QUANTITATIVE TRAIT LOCI are regions of the genome associated with traits such as height, BMI etc.. If the trait is an expression of a gene then we are faced with an eQTL.

Imports and Consts

```
import pandas as pd
import numpy as np
import statsmodels.api
import scipy.stats
import seaborn as sns
import matplotlib.pyplot as plt
```

```
DATASET_PAH = "./Dataset/"
CHOSEN_CHROMOSOME = 19
```

Load and preprocess data

```
rnaSeqData = pd.read_csv(DATASET_PAH + "GD660.GeneQuantRPKM.txt", delimiter="\t")
```

```
rnaSeqData
```

	TargetID	Gene_Symbol	Chr	Coord	HG00096.1.M_111124_6	HG00097.7.M_120219_2
0	ENSG00000225538.1	ENSG00000225538.1	11	55850277	0.00000	0.00000
1	ENSG00000237851.1	ENSG00000237851.1	6	143109260	0.00000	0.00000
2	ENSG00000243765.1	ENSG00000243765.1	15	58442766	0.00000	0.00000
3	ENSG00000257527.1	ENSG00000257527.1	16	18505708	0.70561	0.66697
4	ENSG00000212855.5	ENSG00000212855.5	Y	9578193	0.00000	0.00000
...	...	...	...	...	...	...
53929	ENSG00000172297.6	ENSG00000172297.6	Y	27600708	0.13907	0.10224
53930	ENSG00000259738.1	ENSG00000259738.1	15	59157205	0.00000	0.13191
53931	ENSG00000212040.1	ENSG00000212040.1	14	101498324	0.00000	0.00000
53932	ENSG00000125266.5	ENSG00000125266.5	13	107187462	0.12923	0.07601
53933	ENSG00000230711.2	ENSG00000230711.2	6	168690030	0.20363	0.18754

53934 rows × 664 columns

```
f = open(DATASET_PAH + "ALL.wgs.mergedSV.v8.20130502.svs.genotypes.vcf", "r")
columns = None
for l in f:
    if "#CHROM" in l:
        columns = l.split("\t")
        break
```

```
columns[0] = columns[0].replace("#", "")
```

```
columns[len(columns) - 1] = columns[len(columns) - 1].replace("\n", "")
```

```
vcfSourceFile = pd.read_csv(
    DATASET_PAH + "ALL.wgs.mergedSV.v8.20130502.svs.genotypes.vcf",
    sep="\t",
    comment="#",
    names=columns,
    header=None,
)
```

```
/tmp/ipykernel_8673/2618528307.py:1: DtypeWarning: Columns (0) have mixed types. Specify dtype option o
n import or set low_memory=False.
  vcfSourceFile = pd.read_csv(
```

```
vcfSourceFile
```

	CHROM	POS	ID	
0	1	645710	ALU_umary_ALU_2	
1	1	668630	DUP_delly_DUP20532	
2	1	713044	DUP_gs_CNV_1_713044_755966	
3	1	738570	UW_VH_21763	
4	1	766600	UW_VH_5595	
...	...	...	...	
68813	X	155064470	DUP_gs.X_CNV_X_155064470_155081667	
68814	X	155090084	UW_VH_7995	
68815	X	155120139	L1_umary_LINE1_3151	
68816	X	155122541	DEL_pindel_54975	GAGTAACTGGGATGACAGGCGTGTGCCACCACGCCTGG
68817	X	155146395	BI_GS_DEL1_B3_P3053_10	

68818 rows × 2513 columns

```
def processChr(row):
    if row["Chr"] not in ["X", "Y", "M"]:
        row["Chr"] = str(row["Chr"])
    return row


rnaSeqData = rnaSeqData.apply(lambda x: processChr(x), axis=1)
```

```
rnaSeqData["Chr"].unique()
```

```
array(['11', '6', '15', '16', 'Y', '4', '1', '7', '10', '22', '20', '14',
       '2', '8', '9', '12', '19', '17', '21', '5', '13', '3', 'X', '18',
       'M'], dtype=object)
```

```
def processCHROM(row):
    if row["CHROM"] not in ["X", "Y", "M"]:
        row["CHROM"] = str(row["CHROM"])
    return row


vcfSourceFile = vcfSourceFile.apply(lambda x: processCHROM(x), axis=1)

vcfSourceFile["CHROM"].unique()
```

```
array(['1', '2', '3', '4', '5', '6', '7', '8', '9', '10', '11', '12',
       '13', '14', '15', '16', '17', '18', '19', '20', '21', '22', 'X'],
      dtype=object)
```

```
vcfSourceFile.groupby(by="CHROM").size()
```

```
CHROM
1     4671
10    3126
11    3375
12    3299
13    2485
14    2097
15    1867
16    2062
17    1926
18    2005
19    1621
2     5642
20    1569
21     877
22     848
3     4811
4     4780
5     4425
6     4187
7     4200
8     3681
9     3001
X     2263
dtype: int64
```

```
rnaSeqData.groupby(by="Chr").size()
```

```
Chr
1     5172
10    2199
11    3121
12    2747
13    1185
14    2182
15    2021
16    2292
17    2207
18     562
19    1939
2     3872
20    1276
21     694
22    1187
3     2917
4     2494
5     2734
6     2794
7     2770
8     2315
9     2334
M       37
X     2328
Y      555
dtype: int64
```

```
vcfSourceFile.columns
```

```
Index(['CHROM', 'POS', 'ID', 'REF', 'ALT', 'QUAL', 'FILTER', 'INFO', 'FORMAT',
       'HG00096',
       ...
       'NA21128', 'NA21129', 'NA21130', 'NA21133', 'NA21135', 'NA21137',
       'NA21141', 'NA21142', 'NA21143', 'NA21144'],
      dtype='object', length=2513)
```

```
vcfSourceFile["CHROM"] = vcfSourceFile["CHROM"].apply(lambda x: str(x))
# I pick only 1 chromosome to make sure the computation time is reasonable
vcfSourceFile = vcfSourceFile[vcfSourceFile["CHROM"] == str(CHOSEN_CHROMOSOME)].copy()
```

```
vcfSourceFile
```

	CHROM	POS	ID	REF	ALT	QUAL	FILTER	
61640	19	251548	DUP_gs_CNV_19_251548_283564	C	<CN2>	.	PASS	AC=2;AF=0.0003993
61641	19	266428	DUP_uwash_chr19_266428_341459	T	<CN2>	.	PASS	AC=4;AF=0.0007987
61642	19	286841	DUP_gs_CNV_19_286841_307454	T	<CN0>, <CN2>	.	PASS	AC=3,3;AF=0.00
61643	19	293937	BI_GS_DEL1_B2_P2673_28	G	<CN0>	100	PASS	AC=8;AF=0.001597
61644	19	323368	DUP_gs_CNV_19_323368_334484	A	<CN0>, <CN2>	.	PASS	AC=1,8;AF=0.00
...	...	...	...	...	...	...	...	
63256	19	58999008	SVA_umary_SVA_769	T	<INS:ME:SVA>	.	.	AC=6;AF=0.0011980
63257	19	59015810	BI_GS_DEL1_B2_P2732_9	G	<CN0>	100	PASS	AC=2;AF=0.0003993
63258	19	59020256	UW_VH_3019	C	<CN0>	100	PASS	AC=1;AF=0.0001996
63259	19	59021221	BI_GS_DEL1_B5_P2732_16	T	<CN0>	100	PASS	AC=1;AF=0.0001996
63260	19	59035015	YL_CN_JPT_4322	C	<CN0>	100	PASS	AC=2;AF=0.0003993

1621 rows × 2513 columns

```
newColumns = []
for i in range(len(rnaSeqData.columns)):
    newColumns.append(rnaSeqData.columns[i].split(".")[0])
```

```
rnaSeqData.columns = newColumns
```

```
rnaSeqData = rnaSeqData[rnaSeqData["Chr"] == str(CHOSEN_CHROMOSOME)].copy()
```

```
commonColumns = rnaSeqData.columns.intersection(vcfSourceFile.columns)
```

```
commonColumns
```

```
Index(['HG00096', 'HG00097', 'HG00099', 'HG00100', 'HG00101', 'HG00102',
       'HG00103', 'HG00105', 'HG00106', 'HG00108',
       ...
       'NA20809', 'NA20810', 'NA20811', 'NA20812', 'NA20813', 'NA20814',
       'NA20815', 'NA20819', 'NA20826', 'NA20828'],
      dtype='object', length=445)
```

```
# Data preprocessing: ensuring proper types, mapping

vcfSourceFile["QUAL"] = vcfSourceFile["QUAL"].apply(lambda x: int(x) if x != "." else 0)
```

```
for column in commonColumns:
    vcfSourceFile[column] = vcfSourceFile[column].apply(
        # lambda x: sum([int(n) for n in x.split("|")]) if x != "." else 0
        lambda x: max([int(n) for n in x.split("|")]) if x != "." else 0
    )
for column in commonColumns:
    vcfSourceFile[column] = vcfSourceFile[column].apply(lambda x: int(x))

vcfSourceFile["ALT"] = vcfSourceFile["ALT"].apply(lambda x: x.split(","))


def processRow(row):
    for column in commonColumns:
        if row[column] != 0:
            row[column] = row["ALT"][row[column] - 1]
    return row


vcfSourceFile = vcfSourceFile.apply(lambda x: processRow(x), axis=1)

# 1. DONE: For each row count the number of occurences of each ALT, save it to a dictionary and turn to
# 2. Find a way to merge rna and vcf dataframes, perhaps on closest position
```

```
def countUniqueALTs(row, d):
    uniqueALTs = row[commonColumns].unique()
    rowId = row.name
    d[rowId] = {}
    for alt in uniqueALTs:
        if alt == 0:
            continue
        d[rowId][alt] = 0
        for column in commonColumns:
            if row[column] == alt:
                d[rowId][alt] = d[rowId][alt] + 1
    for alt in d[rowId]:
        row[alt] = d[rowId][alt]
    return row


d = {}
vcfSourceFileCountedAlts = vcfSourceFile.apply(lambda x: countUniqueALTs(x, d), axis=1)
```

```
vcfSourceFileCountedAlts["POS"]
```

```
61640      251548
61641      266428
61642      286841
61643      293937
61644      323368
           ...
63256    58999008
63257    59015810
63258    59020256
63259    59021221
63260    59035015
Name: POS, Length: 1621, dtype: int64
```

```
rnaSeqData.columns
```

```
Index(['TargetID', 'Gene_Symbol', 'Chr', 'Coord', 'HG00096', 'HG00097',
       'HG00099', 'HG00099', 'HG00100', 'HG00101',
       ...
       'NA20810', 'NA20811', 'NA20812', 'NA20813', 'NA20814', 'NA20815',
       'NA20816', 'NA20819', 'NA20826', 'NA20828'],
      dtype='object', length=664)
```

```
rnaSeqData["Coord"]
```

```
33       58581338
51       32417479
86       50706885
140      55583407
193      45681485
           ...
53781    19384074
53798    40079595
53806    17666460
53807    58790318
53863    52196593
Name: Coord, Length: 1939, dtype: int64
```

```
rnaSeqData = rnaSeqData.reset_index()
```

```
vcfSourceFileCountedAlts = vcfSourceFileCountedAlts.reset_index()
```

```
def findClosestRna(rnaRow, closestRna, vcfRow):
    distance = np.abs(vcfRow["POS"] - rnaRow["Coord"])
    if distance < closestRna["distance"]:
        closestRna["closest"] = rnaRow.name
        closestRna["distance"] = distance


def mergeRnaOnClosest(row, rna):
    closestRna = {"closest": -1, "distance": np.inf}
    rna.apply(lambda x: findClosestRna(x, closestRna, row), axis=1)
    # Merge on closestRna
    # print(closestRna["closest"])
    return pd.concat([row, rna.iloc[closestRna["closest"]]], axis=0)


rnaSeqData["mean"] = rnaSeqData[commonColumns].mean(axis=1)
vcfRnaMerged = vcfSourceFileCountedAlts.apply(
    lambda x: mergeRnaOnClosest(x, rnaSeqData), axis=1
).reset_index()
```

```
vcfRnaMerged
```

	level_0	index	<CN0>	<CN2>	<CN3>	<INS:ME:ALU>	<INS:ME:LINE1>	<INS:ME:SVA>	<INS:MT>	<IN
0	0	61640	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
1	1	61641	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
2	2	61642	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
3	3	61643	3.0	NaN	NaN	NaN	NaN	NaN	NaN	N
4	4	61644	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
...	...	...	...	...	...	...	...	...	...	
1616	1616	63256	NaN	NaN	NaN	NaN	NaN	1.0	NaN	N
1617	1617	63257	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
1618	1618	63258	1.0	NaN	NaN	NaN	NaN	NaN	NaN	N
1619	1619	63259	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
1620	1620	63260	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N

1621 rows × 3193 columns

```
vcfRnaMerged["HG00108"].take([1], axis=1)
```

	HG00108
0	0.00000
1	0.10209
2	0.10209
3	0.10209
4	5.04683
...	...
1616	0.00000
1617	1.97912
1618	1.97912
1619	1.97912
1620	3.68633

1621 rows × 1 columns

```
vcfRnaMergedFilteredQuality = vcfRnaMerged[vcfRnaMerged["QUAL"] > 90]
```

```
vcfRnaMergedFilteredQuality
```

	level_0	index	<CN0>	<CN2>	<CN3>	<INS:ME:ALU>	<INS:ME:LINE1>	<INS:ME:SVA>	<INS:MT>	<IN
3	3	61643	3.0	NaN	NaN	NaN	NaN	NaN	NaN	N
6	6	61646	1.0	NaN	NaN	NaN	NaN	NaN	NaN	N
8	8	61648	354.0	NaN	NaN	NaN	NaN	NaN	NaN	N
10	10	61650	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
11	11	61651	142.0	NaN	NaN	NaN	NaN	NaN	NaN	N
...	...	...	...	...	...	...	...	...	...	
1614	1614	63254	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
1617	1617	63257	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
1618	1618	63258	1.0	NaN	NaN	NaN	NaN	NaN	NaN	N
1619	1619	63259	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
1620	1620	63260	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N

968 rows × 3193 columns

```
rnaBaseColumns = ["TargetID", "Gene_Symbol", "Chr", "Coord"]
vcfBaseColumns = [
    "CHROM",
    "POS",
    "ID",
    "REF",
    "ALT",
    "QUAL",
    "FILTER",
    "INFO",
    "FORMAT",
]
```

```
vcfSourceFile[vcfBaseColumns + list(commonColumns)][
    vcfSourceFile["CHROM"] == str(CHOSEN_CHROMOSOME)
]
```

	CHROM	POS	ID	REF	ALT	QUAL	FILTER	
61640	19	251548	DUP_gs_CNV_19_251548_283564	C	[<CN2>]	0	PASS	AC=2;AF=0.000399
61641	19	266428	DUP_uwash_chr19_266428_341459	T	[<CN2>]	0	PASS	AC=4;AF=0.000798
61642	19	286841	DUP_gs_CNV_19_286841_307454	T	[<CN0>, <CN2>]	0	PASS	AC=3,3;AF=0.0
61643	19	293937	BI_GS_DEL1_B2_P2673_28	G	[<CN0>]	100	PASS	AC=8;AF=0.0015
61644	19	323368	DUP_gs_CNV_19_323368_334484	A	[<CN0>, <CN2>]	0	PASS	AC=1,8;AF=0.0
...	...	...	...	...	...	...	...	
63256	19	58999008	SVA_umary_SVA_769	T	[<INS:ME:SVA>]	0	.	AC=6;AF=0.001198
63257	19	59015810	BI_GS_DEL1_B2_P2732_9	G	[<CN0>]	100	PASS	AC=2;AF=0.000399
63258	19	59020256	UW_VH_3019	C	[<CN0>]	100	PASS	AC=1;AF=0.000199
63259	19	59021221	BI_GS_DEL1_B5_P2732_16	T	[<CN0>]	100	PASS	AC=1;AF=0.000199
63260	19	59035015	YL_CN_JPT_4322	C	[<CN0>]	100	PASS	AC=2;AF=0.000399

1621 rows × 454 columns

```
rnaSeqData[rnaBaseColumns + list(commonColumns)][
    rnaSeqData["Chr"] == str(CHOSEN_CHROMOSOME)
]
```

	TargetID	Gene_Symbol	Chr	Coord	HG00096	HG00097	HG00099	HG00099	HG00100
0	ENSG00000243642.1	ENSG00000243642.1	19	58581338	0.00000	0.00000	0.00000	0.00000	0.00000
1	ENSG00000221504.1	ENSG00000221504.1	19	32417479	0.00000	0.00000	0.00000	0.00000	0.00000
2	ENSG00000105357.9	ENSG00000105357.9	19	50706885	0.00000	0.00893	0.01107	0.00000	0.00000
3	ENSG00000131037.8	ENSG00000131037.8	19	55583407	0.30019	0.36200	0.43385	0.25589	0.35353
4	ENSG00000007255.5	ENSG00000007255.5	19	45681485	20.41195	19.88715	19.50962	9.02891	13.88066
...	...	...	...	...	...	...	...	...	...
1934	ENSG00000213996.3	ENSG00000213996.3	19	19384074	0.00000	0.00000	0.00000	0.02572	0.00000
1935	ENSG00000244253.1	ENSG00000244253.1	19	40079595	0.00000	0.00000	0.00000	0.00000	0.00000
1936	ENSG00000130309.4	ENSG00000130309.4	19	17666460	11.26085	10.45836	13.02215	7.23676	11.28219
1937	ENSG00000083842.6	ENSG00000083842.6	19	58790318	2.14528	2.32057	2.45383	1.12668	1.55192
1938	ENSG00000182310.7	ENSG00000182310.7	19	52196593	0.67427	0.45852	0.22117	0.00000	0.13868

1939 rows × 642 columns

Perform linear regression

```
altColumns = []
for column in list(vcfRnaMerged.columns):
```

```
    if "<" in column:
        altColumns.append(column)
```

```
altColumns
```

```
['<CN0>',
 '<CN2>',
 '<CN3>',
 '<INS:ME:ALU>',
 '<INS:ME:LINE1>',
 '<INS:ME:SVA>',
 '<INS:MT>',
 '<INV>']
```

```
p_values = {}
for column in commonColumns:
    mod = statsmodels.api.OLS(
        vcfRnaMerged[column].take([1], axis=1)[column],
        vcfRnaMerged[altColumns + ["A", "C", "T", "G"]].fillna(value=0),
    )
    fii = mod.fit()
    p_values[column] = fii.summary2().tables[1]["P>|t|"]
```

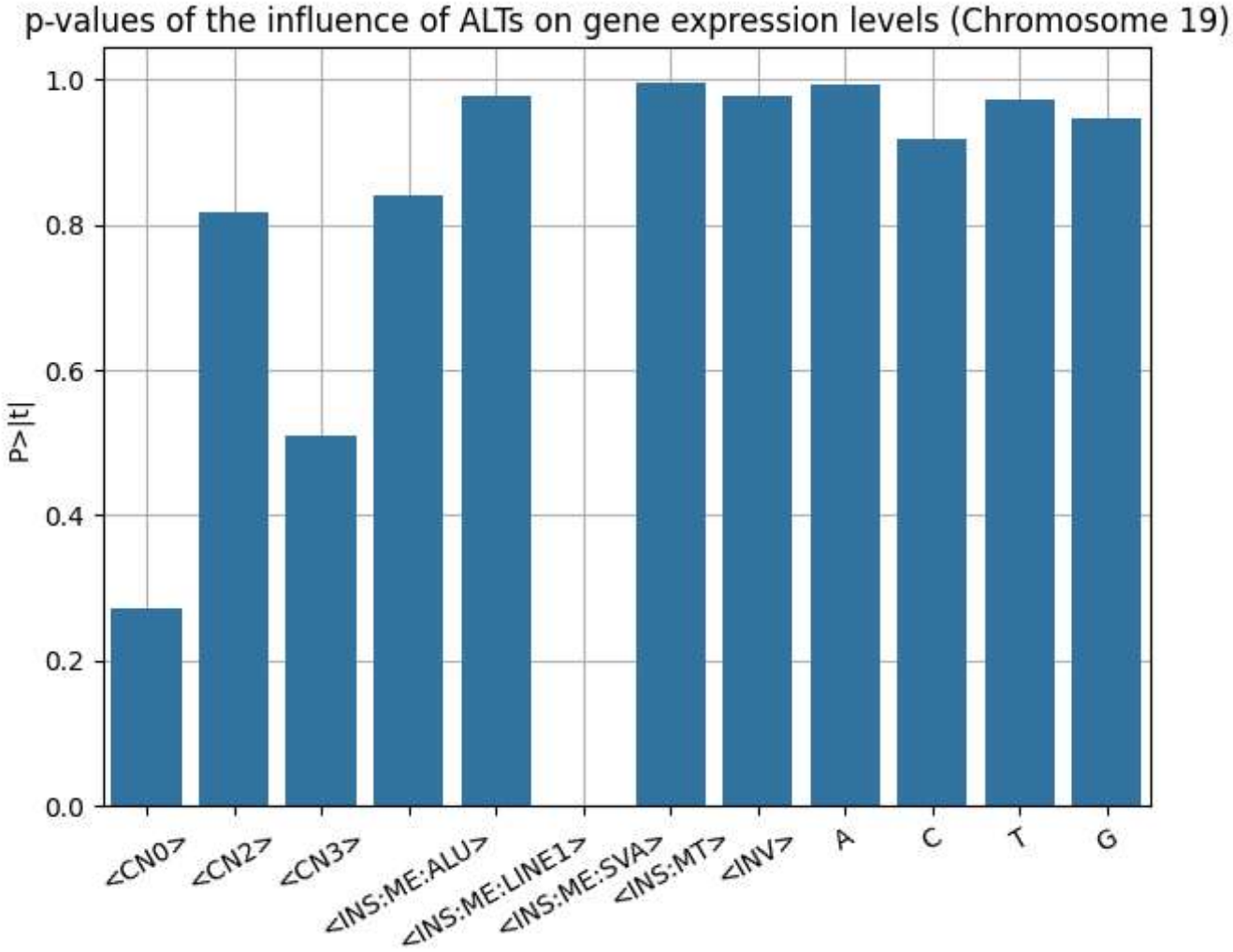
```
mod = statsmodels.api.OLS(
    vcfRnaMerged["mean"],
    vcfRnaMerged[altColumns + ["A", "C", "T", "G"]].fillna(value=0),
)
fii = mod.fit()
pValuesMean = fii.summary2().tables[1]["P>|t|"]
```

Represent the results

```
# p-values for the effect of different ALT values on gene expression levels
print(pValuesMean)
```

```
<CN0>             2.727343e-01
<CN2>             8.181202e-01
<CN3>             5.089327e-01
<INS:ME:ALU>      8.398458e-01
<INS:ME:LINE1>    9.770537e-01
<INS:ME:SVA>      3.872852e-25
<INS:MT>          9.949357e-01
<INV>             9.778868e-01
A                 9.922420e-01
C                 9.175091e-01
T                 9.731486e-01
G                 9.466638e-01
Name: P>|t|, dtype: float64
```

```
fig, ax = plt.subplots()
plt.title(f"p-values of the influence of ALTs on gene expression levels (Chromosome {str(CHOSEN_CHROMOSO
sns.barplot(pValuesMean)
plt.tight_layout()
plt.xticks(rotation=30)
ax.grid()
ax.set_axisbelow(True)
plt.show()
```




```
len(p_values)
```

```
445
```

```
p_values
df = pd.DataFrame(p_values)
```

```
df.mean()
```

```
HG00096    0.746328
HG00097    0.756219
HG00099    0.779400
HG00100    0.769543
HG00101    0.761933
             ...
NA20814    0.814994
NA20815    0.803355
NA20819    0.745504
NA20826    0.782771
NA20828    0.790788
Length: 445, dtype: float64
```

```
df.mean().mean()
```

```
0.7643071966863302
```

```
vcfRnaMerged.to_csv("mergedData.csv")
```

```
df.to_csv("p_values.csv")
```

```
```