ALL DATA COV MATRIX CALCULATION AND PCA

In [1]:
```
import scipy.sparse as sp
import pandas as pd
import numpy as np


data = pd.read_csv('select.freq.filtered', delimiter='\t', header=None)

locations = data[0]  # locations ar1_10.2
alleles = data[1]  # A, C, G, T

freq_data = sp.csr_matrix(data.iloc[:, 2:].values)
```

In [2]:
```
mean_freq = freq_data.mean(axis=1).reshape(-1, 1)
```

In [3]:
```
print(len(mean_freq))
```

```
256
```

In [4]:
```
from scipy.sparse import save_npz

# save
save_npz('freq_data.npz', freq_data)
```

In [5]:
```
import numpy as np
from scipy.sparse import load_npz

freq_data = sp.csr_matrix(data.iloc[:, 2:].values)

dense_data = freq_data.toarray()


num_locations = dense_data.shape[0] // 4

# combine by summing
combined_data = dense_data.reshape(num_locations, 4, -1).sum(axis=1)

print(f"Combined Data shape: {combined_data.shape}")
```

```
Combined Data shape: (64, 1150852)
```

In [6]:
```
print(len(combined_data))
print(combined_data)
```

```
64
[[  2  26   5 ...  14   8 237]
 [  2  18   0 ...  48   3 768]
 [  0   2   0 ...   2   1  53]
 ...
 [  8   0   0 ...   4   0  62]
 [  0   0   0 ...   0   0   0]
 [  0   0   0 ...   0   0   0]]
```

In [7]:
```
cd=pd.DataFrame(combined_data)
```

In [8]:
```
print(cd)
```

```
    0        1        2        3        4        5        6        7        \
0         2       26        5        7       33       13        1        0
1         2       18        0        2       15        2        0        0
2         0        2        0        0        1        1        2        0
3         0        6        4        1        6        2        0        1
4         0        6        1        5       14        5        0        1
..      ...      ...      ...      ...      ...      ...      ...      ...
59        7        3        3        3        7        1        4        1
60        1        0        1        1        1        0        1        0
61        8        0        0        2        2        0        0        2
62        0        0        0        0        0        0        0        0
63        0        0        0        0        0        0        0        0

    8        9        ...  1150842  1150843  1150844  1150845  1150846  \
0        10       23  ...        1       14      525        8       14
1         6        9  ...        0        4     1561        4        4
2         1        2  ...        0        1      117        1        0
3         4        5  ...        1        3      492        1        0
4         2       11  ...        2        7      423        7        1
..      ...      ...  ...      ...      ...      ...      ...      ...
59        0       15  ...       10        6      441        5        4
60        0        0  ...        0        1      401        0        0
61        1        2  ...        0        0      125        0        0
62        0        0  ...        0        0        0        0        0
63        0        0  ...        0        0        0        0        0

    1150847  1150848  1150849  1150850  1150851
0        99      913       14        8      237
1       307     2763       48        3      768
2        26      220        2        1       53
3        98      828       16        1      233
4        84      820       10        7      209
..      ...      ...      ...      ...      ...
59       55      839       15        5      161
60       45      754       17        0      131
61       28      204        4        0       62
62        0        0        0        0        0
63        0        0        0        0        0

[64 rows x 1150852 columns]
```

In [9]:
```
cd_sparse = sp.csr_matrix(combined_data)
```

In [10]:
```
# save
cov_matrix = (cd_sparse @ cd_sparse.T) / (cd_sparse.T.shape[0] - 1)
save_npz('cov_matrix.npz', cov_matrix)
```

In [11]:
```
print(cov_matrix)
```

```
  (0, 0)        7418.638378035037
  (0, 1)        281.54947078292497
  (0, 2)        83.1689523665531
  (0, 3)        138.7054640435643
  (0, 4)        272.5414523687254
  (0, 5)        269.7632299924143
  (0, 6)        29.711636866979305
  (0, 7)        48.7894618851615
  (0, 8)        31.315218911918222
  (0, 9)        181.3734870978085
  (0, 11)       32.06983093380464
  (0, 12)       190.79289412791056
  (0, 13)       120.45026072011059
  (0, 14)       130.78279812069502
  (0, 15)       66.91906771597714
  (0, 16)       440.7237409534336
  (0, 17)       166.5199621845052
  (0, 18)       209.7588541001398
  (0, 20)       67.90968335605565
  (0, 21)       582.5025985118838
  (0, 22)       484.0244784077174
  (0, 23)       161.62408165783407
  (0, 24)       169.388987801201
  (0, 25)       315.1535576716708
  (0, 26)       594.166102301688
  :     :
  (63, 36)      0.0026580330555389014
  (63, 37)      0.001998521094390151
  (63, 38)      0.004441930362835849
  (63, 39)      0.00037102978578460637
  (63, 40)      0.08741618159084016
  (63, 41)      0.0001312072544577882
  (63, 42)      0.004257718853265974
  (63, 43)      0.0006464781279244663
  (63, 44)      0.0012608061338956998
  (63, 45)      0.000507450573532108
  (63, 46)      0.0001181734212335046
  (63, 47)      8.254761042046277e-05
  (63, 48)      0.0008515437706531949
  (63, 49)      0.001946385761493017
  (63, 50)      0.002130597271062892
  (63, 52)      0.001000129469410028
  (63, 54)      0.008491107884513287
  (63, 55)      0.000782898915671968
  (63, 56)      0.0019124977951098796
  (63, 57)      0.016202792542214413
  (63, 58)      0.0003606027192051795
  (63, 59)      0.0010461823468024965
  (63, 60)      0.0009184507812045173
  (63, 61)      0.0016431319084746852
  (63, 63)      1.2164911009331355e-05
```

In [ ]:

In [12]:
```
import numpy as np
from scipy.sparse.linalg import eigsh

eigenvalues, eigenvectors = eigsh(cov_matrix, which='LM')  # largest ev

sorted_indices = np.argsort(eigenvalues)[::-1]
eigenvalues = eigenvalues[sorted_indices]
eigenvectors = eigenvectors[:, sorted_indices]


print("eigenvectors: ", eigenvalues)
print("eigenvalues: \n", eigenvectors)
```

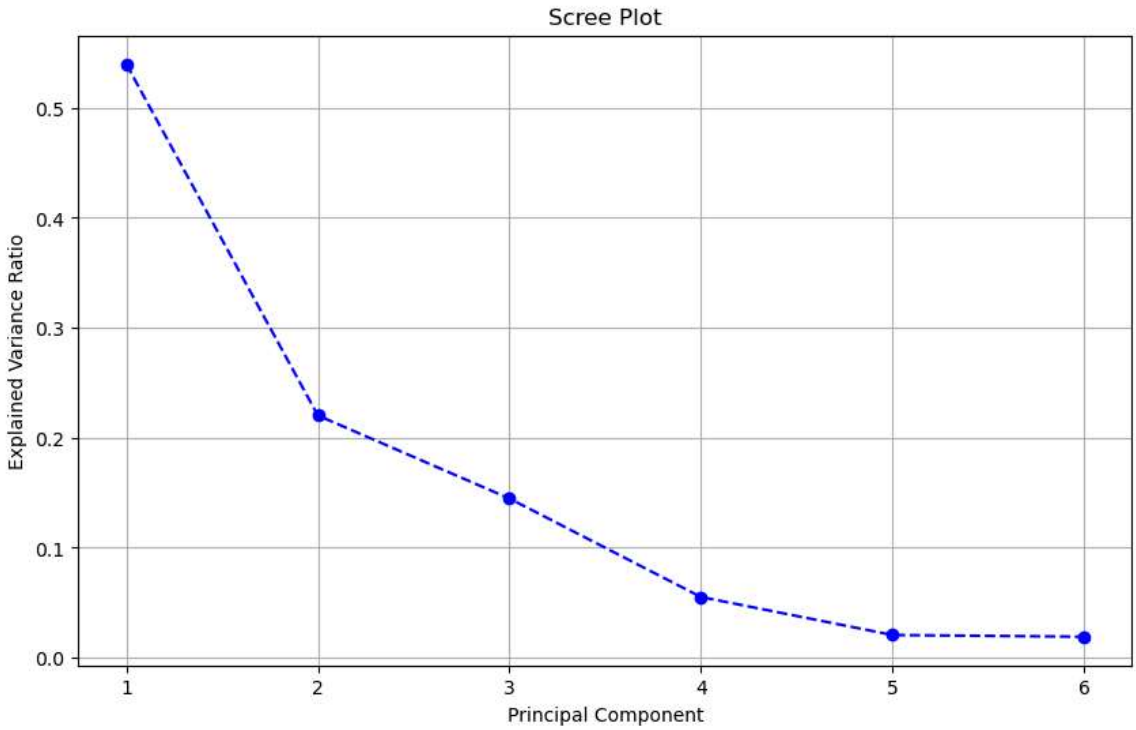
```
eigenvectors: [4328774.99137426 1766937.58514663 1161491.03296366  443426.319850
16
  166153.10002446  153558.5124637 ]
eigenvalues:
 [[ 1.15792651e-03  5.72018763e-04  1.03647555e-03 -2.73339214e-03
   6.30318836e-03  9.00555662e-03]
 [ 3.05464766e-03  3.78330796e-03  1.03418301e-01  8.05813684e-04
   2.49953880e-03  2.71858412e-03]
 [ 5.42294127e-04  4.88785323e-04  4.01826411e-04 -1.22643099e-03
   1.59450496e-03  1.65400713e-03]
 [ 1.57800297e-03  2.20459825e-04  4.57725465e-04 -1.17060659e-03
   2.92024276e-03  2.72433496e-03]
 [ 8.98554324e-04  4.09564066e-04  6.04804087e-04 -2.24165456e-03
   4.44708696e-03  4.99884330e-03]
 [ 6.83971011e-04  3.88660385e-04  9.22956185e-04 -2.58987650e-03
   1.50385137e-02  2.25061681e-02]
 [ 1.39238024e-03  5.40605180e-04  3.81945604e-05 -2.04116667e-04
   1.36548731e-03  6.85449404e-04]
 [ 6.35197477e-04  4.20227567e-04  1.55452987e-04 -9.76623445e-03
   6.13504819e-04  6.84215091e-04]
 [ 5.97714370e-04  2.72526788e-04  1.86262311e-04 -6.50016207e-04
   4.65238525e-03  6.90001461e-03]
 [ 7.58114785e-04  3.39023400e-04  4.90670772e-04 -1.33450734e-03
   2.89366155e-03  3.26885133e-03]
 [-2.11980148e-17  1.66400398e-17  3.88034681e-18 -4.33399705e-19
  -2.91158562e-17 -2.42259469e-18]
 [ 1.68111753e-03  6.61332925e-04  3.88681272e-05 -3.73871674e-04
   8.36414145e-03  2.81585199e-02]
 [ 3.72829755e-03  1.59722250e-03  9.22320321e-04 -3.23015441e-03
   1.21511382e-02  8.91128794e-03]
 [ 7.97329591e-03  2.95664440e-03 -5.16787822e-05 -5.70998008e-04
   6.58212937e-03  2.83147659e-03]
 [ 4.10961935e-04  9.67671904e-05  3.98015734e-04 -1.05120745e-03
   1.87245785e-03  2.17925796e-03]
 [ 1.31096680e-04  1.00623183e-04  1.85061527e-04 -5.55403880e-04
   9.01625127e-04  1.14189790e-03]
 [ 1.32059073e-02 -1.81888609e-02  1.09297249e-03 -1.64687177e-03
  -1.60870436e-03  8.85596243e-03]
 [ 1.07162514e-02  3.99833920e-03 -3.63702292e-05 -8.66987312e-04
   8.97130332e-03  3.89304261e-03]
 [ 3.48992077e-04  1.08826425e-04  7.49918511e-04 -1.78522267e-03
   3.61384794e-03  4.43774929e-03]
 [-1.28071411e-17  1.32551135e-17  3.56239377e-18 -3.88587349e-18
  -2.25893571e-17  2.26377787e-19]
 [ 5.70493523e-03  1.65366483e-02 -3.15360095e-04  7.48223257e-04
  -2.63360220e-03  1.62161757e-03]
 [ 7.78715913e-03  3.91968717e-03  2.98373494e-03 -1.09184111e-02
   8.68452573e-01 -2.27055347e-01]
 [ 1.95504212e-03  2.27843989e-03  2.45452265e-03 -1.57360264e-01
   7.55120076e-04  2.80960337e-03]
 [ 1.47333219e-03  4.78702380e-04  3.57221528e-04 -1.16038394e-03
   2.52952972e-02  9.79620081e-02]
 [ 9.62236907e-04  1.12393259e-03  1.25046833e-03 -8.31265177e-02
  -5.10498453e-04  4.21344228e-05]
 [ 1.70466448e-03  2.59686075e-04  8.32955353e-04 -2.00139765e-03
   5.12104599e-03  5.54845126e-03]
 [ 8.32717083e-02 -6.82890257e-02  1.52033807e-03 -1.63972617e-03
  -4.33509539e-03  1.04295253e-02]
 [ 1.31691610e-11  3.73761742e-11  1.34443988e-12 -9.56832823e-11
   1.82059081e-11  1.31796756e-10]
 [ 1.47611457e-06  7.91378942e-07  9.76399177e-07 -3.29788854e-06
```

```
  1.69278411e-04 -4.26302093e-05]
[ 1.81064536e-03  7.10787578e-04  7.32815813e-05 -3.42396962e-04
  1.86573165e-03  1.03100284e-03]
[ 1.33105961e-02  6.01371869e-03  2.75805004e-03 -9.24025198e-03
  3.50705106e-02  2.35519568e-02]
[ 1.87113804e-07 -2.65164022e-07  3.69759540e-08 -5.36054434e-08
  1.33407074e-08  1.64103671e-07]
[ 2.87771745e-03  5.18864083e-03  1.84577163e-01  3.75624645e-04
  3.99770784e-03  4.56879702e-03]
[ 2.94465765e-04  1.25915631e-04  4.42797105e-05 -1.48506613e-04
  4.22772968e-04  2.81793181e-04]
[ 2.23752799e-03  1.40096580e-03  3.44504319e-03 -1.17136956e-02
  1.76998102e-01  7.04949654e-01]
[ 1.55719098e-02  3.81935607e-03  6.30704737e-04 -2.88224043e-03
  1.46771168e-02  8.26475899e-03]
[ 1.16382855e-03  6.35386994e-04  2.22413325e-03 -7.96381257e-03
  1.12427686e-02  1.17469410e-02]
[ 8.15535941e-03  3.17068038e-02 -2.45615630e-04 -1.89931750e-03
 -3.61363803e-03  7.84524972e-03]
[ 1.42221312e-03  9.04768170e-04  3.49251417e-03 -1.12284037e-02
  5.04467820e-02  7.97758318e-02]
[ 5.82208110e-04  3.31244188e-04  5.49198995e-04 -1.95079911e-03
  2.38483223e-03  3.15718783e-03]
[ 1.44189066e-03  7.00564394e-04  5.84171299e-04 -1.85649744e-03
  6.25842680e-03  5.31047488e-03]
[ 1.58079242e-03  6.08156739e-04  1.30983632e-05 -1.90926417e-04
  1.36790092e-03  5.74791290e-04]
[ 1.97516192e-03  1.11569258e-03  2.53314754e-03 -7.48282966e-03
  1.24621873e-01  5.00474742e-01]
[ 1.78516757e-03  6.81803080e-03 -9.44124517e-05 -2.58811049e-05
 -1.00121304e-03  1.30104652e-03]
[ 1.54026604e-03  1.19292935e-03  1.08301660e-03 -4.87150959e-02
  1.79013197e-03  2.35545607e-03]
[ 8.79383265e-04  2.20269061e-04  5.07494165e-04 -1.53672794e-03
  2.64121652e-03  3.33984087e-03]
[ 3.52185210e-04  1.49210482e-04  6.86824434e-05 -2.32626876e-04
  5.89604813e-04  4.41022482e-04]
[ 8.25564172e-04  3.17087729e-04  6.74012371e-05 -3.09564819e-04
  1.71177911e-03  2.00922496e-03]
[ 1.51641069e-02 -1.66381647e-02  9.16516854e-04 -2.04544501e-03
 -1.87032253e-03  3.85606801e-03]
[ 2.86072995e-04  4.43956388e-04  1.15492234e-03 -5.49630675e-02
  6.15283431e-04  1.96213226e-03]
[ 1.04274987e-03  2.60131561e-04  1.09240481e-03 -2.97376209e-03
  4.59538027e-03  4.88987993e-03]
[ 5.25886209e-10  4.64647188e-10  3.22140890e-09 -1.23480430e-08
  2.15613187e-08  2.87740893e-08]
[ 1.69565461e-02  6.44360204e-03  8.98591309e-04 -4.61914160e-03
  8.32573935e-02  2.78453312e-01]
[ 1.78587567e-09  1.89743310e-09  3.23906867e-08 -5.46223256e-08
  1.10323317e-07  1.58372583e-07]
[ 1.41764901e-02  1.75913491e-02  1.28057528e-02 -9.80487990e-01
 -1.75401393e-02 -1.45466682e-02]
[ 4.02566974e-03  2.34711666e-03  2.44651407e-02 -4.83921550e-04
  3.71025336e-03  3.02386974e-03]
[ 8.38550935e-04  5.61644853e-04  1.78325862e-03 -4.91707345e-03
  7.61522775e-02  3.05018191e-01]
[ 1.32181871e-02  3.04572327e-02  9.76321726e-01  1.43475003e-02
 -5.96065728e-03 -5.49785475e-03]
[ 4.38534972e-02  1.96875402e-02  1.27016331e-04 -4.23429603e-03
  4.23399874e-01 -1.06907848e-01]
```

```
[ 5.04757212e-01  8.59992518e-01 -3.43236705e-02  2.27669132e-02
 -2.52434483e-02 -1.37032062e-03]
[ 8.56993862e-01 -5.01989316e-01  3.51142534e-03  4.13448704e-03
 -1.75686426e-02 -2.58609953e-03]
[ 3.37532203e-04  8.91248626e-05  6.17470707e-04 -1.31885577e-03
  2.26345466e-03  2.74181596e-03]
[ 5.93293719e-18 -8.67116471e-18 -4.32614835e-18  3.61200108e-18
  1.18507739e-17  4.50672166e-19]
[ 2.26689686e-09  2.63681171e-09  1.61148010e-08 -3.30540852e-08
  1.63058902e-07  1.21514902e-07]]
```

In [17]:
```
import matplotlib.pyplot as plt
explained_variance_ratio = eigenvalues / eigenvalues.sum()

# Scree Plot
plt.figure(figsize=(10, 6))
plt.plot(range(1, len(eigenvalues) + 1), explained_variance_ratio, marker='o', lines
plt.title('Scree Plot')
plt.xlabel('Principal Component')
plt.ylabel('Explained Variance Ratio')
plt.grid(True)
plt.show()
```



choose first two PCs and rerun

In [18]:
```
import numpy as np
from scipy.sparse.linalg import eigsh

eigenvalues, eigenvectors = eigsh(cov_matrix, k = 2, which='LM')

# 3. 排序特征值和特征向量（按特征值降序）
sorted_indices = np.argsort(eigenvalues)[::-1]
eigenvalues = eigenvalues[sorted_indices]
eigenvectors = eigenvectors[:, sorted_indices]


print("eval: ", eigenvalues)
print("evect: \n", eigenvectors)
```

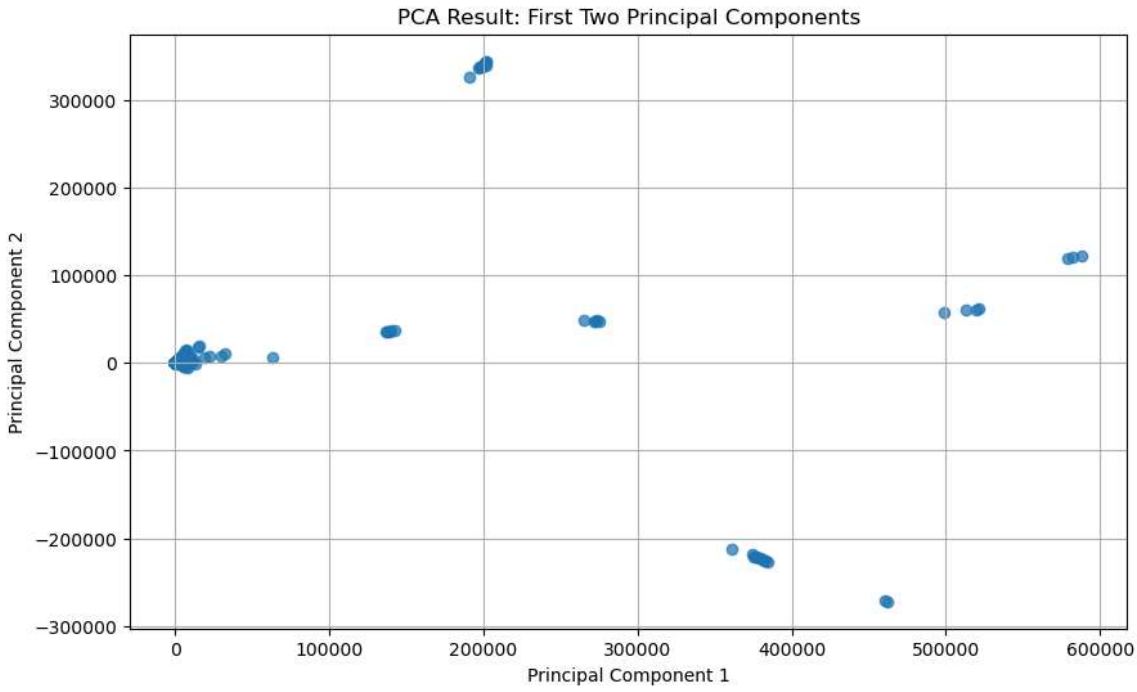
```
eval： [4328774.99137425 1766937.58514663]
evect：
 [[ 1.15792651e-03  5.72018763e-04]
 [ 3.05464766e-03  3.78330796e-03]
 [ 5.42294127e-04  4.88785323e-04]
 [ 1.57800297e-03  2.20459825e-04]
 [ 8.98554324e-04  4.09564066e-04]
 [ 6.83971011e-04  3.88660385e-04]
 [ 1.39238024e-03  5.40605180e-04]
 [ 6.35197477e-04  4.20227567e-04]
 [ 5.97714370e-04  2.72526788e-04]
 [ 7.58114785e-04  3.39023400e-04]
 [ 1.74263686e-18 -1.94690935e-17]
 [ 1.68111753e-03  6.61332925e-04]
 [ 3.72829755e-03  1.59722250e-03]
 [ 7.97329591e-03  2.95664440e-03]
 [ 4.10961935e-04  9.67671904e-05]
 [ 1.31096680e-04  1.00623183e-04]
 [ 1.32059073e-02 -1.81888609e-02]
 [ 1.07162514e-02  3.99833920e-03]
 [ 3.48992077e-04  1.08826425e-04]
 [ 3.08684044e-18 -9.88627375e-18]
 [ 5.70493523e-03  1.65366483e-02]
 [ 7.78715913e-03  3.91968717e-03]
 [ 1.95504212e-03  2.27843989e-03]
 [ 1.47333219e-03  4.78702380e-04]
 [ 9.62236907e-04  1.12393259e-03]
 [ 1.70466448e-03  2.59686075e-04]
 [ 8.32717083e-02 -6.82890257e-02]
 [ 1.31691403e-11  3.73761976e-11]
 [ 1.47611457e-06  7.91378942e-07]
 [ 1.81064536e-03  7.10787578e-04]
 [ 1.33105961e-02  6.01371869e-03]
 [ 1.87113804e-07 -2.65164022e-07]
 [ 2.87771745e-03  5.18864083e-03]
 [ 2.94465765e-04  1.25915631e-04]
 [ 2.23752799e-03  1.40096580e-03]
 [ 1.55719098e-02  3.81935607e-03]
 [ 1.16382855e-03  6.35386994e-04]
 [ 8.15535941e-03  3.17068038e-02]
 [ 1.42221312e-03  9.04768170e-04]
 [ 5.82208110e-04  3.31244188e-04]
 [ 1.44189066e-03  7.00564394e-04]
 [ 1.58079242e-03  6.08156739e-04]
 [ 1.97516192e-03  1.11569258e-03]
 [ 1.78516757e-03  6.81803080e-03]
 [ 1.54026604e-03  1.19292935e-03]
 [ 8.79383265e-04  2.20269061e-04]
 [ 3.52185210e-04  1.49210482e-04]
 [ 8.25564172e-04  3.17087729e-04]
 [ 1.51641069e-02 -1.66381647e-02]
 [ 2.86072995e-04  4.43956388e-04]
 [ 1.04274987e-03  2.60131561e-04]
 [ 5.25886221e-10  4.64647133e-10]
 [ 1.69565461e-02  6.44360204e-03]
 [ 1.78587568e-09  1.89743312e-09]
 [ 1.41764901e-02  1.75913491e-02]
 [ 4.02566974e-03  2.34711666e-03]
 [ 8.38550935e-04  5.61644853e-04]
 [ 1.32181871e-02  3.04572327e-02]
 [ 4.38534972e-02  1.96875402e-02]
```

```
[ 5.04757212e-01   8.59992518e-01]
[ 8.56993862e-01  -5.01989316e-01]
[ 3.37532203e-04   8.91248626e-05]
[-7.37009381e-20   1.32297543e-17]
[ 2.26689685e-09   2.63681167e-09]]
```

In [19]:
```
projected_data = cd.T.dot(eigenvectors)
```

In [20]:
```
np.save('projected_data_combined.npy', projected_data)
```

In [22]:
```
projected_data_np = projected_data.to_numpy()
plt.figure(figsize=(10, 6))
plt.scatter(projected_data_np[:, 0], projected_data_np[:, 1], alpha=0.7)
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.title('PCA Result: First Two Principal Components')
plt.grid(True)
plt.show()
```



In [ ]:
```

```