Read the https://pairtools.readthedocs.io/en/latest/ and https://cooler.readthedocs.io/en/latest/index.html for more information

```
!pip install -q condacolab
import condacolab
condacolab.install()
```

```
WARNING: Running pip as the 'root' user can result in broken
permissions and conflicting behaviour with the system package manager.
It is recommended to use a virtual environment instead:
https://pip.pypa.io/warnings/venv
✨🍰✨ Everything looks OK!
```

Homework report should include:
1) scaling plot in log-log coordinates with description; create correct labels for scaling plot, including units of measurement; make comments on operations in cell starting with ##!!!
2) replicates clusterization for all files (in directory for the lecture) with dendrogram and description; make comments on operations in cells starting with ##!!!

```
%%bash
pip install cooler
pip install hicrep
```

```
Looking in indexes: https://pypi.org/simple, https://us-
python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: cooler in
/usr/local/lib/python3.10/site-packages (0.9.1)
Requirement already satisfied: scipy>=0.16 in
/usr/local/lib/python3.10/site-packages (from cooler) (1.10.1)
Requirement already satisfied: numpy>=1.9 in
/usr/local/lib/python3.10/site-packages (from cooler) (1.24.3)
Requirement already satisfied: asciitree in
/usr/local/lib/python3.10/site-packages (from cooler) (0.3.3)
Requirement already satisfied: multiprocess in
/usr/local/lib/python3.10/site-packages (from cooler) (0.70.14)
Requirement already satisfied: simplejson in
/usr/local/lib/python3.10/site-packages (from cooler) (3.19.1)
Requirement already satisfied: pyfaidx in
/usr/local/lib/python3.10/site-packages (from cooler) (0.7.2.1)
Requirement already satisfied: click>=7 in
/usr/local/lib/python3.10/site-packages (from cooler) (8.1.3)
Requirement already satisfied: pandas>1.0 in
/usr/local/lib/python3.10/site-packages (from cooler) (2.0.1)
Requirement already satisfied: h5py>=2.5 in
/usr/local/lib/python3.10/site-packages (from cooler) (3.8.0)
Requirement already satisfied: pyyaml in
/usr/local/lib/python3.10/site-packages (from cooler) (6.0)
Requirement already satisfied: cytoolz in
/usr/local/lib/python3.10/site-packages (from cooler) (0.12.1)
Requirement already satisfied: python-dateutil>=2.8.2 in
```

```
/usr/local/lib/python3.10/site-packages (from pandas>1.0->cooler)
(2.8.2)
Requirement already satisfied: tzdata>=2022.1 in
/usr/local/lib/python3.10/site-packages (from pandas>1.0->cooler)
(2023.3)
Requirement already satisfied: pytz>=2020.1 in
/usr/local/lib/python3.10/site-packages (from pandas>1.0->cooler)
(2023.3)
Requirement already satisfied: toolz>=0.8.0 in
/usr/local/lib/python3.10/site-packages (from cytoolz->cooler)
(0.12.0)
Requirement already satisfied: dill>=0.3.6 in
/usr/local/lib/python3.10/site-packages (from multiprocess->cooler)
(0.3.6)
Requirement already satisfied: six in /usr/local/lib/python3.10/site-
packages (from pyfaidx->cooler) (1.16.0)
Requirement already satisfied: setuptools in
/usr/local/lib/python3.10/site-packages (from pyfaidx->cooler)
(65.6.3)
Looking in indexes: https://pypi.org/simple, https://us-
python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: hicrep in
/usr/local/lib/python3.10/site-packages (0.2.6)
Requirement already satisfied: cooler in
/usr/local/lib/python3.10/site-packages (from hicrep) (0.9.1)
Requirement already satisfied: h5py in /usr/local/lib/python3.10/site-
packages (from hicrep) (3.8.0)
Requirement already satisfied: pandas in
/usr/local/lib/python3.10/site-packages (from hicrep) (2.0.1)
Requirement already satisfied: numpy>=1.17.0 in
/usr/local/lib/python3.10/site-packages (from hicrep) (1.24.3)
Requirement already satisfied: Deprecated in
/usr/local/lib/python3.10/site-packages (from hicrep) (1.2.13)
Requirement already satisfied: scipy in
/usr/local/lib/python3.10/site-packages (from hicrep) (1.10.1)
Requirement already satisfied: multiprocess in
/usr/local/lib/python3.10/site-packages (from cooler->hicrep)
(0.70.14)
Requirement already satisfied: pyfaidx in
/usr/local/lib/python3.10/site-packages (from cooler->hicrep)
(0.7.2.1)
Requirement already satisfied: cytoolz in
/usr/local/lib/python3.10/site-packages (from cooler->hicrep) (0.12.1)
Requirement already satisfied: pyyaml in
/usr/local/lib/python3.10/site-packages (from cooler->hicrep) (6.0)
Requirement already satisfied: click>=7 in
/usr/local/lib/python3.10/site-packages (from cooler->hicrep) (8.1.3)
Requirement already satisfied: simplejson in
/usr/local/lib/python3.10/site-packages (from cooler->hicrep) (3.19.1)
Requirement already satisfied: asciitree in
```

```
/usr/local/lib/python3.10/site-packages (from cooler->hicrep) (0.3.3)
Requirement already satisfied: tzdata>=2022.1 in
/usr/local/lib/python3.10/site-packages (from pandas->hicrep) (2023.3)
Requirement already satisfied: python-dateutil>=2.8.2 in
/usr/local/lib/python3.10/site-packages (from pandas->hicrep) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in
/usr/local/lib/python3.10/site-packages (from pandas->hicrep) (2023.3)
Requirement already satisfied: wrapt<2,>=1.10 in
/usr/local/lib/python3.10/site-packages (from Deprecated->hicrep)
(1.15.0)
Requirement already satisfied: six>=1.5 in
/usr/local/lib/python3.10/site-packages (from python-dateutil>=2.8.2-
>pandas->hicrep) (1.16.0)
Requirement already satisfied: toolz>=0.8.0 in
/usr/local/lib/python3.10/site-packages (from cytoolz->cooler->hicrep)
(0.12.0)
Requirement already satisfied: dill>=0.3.6 in
/usr/local/lib/python3.10/site-packages (from multiprocess->cooler-
>hicrep) (0.3.6)
Requirement already satisfied: setuptools in
/usr/local/lib/python3.10/site-packages (from pyfaidx->cooler->hicrep)
(65.6.3)

WARNING: Running pip as the 'root' user can result in broken
permissions and conflicting behaviour with the system package manager.
It is recommended to use a virtual environment instead:
https://pip.pypa.io/warnings/venv
WARNING: Running pip as the 'root' user can result in broken
permissions and conflicting behaviour with the system package manager.
It is recommended to use a virtual environment instead:
https://pip.pypa.io/warnings/venv
```

```
import matplotlib.pyplot as plt

import cooler

import numpy as np

from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly
remount, call drive.mount("/content/drive", force_remount=True).
```

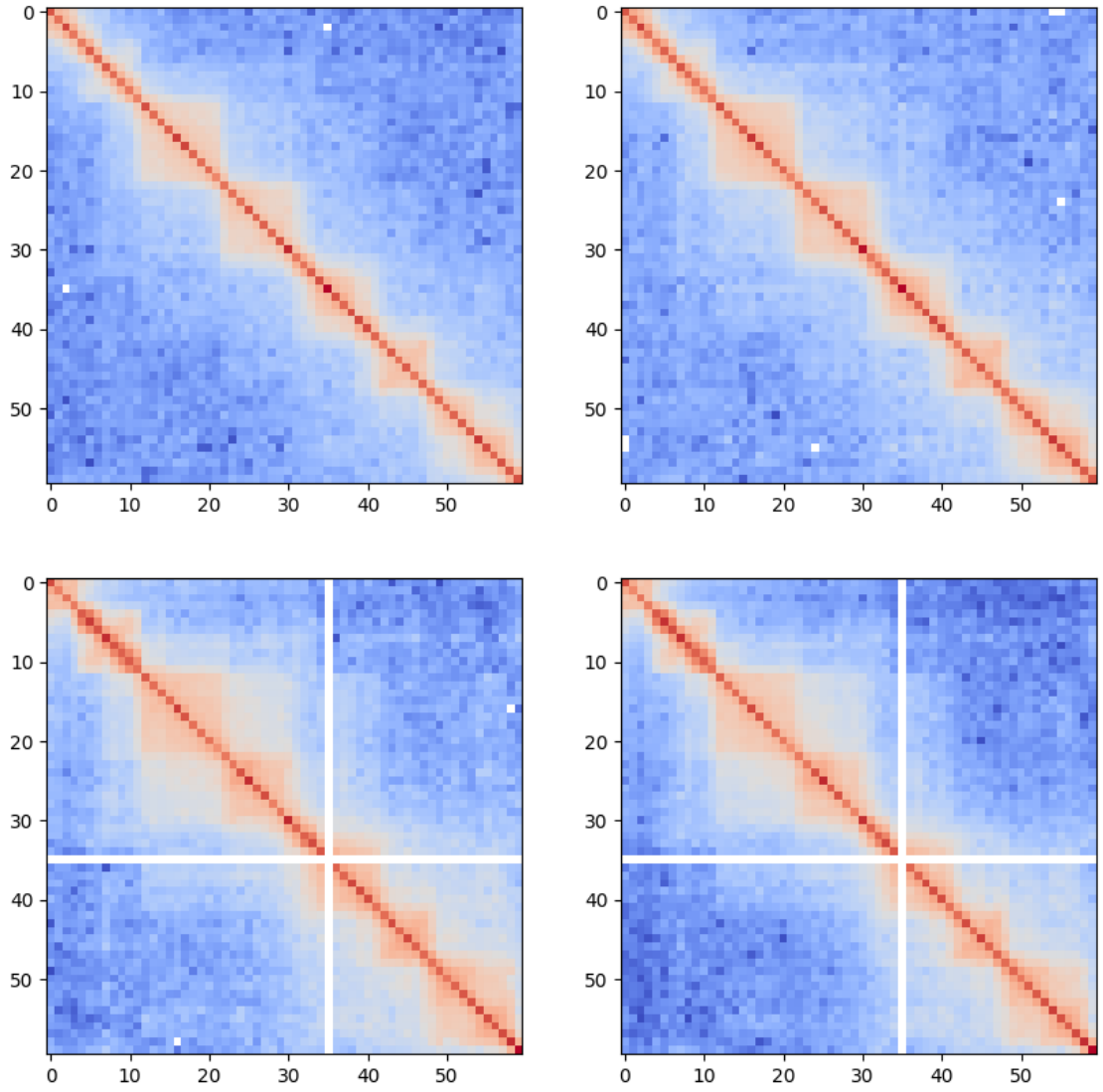
```
import seaborn as sns
import pandas as pd
import hicrep
from hicrep import hicrepSCC
from hicrep.utils import readMcool

mcool_hic1 ='/content/drive/MyDrive/HiC1.dm3.mapq_30.1000.mcool'
mcool_hic2 ='/content/drive/MyDrive/HiC2.dm3.mapq_30.1000.mcool'
```

```
mcool_hic3 ='/content/drive/MyDrive/HiC3.dm3.mapq_30.1000.mcool'
mcool_hic4 ='/content/drive/MyDrive/HiC4.dm3.mapq_30.1000.mcool'

resolution = 20000
clr1 = cooler.Cooler(f'{mcool_hic1}::resolutions/{resolution}')
clr2 = cooler.Cooler(f'{mcool_hic2}::resolutions/{resolution}')
clr3 = cooler.Cooler(f'{mcool_hic3}::resolutions/{resolution}')
clr4 = cooler.Cooler(f'{mcool_hic4}::resolutions/{resolution}')
```

Tasks for seminar:
1) get info and attributes of Hi-C matrix with cooler.info
2) open cooler object as balanced matrix for intrachromosomal contacts
3) open cooler as unbalanced matrix for interchromosomal contacts
4) get table with coordinates and contacts, are they raw or balanced?
5) get the table in command line with command *cooler dump*
6) look at the table with bins, which columns present there?
7) plot a piece of map (log)
8) scaling plot (in log - log coordinates)
9) replicates clusterization

```
m1=clr1.matrix(balance=True).fetch('chrX')
m2=clr2.matrix(balance=True).fetch('chrX')
m3=clr3.matrix(balance=True).fetch('chrX')
m4=clr4.matrix(balance=True).fetch('chrX')

pix1=clr1.pixels(join=True)[:]
pix2=clr2.pixels(join=True)[:]
pix3=clr3.pixels(join=True)[:]
pix4=clr4.pixels(join=True)[:]

bins1=clr1.bins()[:]
bins2=clr2.bins()[:]
bins3=clr3.bins()[:]
bins4=clr4.bins()[:]

fig, ax = plt.subplots(nrows=2, ncols=2, figsize=(10,10))
ax[0,0].imshow(np.log(m1[120:180,120:180]),cmap='coolwarm')
ax[0,1].imshow(np.log(m2[120:180,120:180]),cmap='coolwarm')
ax[1,0].imshow(np.log(m3[120:180,120:180]),cmap='coolwarm')
ax[1,1].imshow(np.log(m4[120:180,120:180]),cmap='coolwarm')
fig.show()
```

```
<ipython-input-35-b79e3378bdd3>:2: RuntimeWarning: divide by zero
encountered in log
  ax[0,0].imshow(np.log(m1[120:180,120:180]),cmap='coolwarm')
<ipython-input-35-b79e3378bdd3>:3: RuntimeWarning: divide by zero
encountered in log
  ax[0,1].imshow(np.log(m2[120:180,120:180]),cmap='coolwarm')
<ipython-input-35-b79e3378bdd3>:4: RuntimeWarning: divide by zero
encountered in log
  ax[1,0].imshow(np.log(m3[120:180,120:180]),cmap='coolwarm')
```

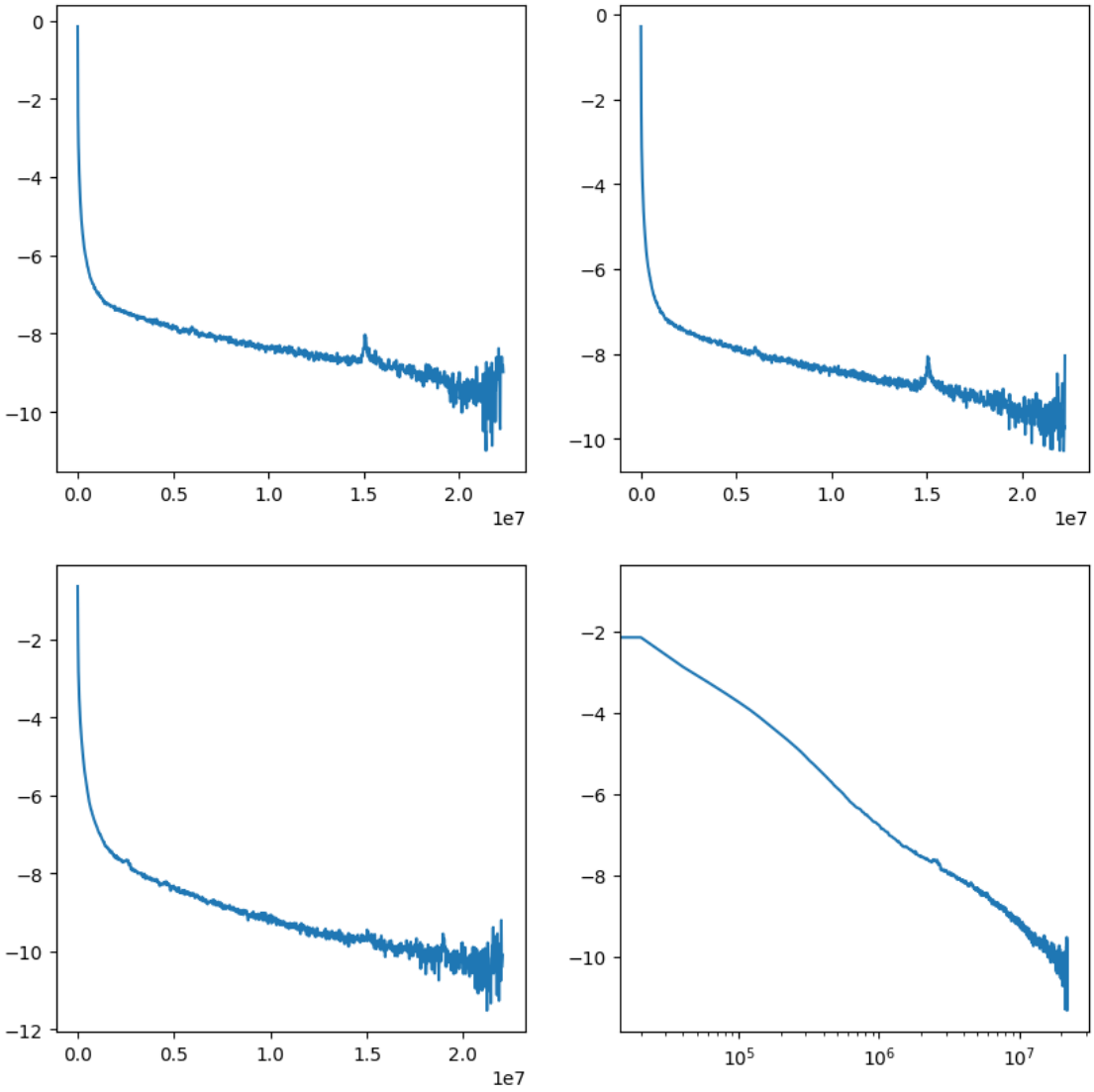



```
##!!! write comments for each row

# create array of our ms
ms = [m1,m2,m3,m4]

# calculate mean on diagonal values for each m and each coordinate
(np.nanmean because we want to get rid of nans)
z=[np.zeros(len(m)) for m in ms]
for j in range(len(ms)):
  for i in range(len(ms[j])):
    z[j][i]=np.nanmean(np.diagonal(ms[j],i))
```

```
<ipython-input-37-53f969f22853>:10: RuntimeWarning: Mean of empty
slice
  z[j][i]=np.nanmean(np.diagonal(ms[j],i))
```

```
##!!! write comments for each row
##!!! why do we paste 20000 below?

# 20000 is resolution value used above
# plot log of calculated values using logscale on x
fig, ax = plt.subplots(nrows=2, ncols=2, figsize=(10,10))
ax[0,0].plot(np.arange(len(m1))*20000,np.log(z[0]))
ax[0,1].plot(np.arange(len(m2))*20000,np.log(z[1]))
ax[1,0].plot(np.arange(len(m3))*20000,np.log(z[2]))
ax[1,1].plot(np.arange(len(m4))*20000,np.log(z[3]))
plt.xscale('log')
fig.show()
```

```
<ipython-input-41-e9aad48048b7>:7: RuntimeWarning: divide by zero
encountered in log
  ax[0,0].plot(np.arange(len(m1))*20000,np.log(z[0]))
<ipython-input-41-e9aad48048b7>:8: RuntimeWarning: divide by zero
encountered in log
  ax[0,1].plot(np.arange(len(m2))*20000,np.log(z[1]))
<ipython-input-41-e9aad48048b7>:9: RuntimeWarning: divide by zero
encountered in log
  ax[1,0].plot(np.arange(len(m3))*20000,np.log(z[2]))
<ipython-input-41-e9aad48048b7>:10: RuntimeWarning: divide by zero
encountered in log
  ax[1,1].plot(np.arange(len(m4))*20000,np.log(z[3]))
```



Replicates clusterization with stratum-adjusted correlation coefficient (scc)

We have replicates for 2 *drosophila* cell lines: Bg3 and Kc167.
Bg3 - nervous cell line (HiC1...., HiC2... files)
Kc167 - embryonic cell line (HiC3...., HiC4... files)
The aim is to conduct replicates clusterization, using scc and demonstrate that replicates of same cell line tend to be closer to each other comparing with different cell types.
Hicrep can only calculate scc for each chromosome separately.
This is why you obtain several values with hicrepSCC function (run code below). The number of the values is equal to the number of chromosomes in cool file. So lets imagine, you take 2 mcool (or cool) files and decide to calculate scc between Hi-C matrices storing in these files only for chromosome 'chr2L', than for 'chr2R', etc. You will obtain as many scc as there are chromosomes in the Hi-C map. This is exactly what hicrepSCC function gives as an output. But then, to get single general scc value for 2 Hi-C maps (2 replicates) you should calculate average value of scc across all the chromosomes.
So you have an scc (averaged across chrs) for each pair of samples (HiC1-HiC2,HiC1-

HiC3,HiC1-HiC4,HiC2-HiC3,HiC2-HiC4,HiC3-HiC4). Now you can use these values as the measure of similarity between each 2 samples and build a dendrogram.
To do this, you should construct symmetric matrix of similarity from calculated SCCs with ones on the diagonal. This matrix should be used as an input for 'linkage' function (see below)

```
### the code is for calculation of scc between
HiC1.dm3.mapq_30.1000.mcool and HiC2.dm3.mapq_30.1000.mcool
##!!! describe the next four parameter (as comments)

# aggregation parameter - there will be 20000=resolutions intervals
binSize = 20000
# we set maximum genomic distance that will be used in our calculation
dBPMax = 5000000
# True means we will subsample our data to reduce sizes
bDownSample = True
# smoothing parameter
h=0
fmcools=[mcool_hic1,mcool_hic2,mcool_hic3,mcool_hic4]
cools, binsizes = [],[]

#read to get values for each mcool
for i in range(4):
  cool, binSize = readMcool(fmcools[i], binSize)
  cools.append(cool)
  binsizes.append(binSize)

sccs = [[np.array([]) for i in range(4)] for j in range(4)]
#calculate scc between each pair
for i in range(4):
  for j in range(i+1, 4):
    sccs[i][j]=hicrepSCC(cools[i], cools[j], h, dBPMax, bDownSample)
```

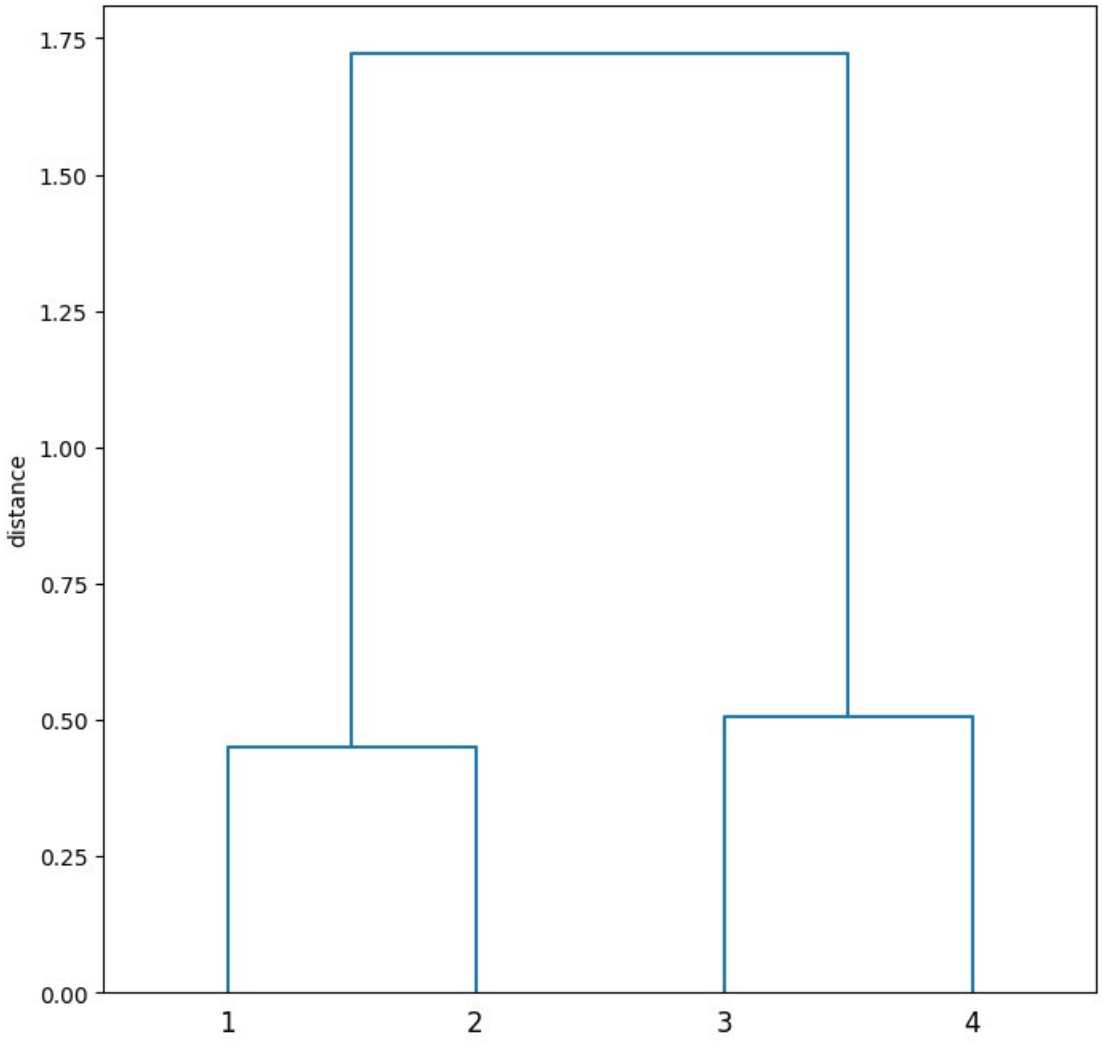
```
/usr/local/lib/python3.10/site-packages/hicrep/hicrep.py:91:
RuntimeWarning: invalid value encountered in double_scalars
  return rhoNan2Zero @ wsNan2Zero / wsNan2Zero.sum()
/usr/local/lib/python3.10/site-packages/hicrep/hicrep.py:91:
RuntimeWarning: invalid value encountered in double_scalars
  return rhoNan2Zero @ wsNan2Zero / wsNan2Zero.sum()
/usr/local/lib/python3.10/site-packages/hicrep/hicrep.py:91:
RuntimeWarning: invalid value encountered in double_scalars
  return rhoNan2Zero @ wsNan2Zero / wsNan2Zero.sum()
/usr/local/lib/python3.10/site-packages/hicrep/hicrep.py:91:
RuntimeWarning: invalid value encountered in double_scalars
  return rhoNan2Zero @ wsNan2Zero / wsNan2Zero.sum()
/usr/local/lib/python3.10/site-packages/hicrep/hicrep.py:91:
RuntimeWarning: invalid value encountered in double_scalars
  return rhoNan2Zero @ wsNan2Zero / wsNan2Zero.sum()
/usr/local/lib/python3.10/site-packages/hicrep/hicrep.py:91:
```

```
RuntimeWarning: invalid value encountered in double_scalars
  return rhoNan2Zero @ wsNan2Zero / wsNan2Zero.sum()
```

```
## A piece of code for the dendrogram generation

#now use this code to generate dendogram, naming labels from 1 to 4.
we use mean values in linkage.
from scipy.cluster.hierarchy import linkage, dendrogram

scc_matrix = [[np.nanmean(sccs[min(i,j)][max(i,j)]) if i!=j else 1 for
i in range(4)] for j in range(4)]
print(scc_matrix)
Z=linkage(scc_matrix, 'single', 'correlation')
plt.figure(figsize=(8,8))
plt.ylabel('distance')

dendrogram(Z, color_threshold=0,labels=['1','2','3','4'])
```

```
[[1, 0.6726052084071487, 0.41941163957022426, 0.3903869717218715],
[0.6726052084071487, 1, 0.4262329504690802, 0.3941956052714497],
[0.41941163957022426, 0.4262329504690802, 1, 0.6552600093802058],
[0.3903869717218715, 0.3941956052714497, 0.6552600093802058, 1]]
```

```
{'icoord': [[5.0, 5.0, 15.0, 15.0],
  [25.0, 25.0, 35.0, 35.0],
  [10.0, 10.0, 30.0, 30.0]],
 'dcoord': [[0.0, 0.45061824540402584, 0.45061824540402584, 0.0],
  [0.0, 0.5045534315552236, 0.5045534315552236, 0.0],
  [0.45061824540402584,
   1.722939031668695,
   1.722939031668695,
   0.5045534315552236]],
 'ivl': ['1', '2', '3', '4'],
 'leaves': [0, 1, 2, 3],
 'color_list': ['C0', 'C0', 'C0'],
 'leaves_color_list': ['C0', 'C0', 'C0', 'C0']}
```


1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
distance
1
2
3
4