A Machine Learning Approach for Prediction of Protein Structure and Function

Tom Wilson

HCI 5903 Capstone

2019.04.15

Objectives

Become familiar with traditional methods of protein structure and function determination

Recognize the utility of protein characterization for potential medical therapies

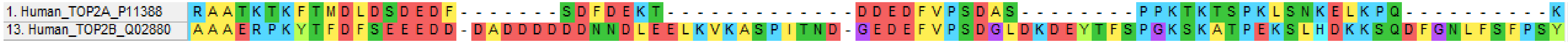
Understand the computational challenge of brute force protein structural prediction

Understand the role of machine learning in protein structure and function prediction

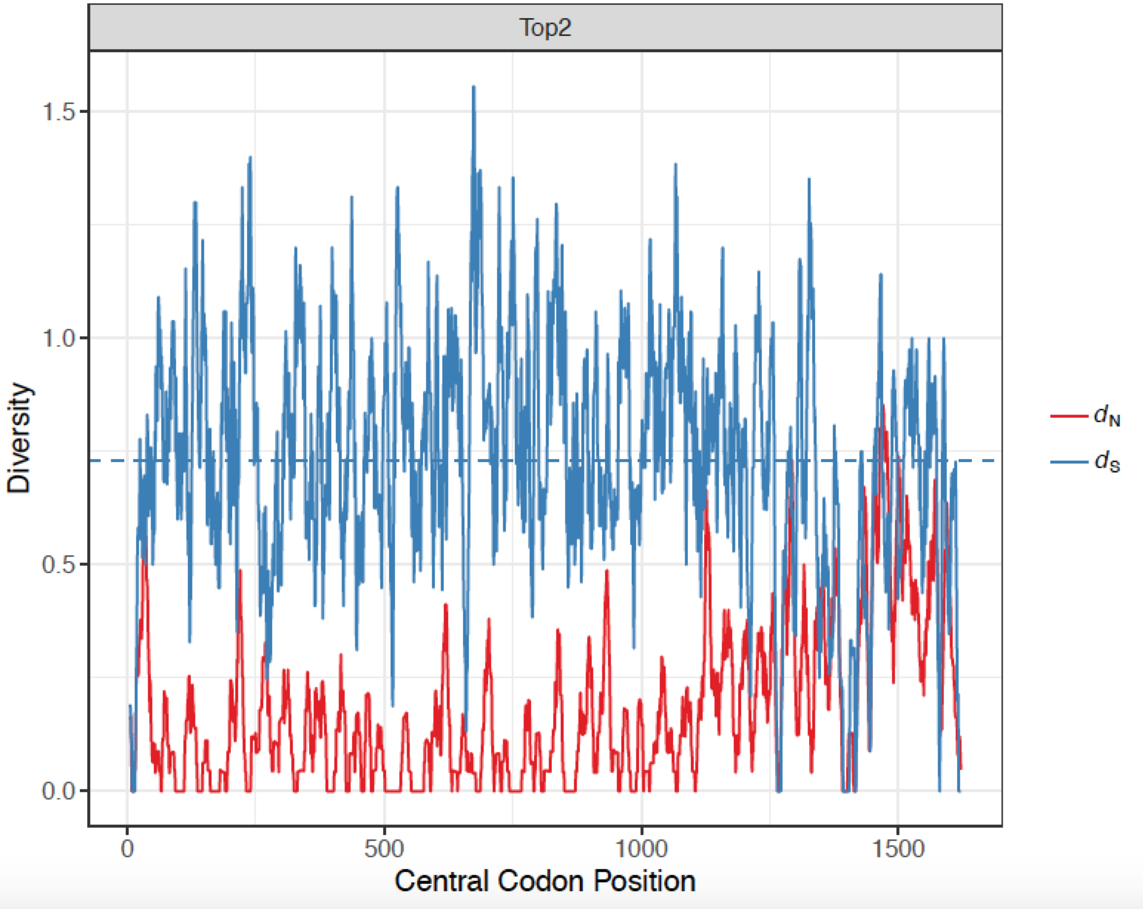
Project Inspiration

- Deweese Laboratory
- Topoisomerase II isoform comparison
- C-terminus is not conserved between the two isoforms[1]
- C-terminus has been resistant to structural determination[1]
- Structure and function determination of the C-terminus for each isoform may allow targeted drug therapies with reduced adverse reactions
  - Secondary leukemias

Differences in C-termini of Topoisomerase II[1]



```
1. Human_TOP2A_P11388    RAATKTKFTMDLDSDEDF-------SDFDEKT-------------DDEDFVPSDAS--------PPKTKTSPKLSNKELKPQ----------K
13. Human_TOP2B_Q02880   AAAERPKYTFDFSEEEDD-DADDDDDDNNDLEELKVKASPITND-GEDEFVPSDGLDKDEYTFSPGKSKATPEKSLHDKKSQDFGNLFSFPSY
```



Traditional Methods of Protein Characterization[2,3]

- X-ray crystallography
- NMR Spectroscopy
- Activity assays
- Agonists vs antagonists

Computational Analysis[4]

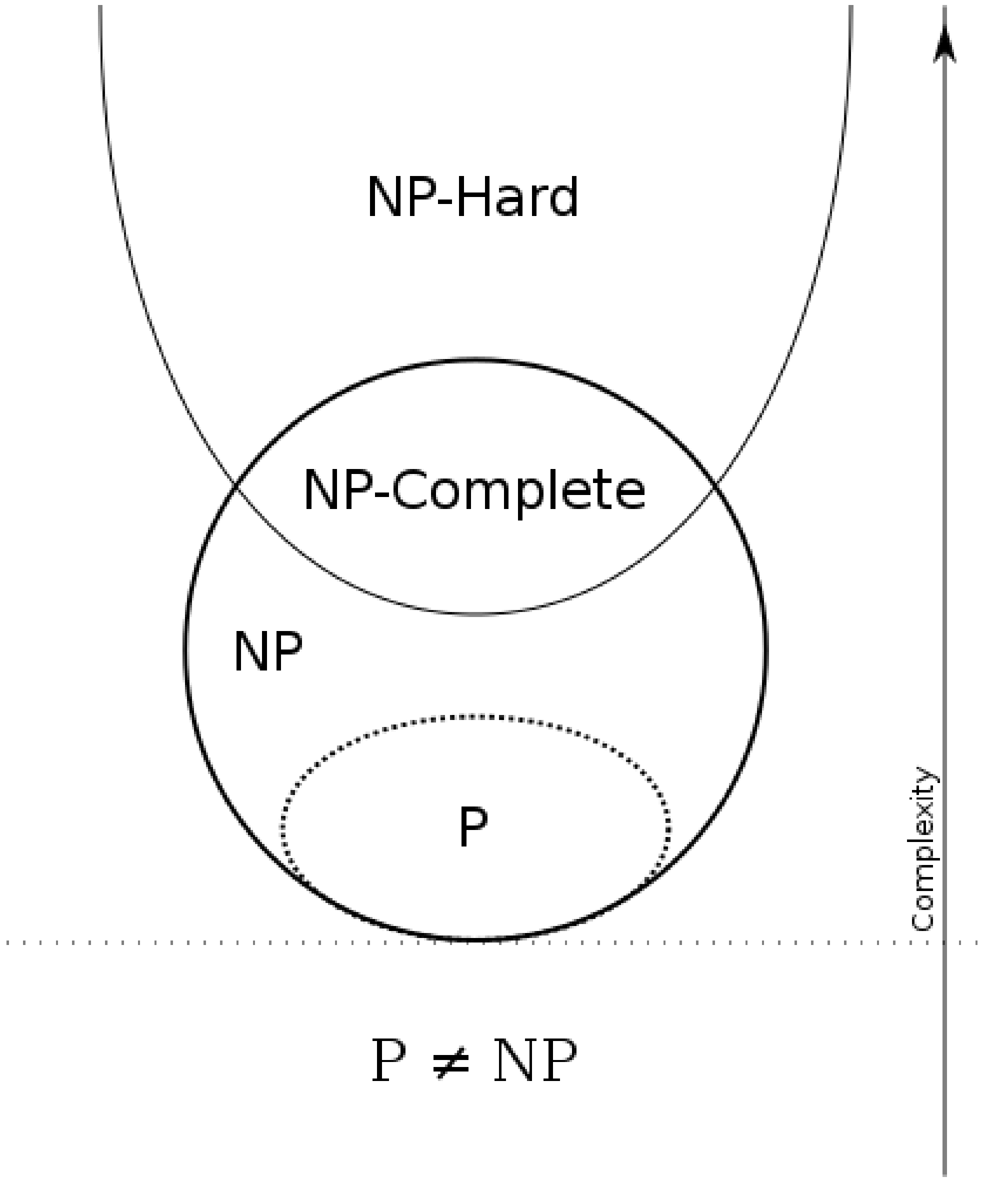
- Computational protein structure prediction (PSP) typically utilizes enumeration or searching strategies with optimization which involve enormous probability spaces

- $U = \sum_{\mathrm{b}} K_{\mathrm{b}}^{bond}(\mathrm{l_b} - l_b^0)^2 + \sum_a K_a^{angle}(\theta_a - \theta_a^0)^2 + \sum_t K_t^{torsion}(1 - \cos[n_t(\varphi_t - \varphi_t^0)]) + \sum_{i>j} K_{ij}^{nonlocal} f(r_{ij} / r_{ij}^0)$
  - related optimization algorithm for the current method of PSP
- Algorithm is NP-Hard

P, NP, and Computational Complexity[5]

- $P = NP$ vs $P \neq NP$
- *P* is the set of all problems that can be solved in polynomial time
- *NP* is the set of all problems that must be solved in non-deterministic polynomial time
- *NP*-Hard is as difficult as the most difficult *NP* problem in the set
- *NP*-Hard problems become computationally intractable if not optimized or abstracted

P, NP, and Computational Complexity[5,9]



Levinthal Paradox

- “In theory a protein is expected to require exponential time to fold, given an arbitrary starting configuration, whereas in practice proteins are observe to fold within seconds to minutes, independent of size… [where] exponential-time folding is expected because of the exponential size of the protein’s conformational space” [4]
- “Exhaustive search of a protein's conformational space is clearly not a feasible algorithmic strategy. The number of possible conformations is exponential in the length of the protein sequence, and powerful computational hardware would not be capable of searching this space for even moderately large proteins”[6]

Machine Learning

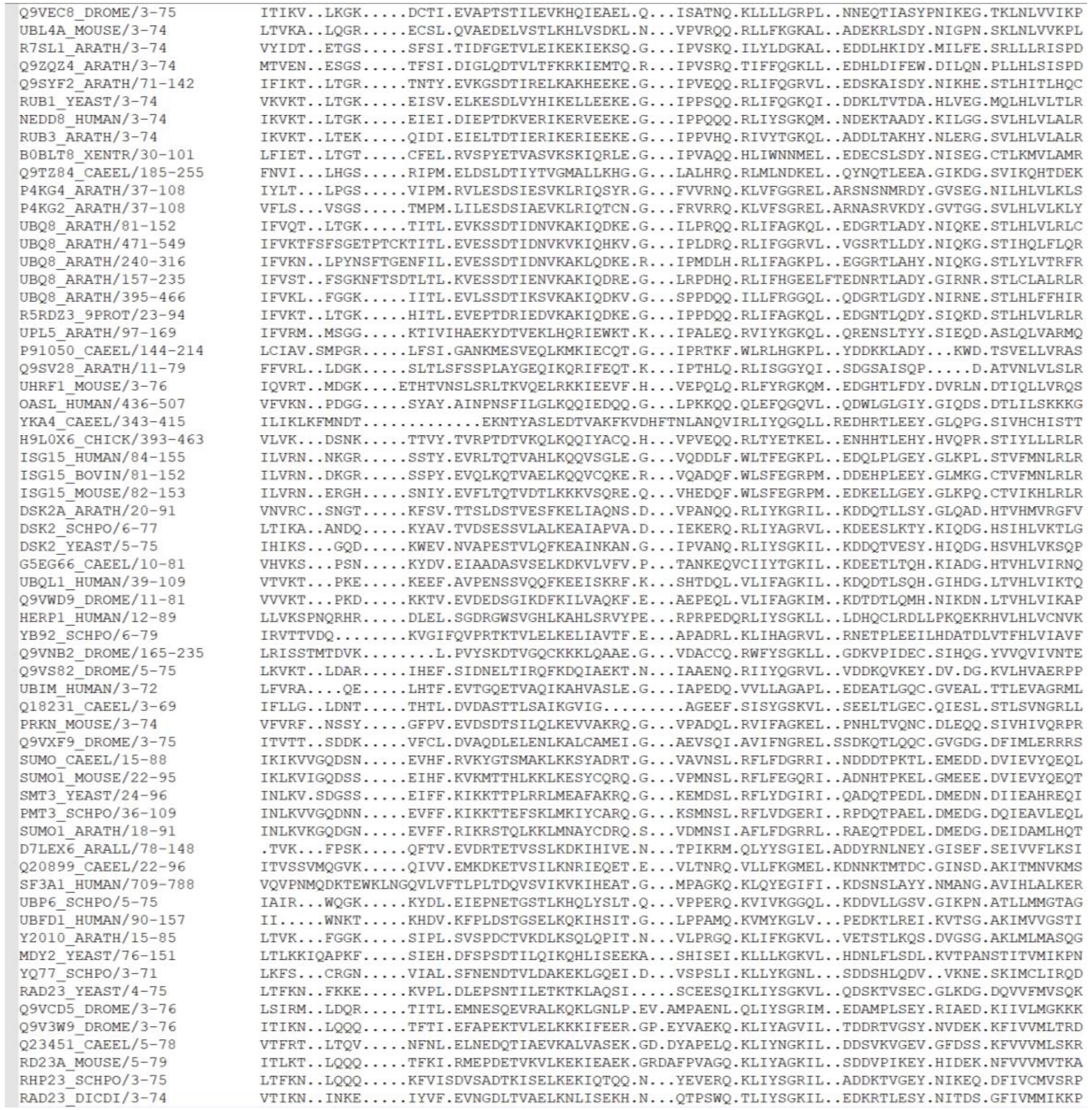
- K-modes Attribute Clustering Algorithm[7]
    - Learning algorithm for non-numeric data
    - take protein sequences and predict a structure in polynomial time based on probability and the mutual similarity of sequences between different groups of species
- Multiple Sequence Alignment
- Complexity[8]
    - $O(knp^2t)$

K-Modes Algorithm[7,8]

- Data-driven, machine learning algorithm for clustering similar data
- Clusters based on how dependent one attribute is on another
- Machine learning has immense potential for use in the biological and medical sciences, especially for process optimization and probabilistic prediction

Multiple Sequence Alignments[10]

- Alignment of a protein’s sequence across multiple organisms
  - Takes into consideration substitution, insertion, and deletion events
- Necessary for probabilistic purposes
  - More sequences allows for better resolution of results



```
Q9VEC8_DROME/3-75          ITIKV..LKGK.....DCTI.EVAPTSTILEVKHQIEAEL.Q...ISATNQ.KLLLLGRPL..NNEQTIASYPNIKEG.TKLNLVVIKP
UBL4A_MOUSE/3-74           LTVKA..LQGR.....ECSL.QVAEDELVSTLKHLVSDKL.N...VPVRQQ.RLLFKGKAL..ADEKRLSDY.NIGPN.SKLNLVVKPL
R7SL1_ARATH/3-74           VYIDT..ETGS.....SFSI.TIDFGETVLEIKEKIEKSQ.G...IPVSKQ.ILYLDGKAL..EDDLHKIDY.MILFE.SRLLLRISPD
Q9ZQZ4_ARATH/3-74          MTVEN..ESGS.....TFSI.DIGLQDTVLTFKRKIEMTQ.R...IPVSRQ.TIFFQGKLL..EDHLDIFEW.DILQN.PLLHLSISPD
Q9SYF2_ARATH/71-142        IFIKT..LTGR.....TNTY.EVKGSDTIRELKAKHEEKE.G...IPVEQQ.RLIFQGRVL..EDSKAISDY.NIKHE.STLHITLHQC
RUB1_YEAST/3-74            VKVKT..LTGK.....EISV.ELKESDLVYHIKELLEEKE.G...IPPSQQ.RLIFQGKQI..DDKLTVTDA.HLVEG.MQLHLVLTLR
NEDD8_HUMAN/3-74           IKVKT..LTGK.....EIEI.DIEPTDKVERIKERVEEKE.G...IPPQQQ.RLIYSGKQM..NDEKTAADY.KILGG.SVLHLVLALR
RUB3_ARATH/3-74            IKVKT..LTEK.....QIDI.EIELTDTIERIKERIEEKE.G...IPPVHQ.RIVYTGKQL..ADDLTAKHY.NLERG.SVLHLVLALR
B0BLT8_XENTR/30-101        LFIET..LTGT.....CFEL.RVSPYETVASVKSKIQRLE.G...IPVAQQ.HLIWNNMEL..EDECSLSDY.NISEG.CTLKMVLAMR
Q9TZ84_CAEEL/185-255       FNVI...LHGS.....RIPM.ELDSLDTIYTVGMALLKHG.G...LALHRQ.RLMLNDKEL..QYNQTLEEA.GIKDG.SVIKQHTDEK
P4KG4_ARATH/37-108         IYLT...LPGS.....VIPM.RVLESDSIESVKLRIQSYR.G...FVVRNQ.KLVFGGREL.ARSNSNMRDY.GVSEG.NILHLVLKLS
P4KG2_ARATH/37-108         VFLS...VSGS.....TMPM.LILESDSIAEVKLRIQTCN.G...FRVRRQ.KLVFSGREL.ARNASRVKDY.GVTGG.SVLHLVLKLY
UBQ8_ARATH/81-152          IFVQT..LTGK.....TITL.EVKSSDTIDNVKAKIQDKE.G...ILPRQQ.RLIFAGKQL..EDGRTLADY.NIQKE.STLHLVLRLC
UBQ8_ARATH/471-549         IFVKTFSFSGETPTCKTITL.EVESSDTIDNVKVKIQHKV.G...IPLDRQ.RLIFGGRVL..VGSRTLLDY.NIQKG.STIHQLFLQR
UBQ8_ARATH/240-316         IFVKN..LPYNSFTGENFIL.EVESSDTIDNVKAKLQDKE.R...IPMDLH.RLIFAGKPL..EGGRTLAHY.NIQKG.STLYLVTRFR
UBQ8_ARATH/157-235         IFVST..FSGKNFTSDTLTL.KVESSDTIENVKAKIQDRE.G...LRPDHQ.RLIFHGEELFTEDNRTLADY.GIRNR.STLCLALRLR
UBQ8_ARATH/395-466         IFVKL..FGGK.....IITL.EVLSSDTIKSVKAKIQDKV.G...SPPDQQ.ILLFRGGQL..QDGRTLGDY.NIRNE.STLHLFFHIR
R5RDZ3_9PROT/23-94         IFVKT..LTGK.....HITL.EVEPTDRIEDVKAKIQDKE.G...IPPDQQ.RLIFAGKQL..EDGNTLQDY.SIQKD.STLHLVLRLR
UPL5_ARATH/97-169          IFVRM..MSGG.....KTIVIHAEKYDTVEKLHQRIEWKT.K...IPALEQ.RVIYKGKQL..QRENSLTYY.SIEQD.ASLQLVARMQ
P91050_CAEEL/144-214       LCIAV.SMPGR.....LFSI.GANKMESVEQLKMKIECQT.G...IPRTKF.WLRLHGKPL..YDDKKLADY...KWD.TSVELLVRAS
Q9SV28_ARATH/11-79         FFVRL..LDGK.....SLTLSFSSPLAYGEQIKQRIFEQT.K...IPTHLQ.RLISGGYQI..SDGSAISQP.....D.ATVNLVLSLR
UHRF1_MOUSE/3-76           IQVRT..MDGK....ETHTVNSLSRLTKVQELRKKIEEVF.H...VEPQLQ.RLFYRGKQM..EDGHTLFDY.DVRLN.DTIQLLVRQS
OASL_HUMAN/436-507         VFVKN..PDGG.....SYAY.AINPNSFILGLKQQIEDQQ.G...LPKKQQ.QLEFQGQVL..QDWLGLGIY.GIQDS.DTLILSKKKG
YKA4_CAEEL/343-415         ILIKLKFMNDT.............EKNTYASLEDTVAKFKVDHFTNLANQVIRLIYQGQLL.REDHRTLEEY.GLQPG.SIVHCHISTT
H9L0X6_CHICK/393-463       VLVK...DSNK.....TTVY.TVRPTDTVKQLKQQIYACQ.H...VPVEQQ.RLTYETKEL..ENHHTLEHY.HVQPR.STIYLLLRLR
ISG15_HUMAN/84-155         ILVRN..NKGR.....SSTY.EVRLTQTVAHLKQQVSGLE.G...VQDDLF.WLTFEGKPL..EDQLPLGEY.GLKPL.STVFMNLRLR
ISG15_BOVIN/81-152         ILVRN..DKGR.....SSPY.EVQLKQTVAELKQQVCQKE.R...VQADQF.WLSFEGRPM..DDEHPLEEY.GLMKG.CTVFMNLRLR
ISG15_MOUSE/82-153         ILVRN..ERGH.....SNIY.EVFLTQTVDTLKKKVSQRE.Q...VHEDQF.WLSFEGRPM..EDKELLGEY.GLKPQ.CTVIKHLRLR
DSK2A_ARATH/20-91          VNVRC..SNGT.....KFSV.TTSLDSTVESFKELIAQNS.D...VPANQQ.RLIYKGRIL..KDDQTLLSY.GLQAD.HTVHMVRGFV
DSK2_SCHPO/6-77            LTIKA..ANDQ.....KYAV.TVDSESSVLALKEAIAPVA.D...IEKERQ.RLIYAGRVL..KDEESLKTY.KIQDG.HSIHLVKTLG
DSK2_YEAST/5-75            IHIKS...GQD.....KWEV.NVAPESTVLQFKEAINKAN.G...IPVANQ.RLIYSGKIL..KDDQTVESY.HIQDG.HSVHLVKSQP
G5EG66_CAEEL/10-81         VHVKS...PSN.....KYDV.EIAADASVSELKDKVLVFV.P...TANKEQVCIIYTGKIL..KDEETLTQH.KIADG.HTVHLVIRNQ
UBQL1_HUMAN/39-109         VTVKT...PKE.....KEEF.AVPENSSVQQFKEEISKRF.K...SHTDQL.VLIFAGKIL..KDQDTLSQH.GIHDG.LTVHLVIKTQ
Q9VWD9_DROME/11-81         VVVKT...PKD.....KKTV.EVDEDSGIKDFKILVAQKF.E...AEPEQL.VLIFAGKIM..KDTDTLQMH.NIKDN.LTVHLVIKAP
HERP1_HUMAN/12-89          LLVKSPNQRHR.....DLEL.SGDRGWSVGHLKAHLSRVYPE...RPRPEDQRLIYSGKLL..LDHQCLRDLLPKQEKRHVLHLVCNVK
YB92_SCHPO/6-79            IRVTTVDQ........KVGIFQVPRTKTVLELKELIAVTF.E...APADRL.KLIHAGRVL..RNETPLEEILHDATDLVTFHLVIAVF
Q9VNB2_DROME/165-235       LRISSTMTDVK........L.PVYSKDTVGQCKKKLQAAE.G...VDACCQ.RWFYSGKLL..GDKVPIDEC.SIHQG.YVVQVIVNTE
Q9VS82_DROME/5-75          LKVKT..LDAR.....IHEF.SIDNELTIRQFKDQIAEKT.N...IAAENQ.RIIYQGRVL..VDDKQVKEY.DV.DG.KVLHVAERPP
UBIM_HUMAN/3-72            LFVRA....QE.....LHTF.EVTGQETVAQIKAHVASLE.G...IAPEDQ.VVLLAGAPL..EDEATLGQC.GVEAL.TTLEVAGRML
Q18231_CAEEL/3-69          IFLLG..LDNT.....THTL.DVDASTTLSAIKGVIG.........AGEEF.SISYGSKVL..SEELTLGEC.QIESL.STLSVNGRLL
PRKN_MOUSE/3-74            VFVRF..NSSY.....GFPV.EVDSDTSILQLKEVVAKRQ.G...VPADQL.RVIFAGKEL..PNHLTVQNC.DLEQQ.SIVHIVQRPR
Q9VXF9_DROME/3-75          ITVTT..SDDK.....VFCL.DVAQDLELENLKALCAMEI.G...AEVSQI.AVIFNGREL.SSDKQTLQQC.GVGDG.DFIMLERRRS
SUMO_CAEEL/15-88           IKIKVVGQDSN.....EVHF.RVKYGTSMAKLKKSYADRT.G...VAVNSL.RFLFDGRRI..NDDDTPKTL.EMEDD.DVIEVYQEQL
SUMO1_MOUSE/22-95          IKLKVIGQDSS.....EIHF.KVKMTTHLKKLKESYCQRQ.G...VPMNSL.RFLFEGQRI..ADNHTPKEL.GMEEE.DVIEVYQEQT
SMT3_YEAST/24-96           INLKV.SDGSS.....EIFF.KIKKTTPLRRLMEAFAKRQ.G...KEMDSL.RFLYDGIRI..QADQTPEDL.DMEDN.DIIEAHREQI
PMT3_SCHPO/36-109          INLKVVGQDNN.....EVFF.KIKKTTEFSKLMKIYCARQ.G...KSMNSL.RFLVDGERI..RPDQTPAEL.DMEDG.DQIEAVLEQL
SUMO1_ARATH/18-91          INLKVKGQDGN.....EVFF.RIKRSTQLKKLMNAYCDRQ.S...VDMNSI.AFLFDGRRL..RAEQTPDEL.DMEDG.DEIDAMLHQT
D7LEX6_ARALL/78-148        .TVK...FPSK.....QFTV.EVDRTETVSSLKDKIHIVE.N...TPIKRM.QLYYSGIEL.ADDYRNLNEY.GISEF.SEIVVFLKSI
Q20899_CAEEL/22-96         ITVSSVMQGVK.....QIVV.EMKDKETVSILKNRIEQET.E...VLTNRQ.VLLFKGMEL.KDNNKTMTDC.GINSD.AKITMNVKMS
SF3A1_HUMAN/709-788        VQVPNMQDKTEWKLNGQVLVFTLPLTDQVSVIKVKIHEAT.G...MPAGKQ.KLQYEGIFI..KDSNSLAYY.NMANG.AVIHLALKER
UBP6_SCHPO/5-75            IAIR...WQGK.....KYDL.EIEPNETGSTLKHQLYSLT.Q...VPPERQ.KVIVKGGQL..KDDVLLGSV.GIKPN.ATLLMMGTAG
UBFD1_HUMAN/90-157         II.....WNKT.....KHDV.KFPLDSTGSELKQKIHSIT.G...LPPAMQ.KVMYKGLV...PEDKTLREI.KVTSG.AKIMVVGSTI
Y2010_ARATH/15-85          LTVK...FGGK.....SIPL.SVSPDCTVKDLKSQLQPIT.N...VLPRGQ.KLIFKGKVL..VETSTLKQS.DVGSG.AKLMLMASQG
MDY2_YEAST/76-151          LTLKKIQAPKF.....SIEH.DFSPSDTILQIKQHLISEEKA...SHISEI.KLLLKGKVL..HDNLFLSDL.KVTPANSTITVMIKPN
YQ77_SCHPO/3-71            LKFS...CRGN.....VIAL.SFNENDTVLDAKEKLGQEI.D...VSPSLI.KLLYKGNL...SDDSHLQDV..VKNE.SKIMCLIRQD
RAD23_YEAST/4-75           LTFKN..FKKE.....KVPL.DLEPSNTILETKTKLAQSI.....SCEESQIKLIYSGKVL..QDSKTVSEC.GLKDG.DQVVFMVSQK
Q9VCD5_DROME/3-76          LSIRM..LDQR.....TITL.EMNESQEVRALKQKLGNLP.EV.AMPAENL.QLIYSGRIM..EDAMPLSEY.RIAED.KIIVLMGKKK
Q9V3W9_DROME/3-76          ITIKN..LQQQ.....TFTI.EFAPEKTVLELKKKIFEER.GP.EYVAEKQ.KLIYAGVIL..TDDRTVGSY.NVDEK.KFIVVMLTRD
Q23451_CAEEL/5-78          VTFRT..LTQV.....NFNL.ELNEDQTIAEVKALVASEK.GD.DYAPELQ.KLIYNGKIL..DDSVKVGEV.GFDSS.KFVVVMLSKR
RD23A_MOUSE/5-79           ITLKT..LQQQ.....TFKI.RMEPDETVKVLKEKIEAEK.GRDAFPVAGQ.KLIYAGKIL..SDDVPIKEY.HIDEK.NFVVVMVTKA
RHP23_SCHPO/3-75           LTFKN..LQQQ.....KFVISDVSADTKISELKEKIQTQQ.N...YEVERQ.KLIYSGRIL..ADDKTVGEY.NIKEQ.DFIVCMVSRP
RAD23_DICDI/3-74           VTIKN..INKE.....IYVF.EVNGDLTVAELKNLISEKH.N...QTPSWQ.TLIYSGKIL..EDKRTLESY.NITDS.GFIVMMIKKP
```

Attributes

- Once the multiple sequence alignment data has been obtained, it needs to be parsed through and manipulated for prediction calculations
  - Array manipulation
- Attributes within the data array
  - An attribute in this case is a column of amino acids within the aligned sequence array

```
data = [
    ['C', 'A', 'R', 'C', 'A', 'W', 'A', 'A'],
    ['C', 'G', 'K', 'C', 'G', 'Y', 'G', 'G'],
    ['C', 'N', 'M', 'C', 'N', 'F', 'N', 'N'],
    ['C', 'D', 'I', 'C', 'D', 'V', 'D', 'D'],
    ['C', 'A', 'L', 'C', 'A', 'H', 'A', 'A'],
    ['C', 'G', 'R', 'C', 'G', 'Q', 'G', 'G'],
    ['C', 'N', 'K', 'C', 'N', 'E', 'N', 'N'],
    ['T', 'D', 'M', 'T', 'D', 'P', 'D', 'D'],
    ['T', 'A', 'I', 'T', 'A', 'W', 'A', 'A'],
    ['T', 'G', 'L', 'T', 'G', 'Y', 'G', 'G'],
    ['T', 'N', 'R', 'T', 'N', 'F', 'N', 'N'],
    ['T', 'D', 'K', 'T', 'D', 'V', 'D', 'D'],
    ['S', 'A', 'M', 'S', 'A', 'H', 'A', 'A'],
    ['S', 'G', 'I', 'S', 'G', 'Q', 'G', 'G'],
    ['S', 'N', 'L', 'S', 'N', 'E', 'N', 'N']
]
```

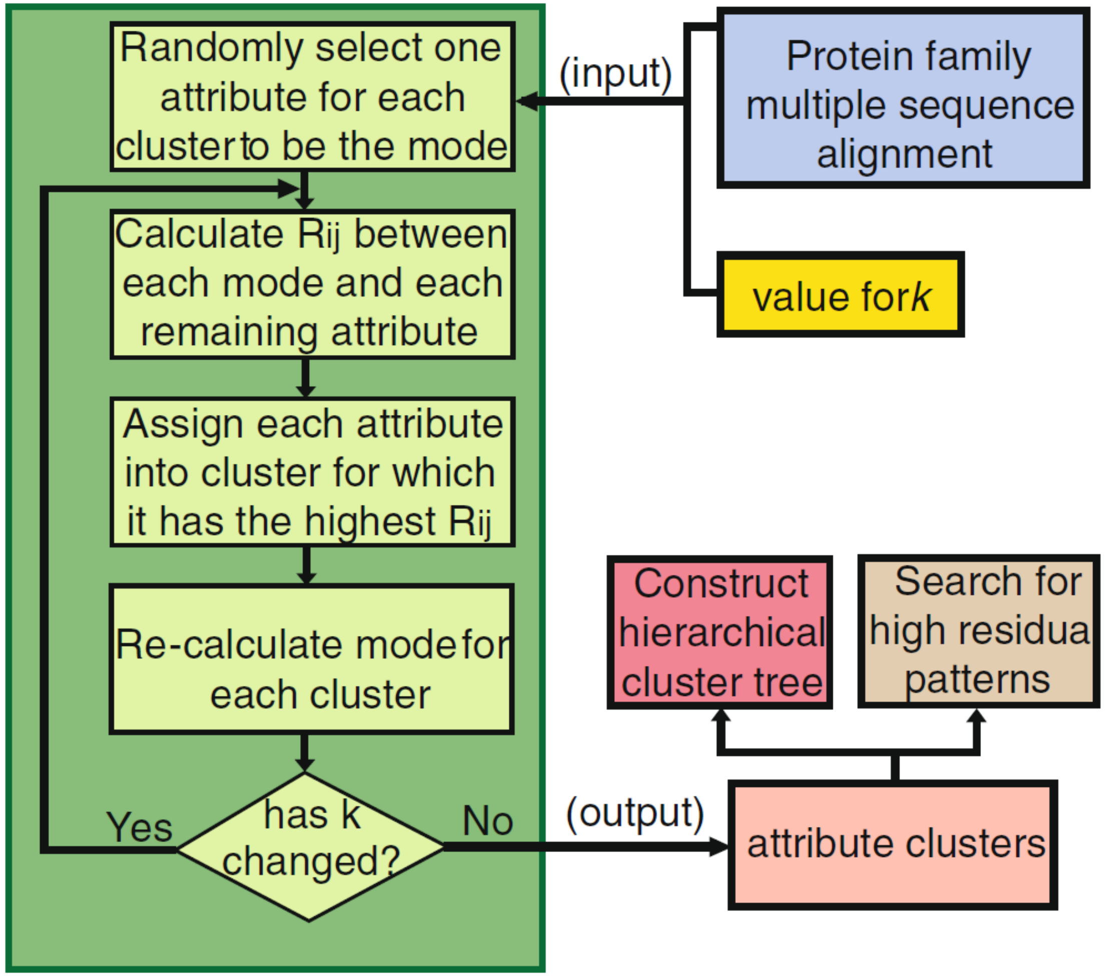
Interdependence and Mutual Information[7,11,12]

- Mutual information is calculated to determine the interdependency relationship between two attributes
- Mutual information is normalized by the joint entropy (Shannon entropy) of the sequences as well to yield a more interpretable result
- $I(X_i, X_j) \ = \sum_{x \in X_i} \sum_{y \in X_j} p(x,y) log(\frac{p(x,y)}{p(x)p(y)})$
  - Mutual Information
- $H(X_i, X_j) \ = \ -\sum_{x \in X_i} \sum_{y \in X_j} p(x,y) log(p(x,y))$
  - Shannon Entropy
- $R_{ij} \ = \ \frac{I(X_i, X_j)}{H(X_i, X_j)}$
  - Normalized Mutual Information
  - Interdependency Redundancy

Algorithmic Approach[13]

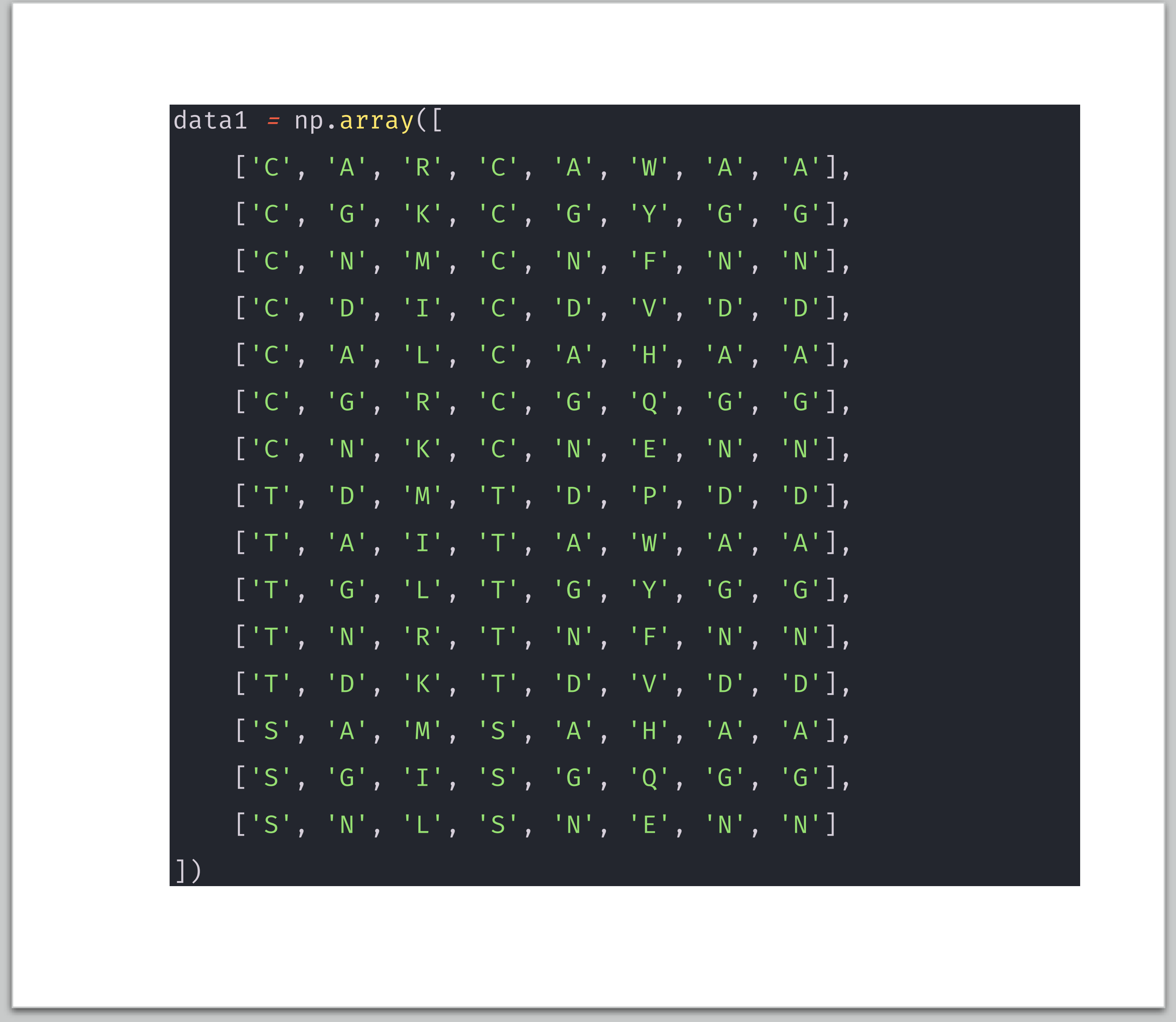
- Python, Scikit-learn, and Numpy
- Once a multiple sequence alignment has been input into the program, a number of steps take place in a loop-wise fashion until the prescribed value for *k* has been reached
    - algorithm will run for a number of iterations from *k* – 1 to *k* = 2
- Output from each iteration will be stored in a file which is then used to construct a cluster tree graph

Algorithm[7]


Randomly select one attribute for each clusterto be the mode
Calculate Rij between each mode and each remaining attribute
Assign each attribute into cluster for which it has the highest Rij
Re-calculate mode for each cluster
has k changed?
Yes
No
(input)
Protein family multiple sequence alignment
value fork
(output)
attribute clusters
Construct hierarchical cluster tree
Search for high residua patterns

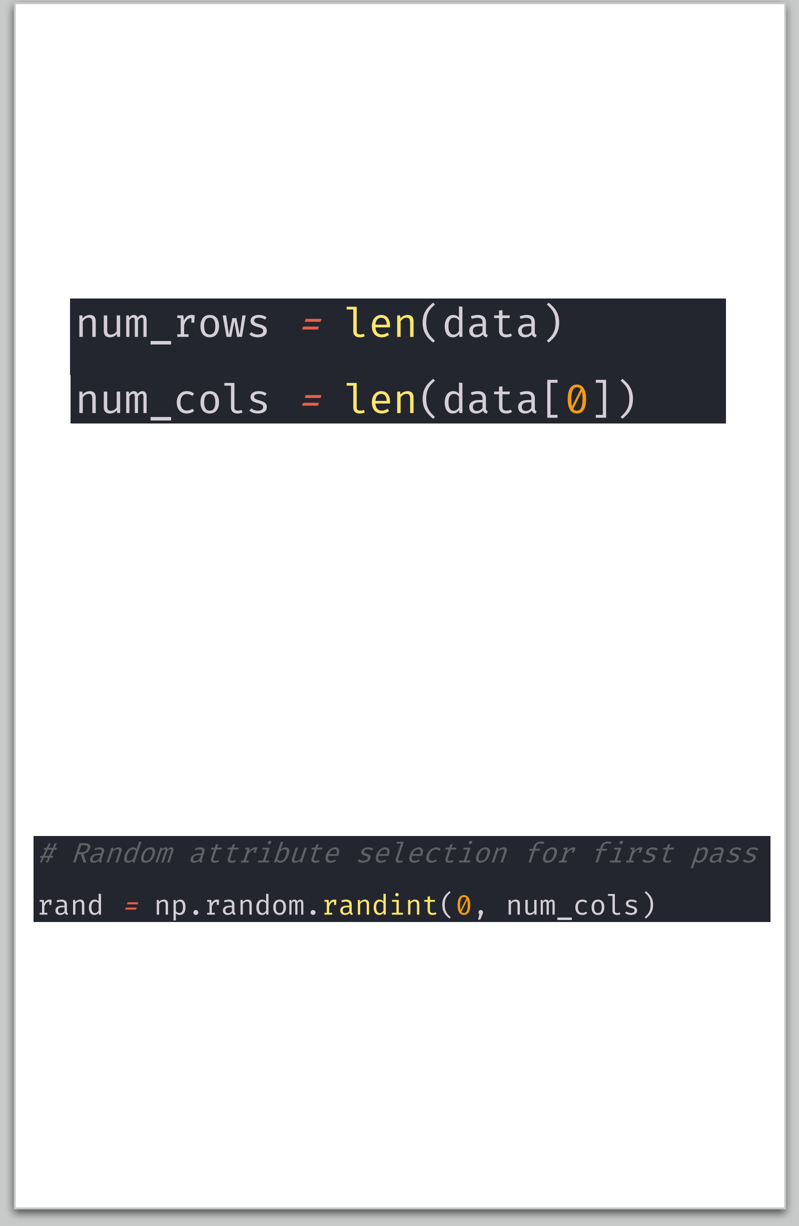
Code

- Hard-coded test sequence
  - In the future, would allow for upload of code via file parsing



```
data1 = np.array([
    ['C', 'A', 'R', 'C', 'A', 'W', 'A', 'A'],
    ['C', 'G', 'K', 'C', 'G', 'Y', 'G', 'G'],
    ['C', 'N', 'M', 'C', 'N', 'F', 'N', 'N'],
    ['C', 'D', 'I', 'C', 'D', 'V', 'D', 'D'],
    ['C', 'A', 'L', 'C', 'A', 'H', 'A', 'A'],
    ['C', 'G', 'R', 'C', 'G', 'Q', 'G', 'G'],
    ['C', 'N', 'K', 'C', 'N', 'E', 'N', 'N'],
    ['T', 'D', 'M', 'T', 'D', 'P', 'D', 'D'],
    ['T', 'A', 'I', 'T', 'A', 'W', 'A', 'A'],
    ['T', 'G', 'L', 'T', 'G', 'Y', 'G', 'G'],
    ['T', 'N', 'R', 'T', 'N', 'F', 'N', 'N'],
    ['T', 'D', 'K', 'T', 'D', 'V', 'D', 'D'],
    ['S', 'A', 'M', 'S', 'A', 'H', 'A', 'A'],
    ['S', 'G', 'I', 'S', 'G', 'Q', 'G', 'G'],
    ['S', 'N', 'L', 'S', 'N', 'E', 'N', 'N']
])
```

Code



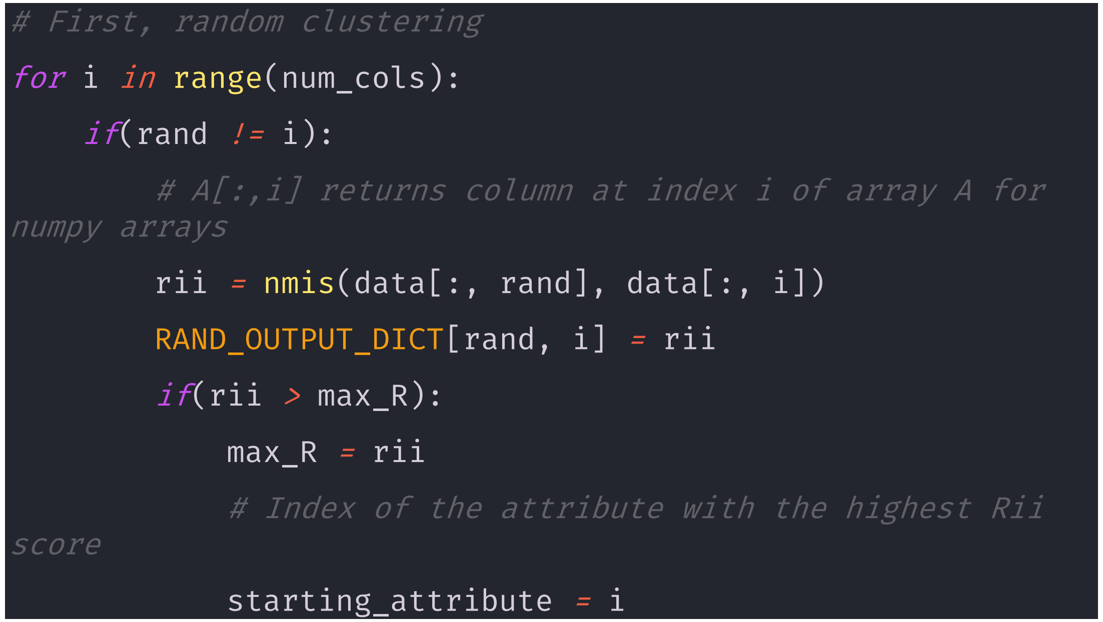
```
num_rows = len(data)
num_cols = len(data[0])
```

```
# Random attribute selection for first pass
rand = np.random.randint(0, num_cols)
```

- Computing the length of the data array to assign to k
- Random attribute selection for first iteration

Code

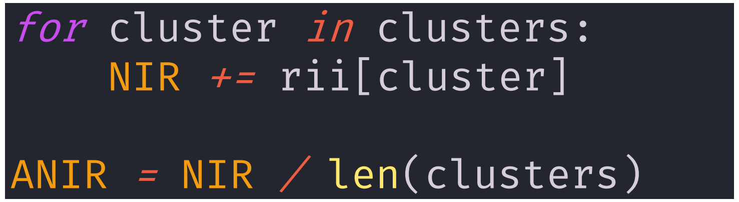
- Looping through the data array to calculate the interdependency redundancy between attributes



```python
# First, random clustering
for i in range(num_cols):
    if(rand != i):
        # A[:,i] returns column at index i of array A for numpy arrays
        rii = nmis(data[:, rand], data[:, i])
        RAND_OUTPUT_DICT[rand, i] = rii
        if(rii > max_R):
            max_R = rii
            # Index of the attribute with the highest Rii score
            starting_attribute = i
```

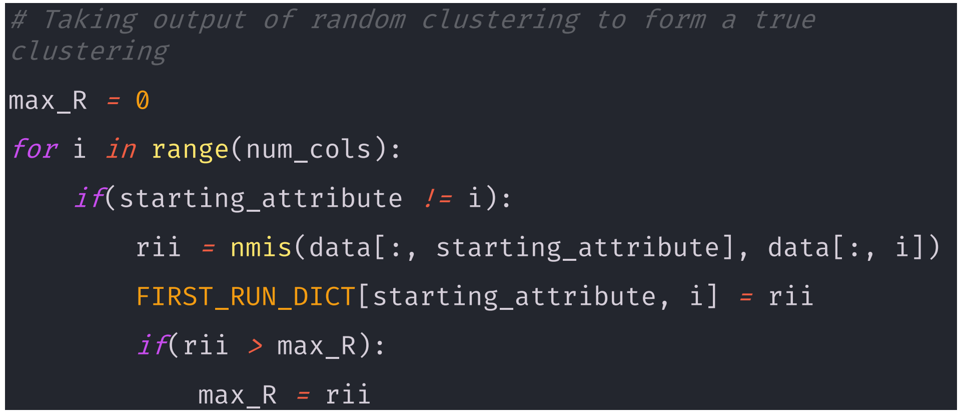
Code

- Calculating the Normalized Interdependency Redundancy (NIR) and Average Normalized Interdependency Redundancy (ANIR)



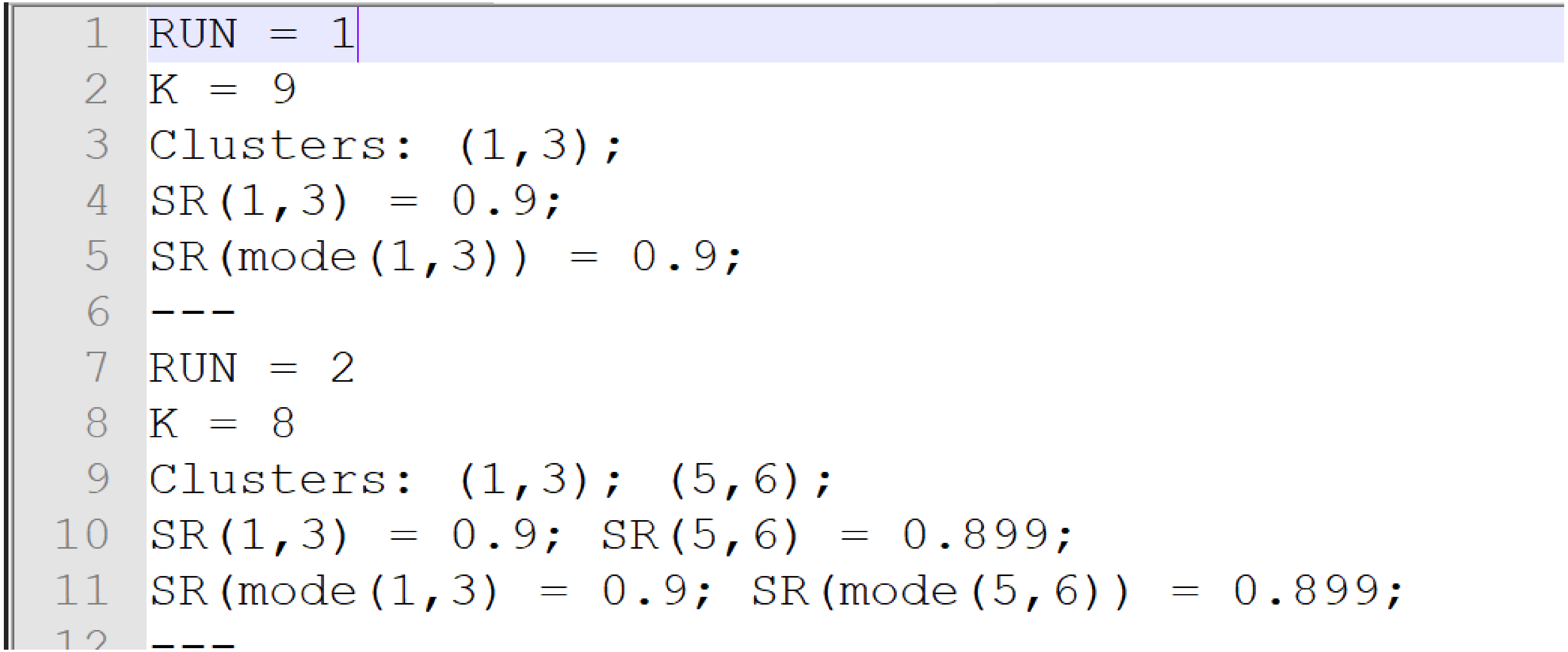
```
for cluster in clusters:
    NIR += rii[cluster]

ANIR = NIR / len(clusters)
```

- Performing clustering again without random selection for remaining iterations



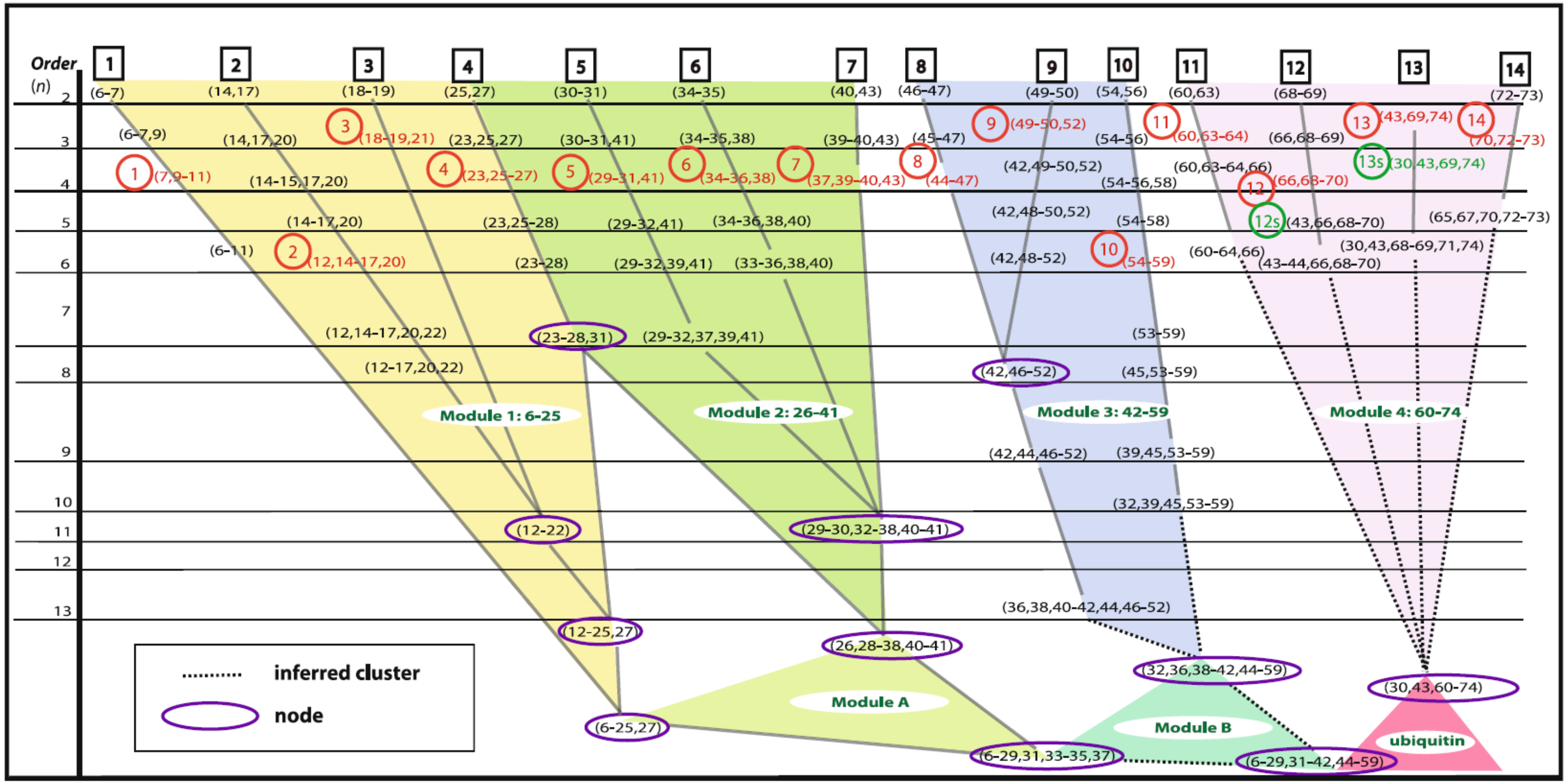
```
# Taking output of random clustering to form a true
clustering
max_R = 0
for i in range(num_cols):
    if(starting_attribute != i):
        rii = nmis(data[:, starting_attribute], data[:, i])
        FIRST_RUN_DICT[starting_attribute, i] = rii
        if(rii > max_R):
            max_R = rii
```

Code Output



```
RUN = 1
K = 9
Clusters: (1,3);
SR(1,3) = 0.9;
SR(mode(1,3)) = 0.9;
---
RUN = 2
K = 8
Clusters: (1,3); (5,6);
SR(1,3) = 0.9; SR(5,6) = 0.899;
SR(mode(1,3) = 0.9; SR(mode(5,6)) = 0.899;
---
```

Cluster Trees[7]



Next Steps

- Verification and repeatability are of the utmost importance if this tool is to be used by other researchers
- Next steps should focus on validating this algorithm against protein sequences with known and verified crystallographic structures
- Algorithmic scalability is also an area ripe for optimization
  - Protein sequence and probability space grows exponentially with the length of the sequence
  - Intractable program runtime
  - Computational Clusters

Reiteration

- The importance of the scope of this project lies in the utility of proteins as medicinal targets for various therapeutic strategies
- Having a tool that can accurately and efficiently predict protein structure and function from a sequence would be useful for new drug design and targeting techniques as well as understanding the pathophysiology of disease states

References

1. Deweese, J., et al. (2019). The variable C-terminal domain of human type II topoisomerases as a functionally relevant therapeutic target. American Society for Biochemistry and Molecular Biology, Orlando, FL.

2. Carpenter, E. P., et al. (2008). "Overcoming the challenges of membrane protein crystallography." Curr Opin Struct Biol **18**(5): 581-586.

3. Lacapere, J. J., et al. (2007). "Determining membrane protein structures: still a challenge!" Trends Biochem Sci **32**(6): 259-270.

4. Ngo, J. T., et al. (1994). "Computational Complexity, Protein Structure Prediction, and the Levinthal Paradox." The Protein Folding Problem and Tertiary Structure Prediction: 433-506.

5. Knuth, D. E. (1974). "Postscript about NP-hard problems." ACM SIGACT News **6**(2): 15-16.

6. Newman, A. and W. E. Hart (2001). "The Computational Complexity of Protein Structure Prediction in Simple Lattice Models." CRC Press.

7. Durston, K. K., et al. (2012). "Statistical discovery of site inter-dependencies in sub-molecular hierarchical protein structuring." EURASIP J Bioinform Syst Biol **2012**(1): 8.

8. Au, W. H., et al. (2005). "Attribute clustering for grouping, selection, and classification of gene expression data." IEEE/ACM Trans Comput Biol Bioinform **2**(2): 83-101.

9. Esfahbod, B. (2007). N. Euler diagram for P, NP-Complete, and NP-Hard set of problems. Wikimedia Commons.

10. Laboratory, E. M. B. (2019). "Family: ubiquitin (PF00240)." 2019, from https://pfam.xfam.org/family/PF00240#tabview=tab3.

11. Wong, A. K. C. and G. C. L. Li (2008). "Simultaneous Pattern and Data Clustering for Pattern Cluster Analysis." IEEE Transactions on Knowledge and Data Engineering **20**(7): 911-923.

12. Wong, A. K. C., et al. (1976). "Statistical analysis of residue variability in cytochrome c." Journal of Molecular Biology **102**(2): 287-295.

13. Pedregosa, F., et al. (2013). "Scikit-learn: Machine Learning in Python." Journal of Machine Learning Research **12**: 2825-2830.

Questions

A Machine Learning Approach for Prediction of Protein Structure and Function