011351
0.9257503044782515

```

Evaluate sensitivity prediction (General)

In [74]:

```

ri1_all= [r.item() for ri in ri1_all for r in ri]
ri1_true_all = [r for ri in ri1_true_all for r in ri]

ri2_all= [r.item() for ri in ri2_all for r in ri]
ri2_true_all = [r for ri in ri2_true_all for r in ri]

ri_all=ri1_all+ri2_all
ri_true_all=ri1_true_all+ri2_true_all
#NDCG
dcg_score(ri_all,ri_true_all, k=20)/dcg_score(ri_true_all,ri_true_all, k=20)
#AUC
metrics.roc_auc_score(ri_true_all, 1/(1 + np.exp(-np.array(ri_all))))

```

Out[74]: 0.8462647050183483

Transfer the general model to specific model

If you want to boost a bit more with general model's test set

```

In [ ]: #Use major's test set
          training(False, test_loader)
          training(True, test_loader)

epoch 10 | 100/ 17 batches | ms/batch 87.88 | loss  0.23159
epoch 10 | 200/ 17 batches | ms/batch 87.01 | loss  0.20515
epoch 10 | 300/ 17 batches | ms/batch 85.83 | loss  0.21154
epoch 10 | 400/ 17 batches | ms/batch 89.06 | loss  0.20682
epoch 10 | 500/ 17 batches | ms/batch 88.33 | loss  0.19914
epoch 10 | 100/ 17 batches | ms/batch 89.85 | loss  0.16814
epoch 10 | 200/ 17 batches | ms/batch 85.97 | loss  0.16068
epoch 10 | 300/ 17 batches | ms/batch 88.43 | loss  0.16447
epoch 10 | 400/ 17 batches | ms/batch 88.77 | loss  0.15749
epoch 10 | 500/ 17 batches | ms/batch 84.70 | loss  0.15284

```

Freeze layers

Examine the layer's ID that we'd like to fix or free

```

In [ ]: for i, param in enumerate(model.parameters()):
          print(i, param.size(), param.requires_grad)
release_after = 46
for i, param in enumerate(model.parameters()):
    if i>=release_after:
        param.requires_grad=True
    else:
        param.requires_grad=False

0 torch.Size([52, 52]) True
1 torch.Size([156, 52]) True
2 torch.Size([156]) True
3 torch.Size([52, 52]) True
4 torch.Size([52]) True
5 torch.Size([2048, 52]) True
6 torch.Size([2048]) True
7 torch.Size([52, 2048]) True
8 torch.Size([52]) True
9 torch.Size([52]) True
10 torch.Size([52]) True
11 torch.Size([52]) True
12 torch.Size([52]) True
13 torch.Size([156, 52]) True
14 torch.Size([156]) True
15 torch.Size([52, 52]) True
16 torch.Size([52]) True
17 torch.Size([2048, 52]) True
18 torch.Size([2048]) True
19 torch.Size([52, 2048]) True
20 torch.Size([52]) True
21 torch.Size([52]) True
22 torch.Size([52]) True
23 torch.Size([52]) True
24 torch.Size([52]) True
25 torch.Size([32, 167]) True
26 torch.Size([32]) True
27 torch.Size([32, 64]) True
28 torch.Size([32]) True
29 torch.Size([1, 1, 52]) True

```

```

27 torch.Size([1, 1, 1, 1]) True
30 torch.Size([1]) True
31 torch.Size([352]) True
32 torch.Size([352]) True
33 torch.Size([176, 352]) True
34 torch.Size([176]) True
35 torch.Size([64, 176]) True
36 torch.Size([64]) True
37 torch.Size([28, 28]) True
38 torch.Size([64, 128]) True
39 torch.Size([64]) True
40 torch.Size([92]) True
41 torch.Size([92]) True
42 torch.Size([46, 92]) True
43 torch.Size([46]) True
44 torch.Size([64, 46]) True
45 torch.Size([64]) True
46 torch.Size([192]) True
47 torch.Size([192]) True
48 torch.Size([96, 192]) True
49 torch.Size([96]) True
50 torch.Size([1, 96]) True
51 torch.Size([1]) True
52 torch.Size([128]) True
53 torch.Size([128]) True
54 torch.Size([64, 128]) True
55 torch.Size([64]) True
56 torch.Size([1, 64]) True
57 torch.Size([1]) True

```

Prostate or bone

```
In [40]: _train_loader_minor=_train_loader_prostate
        _test_loader_minor=_test_loader_prostate
        _test_minor=_test_prostate
# _train_loader_minor=_train_loader_bone
# _test_loader_minor=_test_loader_bone
# _test_minor=_test_bone
```

```
In [48]: #Use minor's train set
try:
    for epoch in range(1, epochs+1):
        epoch_start_time = time.time()
        training(False, _train_loader_minor)
        training(True, _train_loader_minor)
        evaluate(_test_loader_minor)
        print('*'*89)
except KeyboardInterrupt:
    print('*'*89)
    print('Existing from training early')
```

```
syn mse 0.18647881392594223
sen_mse 0.05407539304796156
```

```
-----
```

```
syn mse 0.13272172278100317
sen_mse 0.05336044122884562
```

```
-----
```

```
syn mse 0.12024202451601133
sen_mse 0.05167775625710959
```

```
-----
```

```
syn mse 0.11358097621372767
sen_mse 0.05193812506539481
```

```
-----
```

```
syn mse 0.10983221871512276
sen_mse 0.051680072323306576
```

```
-----
```

```
syn mse 0.10576765877859932
sen_mse 0.05199513592562833
```

```
-----
```

```
syn mse 0.10389151939978966
sen_mse 0.05193541600153996
```

```
-----
```

```
syn mse 0.10244132660247468
sen_mse 0.05179897245469984
```

```
-----
```

```
syn mse 0.10027597238729288
sen_mse 0.05152076679271656
```

```
-----
```

```
syn mse 0.0995721188220349
sen_mse 0.051157469277853494
```

Evaluate synergy prediction (No transfer Prostate)

```
In [75]: model.load_state_dict(torch.load(data_path+'15-Pooled-no-transfer-prostate.p'))  
  
Out[75]: <All keys matched successfully>  
  
In [76]: syn_all, syn_true_all, ri1_all, ri1_true_all, ri2_all, ri2_true_all= evaluate_accuracy(_test_loader_minor)  
  
syn_all= [s.item() for syn in syn_all for s in syn]  
syn_true_all = [s for syn in syn_true_all for s in syn]  
#NDCG  
print(dcg_score(syn_all,syn_true_all, k=20)/dcg_score(syn_true_all,syn_true_all, k=20))  
#AUROC  
print(metrics.roc_auc_score(syn_true_all, 1/(1 + np.exp(-np.array(syn_all)))))  
#AUPRC  
print(metrics.average_precision_score(syn_true_all, 1/(1 + np.exp(-np.array(syn_all)))))  
  
1.0  
0.986630348869761  
0.9114650739035156
```

Evaluate sensitivity prediction (No transfer Prostate)

```
In [77]: ri1_all= [r.item() for ri in ri1_all for r in ri]  
ri1_true_all = [r for ri in ri1_true_all for r in ri]  
  
ri2_all= [r.item() for ri in ri2_all for r in ri]  
ri2_true_all = [r for ri in ri2_true_all for r in ri]  
  
ri_all=ri1_all+ri2_all  
ri_true_all=ri1_true_all+ri2_true_all  
#NDCG  
dcg_score(ri_all,ri_true_all, k=20)/dcg_score(ri_true_all,ri_true_all, k=20)  
#AUC  
metrics.roc_auc_score(ri_true_all, 1/(1 + np.exp(-np.array(ri_all))))
```

Out[77]: 0.7897974169720681

Evaluate synergy prediction (Transfer Retrain All Parameters Prostate)

```
In [78]: model.load_state_dict(torch.load(data_path+'15-Pooled-transfer-retrain-prostate.p'))  
  
Out[78]: <All keys matched successfully>  
  
In [79]: syn_all, syn_true_all, ri1_all, ri1_true_all, ri2_all, ri2_true_all= evaluate_accuracy(_test_loader_minor)  
  
syn_all= [s.item() for syn in syn_all for s in syn]  
syn_true_all = [s for syn in syn_true_all for s in syn]  
#NDCG  
print(dcg_score(syn_all,syn_true_all, k=20)/dcg_score(syn_true_all,syn_true_all, k=20))  
#AUROC  
print(metrics.roc_auc_score(syn_true_all, 1/(1 + np.exp(-np.array(syn_all)))))  
#AUPRC  
print(metrics.average_precision_score(syn_true_all, 1/(1 + np.exp(-np.array(syn_all)))))  
  
1.0  
0.9969655452679639  
0.9813636350945647
```

Evaluate sensitivity prediction (Transfer Retrain All Parameters Prostate)

```
In [80]: ri1_all= [r.item() for ri in ri1_all for r in ri]  
ri1_true_all = [r for ri in ri1_true_all for r in ri]  
  
ri2_all= [r.item() for ri in ri2_all for r in ri]  
ri2_true_all = [r for ri in ri2_true_all for r in ri]  
  
ri_all=ri1_all+ri2_all  
ri_true_all=ri1_true_all+ri2_true_all  
#NDCG  
dcg_score(ri_all,ri_true_all, k=20)/dcg_score(ri_true_all,ri_true_all, k=20)  
#AUC
```

```
#AUC
metrics.roc_auc_score(ri_true_all, 1/(1 + np.exp(-np.array(ri_all))))
```

Out[80]: 0.8036510078963031

Select the top ranked drug combinations

```
In [ ]:
syn_all_prob=1/(1 + np.exp(-np.array(syn_all)))
order = np.argsort(syn_all_prob)[::-1]
syn_true_all_order = np.take(syn_true_all, order[:20])
for k in range(20):
    comb=_test_minor[order[k]]
    print(codes['drugs'].idx2item[comb['d1']], ',',
          codes['drugs'].idx2item[comb['d2']], ',',
          codes['cell'].idx2item[comb['cell1']], ',',
          codes['tissue'].idx2item[comb['c_ts']], ',',
          codes['disease'].idx2item[comb['c_disease']], ',',
          comb['ri_d1'], ',',
          comb['ri_d2'], ',',
          syn_true_all_order[k])
```

5-Fluoro-2'-deoxyuridine , Trisenox , DU-145 , prostate , prostate , 0 , 0 , 1.0
 cyclophosphamide , Trisenox , DU-145 , prostate , prostate , 0 , 0 , 1.0
 Nilotinib , Trisenox , DU-145 , prostate , prostate , 0 , 0 , 1.0
 Pralatrexate , Trisenox , DU-145 , prostate , prostate , 0 , 0 , 1.0
 Procarbazine hydrochloride , Trisenox , DU-145 , prostate , prostate , 0 , 0 , 1.0
 6-Mercaptopurine , Trisenox , DU-145 , prostate , prostate , 0 , 0 , 1.0
 5-Fluoro-2'-deoxyuridine , Trisenox , PC-3 , prostate , prostate , 0 , 0 , 1.0
 MK-2206 , topotecan , LNCAP , prostate , prostate , 0 , 1.0 , 1.0
 Bortezomib , Trisenox , DU-145 , prostate , prostate , 0 , 0 , 1.0
 122111-05-1 , Trisenox , DU-145 , prostate , prostate , 0 , 0 , 1.0
 Eloxatin (TN) (Sanofi Synthelab) , Trisenox , DU-145 , prostate , prostate , 0 , 0 , 1.0
 Antibiotic AY 22989 , Trisenox , DU-145 , prostate , prostate , 0 , 0 , 1.0
 Nilotinib , Trisenox , PC-3 , prostate , prostate , 0 , 0 , 1.0
 6-Mercaptopurine , Ixabepilone , PC-3 , prostate , prostate , 0 , 0 , 1.0
 cyclophosphamide , Ixabepilone , PC-3 , prostate , prostate , 0 , 0 , 1.0
 Pralatrexate , Trisenox , PC-3 , prostate , prostate , 0 , 0 , 1.0
 cyclophosphamide , Trisenox , PC-3 , prostate , prostate , 0 , 0 , 1.0
 6-Mercaptopurine , Trisenox , PC-3 , prostate , prostate , 0 , 0 , 1.0
 Procarbazine hydrochloride , Trisenox , PC-3 , prostate , prostate , 0 , 0 , 1.0
 cyclophosphamide , Ixabepilone , DU-145 , prostate , prostate , 0 , 0 , 1.0

Results

This section will go over the results of the project in comparison to the results of the paper.

Model	AUROC	AUPRC
Original General	.9577	.8335
Reproduced General	.9258	.7270
Original Prostate No Transfer (ALMANC,ONEIL)	.9775	.8575
Reproduced Prostate No Transfer (ALMANC,ONEIL)	.9866	.9115
Original Prostate Transfer Retrain (ALMANC,ONEIL)	.9928	.9628
Reproduced Prostate Transfer Retrain (ALMANC,ONEIL)	.9970	.9814

As shown by the table we have achieved similar results with the ALMANAC and ONEIL datasets. The original paper hypothesis was given data rich tissue we can use transfer learning on the model in order to improve the results of drug synergy prediction. The original paper showed that with an increase in the AUROC and AUPRC scores. We have also shown an increase in both of those metrics after the transfer learning. Therefore this shows as evidence that transfer learning can be applied to drug synergy prediction models.

The goal of the paper was to train a general model and use whatever data was available for a data poor tissue and use transfer learning to make a drug synergy prediction model for that data poor tissue. We have shown strong evidence that supports the hypothesis of the original paper. With these evaluation metrics we have proof that our model can perform effective drug synergy prediction on prostate tissue which is data poor.

Ablation Study

We plan to remove the dropout layer for the model. We believe that over-fitting is not necessarily possible with this scenario as this model can be trained and targeted to different forms of human body tissue. Therefore we do not need to worry about this being a general model in

totally, we can tailor it to one tissue at a time.

Ablation Results

When performing this ablation without a dropout in the encoder models we found that the model became overfitted to the pooled databases. To reiterate, because the data pool was varied in the tissues included we believe that the encoding for the drug and tissues would represent the data well. However we observed that the model did not perform as well at all.

Our results for the general model appear to be roughly the same, our AUROC was .9356 and our AUPRC was .8122. This appeared to be fine but after transfer learning we saw a decrease in results. We tested the transfer learning in the same manner the general model was transferred and retrained and retested again prostate tissue. We found that our AUROC was .7148 and our AUPRC was .6520. This is roughly 30% decrease in score from the original papers results.

The takeaway is that the dropout is a necessary hyperparameter to the performance of the model. We suspect that because the data rich tissue is multiple times more abundant than the data poor datasets overfitting is all but guaranteed to happen. Transfer learning relies on the model being general enough such that new hyperparameters can be trained. However, we see that the general model requires the dropout to not be overfitted.

Discussion

The original paper was able to be somewhat reproduced. The paper used many databases and applied different learning methods to obtain drug synergy score. We were able to use and demonstrate a few databases namely those using prostate tissue. We were able to achieve results close enough to the original paper results. There was some discrepancy which will be discussed further.

Predominantly the paper was reproducible and with more time, we may have achieved the exact same results of the original author. Using Google Colab we had all the hardware we need in order to run the code and evaluate the model. All the data was accessible and even the author was able to be reached via communication. The source code was up to date and fit into modern python and programming standards, and the github was public. For the purpose of reproduction this paper had all the fundamentals to be reproducible.

However there are some things that could not be reproduced given the time spent on the project. We could not use all the datasets used to train different models on different tissue. The reason for this is because the datasets provided and the code expectation of the input were not synchronized. There were elements in the input that the code looked for that were not in the datasets themselves. Furthermore, there was an expectation that the reader would know where to load in the data, and where to make changes, in order to run different datasets. This was not clear however, and would take a lot of experience in the deep learning for healthcare field for it to be recognizable.

One of the easiest parts was acquiring the data and building the preprocessing file. The preprocessing file was easy because the data that we got from the author slipped fairly easily into the provided code. Using the graphs we were able to understand and report on the data. Another part that was fairly easy was running the given code. There was minimal editing required on our part once the data had undergone the necessary preprocessing.

One of the difficult parts was using the data in computation. The lack of clarity on how to use the different datasets was a large challenge. There was also some additional preprocessing on our end that needed to be done. For example disease name and tissue name columns were missing in the cell file. We had to incorporate these through some adhoc method.

To improve the reproducibility we would ask the authors to add more information on how to use the data in various ways, and tell the readers what parts of the dataset they are targeting. Another possible improvement would be storing versioned files. The current drugcomb file we used is several versions ahead of the one the paper used. Having the original version would have helped compare results more accurately.

References

1. Jaeger, S., Duran-Frigola, M. & Aloy, P. Drug sensitivity in cancer cell lines is not tissue-specific. *Mol Cancer* 14, 40 (2015). <https://doi.org/10.1186/s12943-015-0312-6>
2. Kim, Y., Zheng, S., Tang, J., Jim Zheng, W., Li, Z., & Jiang, X. (2021). Anticancer drug synergy prediction in understudied tissues using transfer learning. *Journal of the American Medical Informatics Association : JAMIA*, 28(1), 42–51. <https://doi.org/10.1093/jamia/ocaa212>
3. Yao F, Madani Tonekaboni SA, Safikhani Z, Smirnov P, El-Hachem N, Freeman M, Manem VSK, Haibe-Kains B. Tissue specificity of in vitro drug sensitivity. *J Am Med Inform Assoc.* 2018 Feb 1;25(2):158–166. doi: 10.1093/jamia/ocx062. PMID: 29016819; PMCID: PMC6381764.
4. Zheng, S., Aldahdooh, J., Shadbahr, T., Wang, Y., Aldahdooh, D., Bao, J., Wang, W., & Tang, J. (2021). Drugcomb update: A more comprehensive drug sensitivity data repository and Analysis Portal. *Nucleic Acids Research*, 49(W1). <https://doi.org/10.1093/nar/gkab438>
5. Kim, Y. (2024). Genetic Features, https://drive.google.com/drive/folders/1Dg8X_R6-qyfjEtgAYGw7RAo94-6JhfNp?usp=sharing