(1)
```
('new_X,Y_train', 0)
SEQUENCE ONLY RESULTS
score, sequence only
('TRAINING ACCURACY', 0.84709985315712188)
('TEST ACCURACY', 0.82831988261188549)
DSSP RESULTS
score, ss and sequence
('TRAINING ACCURACY', 0.84177679882525702)
('TEST ACCURACY', 0.84959647835656638)
TM RESULTS
score, ss and sequence
('TRAINING ACCURACY', 0.84196035242290745)
('TEST ACCURACY', 0.84886280264123259

('new_X,Y_train', 1)
SEQUENCE ONLY RESULTS
score, sequence only
('TRAINING ACCURACY', 0.53749999999999998)
('TEST ACCURACY', 0.29493763756419661)
DSSP RESULTS
score, ss and sequence
('TRAINING ACCURACY', 0.55729794933655008)
('TEST ACCURACY', 0.60161408657373439)
TM RESULTS
score, ss and sequence
('TRAINING ACCURACY', 0.5625)
('TEST ACCURACY', 0.42993396918561994)
```

This question was probably the hardest of all of them. At first, my problem was due to trying to make a `numpy` array instead of using the native lists. The program would make my computer run out of memory at the first sequence with a length over 2000. I was then told by one of the three students who helped me with the Advanced Biological Background that the `split_dataset` function had to be edited. That helped correct a few things. As we can see from both versions, the Training Accuracy is higher with `TMHMM`, but the Test Accuracy is higher with `DSSP`.

(2) Let's say that one takes an index p such that at least half the odd `H` are on one side – let's call it Side 1 to not invoke Left and Right. Since at least half the odd are on one side, it means that less than half the even `H`'s are on the Side 1. Therefore, at least half the even `H`'s are on the other side – Side 2. Proving that this index p always exists.[1]

Secondly, this questions is very similar to the Balanced Sequence Cut Combinatorial

1. Sorin Istrail and Fumei Lam, "Combinatorial Algorithms for Protein Folding in Lattice Models: A Survey of Mathematical Results", 9 (January 2009), doi:10.4310/CIS.2009.v9.n4.a2, https://www.brown.edu/Research/Istrail_Lab/papers/pfoldingsurvey.pdf.

Problem. This proof summarizes what is presented in the paper:[2] "Suppose we are given a sequence of positive integers for which we consider two subsequences: the subsequence of integers occurring at odd positions and the subsequence of integers occurring at even positions. Suppose that the sums of these two subsequences are equal, say σ. Now suppose we cut the original sequence into two halves and consider the sum of the integers at odd positions on one half and the sum of the integers at even positions on the other half."

$$S = \alpha_1, \ldots, \alpha_k$$
$$|S| = \sum_{i=1}^{k} \alpha_i$$

A cut is denoted (u, v) where:

$$u = \sum_{i=1}^{p} \alpha_i$$
$$v = \sum_{i=p+1}^{k} \alpha_i$$

If $u = v$, the cut (u, v) is known as an equal cut. Let $L^0 = 0$, $R^k = 0$, and for $j = 1, \ldots, k$, $L^j = \sum_{i=1}^{k} \alpha_i$, $R^j = \sum_{i=j+1}^{k} \alpha_i$. A pair (L^t, R^t) is referred to as a peak of S_α if $L^t \geq \frac{|S_\alpha|}{2}$ and if $R^t \geq \frac{|S_\alpha|}{2}$

It can be trivially proven that every sequence S_α has at least one peak and at most two peaks. It has at least one peak because the sequence $L^0 < L^1 < \ldots < L^k = |S_\alpha|$ is strictly increasing and $L^k \geq \frac{|S_\alpha|}{2}$. This shows that that there is a minimum j such that $L^j \geq \frac{|S_\alpha|}{2}$. Let's assume 3 peaks:

$$(L^t, R^t), (L^{t+1}, R^{t+1}), (L^{t+2}, R^{t+2})$$
$$L^t, R^t \geq \frac{|S_\alpha|}{2}$$
$$L^t + R^t = |S_\alpha|$$
$$\therefore L^t = R^{t+1} = \frac{|S_\alpha|}{2} \text{ and } L^{t+1} = R^{i+2} = \frac{|S_\alpha|}{2}$$
$$L^t < L^{t+1} \qquad \text{(Contradiction)}$$

Essentially, one sequence is strictly increasing and the second sequence is strictly decreasing. The same reasoning can be used to prove that no more than two peaks can exist.

2. William E. Hart and Sorin C. Istrail, "Fast Protein Folding in the Hydrophobic–Hydrophilic Model within Three-Eighths of Optimal", PMID: 8697239, *Journal of Computational Biology* 3, no. 1 (1996): 53–96, doi:10.1089/cmb.1996.3.53, eprint: https://doi.org/10.1089/cmb.1996.3.53, https://doi.org/10.1089/cmb.1996.3.53.

Definition: A pair (L^t, R^t) is considered a strong peak when

$$L^t = \frac{|S_\alpha|+1}{2}, R^t = \frac{|S_\alpha|+1}{2}$$

It can also be trivially proven that a sequence S_α has either one strong peak or twin peaks. "If $|S_\alpha|$ is even and S_α has an equal cut, then S_α has twin peaks. If S_α is even but with no equal cut or if S_α is odd, then no L^i or R^i equals $\frac{|S_\alpha|}{2}$. Therefore, in this case, the existence of a peak implies the existence of its stronger version, namely, a strong peak.

Finally, it can be proven that every alternating sequence has a balanced cut. "Let S be an alternating sequence and S_α and S_β be the corresponding subsequences of S. If one subsequence, say S_α has a strong peak, then every cut at a position of S between the peak of S_α and a peak of S_β is a balanced cut. If both sequences have twin peaks, there exists a cut that separates a peak of S_α and a peak of S_β. The leftmost such cut is a balanced cut."

Furthermore using the notation found on the slides[3], we can prove the approximation. The resulting fold is at least $\frac{1}{2}\min\{\mathcal{E}(S), \mathcal{O}(S)\}$

$$\frac{\frac{1}{2}\min\{\mathcal{E}(S), \mathcal{O}(S)\}}{2\min\{\mathcal{E}(S), \mathcal{O}(S)\}} = \frac{1}{4} \quad \text{(Ratio)}$$

(3) I removed the header from the xvg file.

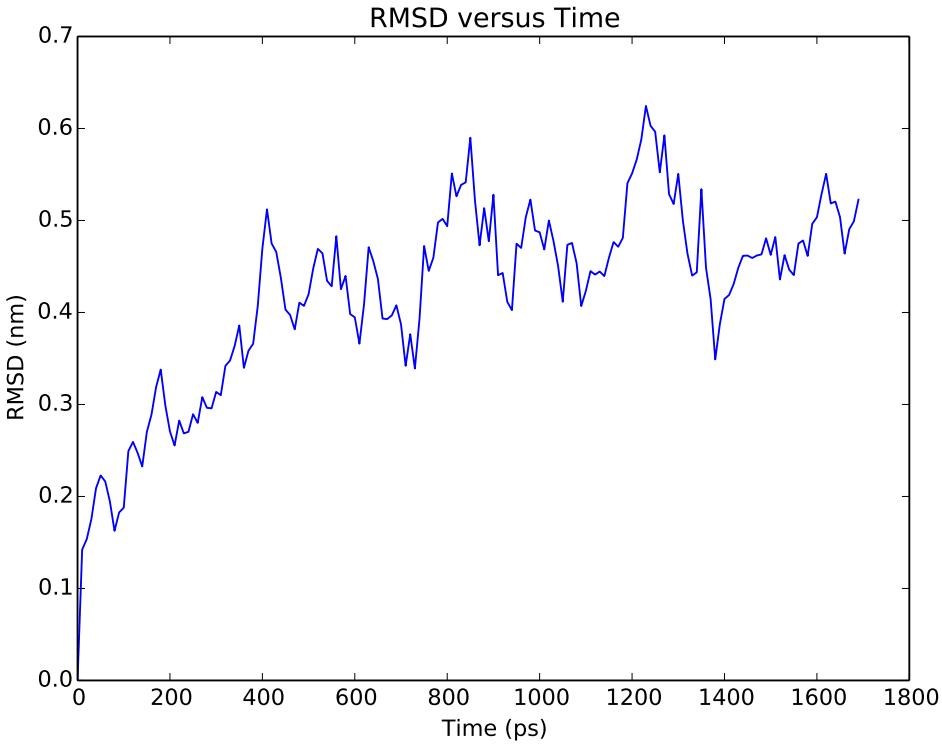
```
import numpy as np
import matplotlib.pyplot as plt

x, y = [], []

with open("molecule-bkbone-rmsd.xvg") as file:
  for line in file:
    cols = line.split()
    x.append(cols[0])
    y.append(cols[1])

plt.title("RMSD versus Time")
plt.xlabel('Time (ps)')
plt.ylabel('RMSD (nm)')
plt.plot(x, y)
plt.savefig("RMSD.pdf")
```

3. https://www.cs.mcgill.ca/~jeromew/teaching/564/W2019/COMP564_Lecture_17_W2019.pdf



(4b)
```
import numpy as np
import collections
import networkx as nx
import matplotlib.pyplot as plt
from scipy import sparse as sp

def createGraph(file):

  g = nx.Graph()

  for line in open(file, 'r' ):
    line = line.replace('\n', '')
    line = line.split()
    g.add_edge(line[0], line[1])

  gnodes = np.sort(g.nodes())
  gneighbours = collections.OrderedDict()

  for item in gnodes:
    gneighbours[item] = list(g.neighbors(item))

  return g, gneighbours, gnodes
```

```
def greedy(RList, combo, min_len, min_eigVec):
  subgraph = nx.Graph()
  for i in range(len(RList)):
    if(min_len < 0): break
    max_index = RList.index(max(RList))
    max_Rvalue = RList[max_index]
    RList[max_index] = min_eigVec - 1
    edge = combo[max_index]

    if(not subgraph.has_node(edge[0]) and
      not subgraph.has_node(edge[1])):
      print(edge[0], edge[1], max_Rvalue)
      subgraph.add_edge(*edge)
      min_len -= 1

def data(mFile, nFile, threshold):
  m, mNeigh, mSorted = createGraph(mFile)
  n, nNeigh, nSorted = createGraph(nFile)

  mn = len(m.nodes()) * len(n.nodes())

  A = sp.dok_matrix((mn,mn), dtype = float)

  for i_ind, i in enumerate(m.nodes()):
    for j_ind, j in enumerate(n.nodes()):
      for u_ind, u in enumerate(list(mNeigh.values())[i_ind]):
        for v_ind, v in enumerate(list(nNeigh.values())[j_ind]):
          N_u = len(list(mNeigh.values())[u_ind])
          N_v = len(list(nNeigh.values())[v_ind])
          ind_x = i_ind*len(n.nodes())+j_ind
          ind_y = u_ind*len(n.nodes())+v_ind
          A[ind_x, ind_y] = (1.0 / (N_u * N_v))

  R_lambda, R = sp.linalg.eigs(A, k=1)

  name = mFile.split('.')[0] + nFile.split('.')[0]
  print(name)
  print("Eigenvalue", R_lambda)

  RList = [abs(e.real) for e in R]
  RList = np.asarray(RList, dtype=float)
  RList = np.extract(RList > threshold, RList)
  RList = list(RList)

  combo = []
```

```
    for i in range(len(mSorted)):
      for j in range(len(nSorted)):
        combo.append((mSorted[i],nSorted[j]))
    min_eigVec = min(RList)
    min_len = min(len(m.nodes()), len(n.nodes()))

    greedy(RList, combo, min_len, min_eigVec)

if __name__ == '__main__':
    data("N1.txt","N2.txt", 0.0005)
    data("N1.txt","N3.txt", 0.0005)
    data("N2.txt","N3.txt", 0.0005)
```

(4c) N1N2

```
('Eigenvalue', array([ 1.+0.j]))
('CG7033', 'GTF2F1', 0.11656150732215094)
('CG4896', 'PTPRF', 0.11249540822951759)
('CG6937', 'PKD1', 0.11087558013570448)
('CG4849', 'YWHAB', 0.10700782734027274)
('Hsp68', 'PA', 0.07861124912424107)
('Fmr1', 'COPG', 0.077255882760030134)
('Hsc70-4', 'STUB1', 0.074776554045009921)
('Fib', 'LTA4H', 0.073487303113199412)
('CG6197', 'RPL18', 0.073189783667396882)
('Lam', 'RPL13', 0.07183441730318596)
('MTA1-like', 'BCAP31', 0.067768318210552833)
('MED7', 'HBEGF', 0.064462546590525791)
('MED26', 'RRAS', 0.063702219117919637)
('Det', 'ECHS1', 0.04743782274738699)
('Dcr-2', 'MST1R', 0.04512378261336792)
('Hrb87F', 'ARF1', 0.036594891833698448)
('CG6418', 'A2M', 0.035702333496291164)
('Ef1gamma', 'TERT', 0.035239525469487415)
('Hrb27C', 'FYN', 0.034809775158883893)
('Act42A', 'BCL2L1', 0.033884159105276271)
('Aats-glupro', 'HCK', 0.032231273295262909)
('Hlc', 'LRPAP1', 0.029818060012643203)
('CG5525', 'RPL36', 0.028462693648432198)
('DNApol-alpha73', 'RAP2A', 0.027636250743425433)
('Hel25E', 'UBE2Z', 0.027636250743425399)
('AGO2', 'RARRES3', 0.023041228191587961)
('CG32479', 'IKBKB', 0.014909030006321619)
('CG2807', 'RELA', 0.014181760249915677)
('MBD-R2', 'ACTG1', 0.014016471668914338)
('CG5641', 'ABL1', 0.01388424080411324)
```

```
('Cpsf100', 'TPM2', 0.013553663642110544)
('Hsc70-3', 'PJA1', 0.012892509318105157)
('CkIIalpha', 'TOLLIP', 0.012198297277899511)
('Caf1', 'STAU1', 0.01219829727789948)
('CG34422', 'HLA-A', 0.012198297277899475)
('CkIIbeta', 'STAT3', 0.010842930913688445)
('CG3605', 'HSPB1', 0.01084293091368844)
('CG16941', 'LGALS9', 0.01084293091368844)
('CG13900', 'MIF', 0.010314007454484138)
('CG1542', 'RPS6KA2', 0.01031400745448413)
('DNApol-alpha180', 'CCND1', 0.0095867376980781922)
('CG9667', 'CAV1', 0.009487564549477398)
('MED17', 'FOS', 0.0094875645494773911)
('Akt1', 'target', 0.0094875645494773894)
('Hsc70-1', 'ITGB2', 0.0094875645494773894)
('CoRest', 'NEDD9', 0.0094875645494773876)
('CG8801', 'RBPMS', 0.0094875645494773859)
('14-3-3zeta', 'SMARCC2', 0.0094875645494773841)
('Gem3', 'IFI35', 0.0094875645494773841)
('Cdc5', 'EDC4', 0.0094875645494773824)
('Ef1alpha100E', 'KLF4', 0.0094875645494773755)
('CG9143', 'PSMB3', 0.0090247565226736146)
('Cctgamma', 'SERF2', 0.0090247565226736111)
('MED1', 'TAP1', 0.0090247565226736094)
('Capr', 'BHLHE40', 0.0086280639282703791)
('Gp93', 'POLR2G', 0.0081321981852663372)
('CG4266', 'STAT6', 0.0081321981852663355)
('CG7911', 'LAMP2', 0.0081321981852663355)
('CG8142', 'NCAPH2', 0.0081321981852663337)
('Hrb98DE', 'SMARCA4', 0.0081321981852663337)
('Hsp83', 'RRAS2', 0.0081321981852663337)
('Cct5', 'PHB', 0.0077355055908630991)
('CG13097', 'LMNA', 0.0077355055908630982)
('CG7185', 'NP', 0.0075371592936614778)
('MED21', 'BAT3', 0.007537159293661477)
('Bap55', 'PPAP2B', 0.0067768318210552798)
('CG10754', 'PTMA', 0.0067768318210552798)
('CG8963', 'RNF126', 0.0067768318210552772)
('B4', 'CALM3', 0.0067768318210552755)
('Droj2', 'ADRB2', 0.0067768318210552746)
('Arf102F', 'CCL5', 0.0067107163886547423)
('Arp2', 'RNF10', 0.0064462546590525827)
('Gnf1', 'RPS14', 0.0062809660780512337)
('CG8636', 'YY1', 0.0059503889160485357)
('Hsc70-2', 'CEBPB', 0.0054214654568442268)
```

```
('Arf79F', 'VCP', 0.0054214654568442216)
('E(bx)', 'FAM46A', 0.0054214654568442216)
('MED24', 'PHB2', 0.0054214654568442216)
('CG10333', 'MAP1LC3B', 0.0054214654568442199)
('Iswi', 'PABPN1', 0.005421465456844219)
('14-3-3epsilon', 'SNF8', 0.0051570037272420707)
('CG11107', 'APH1A', 0.005157003727242069 8)
('Ef1alpha48D', 'PRKRA', 0.0051570037272420689)
('Dsor1', 'TSPAN4', 0.0051570037272420681)
('Mcm6', 'ATF4', 0.0040660990926331669)
('CG3689', 'ATN1', 0.0040660990926331643)
('Act5C', 'EZR', 0.0027107327284221112)
('CG4364', 'MYH9', 0.0027107327284221112)
```

```
N1N3
('Eigenvalue', array([ 1.+0.j]))
('Det', 'N194', 0.059496109283472709)
('CkIIalpha', 'N148', 0.057420663610793163)
('Droj2', 'N120', 0.050342861701399774)
('RpL18', 'N114', 0.040125283005132593)
('Parp', 'N51', 0.039433467780906249)
('DNApol-alpha180', 'N212', 0.037358022108226953)
('RpL17', 'N90', 0.037038722773968621)
('RpL23', 'N40', 0.036666206884000616)
('RpS3', 'N128', 0.034590761211321326)
('Pdk1', 'N209', 0.03336678042999764)
('RpS11', 'N207', 0.032515315538642009)
('DNApol-alpha73', 'N14', 0.031610634091576689)
('RpL26', 'N199', 0.031025251978769607)
('RpS13', 'N134', 0.027512959301927865)
('Mcm6', 'N235', 0.024213532847924889)
('Mcm7', 'N162', 0.020488373948244133)
('RfC38', 'N110', 0.018679011054113487)
('Nup98-96', 'target', 0.017987195829887032)
('Act5C', 'source', 0.017295380605660614)
('Ref1', 'N87', 0.017242164049950902)
('Nurf-38', 'N27', 0.015219934932981382)
('Pvr', 'N158', 0.015219934932981362)
('Akt1', 'N26', 0.014634552820174403)
('CkIIbeta', 'N156', 0.014528119708754931)
('MTA1-like', 'N208', 0.014049170707367395)
('Act42A', 'N168', 0.011760858811849234)
('CoRest', 'N103', 0.0078228336893295738)
('CG7911', 'N135', 0.0070245853536837028)
('MED21', 'N96', 0.0069181522422642463)
```

```
('RpL15', 'N167', 0.0063859866851669949)
('CG8142', 'N161', 0.0062263370180378284)
('Hrb98DE', 'N83', 0.0062263370180378223)
('MBD-R2', 'N147', 0.0062263370180378197)
('MED24', 'N253', 0.0058538211280697522)
('MED1', 'N137', 0.0055345217938114035)
('CG6937', 'N206', 0.0055345217938114026)
('CG6197', 'N34', 0.0055345217938114009)
('CG8636', 'N16', 0.005534521793811 4)
('Dcr-2', 'N202', 0.0055345217938113983)
('CG10333', 'N183', 0.0048427065695849777)
('RpL10', 'N180', 0.0048427065695849777)
('RpL4', 'N92', 0.0048427065695849777)
('Aats-glupro', 'N177', 0.0048427065695849768)
('Hrb27C', 'N149', 0.0048427065695849768)
('Ns1', 'N170', 0.0048427065695849751)
('MED17', 'N127', 0.0048427065695849742)
('Gem3', 'N37', 0.0048427065695849734)
('Hsp83', 'N254', 0.0048427065695849734)
('Pi3K92E', 'N30', 0.0048427065695849734)
('Hsc70-1', 'N155', 0.0047894900138752529)
('CG6418', 'N187', 0.0046830569024557947)
('Pi3K21B', 'N244', 0.004470190679616899)
('CG7033', 'N105', 0.0042573244567779963)
('RpL18A', 'N91', 0.004150891345358552)
('CG5525', 'N77', 0.0041508913453585502)
('RpL6', 'N5', 0.0041508913453585502)
('CG9667', 'N222', 0.0041508913453585494)
('Hrb87F', 'N139', 0.0041508913453585494)
('Hsc70-4', 'N43', 0.0041508913453585494)
('Ef1gamma', 'N104', 0.0041508913453585485)
('Rm62', 'N190', 0.0041508913453585485)
('Ef1alpha100E', 'N124', 0.0041508913453585476)
('Fmr1', 'N47', 0.0041508913453585476)
('Hsp68', 'N33', 0.0041508913453585468)
('Prp19', 'N224', 0.0041508913453585468)
('Gnf1', 'N19', 0.0040976747896488246)
('RpL10Ab', 'N101', 0.0040976747896488246)
('Iswi', 'N176', 0.0040976747896488211)
('E(bx)', 'N107', 0.0038315920111002006)
('Ef1alpha48D', 'N80', 0.0038315920111001993)
('Pitslre', 'N160', 0.0037251588996807506)
('RpL5', 'N195', 0.0037251588996807506)
('Lam', 'N166', 0.0035122926768418523)
('RpL19', 'N243', 0.0035122926768418523)
```

```
('Caf1', 'N144', 0.0035122926768418519)
('RpL7', 'N201', 0.0035122926768418514)
('Prp8', 'N146', 0.003512292676841851)
('CG5641', 'N229', 0.0035122926768418501)
('MED26', 'N85', 0.0034590761211321271)
('Gp93', 'N232', 0.0034590761211321266)
('CG34422', 'N49', 0.0034590761211321253)
('CG4364', 'N142', 0.0034590761211321249)
('Cpsf100', 'N221', 0.0034590761211321245)
('Hsc70-3', 'N53', 0.0034590761211321245)
('Hsc70-2', 'N63', 0.0034590761211321227)
('Pp2A-29B', 'N226', 0.0034590761211321227)
('Ns2', 'N1', 0.0033526430097126753)
('CG10754', 'N111', 0.0033526430097126736)
('CG2807', 'N100', 0.0031929933425834992)
('CG8801', 'N10', 0.0029801271197445979)
('Hel25E', 'N154', 0.0029269105640348752)
('CG7185', 'N118', 0.0029269105640348744)
('MED7', 'N237', 0.0029269105640348744)
('Mes2', 'N172', 0.0029269105640348744)
('RpL3', 'N192', 0.0029269105640348735)
('CG3605', 'N200', 0.0029269105640348731)
('Dsor1', 'N97', 0.00287369400832515)
('Fib', 'N69', 0.0027672608969057013)
('CG13900', 'N179', 0.0027672608969057)
('CG32479', 'N130', 0.0027672608969057)
('CG4849', 'N153', 0.0027672608969057)
('CG4896', 'N82', 0.0027672608969057)
('Nc73EF', 'N9', 0.0027672608969057)
('RpL8', 'N46', 0.0027672608969056996)
('CG3689', 'N171', 0.0025543946740668008)
('Nup160', 'N98', 0.0025543946740668008)
('RfC4', 'N45', 0.0025543946740668004)
('Nnp-1', 'N23', 0.0023415284512279025)
('RpL21', 'N250', 0.0023415284512279025)
('CG11107', 'N233', 0.0023415284512279008)
('CG16941', 'N94', 0.0021286622283890016)
('AGO2', 'N3', 0.0021286622283890008)
('Hlc', 'N62', 0.0021286622283890008)
('Rheb', 'N136', 0.0021286622283890008)
('RpL14', 'N239', 0.0021286622283890008)
('RpLP0', 'N113', 0.0021286622283890008)
('RpL13', 'N178', 0.0021286622283890003)
('Arf102F', 'N214', 0.0021286622283889999)
('14-3-3epsilon', 'N138', 0.0020754456726792755)
```

```
('RpS3A', 'N12', 0.0020754456726792755)
('Nopp140', 'N211', 0.0020754456726792742)
('CG8963', 'N11', 0.0020754456726792738)
('Pten', 'N117', 0.0019157960055501016)
('RpL13A', 'N213', 0.0019157960055501007)
('Cctgamma', 'N246', 0.0019157960055501005)
('RpL7A', 'N7', 0.0019157960055501001)
('CG4266', 'N99', 0.0018625794498403746)
('Nup107', 'N248', 0.0017561463384209261)
('Mi-2', 'N133', 0.0017561463384209259)
('NAT1', 'N102', 0.0017561463384209255)
('CG9143', 'N227', 0.0017561463384209253)
('14-3-3zeta', 'N175', 0.0015964966712917509)
('Cdc5', 'N141', 0.0015964966712917509)
('CG13097', 'N169', 0.0015964966712917507)
('RpS4', 'N173', 0.00159649667129175)
('CG1542', 'N109', 0.0014900635598723013)
('B4', 'N193', 0.0013836304484528511)
('Arf79F', 'N205', 0.0013836304484528509)
('Arp2', 'N41', 0.0013836304484528509)
('Bap55', 'N251', 0.0013836304484528509)
('RpL35A', 'N245', 0.0013836304484528509)
('RpL30', 'N78', 0.0013836304484528507)
('Capr', 'N122', 0.0013836304484528504)
('Cct5', 'N6', 0.0011707642256139513)
('RpS6', 'N76', 0.0011707642256139506)
('NHP2', 'N95', 0.0011175476699042254)
```

N2N3

```
('Eigenvalue', array([ 1.+0.j]))
('PJA1', 'N30', 0.037118330889366447)
('ACOT9', 'N41', 0.035307680602080282)
('CLU', 'N21', 0.034402355458437173)
('PKD1', 'N189', 0.031407818444848537)
('ACTB', 'N2', 0.029875729740221721)
('CNNM3', 'N137', 0.029109685387908361)
('HSF1', 'N132', 0.027159754309292534)
('SHC1', 'N107', 0.026254429165649452)
('FGFR1', 'N225', 0.02444377887836325)
('SFN', 'N83', 0.024234857691368728)
('RIN3', 'N105', 0.020892118699455793)
('FN1', 'N152', 0.02068319751246123)
('M2', 'N53', 0.019917153160147864)
('EFEMP2', 'N101', 0.019917153160147843)
('RPA2', 'N28', 0.019011828016504782)
```

```
('EWSR1', 'N5', 0.019011828016504757)
('HRAS', 'N94', 0.018802906829510208)
('EEF1G', 'N78', 0.0183850644555211)
('SNF8', 'N165', 0.01810650287286171)
('RAC1', 'N162', 0.018106502872861689)
('BAG1', 'N19', 0.018106502872861686)
('RNF126', 'N122', 0.018106502872861682)
('MAGED1', 'N210', 0.016852975750894339)
('RNF10', 'N99', 0.016713694959564627)
('COBRA1', 'N183', 0.01671369495956462)
('RPL10', 'N186', 0.016086931398580948)
('EZR', 'N207', 0.016086931398580945)
('PB2', 'N219', 0.015390527441932436)
('BAIAP2', 'N116', 0.01532088704626757)
('PECAM1', 'N150', 0.014485202298289351)
('DNM2', 'N62', 0.014485202298289342)
('RHOA', 'N98', 0.013928079132970535)
('COL4A2', 'N111', 0.013928079132970521)
('CLCN7', 'N206', 0.013788798341640818)
('JUN', 'N230', 0.013579877154646262)
('KAT5', 'N133', 0.013579877154646258)
('HSPA1B', 'target', 0.013579877154646253)
('ATF3', 'N50', 0.013370955967651709)
('ROCK2', 'N231', 0.0133709559676517)
('PHC2', 'N234', 0.013370955967651699)
('ERBB2', 'N253', 0.013370955967651692)
('PDCD6IP', 'N146', 0.0130227539893274 31)
('EDC4', 'N192', 0.012535271219673474)
('SNRPA', 'N1', 0.012535271219673469)
('DTX2', 'N22', 0.012256709637014064)
('ATF4', 'N235', 0.012256709637014061)
('CD68', 'N212', 0.011769226867360094)
('NEDD9', 'N179', 0.011769226867360089)
('GMEB2', 'N97', 0.011769226867360087)
('PXN', 'N39', 0.011769226867360087)
('AKAP8L', 'N194', 0.011769226867360085)
('HSPB1', 'N27', 0.011490665284700686)
('JUNB', 'N158', 0.011490665284700679)
('RALGDS', 'N103', 0.011142463306376419)
('BCL3', 'N120', 0.01086390172371701)
('LRPAP1', 'N255', 0.010863901723717007)
('CD93', 'N154', 0.01086390172371 7005)
('RARA', 'N8', 0.010028216975738776)
('CDK4', 'N204', 0.010028216975738774)
('CDC25B', 'N81', 0.010028216975738772)
```

```
('DVL1', 'N244', 0.0099585765800739266)
('RXRA', 'N113', 0.0099585765800739266)
('QARS', 'N197', 0.0099585765800739249)
('JUND', 'N182', 0.0099585765800739214)
('S100A4', 'N68', 0.0099585765800739197)
('CD81', 'N14', 0.009958576580073918)
('GNA11', 'N254', 0.009958576580073918)
('NP', 'N104', 0.0099585765800739162)
('CIB1', 'N88', 0.0097496553930793602)
('JUP', 'N95', 0.0094014534147551093)
('EDN1', 'N142', 0.0094014534147551024)
('NFKBIA', 'N80', 0.0091925322277605515)
('CDC42', 'N35', 0.0091925322277605446)
('RRAS2', 'N9', 0.0091925322277605446)
('RARRES3', 'N10', 0.0091925322277605429)
('CDK9', 'N159', 0.0091925322277605411)
('AATF', 'N90', 0.0090532514364308429)
('MX1', 'N56', 0.0090532514364308411)
('ADRB2', 'N252', 0.0090532514364308394)
('BBS1', 'N130', 0.0090532514364308394)
('HA', 'N190', 0.0090532514364308377)
('ECHS1', 'N171', 0.0084264878754471677)
('GPS2', 'N193', 0.0083568474797823152)
('LTA4H', 'N0', 0.0083568474797823117)
('NFIL3', 'N112', 0.0081479262927877608)
('AKT1', 'N128', 0.0081479262927877591)
('NUDC', 'N33', 0.0081479262927877591)
('EDNRB', 'N160', 0.0081479262927877556)
('BCL2L1', 'N117', 0.0076604435231337926)
('ARHGDIA', 'N74', 0.0076604435231337909)
('SAT1', 'N139', 0.0076604435231337909)
('MYD88', 'N213', 0.0076604435231337892)
('GRB2', 'N147', 0.0076604435231337883)
('ABL1', 'N248', 0.0076604435231337857)
('AIMP2', 'N18', 0.0076604435231337831)
('PFKL', 'N123', 0.0075211627318040805)
('ICT1', 'N63', 0.0075211627318040736)
('SF1', 'N166', 0.0072426011491446736)
('COPS6', 'N151', 0.007242601149144671)
('RPL28', 'N16', 0.0072426011491446701)
('SDC4', 'N25', 0.0072426011491446701)
('HLA-A', 'N29', 0.0072426011491446693)
('CTBP2', 'N155', 0.0072426011491446658)
('P4HB', 'N191', 0.006894399170820408)
('POLR2E', 'N240', 0.0068943991708204071)
```

```
('SREBF1', 'N52', 0.0066854779838258528)
('HGS', 'N232', 0.006685477983825849 3)
('CALR', 'N148', 0.0063372760055015916)
('HEATR2', 'N38', 0.0063372760055015907)
('ILK', 'N20', 0.0063372760055015907)
('IRF1', 'N180', 0.0063372760055015898)
('BCAP31', 'N13', 0.0063372760055015881)
('CBR1', 'N72', 0.0063372760055015864)
('GATA2', 'N187', 0.0063372760055015864)
('GRN', 'N209', 0.0063372760055015864)
('RPLP0', 'N149', 0.0063372760055015838)
('IFI27', 'N196', 0.0062676356098367372)
('BATF', 'N92', 0.0062676356098367355)
('MT2A', 'N185', 0.0062676356098367346)
('HBEGF', 'N124', 0.0062676356098367312)
('NS1', 'N34', 0.006128354818507032)
('CAPN1', 'N135', 0.0061283548185070312)
('NR2F2', 'N100', 0.0061283548185070303)
('SEC61B', 'N177', 0.0061283548185070286)
('STAT3', 'N237', 0.0061283548185070286)
('CCND1', 'N71', 0.0058497932358476216)
('FAU', 'N256', 0.0058497932358476216)
('GTF2F1', 'N172', 0.0058497932358476208)
('FYN', 'N17', 0.005849793235847619)
('PSMB1', 'N60', 0.005849793235847619)
('RAP2A', 'N11', 0.0058497932358476164)
('CHMP6', 'N169', 0.0056408720488530656)
('POLR2C', 'source', 0.0056408720488530647)
('EGLN2', 'N161', 0.0055712316531882104)
('HLA-B', 'N134', 0.0055712316531882078)
('ARF1', 'N87', 0.0054319508618585043)
('BCL6', 'N12', 0.0054319508618585043)
('CCL5', 'N224', 0.0053623104661936517)
('GLB1', 'N108', 0.0053623104661936517)
('IRAK1', 'N121', 0.00536231046619365)
('PSMB3', 'N24', 0.00536231046619365)
('IFI35', 'N241', 0.0053623104661936491)
('CCT8', 'N250', 0.005362310466193647 4)
('NCAPH2', 'N86', 0.0050141084878693887)
('GLUL', 'N79', 0.0050141084878693879)
('EGFR', 'N245', 0.005014108487869387)
('MAPK3', 'N47', 0.0045962661138802749)
('MCC', 'N202', 0.004596266113880274)
('NCOR2', 'N106', 0.0045962661138802723)
('PDPK1', 'N156', 0.0045962661138802714)
```

```
('HCK', 'N110', 0.0045962661138802706)
('KDR', 'N129', 0.0045962661138802697)
('C12orf44', 'N236', 0.004526625718215424)
('ELF3', 'N153', 0.004526625718215424)
('HSPA1A', 'N131', 0.004526625718215424)
('PPAP2B', 'N239', 0.004526625718215424)
('AXIN1', 'N223', 0.0045266257182154232)
('BRD1', 'N246', 0.0045266257182154232)
('SRC', 'N89', 0.0045266257182154223)
('C20orf30', 'N226', 0.0045266257182154214)
('CASC3', 'N138', 0.0045266257182154214)
('CALM1', 'N216', 0.0045266257182154206)
('HMOX1', 'N249', 0.0045266257182154197)
('FBLN1', 'N173', 0.0043873449268857162)
('IRF9', 'N109', 0.0043873449268857145)
('PDIA3', 'N136', 0.0043873449268857145)
('AQP1', 'N7', 0.0041784237398911602)
('EDNRA', 'N44', 0.0041784237398911584)
('PABPN1', 'N93', 0.0041784237398911584)
('PSMB9', 'N40', 0.0041784237398911584)
('PTMA', 'N164', 0.0041784237398911567)
('RPS2', 'N6', 0.0041784237398911558)
('EIF1B', 'N127', 0.0041784237398911541)
('NA', 'N141', 0.0041784237398911541)
('DAZAP2', 'N102', 0.0038998621572317 45)
('SF3B4', 'N115', 0.0038998621572317 45)
('CREB3', 'N75', 0.0038302217615668955)
('HNRNPD', 'N170', 0.0038302217615668895)
('MYO1F', 'N175', 0.0038302217615668895)
('B2M', 'N145', 0.0038302217615668946)
('EEF1D', 'N73', 0.0037605813659020424)
('MYH9', 'N176', 0.0037605813659020424)
('CD2BP2', 'N4', 0.0036213005745723398)
('SLC25A1', 'N242', 0.0036213005745723381)
('LTBP4', 'N157', 0.0036213005745723372)
('CD9', 'N163', 0.0036213005745723368)
('MCF2L', 'N222', 0.0036213005745723368)
('SMARCC2', 'N174', 0.0036213005745723368)
('ATP1A1', 'N233', 0.0036213005745723359)
('DDX17', 'N119', 0.0036213005745723359)
('SQSTM1', 'N144', 0.0036213005745723355)
('CAV1', 'N3', 0.0034820197832426338)
('ACTG1', 'N65', 0.0034820197832426325)
('PSMD4', 'N198', 0.0034820197832426312)
('RPS6KA2', 'N118', 0.0034820197832426312)
```

```
('CALM2', 'N48', 0.0034820197832426303)
('FASN', 'N84', 0.0033427389919129286)
('PRPSAP1', 'N70', 0.0033427389919129273)
('SMAD1', 'N205', 0.0033427389919129273)
('NDUFB11', 'N82', 0.0031338178049183682)
('PLAUR', 'N140', 0.0031338178049183682)
('PPP1CA', 'N168', 0.0031338178049183682)
('MIF', 'N188', 0.0031338178049183677)
('NDUFS3', 'N126', 0.0031338178049183677)
('MCL1', 'N218', 0.0031338178049183669)
('PALM2-AKAP2', 'N200', 0.0031338178049183669)
('DDA1', 'N125', 0.003064177409253516)
('FASTK', 'N32', 0.0030641774092535156)
('MAP7D1', 'N57', 0.0030641774092535156)
('CHD3', 'N215', 0.0030641774092535147)
('GNAI2', 'N43', 0.0030641774092535147)
('BAT3', 'N96', 0.0029248966179238104)
('A2M', 'N217', 0.0027856158265941065)
('CDH5', 'N85', 0.0027856158265941065)
('RPS17', 'N58', 0.0027856158265941065)
('SETDB1', 'N247', 0.0027856158265941065)
('KEAP1', 'N195', 0.0027856158265941061)
('MALL', 'N228', 0.0027856158265941052)
('GNB2L1', 'N15', 0.0027856158265941048)
('SPG7', 'N229', 0.0027856158265941048)
('NS2', 'N54', 0.0027159754309292522)
('STAT5B', 'N26', 0.0027159754309292509)
('PRKAB1', 'N59', 0.0025070542439346965)
('RAB4A', 'N143', 0.0025070542439346961)
('PHB', 'N77', 0.0025070542439346957)
('RPL36', 'N201', 0.0025070542439346957)
('RPS13', 'N221', 0.0025070542439346952)
('IFI6', 'N220', 0.0025070542439346948)
('RPS14', 'N66', 0.0025070542439346939)
('PB1', 'N199', 0.0022981330569401383)
('CDKN1A', 'N91', 0.002298133056940137)
('ISG15', 'N114', 0.002298133056940137)
('ATN1', 'N251', 0.0022981330569401366)
('STAT6', 'N167', 0.0022981330569401361)
('AKAP2', 'N178', 0.0022284926612752844)
('CEBPB', 'N227', 0.0020892118699455792)
('CSNK2B', 'N64', 0.0020892118699455792)
('MST1R', 'N203', 0.0020892118699455792)
('SERF2', 'N36', 0.0020892118699455792)
('APH1A', 'N31', 0.0020892118699455784)
```

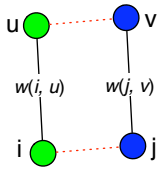
```
('PRKCZ', 'N76', 0.0020892118699455766)
('DERL1', 'N23', 0.0018802906829510221)
('FOS', 'N211', 0.0018802906829510219)
('POLR2G', 'N45', 0.0018802906829510208)
('LGALS9', 'N61', 0.0018802906829510203)
('POLR2A', 'N67', 0.0018802906829510203)
('RPL18', 'N184', 0.0018802906829510197)
('EIF4ENIF1', 'N243', 0.0018106502872861695)
('CDC42EP4', 'N51', 0.001810650287286168)
('RNF5', 'N69', 0.0018106502872861677)
('LMO2', 'N46', 0.0016713694959564647)
('RRAS', 'N181', 0.0016713694959564643)
('PARP12', 'N55', 0.001671369495956463)
('CD4', 'N208', 0.0016713694959564619)
('MUC1', 'N37', 0.0015320887046267589)
('ITGAM', 'N214', 0.001532088704626758)
('PRKRA', 'N238', 0.0014624483089619052)
('FADD', 'N42', 0.0013928079132970535)
('COPG', 'N49', 0.0012535271219673472)
```

(4d) As one can see from the Carnegie Mellon Lecture Slides on IsoRank[4], it is shown that the weighted case only needs an adjustment of the equation R_{ij}.

The Weighted Cases

Unweighted case:

$$R_{ij} := \sum_{u \in N(i)} \sum_{v \in N(j)} \frac{1}{|N(u)||N(v)|} R_{uv}$$



Weighted case:

$$R_{ij} := \sum_{u \in N(i)} \sum_{v \in N(j)} \frac{w(i,u)w(j,v)}{W(u)W(v)} R_{uv}$$

where

$$W(u) = \sum_{x \in N(u)} w(x,u) \qquad \leftarrow \text{“weighted degree”}$$

4. https://www.cs.cmu.edu/~ckingsf/bioinfo-lectures/isorank.pdf