

Summer 8-2021

Automated Parsing of Flexible Molecular Systems using Principal Component Analysis and K-Means Clustering Techniques

Matthew J. Nwerem

Chapman University, m.nwerem@gmail.com

Follow this and additional works at: https://digitalcommons.chapman.edu/cads_theses

 Part of the [Organic Chemistry Commons](#), [Other Computer Sciences Commons](#), and the [Structural Biology Commons](#)

Recommended Citation

M. Nwerem, "Automated parsing of flexible molecular systems using principal component analysis and K-means clustering techniques," M. S. thesis, Chapman University, Orange, CA, 2021. <https://doi.org/10.36837/chapman.000293>

This Thesis is brought to you for free and open access by the Dissertations and Theses at Chapman University Digital Commons. It has been accepted for inclusion in Computational and Data Sciences (MS) Theses by an authorized administrator of Chapman University Digital Commons. For more information, please contact laughtin@chapman.edu.

Automated Parsing of Flexible Molecular Systems using
Principal Component Analysis and K-Means Clustering
Techniques

A Thesis by

Matthew Jonathan Chukwunenye Nwerem

Chapman University

Orange, CA

Schmid College of Science and Technology

Submitted in partial fulfillment of the requirements for the degree of

Master of Science in Computational and Data Sciences

August, 2021

Committee in charge:

O. Maduka Ogba, Ph.D., Chair

Gennady Verkhivker, Ph.D.

Lindsay Waldrop, Ph.D.



CHAPMAN UNIVERSITY
SCHMID COLLEGE OF SCIENCE AND TECHNOLOGY

Computational and Data Sciences

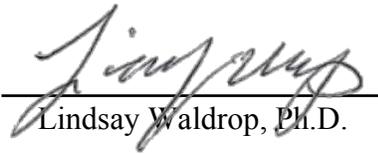
The thesis of Matthew Jonathan Chukwunenye Nwerem
is approved.



O. Maduka Ogba, Ph.D., Chair



Gennady Verkhivker, Ph.D.



Lindsay Waldrop, Ph.D.

June, 2021

Automated Parsing of Flexible Molecular Systems using Principal Component Analysis and K-Means Clustering Techniques

Copyright © 2021

by Matthew Jonathan Chukwunenye Nwerem

ACKNOWLEDGEMENTS

I would like to thank my family, for helping me reach my goals. I would also like to thank Dr. Ogba and Dr. Schwartz, as I believe their guidance has been instrumental to my successes.

To my committee, thank you for your support throughout the thesis defense process.

To my friends in the CADS program, and my friends I met throughout my five years here at Chapman, I cannot thank you enough for giving me such a fruitful experience.

ABSTRACT

Automated Parsing of Flexible Molecular Systems using Principal Component Analysis and K-Means Clustering Techniques

by Matthew Jonathan Chukwunenye Nwerem

Computational investigation of molecular structures and reactions of biological and pharmaceutical interests remains a grand scientific challenge due to the size and conformational flexibility of these systems. The work requires parsing and analyzing thousands of conformations in each molecular state for meaningful chemical information and subjecting the ensemble to costly quantum chemical calculations. The current status quo typically involves a manual process where the investigator must look at each conformation, separating each into structural families. This process is time-intensive and tedious, making this process infeasible in some cases, and limiting the ability of theoreticians to study these systems. However, the use of computational software allows for the necessary exhaustive investigation without the bottlenecks of a brute force approach to each flexible system.

I aim to create the solution to this problem. In my thesis project, I seek to develop a Python software that will (i) automate the parsing of each conformation within a conformational ensemble, (ii) use principal component analysis (PCA) and clustering to find and investigate conformational families within the ensemble, (iii) separate and visualize conformational families in a user-friendly manner, and (iv) convey to the user how conformational families were delineated by way of features found within data. Results explored this work show that the program has the ability to separate conformational families with varying ranges of difficulty.

TABLE OF CONTENTS

	<u>Page</u>
ACKNOWLEDGEMENTS	IV
ABSTRACT	V
LIST OF FIGURES	VIII
LIST OF TABLES	XI
LIST OF ABBREVIATIONS	XII
LIST OF SYMBOLS	XIII
1 INTRODUCTION	1
1.1 What is Conformational Analysis?	1
1.2 Modes of Performing Conformational Analysis	2
1.3 Current State of Conformational Analysis Field	6
1.4 My Goal in Conformational Analysis Field.....	7
2 METHODS	8
2.1 Software	8
2.1.1 Packages Used	9
2.1.2 PyMOL Integration	9
2.2 Algorithm	10
2.2.1 Inter-Atomic Calculation Functions	11
2.2.2 PCA.....	14
2.2.3 Clustering.....	15
2.3 Installation and How-To	18
2.4 Testing of Known Conformational Families	19
3 RESULTS/DISCUSSION	20
3.1 Conformational Analysis of Substituted Cyclohexanes	20
3.1.1 Substituted Cyclohexane Test Case: <i>Cis</i> -1-Flouro-4-Propylcyclohexane.....	22
3.2 Conformational Analysis of Tri-Valine Peptides	28
3.2.1 Tri-Glycine Peptides Test Case: Ac-(Gly) ₃ -NHMe.....	30
3.2.2 Tri-Valine Peptides Test Case: Ac-(Val) ₃ -NHMe	34
3.3 Conformational Analysis of Hexacoordinated Ca ²⁺ Complexes	41
3.3.1 Hexacoordinated Ca ²⁺ Complex Test Case: [Ca(NH ₃) ₂ (THF) ₄] ²⁺	42

4 CONCLUSION	46
4.1 Pros and Cons of Code based on Test Cases	46
4.2 Future Developments	49
4.3 Concluding Thoughts.....	49
REFERENCES.....	51
APPENDICES.....	61

LIST OF FIGURES

	<u>Page</u>
Figure 1-1 Conformational Analysis Technique Timeline	4
Figure 2-1 Example XYZ file.....	10
Figure 2-2 Conformational Analysis Algorithm Flowchart	11
Figure 2-3 Example Data Frame.....	11
Figure 2-4 Positive Angles have Negative Equal	13
Figure 2-5 Example Hierarchical Clustering Plot.....	16
Figure 2-6 Example Density-Based Clustering Plot; Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.....	16
Figure 2-7 Example Centroid Clustering Plot with Centroids 0 and 1	18
Figure 3-1 Energetic preference of cis-1,4 disubstituted cyclohexanes	22
Figure 3-2 Ring Flip Between Two Conformational Families of Cis-1-Flouro-4- Propylcyclohexane.....	22
Figure 3-3 2D Cis-1-Flouro-4-Propylcyclohexane Chair Flip Diagram	23
Figure 3-4 2D Cis-1-Flouro-4-Propylcyclohexane C2-C5 Ring Flip Structures.....	24
Figure 3-5 2D Cis-1-Flouro-4-Propylcyclohexane All Axial Ring Flip Structures	24
Figure 3-6 Inertia v. # Clusters of Cis-1-Flouro-4-Propylcyclohexane (Dihedral data only)	25
Figure 3-7 2D Clustering Families of Cis-1-Flouro-4-Propylcyclohexane (4 clusters)	26
Figure 3-8 2D Clustering of Cis-1-Flouro-4-Propylcyclohexane (4 clusters).....	26
Figure 3-9 2D Clustering of Cis-1-Flouro-4-Propylcyclohexane (5 clusters).....	27
Figure 3-10 Example of alpha-helices with hydrogen bonding.....	29
Figure 3-11 Beta-turn example	29
Figure 3-12 Extended conformation example	30
Figure 3-13 Intertias v. # of Clusters of Ac-(Gly) ₃ -NHMe (all atoms; distances only)....	31
Figure 3-14 Clustering of Ac-(Gly) ₃ -NHMe (all atoms; distances only)	31

Figure 3-15 Clustering of Ac-(Gly) ₃ -NHMe (O, N, and H atoms; distances only).....	32
Figure 3-16 Conformational families of Ac-(Gly) ₃ -NHMe.....	33
Figure 3-17 Example structure of two gamma-turns in one conformation.....	34
Figure 3-18 2D Clustering of Ac-(Val) ₃ -NHMe (all atoms; distances only)	36
Figure 3-19 2D Clustering of Ac-(Val) ₃ -NHMe (O, N, and H atoms; distances only)	37
Figure 3-20 2D Clustering of Ac-(Val) ₃ -NHMe (all atoms; dihedrals only)	38
Figure 3-21 Conformational families of Ac-(Val) ₃ -NHMe	39
Figure 3-22 Example structure of conformational family 1 (gamma + extended)	40
Figure 3-23 Conformational family 3 Variation (gamma-turn).....	40
Figure 3-24 Elbow Graph of [Ca(NH ₃) ₂ (THF) ₄] ₂ ⁺ (all atoms; angles only)	43
Figure 3-25 Clustering Graph of [Ca(NH ₃) ₂ (THF) ₄] ₂ ⁺ (all atoms; angles only)	43
Figure 3-26 Side-by-side comparison of most different structures between cis [Ca(NH ₃) ₂ (THF) ₄] ₂ ⁺ families (all atoms; angles only).....	44
Figure 3-27 Elbow Graph of [Ca(NH ₃) ₂ (THF) ₄] ₂ ⁺ (O, N, C, F atoms; angles only)	45
Figure 3-28 Clustering Graph of [Ca(NH ₃) ₂ (THF) ₄] ₂ ⁺ (O, N, C, F atoms; angles only)	45
Figure 3-29 Clustering Graph of [Ca(NH ₃) ₂ (THF) ₄] ₂ ⁺ (O, N, C, F atoms; angles only)	46

LIST OF TABLES

Page

Table 1 Python Packages Used	9
------------------------------------	---

LIST OF ABBREVIATIONS

<u>Abbreviation</u>	<u>Meaning</u>
<i>NMR</i>	Nuclear Magnetic Resonance
<i>PCA</i>	Principal Component Analysis
<i>ML</i>	Machine Learning
<i>DL</i>	Deep Learning
<i>NMR</i>	Nuclear Magnetic Resonance
<i>VCD</i>	Vibrational Circular Dichroism
<i>ECD</i>	Electronic Circular Dichroism

LIST OF SYMBOLS

<u>Symbol</u>	<u>Meaning</u>
Å	Angstrom, a unit of length equal to one hundred-millionth of a centimeter, 10^{-10} meter, used mainly to express wavelengths and interatomic distances.

1 Introduction

1.1 What is Conformational Analysis?

Conformational analysis is an area within chemistry that has taken time to evolve into what it is now. It first began in the 19th century with the development of structural theory¹ by August Kekulé, an organic chemist known as the creator of the Kekulé structure of benzene. Kekulé and others pushed the idea of constitution, specifying which atoms were connected to the other. This idea led to the necessity of stereochemistry and configuration because molecules could have the same connectivity but different projections in space. Le Bel and Van't Hoff were pioneers on this front, explaining that a molecule's point in space coincides with its given optical activity²⁻⁴ compared to its counterparts.

The word “conformation,” defined as any one of the infinite numbers of possible spatial arrangements of atoms a molecule can have, began to be used in the early 1900s. Barton and Cookson spread the idea that although there was an infinite number of possible arrangements, only a few were *energetically preferred*⁵. They also explained that these less energetically preferred conformers could be isolated by lowering the temperature. The two English chemists gave reasons for the existence of preferred conformations, such as repulsive intramolecular interactions, and shed light on the importance of understanding a chemical structure past its initial conformer. They stated that a molecule's preferred conformation, physical properties such as IR and UV absorption bands, chemical effects dominated by steric hindrance, and overall geometrical requirements of transition states could all be investigated deeper by conformational analysis⁵.

An example from this time that ratified Barton and Cookson's statements on energetically preferred conformations was the conformational investigation of cyclohexane. Before its investigation, the distinction between the chair and boat cyclohexane conformations had little to no importance to chemists. However, Hassel investigated the molecule and found that the 'chair' conformation was considerably preferred to its 'boat' counterpart^{5,6}. This was due to the large H-H distances (2.5 Å) found in the conformer, a distance twice the van der Waals radius of hydrogen⁷. On the other hand, Boat conformers consistently had smaller H-H distances, resulting in a less energetically favorable conformation. At the time, this finding was crucial to the burgeoning steroid chemistry field because it began the relationship between single and fused ring conformational analysis and accentuated their similarities. Barton championed this, uncovering direct relationships between configuration and conformation for single rings and their steroid and steroid-like counterparts.⁸ For example, the steric hindrance trends found in the conformational analysis of cyclohexane derivatives remain for steroids, as does the preference of axial/equatorial location for substituents⁸. Analyses also gave credence to rearrangements in steroids that could release steric compression when forming products. Since steroids are fused rings, their conformation becomes fixed. Understanding the possible steric hindrances in one conformation allowed chemists to better predict the product in their reactions and increase the overall understanding of steric hindrance in molecules as a whole⁹.

1.2 Modes of Performing Conformational Analysis

Conformational analysis has been performed in a myriad of ways since its inception (Figure 1). These techniques are in two categories: experimental and computational. Conformational analysis began with experimental techniques, but as the exponential increase of computational power took place, many computational methods were created to substitute or expedite older

practices. Experimental approaches include gas-phase electron diffraction (GED), electronic circular dichroism (ECD), vibrational circular dichroism (VCD), and nuclear magnetic resonance (NMR). GED was the first powerful technique used in the 1960s to analyze conformers, most notably for cyclohexane^{6,10}. GED's downside at the time was the challenging interpretation of data¹¹. This is because the initial data is an intensity curve consisting of damped sine-waves, where the position of each peak is determined by the length of the distance between each pair of atoms, r_{ij} , the width of the peak is determined by the root-mean-square vibrational amplitude, l_{ij} , and the area under the peak is approximately proportional to $Z_i Z_j / r_{ij}$, where Z are atomic numbers¹². This once challenging calculation and interpretation has since been aided with the help of computational hardware.

ECD and VCD were also techniques used in the 1960s. Circular dichroism, which uses the difference between the absorption of left and right circularly polarized lights in molecules, can provide information on overall molecular stereochemistry, conformation, and configuration with high sensitivity¹³. CD has been used to investigate the conformation of larger moieties such as biomolecules¹⁴⁻¹⁶. NMR is widely accepted as one of the most used tools for analyzing small- and medium-sized organic molecules since its invention. Its use in carbohydrate chemistry in the 1970s proved to be imperative to understand hexulose and hexulose derivatives¹⁷. NMR is a prime example of how improvements in technology have pushed the field of conformational analysis forward. Compared to when it was first introduced, NMR sensitivity has increased by more than four orders of magnitude¹⁸. This is a crucial factor that allows it to still be such a powerful tool many years after its creation. Its extreme sensitivity of chemical shifts and their subsequent conformational change, as well as its ability to use ¹H, ¹³C, and other isotopes to

perform its analysis are just a few unique features that allow NMRs relevance to continue in a world of computational conformational analysis.

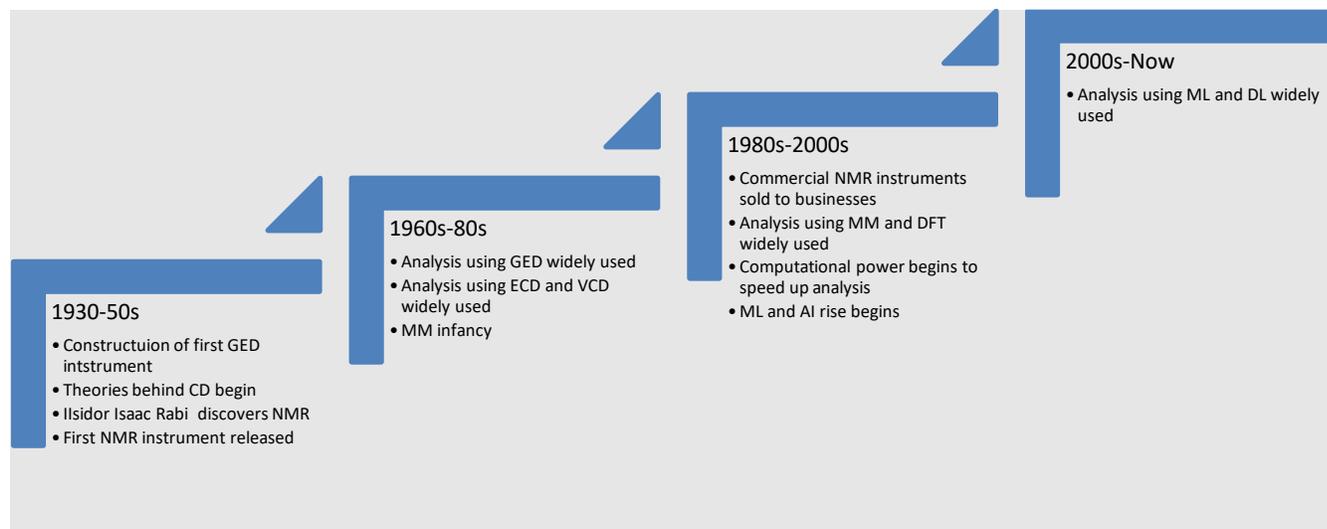


Figure 1-1 Conformational Analysis Technique Timeline

Computational techniques that can be used for conformational analysis include molecular mechanics (MM), Quantum Mechanics (QM), Machine Learning (ML), and Deep Learning (DL). The first MM-based conformational search approach began in the 1960s. At the time, it was a simpler and more efficient way to analyze small organic molecules. Additionally, as commercial PCs became less expensive, MM permitted researchers to conduct conformational analyses without the need of supercomputers. This allowed MM to dominate the late 20th century. The development of molecular biology and biophysics in the 1960s required a more powerful tool to analyze the conformational space of large systems. MM simulations solved the construction of 3D structures of macromolecules and predicted the pathways of protein folding and refinement of experimental structures¹⁹. MM also allowed for researchers to carry out conformational searches to find all low-energy conformers. Conformational searches work by rotating bonds within a molecule of interest, finding local minima. By repeating the rotations

throughout the molecule, conformations of varying energies are produced. The accuracy and speed at which MM can produce conformations of a molecule with a wide energy window are why it is one of the most useful tools for conformational analysis. However, MM for conformational analysis does have its shortcomings. One's conformational search is only as good as the force field used. If a force field used does not encase the necessary accuracy one needs for the given atoms, the results will be subpar^{20,21}, or not work at all. This is where QM improved on MM.

QM can manage larger molecules at higher—albeit affordable—computational cost with higher accuracy and has evolved to be available in many high-performance servers and software. Overall, QM provided more reliable results than MM and could be used with more complex molecules resulting only in additional computational cost. QM has since been the gold standard in producing the accurate structures necessary for conformational analysis, as well as computed energies used to separate structures from each other. The library of methods and basis sets now at chemists' disposal allow for extremely accurate optimizations that are curated based on the molecule in question and the researcher's goal. ML and DL first began to see use in conformational analysis in the 1990s and started to take off in the 21st century. Of the many techniques available, neural networks, clustering, and dimensionality reduction applications are the most widely used. This is due in part because of the large amount of data necessary to complete accurate conformational analysis. Dimensionality reduction techniques such as principal component analysis (PCA) and t-distributed stochastic neighbor embedding (t-SNE) have both been used as a means of reducing the complexity and dimension of molecular data^{22,23}, while clustering provides visual analysis explaining similarities found within groups and neural networks can predict similarities between structures. Compared to all other methods discussed,

ML and DL are the most accessible conformational analysis techniques, as all it takes is a computer equipped with Python.

1.3 Current State of Conformational Analysis Field

Conformational analysis continues to be one of the most challenging jobs chemists must do in order to understand and conduct chemistry-related research. For a chemist to explore a given molecule, one attempts to find all of its potential arrangements in three-dimensional space. This action ensures an exhaustive conformational analysis. In doing so, a multitude of conformations are created for the system of study. Making sense of each conformation is no simple feat. Side-by-side visual comparisons of each conformation may help in the grouping structures, but differences between conformational families vary, and are easily missed even for the trained eye. Additionally, the more flexible the chemical system, the greater the number of conformations. Thus, the tougher it becomes to find distinguishing and meaningful structural and electronic features, and the more costly and tiresome the task at hand becomes. Computational approaches in conformational analysis can also suffer from the same problems. Size and flexibility of a system force algorithms to incorporate more data without any guarantee that its inclusion is important for accurate analysis. Thus, computational investigation of molecular structures and reactions of biological and pharmaceutical interests remain a grand scientific challenge due to these systems' size and conformational flexibility. The investigation of these structures requires parsing and analyzing thousands of conformations in each molecular state for meaningful chemical information and subjecting the ensemble to costly quantum chemical calculations. The current status quo is a time-intensive and tedious process that is infeasible in some cases, limiting the ability of theoreticians to study these systems.

Since their initial uses in chemistry-related research, ML and DL look to solve grand scientific challenges at hand. Conformational analysis using ML techniques neural networks²⁴⁻²⁶, trees/forests^{27,28}, clustering^{29,30}, and principal component analysis³¹⁻³⁴ (PCA) have been seen in literature. ML and DL have also been useful in other chemistry-related research, including investigations of drug design/catalysis³⁵⁻³⁹, activation energies⁴⁰, and transition states⁴¹. ML's ability to create automated and semi-automated data analysis methods creates compelling use cases, allowing chemists to produce more results than before. The speed at which these results are produced can be markedly faster than previous techniques⁴²⁻⁴⁴. In this work, clustering and PCA were used.

1.4 My Goal in Conformational Analysis Field

The aim was to develop code that will automate conformational analysis of flexible, chemical systems. In doing so, I created a methodology that answers how the code was going to successfully complete its objective. The first question answered was what was going to be extracted (data-wise) to complete the conformational analysis of ensembles. We knew that each in every structure, each atom had a specific three-dimensional (also known as XYZ-) coordinate, and since every coordinate was known, we could calculate the distances, angles, and dihedrals between atoms.

Once we knew what we wanted to extract/calculate, the next question was how we planned to stratify the conformational ensembles. We decided to use PCA and K-means clustering as a way to reduce dimensionality of data, stratify ensembles and classify conformational families. As stated in section 1.3, PCA and clustering specifically have been utilized to conduct conformational analyses and classify conformational families found within structures with

success. PCA is a valuable ML tool that reduces the dimensionality of data while simultaneously minimizing information loss. It does this by creating principal components, linear combinations of the initial variables orthogonal to one another. Clustering is an unsupervised learning algorithm that finds structure or classification in unclassified data. In our case, K-means clustering's goal is to create classified families by minimizing the variation within each cluster. Existing uses of PCA and clustering vary. Current PCA literature uses it to tease out molecular dynamics within large lipids³³, large molecules³³, or proteins^{34,45-47}, and have not been used in a generalizable fashion. Additionally, there are no programs that are able to find conformational families within an ensemble of structures in a rapid and automated fashion. I would like to build on this existing work to create a generalizable program that uses PCA and clustering to classify conformational ensembles into their respective conformational families with speed and accuracy that is currently not seen in the field.

2 Methods

For the program to work, it is imperative that the structures to be analyzed are of XYZ file type. The XYZ file format is a chemical file format that specifies the molecular geometry by giving each atom a location in three-dimensional space via Cartesian coordinates. The first line of the file gives the number of atoms, the second line is the name of the molecule, and the remaining lines describe the atomic coordinates for each atom.

2.1 Software

General hardware requirements include a computer that runs MacOS X 10.9+ or Windows with at least 512 MB of RAM. Software requirements include PyMOL, Python 3.0+ and all python

packages described in section 2.2. Users access the program through [GitHub](#). They may download the files as a zip or clone the files if they have git commands downloaded on their computer. Downloading git commands are linked [here](#). Once the program is downloaded, the user should open a terminal and change directories until within the software scripts. Python⁴⁸ and Python-related packages⁴⁹⁻⁵² were used to create the software. Users can run the python script with the command “python3 main.py” or “python main.py”. Directions regarding running the analysis are built into the program. Users can (1) use PCA to reduce the dimensionality of data and find all conformational families within conformational searches, (2) print out explanations as to how conformational families were separated by way of most important features for each principal component, (3) separate structural files (ex. XYZ) by conformational family on computer filing system, and (4) visualize conformational families using PyMol⁵³.

2.1.1 Packages Used

<i>Packages Used</i>	<i>Purpose</i>	<i>Location</i>
<i>os</i>	Interfacing with operating system	pymol.py
<i>shutil</i>	Interfacing with operating system	pymol.py
<i>subprocess</i>	Invoking PyMOL software open	pymol.py
<i>sklearn.decomposition</i>	Principal component analysis	pca.py
<i>NumPy</i>	Data structures and calculations	pca.py, elements.py, xyz.py
<i>plotly.express</i>	Interactive plots	pca.py
<i>pandas</i>	Data structures	pca.py, xyz.py
<i>stat</i>	Interfacing with directory	main.py

Table 1 Python Packages Used

2.1.2 PyMOL Integration

PyMOL is a comprehensive molecular visualization software for rendering and animating 3D molecular structures. Integration of PyMOL consists of the user being able to interact and view the analyzed structures in real-time. A PyMOL session can be started after selecting the desired family to be visualized. Folders for each conformational family are found in the same directory

as the original XYZ files. All features of PyMOL can be done on structures. *Note:* It is necessary to have PyMOL software already installed on a computer for integration to work correctly. To install PyMOL, download the installer to your computer via the [PyMOL website](#).

2.2 Algorithm

The goal of this algorithm is to find and classify conformational families within a given set of structures using PCA and clustering. There are requirements for the algorithm to work. (1) The XYZ files must be structural isomers with the same chemical formula, and (2) The same atom across all isomers must be numbered identically (i.e., same atom ID). Each of these requirements ensure that the atom-to-atom standardized comparisons between conformers are occurring only. Major software such as Schrodinger Maestro satisfies the given criteria. Structures to be analyzed start as XYZ files in a folder.

```
p8
t-amy1OH
C      0.41780      -0.34820      -0.39180
H     -0.63100     -0.42630     -0.08970
H      0.57520     -0.97970     -1.27090
H      1.03090     -0.75500      0.41920
C      0.81350      1.09840     -0.66510
H      1.86770      1.12740     -0.96650
H      0.73750      1.64080      0.28630
C     -0.05550      1.81070     -1.72050
O     -1.42050      1.79960     -1.29750
C      0.00610      1.12320     -3.08530
H     -0.58000      1.67960     -3.82630
H      1.03500      1.04370     -3.45040
H     -0.43360      0.12080     -3.04960
C      0.38320      3.27210     -1.85790
H     -0.26440      3.81010     -2.55980
H      0.30090      3.79720     -0.89940
H      1.41750      3.35120     -2.20820
H     -1.46900      2.23730     -0.43040
```

Figure 2-1 Example XYZ file

These XYZ files are then parsed, creating a data frame consisting of cartesian coordinates for each atom on a column-by-column basis and each structure on a row-by-row basis. Files are parsed line-by-line in the program, creating element-specific objects for each atom read. Depending on the type of atom read, the atom object created has specific ranges for what it considers a bond or interaction. For example, a carbon object will accept 2.10 Å from another atom object as a bond, while a nitrogen object will not. Atomic distance min/max settings are

described in Appendix B. Atom objects also include the atom number and cartesian coordinates. After each line in the XYZ file is read and creates its atom object, a data frame containing all atoms and their cartesian coordinates for every conformer read is created.

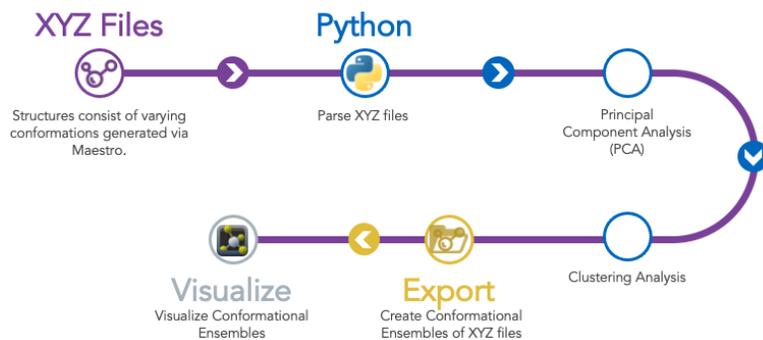


Figure 2-2 Conformational Analysis Algorithm Flowchart

The data frame can be manipulated to calculate all inter-atom distances, angles, and proper dihedrals. The result produces a data frame tailored to the user's desire, with the ability of including the aforementioned chemically relevant calculations.

	Bonds						Dihedrals			
	C ₁ -O ₁	C ₁ -O ₂	C ₁ -C ₂	C ₂ -O ₁	C ₂ -O ₂	C ₂ -C ₃	...	C ₁ -C ₂ -C ₃ -C ₄	C ₁ -C ₂ -C ₃ -C ₅	...
tamyOH_1.xyz										
tamyOH_2.xyz										
...										
tamyOH_10.xyz										

Figure 2-3 Example Data Frame

2.2.1 Inter-Atomic Calculation Functions

Inter-atom distance calculations take the following form:

$$Distance_{a,b} = \sum \sqrt{(b - a)^2}$$

Where a and b are the cartesian coordinate vectors of atoms in question. In the program, NumPy is used to store each cartesian coordinate vector via arrays and find the Euclidean distance via the norm function. Angles are calculated in a similar manner:

$$Angle_{a,b,c} = \cos^{-1} \frac{(b - a) \cdot (c - b)}{\sum \sqrt{(b - a)^2} \cdot \sum \sqrt{(c - b)^2}}$$

$$Angle \text{ (degrees)}_{a,b,c} = (Angle_{a,b,c}) \left(\frac{180^\circ}{\pi} \right)$$

Where a , b , and c are the cartesian coordinate vectors of each atom. In addition to its same responsibilities when calculating distances, NumPy is used to calculate the angles using an arccosine function. When calculating dihedrals, only proper dihedrals can be calculated. Proper dihedral angles are defined as the angle between ijk and jkl planes, where i , j , k and l are atoms in the molecule being investigated and atom i is covalently bonded to j , j is covalently bonded to k , and k is covalently bonded to l . Only proper dihedrals were calculated to optimize the algorithm to only calculate chemically relevant parameters. Proper dihedrals are also backed by the foundations of organic chemistry; if non-proper dihedrals were calculated, understanding their importance (or lack of) could not be done using chemical principles. To ensure only proper dihedrals completed, a series of checks to ensure that the four atoms currently selected are within distance of each other to be described as bond (Appendix C, xyz.py, lines 372-399). Those that qualify are calculated in the form below:

$$Dihedral_{a,b,c,d} = \cos^{-1} \frac{-[(b - a) \times (c - b)] \cdot [(b - c) \times (d - c)]}{\sum \sqrt{[(b - a) \times (c - b)]^2} \cdot \sum \sqrt{[(b - c) \times (d - c)]^2}}$$

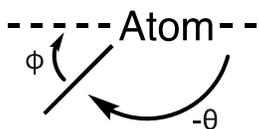
Additionally, if $Dihedral_{a,b,c,d}$ produced an angle between 90° and 180° , or between $(-)90^\circ$ and $(-)180^\circ$ it was modified.

$$Dihedral_{a,b,c,d} > 90^\circ \text{ modified to } 180^\circ - Dihedral_{a,b,c,d}$$

Or

$$Dihedral_{a,b,c,d} > -90^\circ \text{ modified to } -180^\circ - Dihedral_{a,b,c,d}$$

Modifications were completed to ensure all dihedrals can be properly compared across all conformations. This is because a dihedral angle of 100° is the same as -260° . Although the modification turns real dihedrals into mock dihedrals, the results produce an output that verifies a positive dihedral has an equal negative reciprocal. In turn, this modification is not problematic, rather, it ensures that each dihedral, whether negative or positive, will be compared in a standardized fashion.



Both ϕ and $-\theta$ equate to the same angle

Figure 2-4 Positive Angles have Negative Equal

All inter-atomic distances, angles, and dihedrals are calculated at the user's discretion.

2.2.2 PCA

PCA's general objectives are data/dimensionality reduction and interpretation. Principal component can be described algebraically as particular linear combinations of p random variables X_1, X_2, \dots, X_p . Geometrically, these linear combinations create a new coordinate system based on the X_1, X_2, \dots, X_p axes. These X axes represent the directions with maximum variance found within the original data. It is also known that principal components depend on the covariance matrix Σ of X_1, X_2, \dots, X_p . To create the covariance matrix, we must take a matrix of observations (m) and samples (n) to make an $m \times n$ data matrix. For example, in this research the observations would be each distinct structure to be conformationally analyzed and each sample would be a cartesian coordinate, inter-molecular distance, angle or dihedral. In an effort to seek how the variance of one sample correlated with the variance of another sample, the covariance must also be calculated:

$$Cov(x, y) = \sum_{i=1}^N \frac{(x_i - \bar{x}_i)(y - \bar{y}_i)}{N - 1}$$

Where x_i is a single observation of one sample, \bar{x}_i is the mean of all the observations of that sample, y_i is a single observation of another sample, and \bar{y}_i is the mean of all the observations of that sample. This allows us to define all our principal components in the following form:

Principal Component 1 = $\mathbf{a}'_1\mathbf{X}$ that maximizes variance($\mathbf{a}'_1\mathbf{X}$)

subject to $\mathbf{a}'_1\mathbf{a}_1 = 1$

Principal Component 2 = $\mathbf{a}'_2\mathbf{X}$ that maximizes variance($\mathbf{a}'_2\mathbf{X}$)

subject to $\mathbf{a}'_2\mathbf{a}_2 = 1$ and $Cov(\mathbf{a}'_1\mathbf{X}, \mathbf{a}'_2\mathbf{X}) = 0$

Principal Component i = $\mathbf{a}'_i\mathbf{X}$ that maximizes variance($\mathbf{a}'_i\mathbf{X}$)

subject to $\mathbf{a}'_i\mathbf{a}_i = 1$ and $Cov(\mathbf{a}'_i\mathbf{X}, \mathbf{a}'_k\mathbf{X}) = 0$ for $k < i$

It is important to note that each principal component explains the *maximum* amount of variance in its dimensional space. This allows for the best stratification as possible. It is also important that principal components are orthogonal to each other (i.e. covariance = 0, as stated above). If not, explained variances between principal components would overlap, resulting in overall less total explained variance and possibly worse stratification. The PCA function in the scikit-learn python package reduces the dimensionality of the data given and finds the number of principal components the user desires.

2.2.3 Clustering

Clustering analysis takes on the task of dividing up data points into a number of groups such that data points in the same group are more similar than data points in other groups. In doing so, clustering analysis can 'label' data that was not previously defined categorically. There are three main types of clustering: (1) connectivity/hierarchical clustering, (2) density clustering, and (3) centroid clustering. Connectivity/hierarchical clustering starts by treating each point as if each were its own cluster, and then finds the cluster that is closest in distance, this continues until each cluster has become one large cluster, and the user can decide where to 'cut the connectivity' define each as a separate cluster. Hierarchical clustering excels at finding embedded structures/families within data. However, for our purposes hierarchical clustering would be more costly approach, since each point (in our case, conformer) would start as its own family.

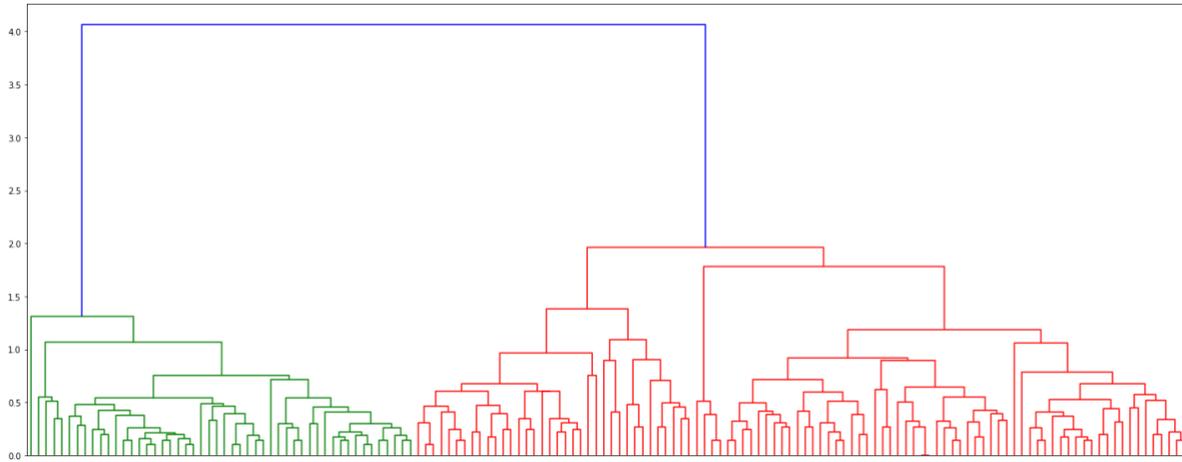


Figure 2-5 Example Hierarchical Clustering Plot

Density-based clustering looks at points that are tightly packed and classifies them accordingly. By calculating the radius of one point to another and the number of points that are at within that given radius, clusters of data are created. Density-based clustering excels at finding an unknown number of clusters of similar density.



Figure 2-6 Example Density-Based Clustering Plot; Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

Although both hierarchical and density-based clustering have their strengths, both do not excel at finding a ‘consensus’ or average data point that can generally explain most points found within the cluster. That is why centroid clustering is the best type of algorithm for the data used in this

program. For centroid clustering techniques such as k-means, the labeling of data points into groups is done mathematically using centroids. A centroid is defined as the average of all data points found within one group. Since data points that are cluster to each other are more similar, centroids will move iteratively in accordance with its surrounding, decreasing the absolute distance to all of its assigned data points. This iterative process stops once centroids can no longer decrease the additive distance from its assigned points, thus being stabilized. K-means clustering does this is the steps below:

1. Partition the items (in our case, conformers) into K initial clusters
2. Assign each item to the cluster whose centroid is nearest via Euclidian distance.

Recalculate the centroid for each cluster receiving a new item and for each cluster losing an item.

3. Repeat step 2 until no reassignments take place and the centroids stabilize.

Minimization of intra-cluster variation is described in following form:

$$\min_{c_1, \dots, c_k} \left\{ \sum_{p=1}^k \frac{1}{|c_p|} \sum_{j \in c_p} (\vec{x}_i - \vec{x}_j)^2 \right\}$$

Where c are the clusters, $|c_p|$ is the number of elements in the p^{th} cluster, and $(\vec{x}_i - \vec{x}_j)^2$ is the L_2 Norm that results in the computation of length in Euclidean space.

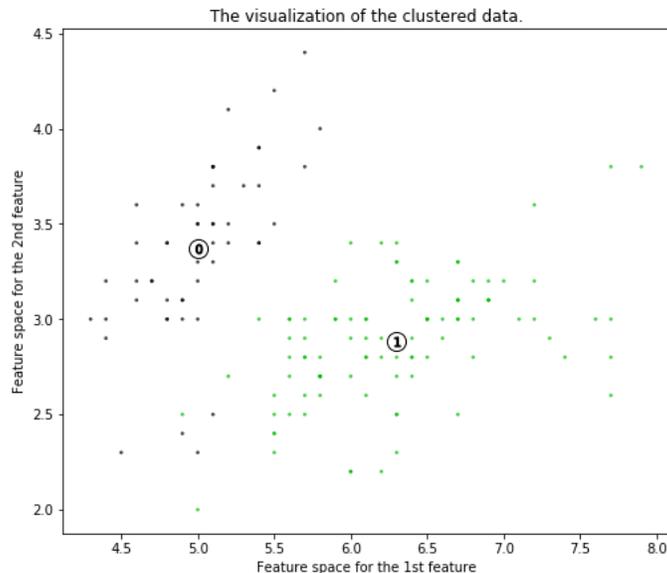


Figure 2-7 Example Centroid Clustering Plot with Centroids 0 and 1

2.3 Installation and How-To

Python is necessary to run program. To install Python, go to python.org/downloads/, and download the latest version. During the installation process, be sure to click the option to add Python to PATH. The program works with Python 3.0+ versions. External python libraries necessary to run program are scikit-learn, pandas, and plotly, and NumPy. Additionally, PyMOL can be used for visualization of families. The easiest way to use the program is to install it as a package using pip, a standard package manager for Python that allows you to install and manage additional packages that are not part of the Python standard library.

To install pip, go to <https://bootstrap.pypa.io/get-pip.py>, and download the file on the page selecting 'save page as..' on the browser. The name of the file should be 'get-pip.py'. Open the computer's command terminal, change directories to the folder where the file was downloaded, and run the following command:

```
python get-pip.py
```

At this point, Python and pip are installed. To install this package, enter the following code in the command terminal:

```
pip install omolab-conf-analysis
```

To use the program, open a python file, import, use, and run the package using the following python code:

```
Import ConformationalAnalysis  
Test1 = ConformationalAnalysis()
```

A second method to install the software is to pull the project from GitHub. Using the command line, change the directory to a location where you would like the software to be downloaded.

Next, enter the following command into the terminal:

```
git clone https://github.com/mattnw1/Conformational_Analysis.git
```

Once you are in the folder in which the GitHub package was downloaded, run the script using:

```
python ConformationalAnalysis.py  
  
or  
  
python3 ConformationalAnalysis.py
```

2.4 Testing of Known Conformational Families

Testing my program is essential to ensure that its abilities work as intended, and that its goal of creating an automated way to find conformational families is met. Literature searches of known conformational families were completed. Criteria necessary for being tested included the

following: (1) there must be well-defined conformational families stated and/or depicted within the published paper that cannot be disputed, (2) there are detailed explanations behind each conformational difference found within the structures, and (3) the structures can be recreated using MM. (1) and (2) are crucial because it allows me to know the baseline of what I am testing, and important distinguishing characteristics I should find within the structures. (3) Allows me to recreate the structures for testing. Due to the program's ability to compute distances, angles, and dihedrals for a given molecule, the selected systems will each test one option, as well as the programs ability to use clustering to still find conformational families after dimensionality reduction. Tri-valine peptide (AC-3VAL-NHMe) was used to test distances, as polypeptides form helical structures where stability is greatly affected by hydrogen and nonbonding interactions. Solvated calcium fluoride complex $[\text{CaF}_2(\text{H}_2\text{O})_4]$ was used to test angles. Lastly, cis-1-fluoro-4-propylcyclohexane and 1,2-dicyclohexylethane, were used to test dihedrals since both have the ability to produce separate 'ring flipped' conformational families, a phenomenon that can be explained through the dihedral angles found within a given conformation.

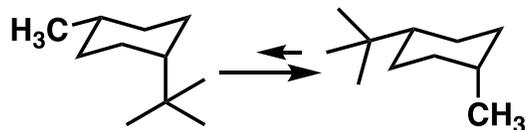
3 Results/Discussion

3.1 Conformational Analysis of Substituted Cyclohexanes

Cyclohexane is a cyclic, non-aromatic organic hydrocarbon comprised of six carbon atoms with a molecular formula of C_6H_{12} . Cyclohexane is used as a nonpolar solvent for the chemical industry, and as a raw material for the industrial production of adipic acid and caprolactam, intermediates used in the production of nylon⁵⁴. Cyclohexanes can be found in four conformations: the chair, the half-chair, the boat, and the twist-boat. The predominant

conformation is the chair, as it reduces the molecules overall torsional strain. Additionally in the chair form of cyclohexanes, hydrogens or other substituents on the ring can either be axial or equatorial⁵⁵. In the most stable structure of a non-substituted cyclohexane, half of the hydrogens are parallel to the ring, or equatorial, while the other half are perpendicular to the plane, or axial. The cyclohexane ring can also be inverted, or flip, from one chair form to another, effectively changing each axial hydrogen into equatorial and vice-versa. This chair-chair interconversion phenomenon is called “ring inversion” or “ring reversal” and proceeds through a transition state to a twist-boat intermediate⁵⁵.

Monosubstituted chair conformations of cyclohexanes produce two conformers of their own, with the equatorial conformation being predominant.⁵⁵ The equatorial conformation is more stable in part due to van der Waals repulsions occurring between the substituent and the two axial hydrogens on the same face of the ring. These 1,3-diaxial interactions destabilize the axial conformation relative to its equatorial counterpart. By way of example, the standard Gibbs free energy difference between axial and equatorial conformations of *tert*-butylcyclohexane is 20 kJ/mol (5 kcal/mol)⁵⁶. This energy difference creates a population in which over 99% of the *tert*-butylcyclohexane are the equatorial conformation. Disubstituted cyclohexanes exist in *cis*- and *trans*- isomers. For 1,4-disubstituted cyclohexanes, each *cis* and *trans* isomer produces its own preference within their respective conformations. *Trans* isomers follow the previous trend of equatorial conformers being preferred. As for *cis* isomers, since one substituent must be equatorial and the other axial, the energetically preferred conformation is dependent on which substituent is most sterically hindered in the axial position.



Enegetically preferred to have steric hindered substituent equatorial

Figure 3-1 Energetic preference of cis-1,4 disubstituted cyclohexanes

3.1.1 Substituted Cyclohexane Test Case: *Cis-1-Flouro-4-Propylcyclohexane*

Conformational analysis of *cis-1-flouro-4-propylcyclohexane* was conducted using PCA and clustering techniques. The conformations of *cis-1-flouro-4-propylcyclohexane* were used to test the dihedrals part of the code and its ability to explain ‘ring flipped’ conformational families at the C1-C4 positions. The two main conformational families in this compound are two forms of a chair conformation: (1) equatorial fluorine, axial propyl disubstituted cyclohexane, and (2) axial fluorine, equatorial propyl disubstituted cyclohexane.

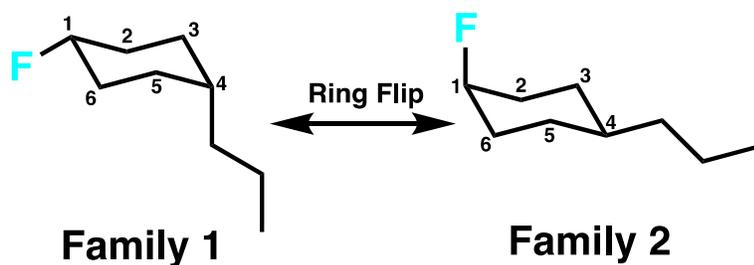


Figure 3-2 Ring Flip Between Two Conformational Families of *Cis-1-Flouro-4-Propylcyclohexane*

The two families follow the aforementioned ring flip, the conversion from one to another requires multiple steps. The first step passes through a half-chair transition state and a twist-boat saddle transition state. Next is the formation of the boat. The boat then undergoes simultaneous internal rotations about all carbon-carbon bonds except those to carbon-1. The result of the is a

second twist-boat that is a mirror of the first. Finally, a second half-chair transition state produces the ring flipped chair conformational family.

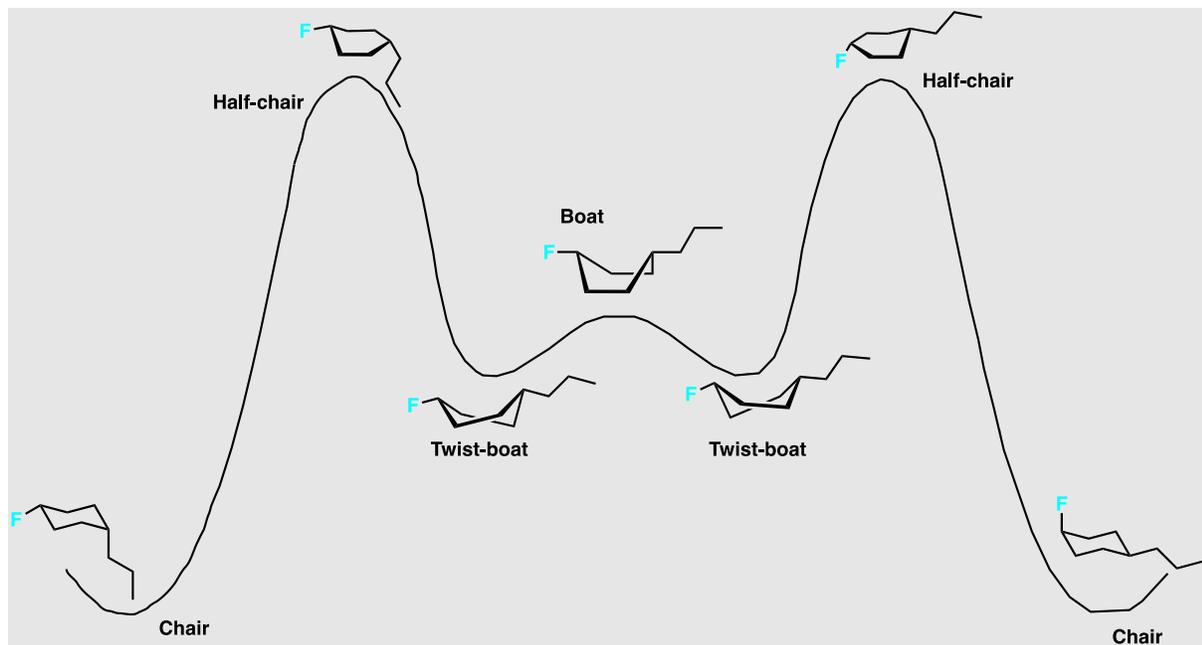


Figure 3-3 2D Cis-1-Fluoro-4-Propylcyclohexane Chair Flip Diagram

However, there are multiple ways for the ring to flip. It is possible for the ring-flip to occur at the C2-C5 positions. This would result in the substituted carbons at the “arms” of the half-chair, twist-boat, and boat conformations. Another possible ring-flip can occur in such a way that produces structures where both substituents are in axial or axial-like positions in the half-chair, twist-boat, and boat conformations. Although these additional structures are less likely to happen due to energetic preferences, it is important to note their possibility when finding conformational families.

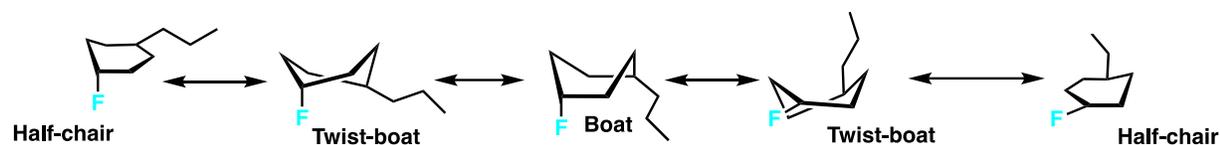


Figure 3-4 2D Cis-1-Fluoro-4-Propylcyclohexane C2-C5 Ring Flip Structures

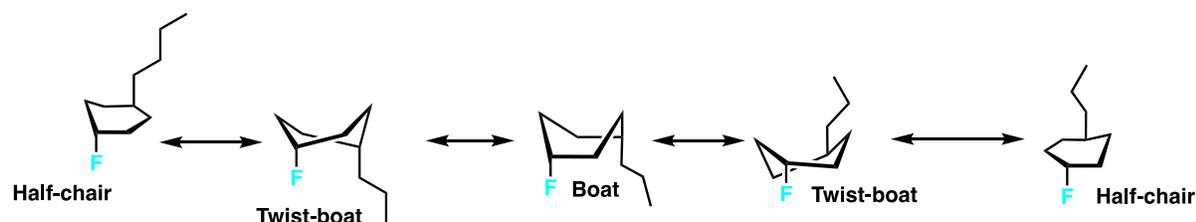


Figure 3-5 2D Cis-1-Fluoro-4-Propylcyclohexane All Axial Ring Flip Structures

We used Schrödinger Maestro to reproduce *cis*-1-fluoro-4-propylcyclohexane structures using molecular-mechanics based conformational searches via Merck Molecular Force Field (MMFF). Mixed torsional/Low-mode was used as the sampling method. The maximum number of steps was set to 10000, the energy window for saving structures was 25.00 kcal/mol, and an RMSD cutoff of 0.05 Å was used for eliminating redundant conformers. The result created 84 structures to investigate. Each structure's XYZ-coordinates were then parsed and placed in a data frame via python functions in xyz.py. Based off the contents in the data frame, calculations of proper dihedrals found within each conformation were computed. The act of parsing each XYZ-coordinate and computing proper dihedrals took the program 16.9 seconds to complete. PCA using two principal components on proper dihedral data produced an explained variance of 27.65% (Appendix-A-1). In order to find the optimal number of clusters, the sum of squares at each number of clusters is calculated and graphed. This allows the user to then choose the number of clusters where there the steep decline in the sum of squares becomes shallow, or the “elbow” of the graph.

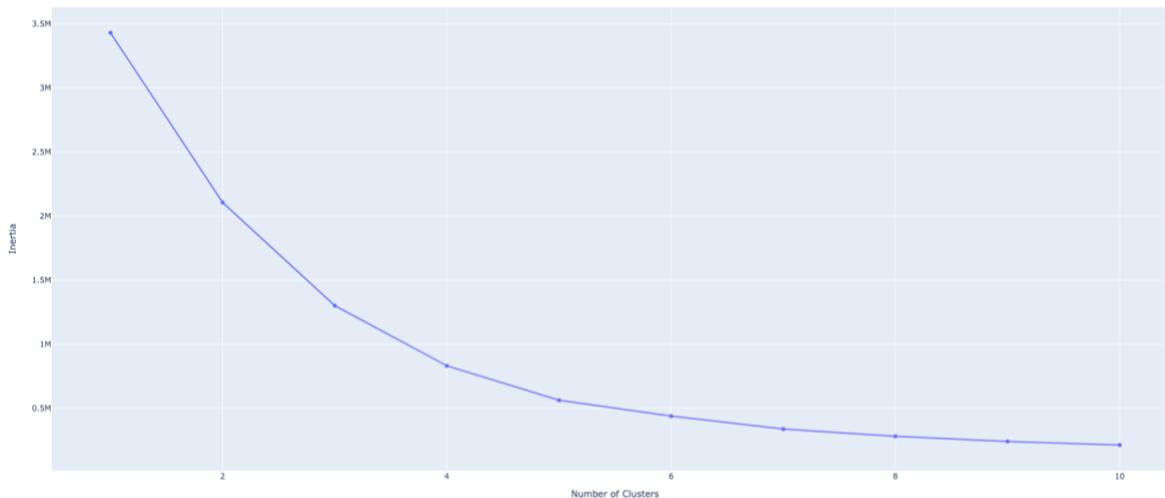


Figure 3-6 Inertia v. # Clusters of Cis-1-Fluoro-4-Propylcyclohexane (Dihedral data only)

In this case, the optimal number of clusters are four and five. When using four clusters, each cluster is split such that cluster one contains the equatorial fluorine, axial propyl family, cluster two contains the axial fluorine, equatorial propyl family, and clusters three and four contain twist-boat intermediates. Cluster three contains mostly twist-boat axial fluorine, axial propyl intermediate structures, while cluster four contains mostly twist-boat equatorial fluorine, equatorial propyl intermediate structures. Additionally, both four contains twist-boat intermediates that are ring flipping at C2-C5 positions rather than C1-C4. For each of these two clusters, the closer to the center of the graph, there were structures that deviated to twist-boat axial fluorine, equatorial propyl structures as well. Cluster four also had one incorrect classification that of the equatorial fluorine, axial propyl family.

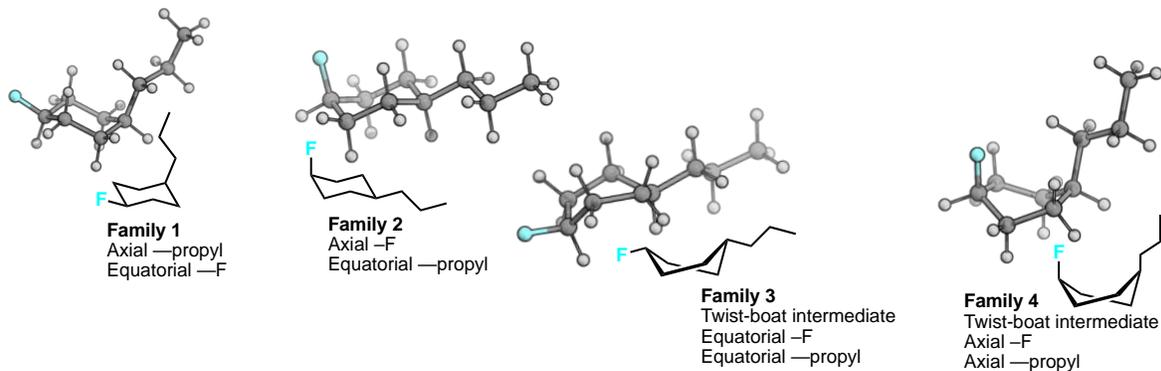


Figure 3-7 2D Clustering Families of Cis-1-Flouro-4-Propylcyclohexane (4 clusters)

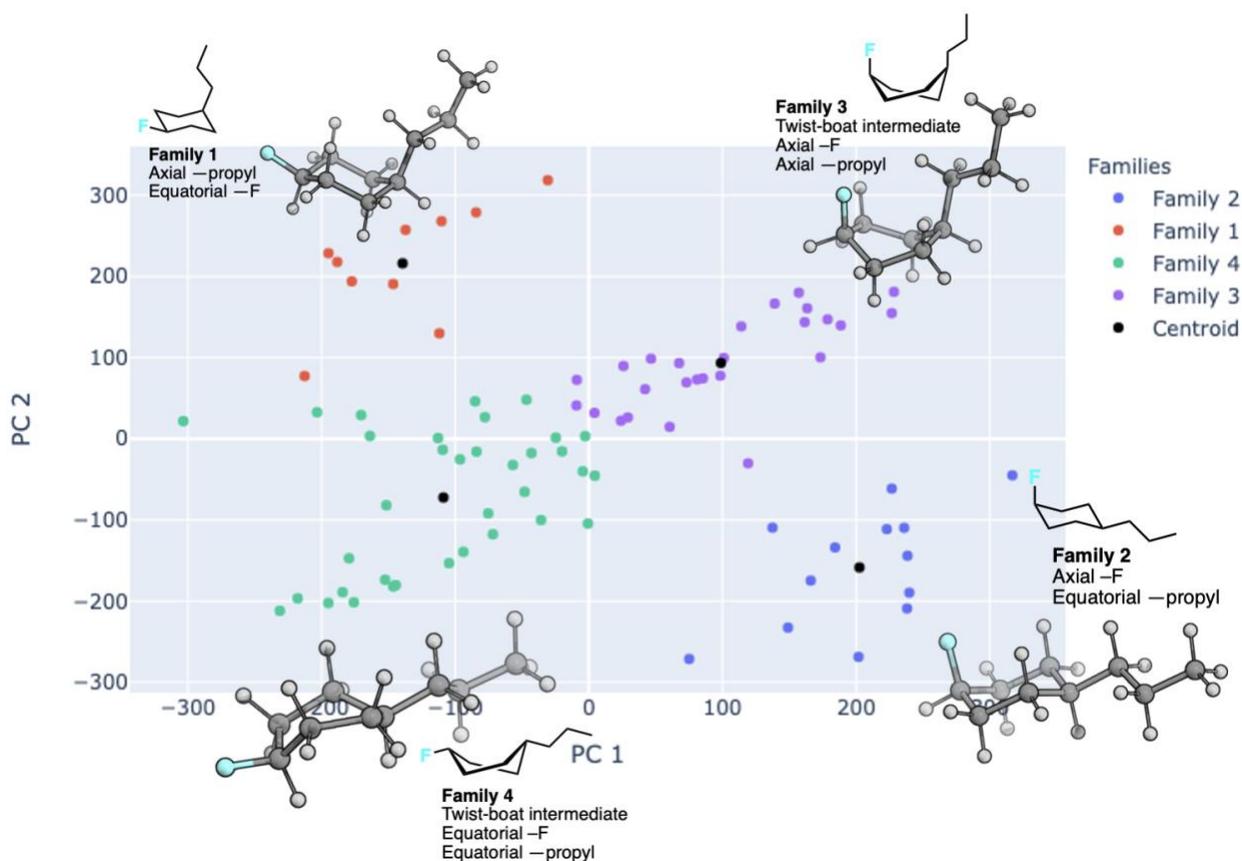


Figure 3-8 2D Clustering of Cis-1-Flouro-4-Propylcyclohexane (4 clusters)

When using five clusters, clusters one and two remained unchanged. In other words, the addition of a cluster did not affect the classification of the equatorial fluorine, axial propyl family, and

the axial fluorine, equatorial propyl family. Instead, the new fifth cluster aimed to reclassify twist-boat intermediate structures. As a result, this led to a similar but finer grained classification, where one cluster contained the twist-boat axial fluorine, axial propyl structures, a second cluster contained the twist-boat equatorial fluorine, equatorial propyl structures, and the final cluster contains a mixture of the two as well as twist boat structures that twist differently than all other conformations. These structures twist one carbon further away from the substituents than the previous structures, creating a subclass of its own. With respect to C2-C5 ring flipping twist-boat intermediates, those structures were only seen in the first clusters, and not the final cluster.

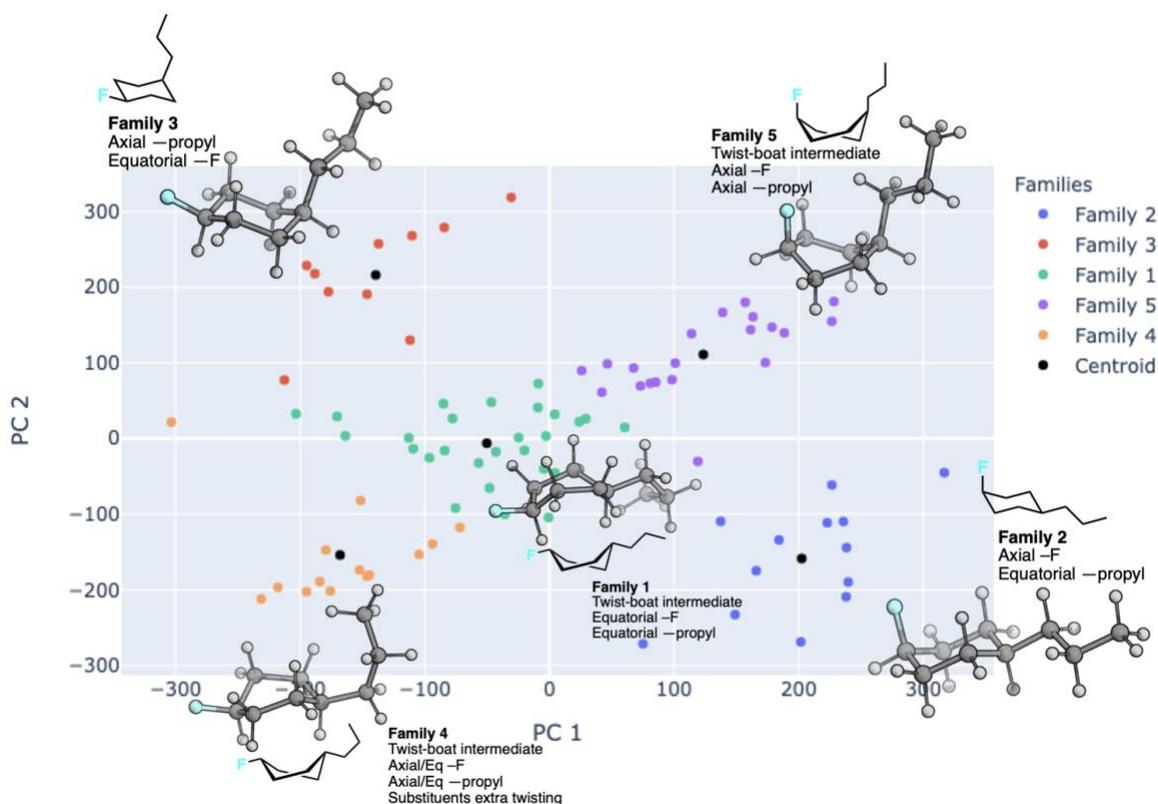


Figure 3-9 2D Clustering of Cis-1-Fluoro-4-Propylcyclohexane (5 clusters)

Overall, the use of two principal components and clustering was able to correctly identify the two ring-flipped chair conformational families of *Cis*-1-fluoro-4-propylcyclohexane, the two

types of twist-boat structures known to be intermediates across a cyclohexane ring flip, as well as different intermediates twisted in varying fashions.

3.2 Conformational Analysis of Tri-Valine Peptides

Polypeptides are defined as a polymer of amino acids, connected to each other by peptide bonds. Peptides have many uses in drugs, as well as endogenously within the body. Endogenously formed peptides, however, have limited availability. Efforts in expanding the uses of peptides as therapeutic targets led to the development of mimetic peptides—peptides that biologically mimic active ligands of hormones, enzyme substrates and other biomolecules. These peptides were created as organic replacements for scarce peptides. Many of these mimetic peptides often look identical to their naturally occurring peptide, with exception of one or two residues replaced by an organic compound⁵⁷, with the goal of retaining major conformational foundations.

Conformational analysis of tri-peptides and their mimetics have been of interest for their medicinal uses as drug-like target molecules⁵⁸ in recent years due to their secondary structure and metabolic stability. These analyses resulted in the finding that the most stable conformations of tri-peptide structures are helical⁵⁹. This result holds true for tri-valine peptides⁶⁰, one of the systems I investigated. I also investigated a simpler peptide, tri-glycine, to investigate how complexity found within the peptide may affect the results. Families found within tri-valine and tri-glycine peptides are alpha-helices, beta-turns (reverse turns), and extended conformations. Alpha-helices conformations are coiled and stabilized by hydrogen bonds between the carbonyl oxygen and an amino hydrogen of residue that is four down the chain.

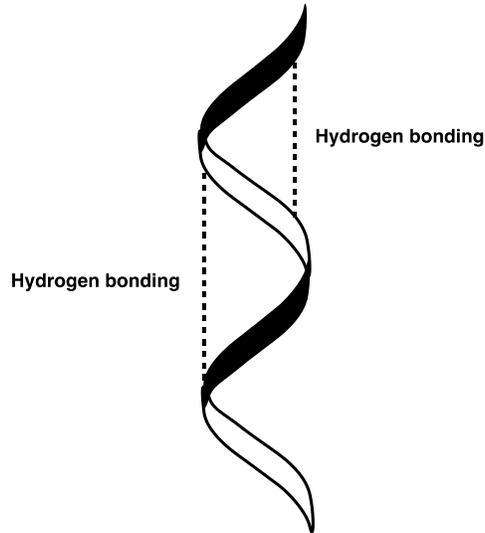


Figure 3-10 Example of alpha-helices with hydrogen bonding

Beta-turn conformations are not coiled, instead, hydrogen bonds form between a carbonyl oxygen and amino hydrogen that is three residues down the chain. The resulting structure looks more cyclic than its helical counterpart. Beta turns can also be divided into two classes. Type I and type II beta turns differ by 180-degree rotation around the bond linking residues two and three.

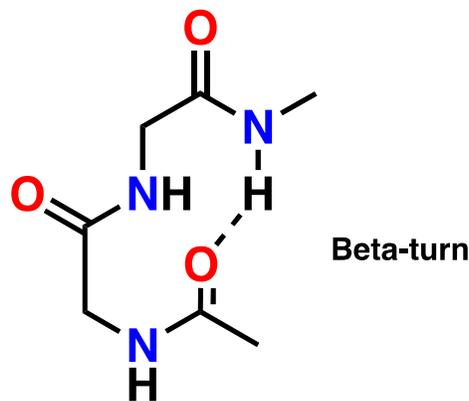


Figure 3-11 Beta-turn example

Extended conformations have 180-degree dihedral rotations around all amines and carbonyls, resulting in a very upright structure with non-bonding interactions between a carbonyl oxygen

and an immediately adjacent amino hydrogen. Extended conformations of peptides are often neglected due to their small proclivity to form in nature.

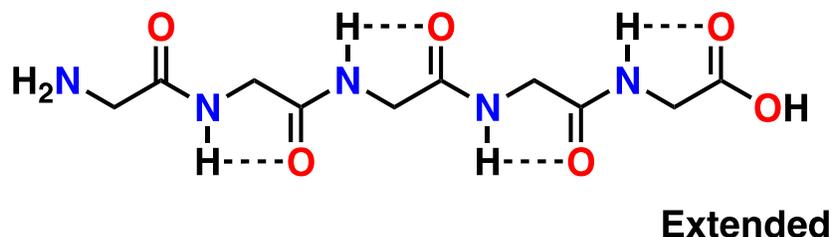


Figure 3-12 Extended conformation example

3.2.1 Tri-Glycine Peptides Test Case: Ac-(Gly)₃-NHMe

In order to see if my program could first deal with a simpler tri-peptide, Ac-(Gly)₃-NHMe was investigated. Molecular-mechanics based conformational searches via OPLS 2005, mixed torsional/Low-mode as the sampling method, the maximum number of steps was set to 1000, the energy window for saving structures was 25.00 kcal/mol, and an RMSD cutoff of 0.05 Å. This produced in 106 structures. The conformations of Ac-(Gly)₃-NHMe were used to test specifically the distances part of the code and its ability to explain conformational families through nonbonding interactions found within small peptides. This case was investigated using two fashions (1) to take all atomic distances, (2) to take atomic distances from just oxygen, hydrogen and nitrogen atoms found in the peptide into consideration for calculated quantities. Each case had their data dimensionality reduced using two principal components. Each produced explained variances of 81.4% and 78.3%, respectively. With such high explained variance percentages, one should expect much cleaner conformational families. This was found to be true. Finding the optimal number of clusters by calculating the sum of squares at each number of clusters and reading the “elbow” of the graph resulted in four clusters.

Inertias for choosing best number of clusters

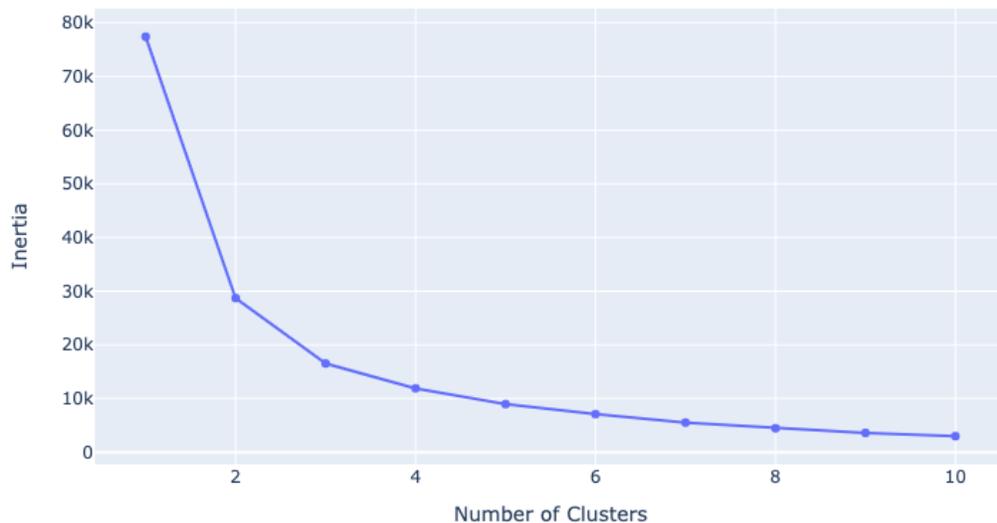


Figure 3-13 Inertias v. # of Clusters of Ac-(Gly)₃-NHMe (all atoms; distances only)

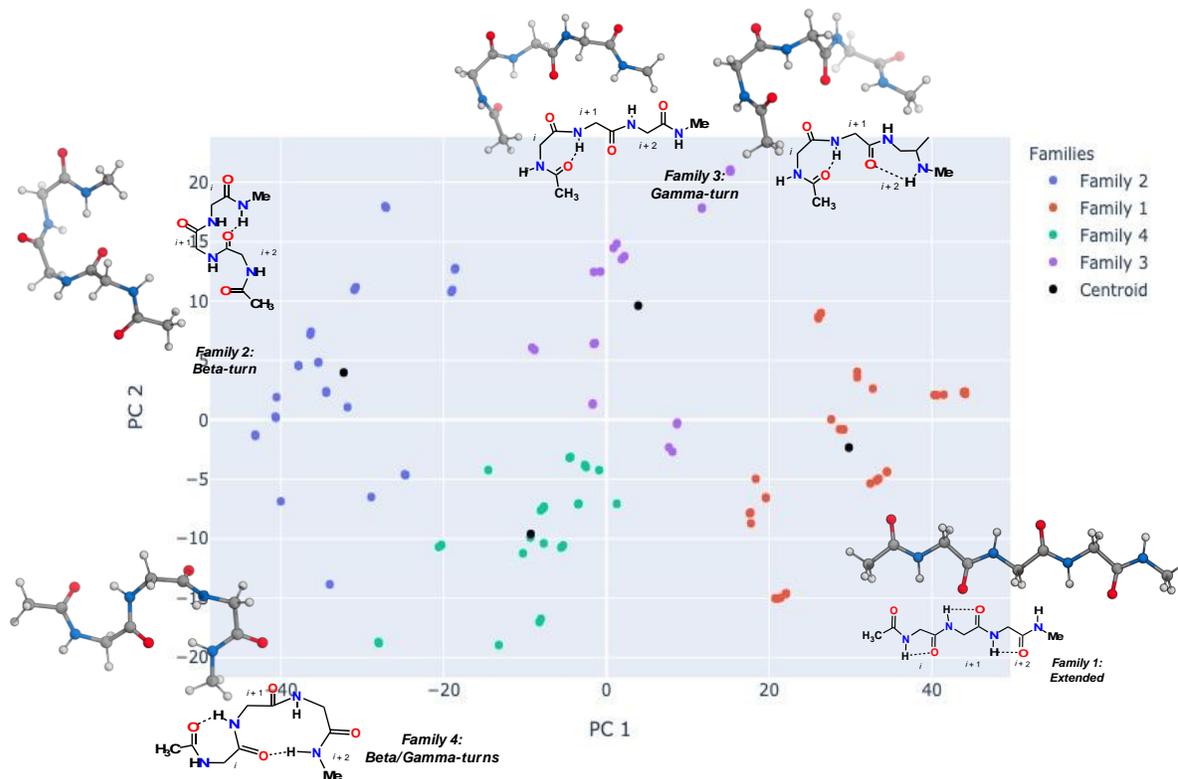


Figure 3-14 Clustering of Ac-(Gly)₃-NHMe (all atoms; distances only)

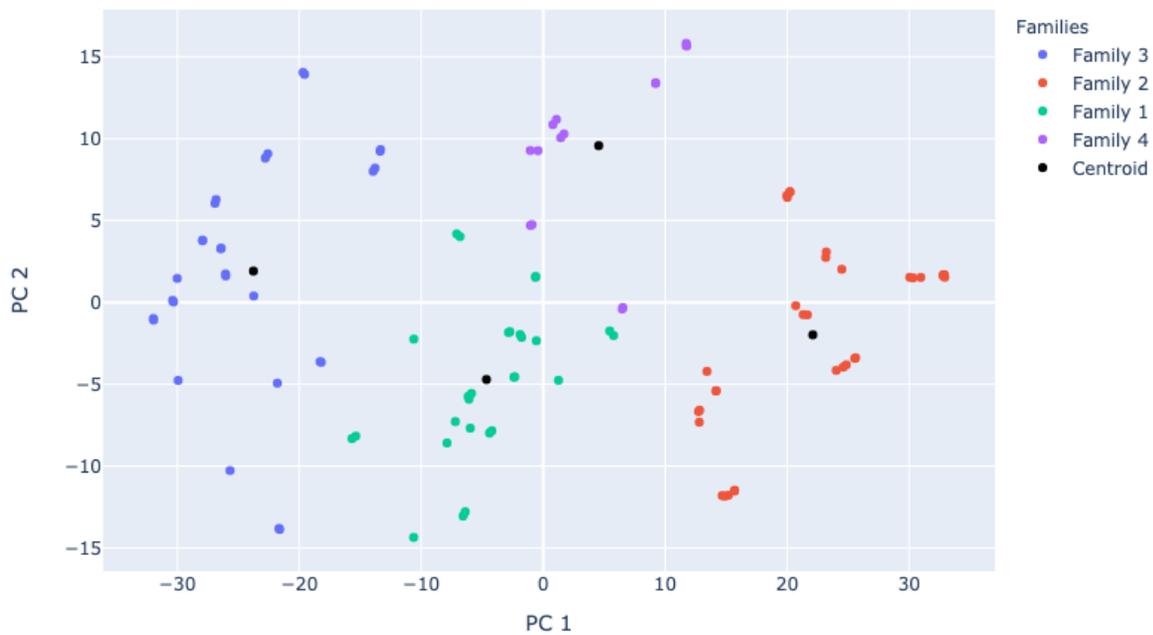


Figure 3-15 Clustering of Ac-(Gly)₃-NHMe (O, N, and H atoms; distances only)

The families for Ac-(Gly)₃-NHMe were (1) extended, (2) beta-turn, (3) gamma-turn, and (4) gamma and beta turn.

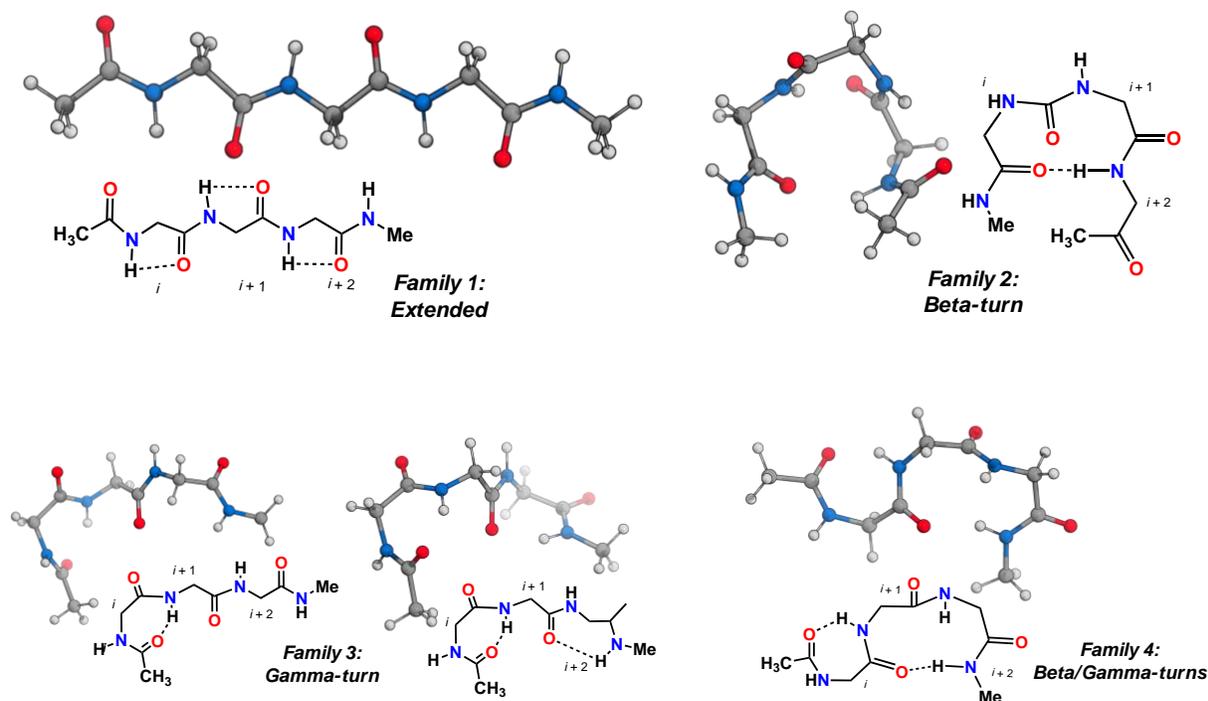


Figure 3-16 Conformational families of Ac-(Gly)₃-NHMe

When looking at cluster one, as predicted, extended families were found and *classified* with a high accuracy. In fact, for all of the clusters, all produced a high accuracy of classification. As for cluster two, beta-turn conformations showed in a classical fashion where the non-bonding interaction occurs with the carbonyl oxygen and the third nearest amino hydrogen. As for cluster three, saw found a previously unexpected conformational family: gamma-turns. Gamma-turns are very similar to beta turns, except the non-bonding interaction occurs with the carbonyl oxygen and the second nearest amino hydrogen (rather than the third). In this specific case gamma turns presented in multiple ways. Some conformations presented with one gamma-turn, while others presented two gamma-turns within the peptide chain.

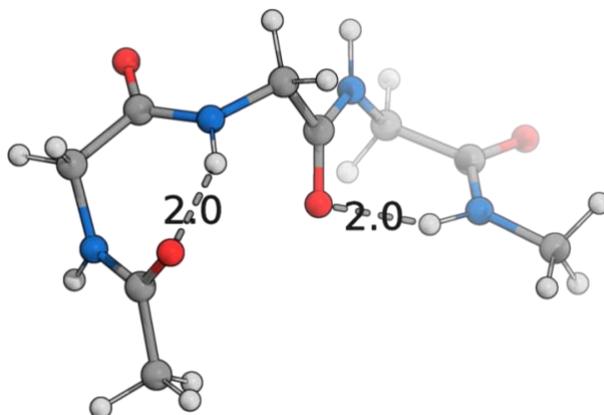


Figure 3-17 Example structure of two gamma-turns in one conformation

Lastly, cluster four found a family that shows both gamma- and beta-turns within a structure.

Overall, the software did a very good job in finding and separating conformational families for the smaller tripeptide chain.

3.2.2 Tri-Valine Peptides Test Case: Ac-(Val)₃-NHMe

We used Schrödinger Maestro to reproduce Ac-(Val)₃-NHMe structures. Molecular-mechanics based conformational searches via OPLS 2005, mixed torsional/Low-mode as the sampling method, the maximum number of steps was set to 10000, the energy window for saving structures was 25.00 kcal/mol, and an RMSD cutoff of 0.05 Å. This produced in 492 structures.

The conformations of Ac-(Val)₃-NHMe were also used to test specifically the distances part of the code and its ability to explain conformational families through nonbonding interactions found within a slightly more complex small peptide. The act of parsing each XYZ-coordinate and computing distances took the program 40.9 seconds to complete. Conformational analysis of Ac-(Val)₃-NHMe was conducted using PCA and K-means clustering techniques. PCA using two principal components on distances data produced an explained variance of 37.44% (Appendix-A-6). The conformations of Ac-(Val)₃-NHMe were used to test specifically the distances part of the code and its ability to explain conformational families through nonbonding interactions found

within small peptides. In this case instructed the code in three separate fashions (1) to take all atomic distances into consideration, (2) to take atomic distances from just oxygen, hydrogen and nitrogen atoms found in the peptide, and (3) take all proper dihedrals into consideration. All were subjected to PCA of two principal components to reduce the dimensionality of the data. For (1), the explained variance for two principal components was 37.4%, (2) was 34.7%, and (3) was 14.2%. A result such as this shows that using distances was the best metric available for finding conformational families within the tri-valine peptide. Finding the optimal number of clusters is completed identically as the cyclohexane test case, by calculating the sum of squares at each number of clusters and reading the “elbow” of the graph. In this test case, the best number of clusters to use was four for all three trials. Clustering graphs for (1) and (2) showed significant aggregation of structures, while (3) did not. Due to its lack in ability to separate structures case (3) was not investigated any further.

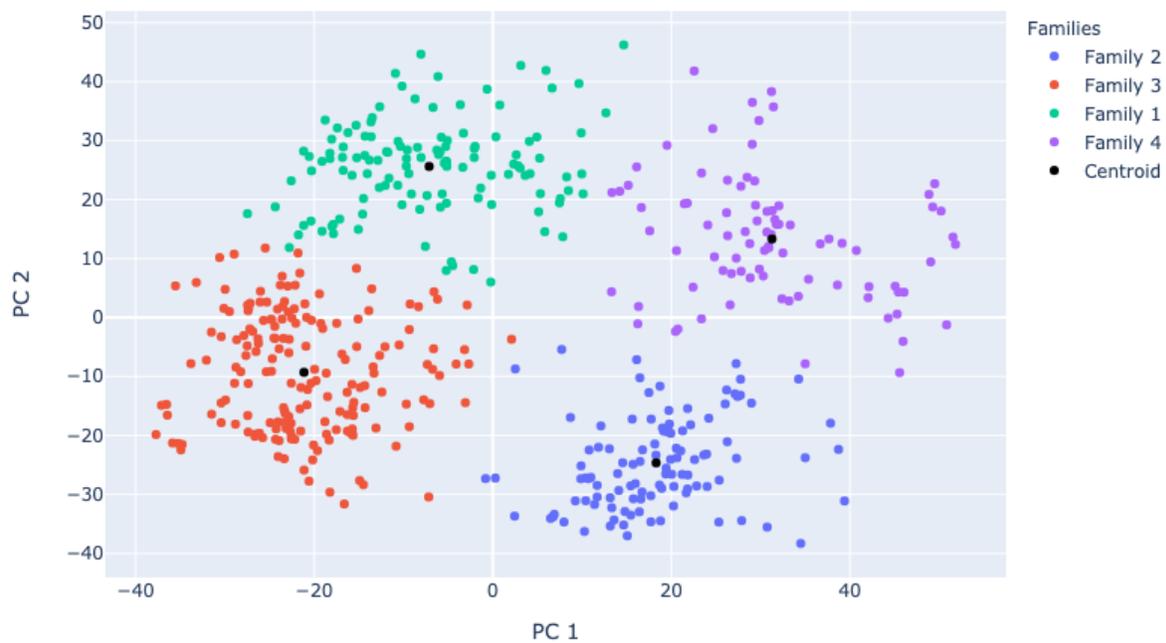


Figure 3-18 2D Clustering of Ac-(Val)₃-NHMe (all atoms; distances only)

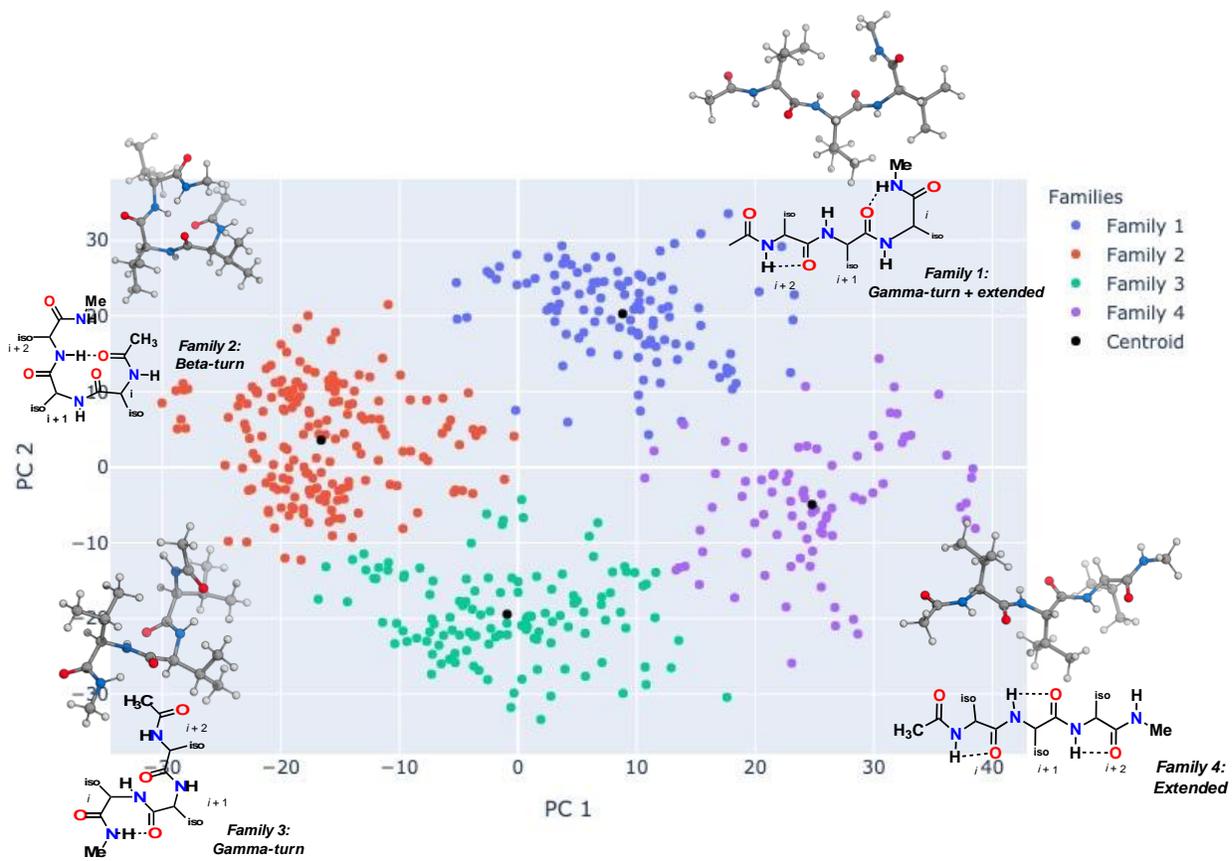


Figure 3-19 2D Clustering of Ac-(Val)₃-NHMe (O, N, and H atoms; distances only)

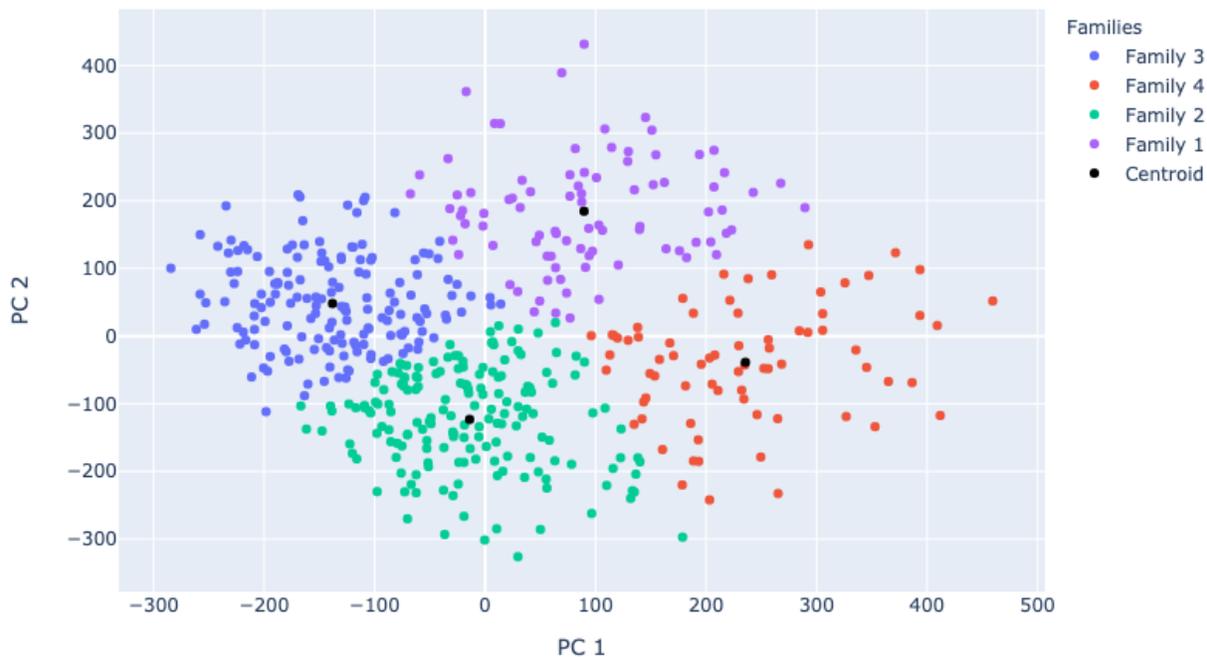


Figure 3-20 2D Clustering of Ac-(Val)₃-NHMe (all atoms; dihedrals only)

When investigating the O, N, H atoms only and all atoms cases, each had similar outcomes. Although there was separation between each cluster, clusters did not contain only one family. Rather, each cluster had a conformational family that was a majority, with other families in smaller quantities. Nonetheless, there were four families found. The four conformational families were (1) gamma-turn with partial extension, (2) beta-turns, (3) gamma-turns, and (4) extended conformation.

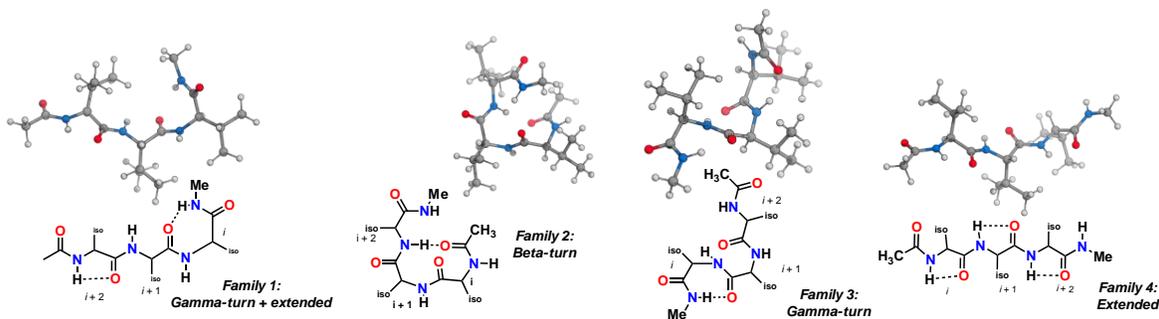


Figure 3-21 Conformational families of Ac-(Val)₃-NHMe

Again, it is important to note that the complete separation of families was not achieved by the algorithm in this case, and manual investigation of each family was necessary to find the majorities within each cluster. Family one, gamma-turn with partial extension shows structures where one side of the conformer has non-bonding interactions between carbonyl oxygen and an amino hydrogen that are two amines away, while the other side of the conformer is linear where a carbonyl oxygen is interacting with an adjacent amino hydrogen. I personally believe that this half and half family was found in part due to the peptide's isopropyl groups. In each instance, the isopropyl group on the side where the gamma-turn is located is in a position such that complete extension is not possible due to steric hindrance. This type of conformation was not seen in our previous short peptide chains investigation with Ac-(Gly)₃-NHMe, a peptide with smaller R groups.

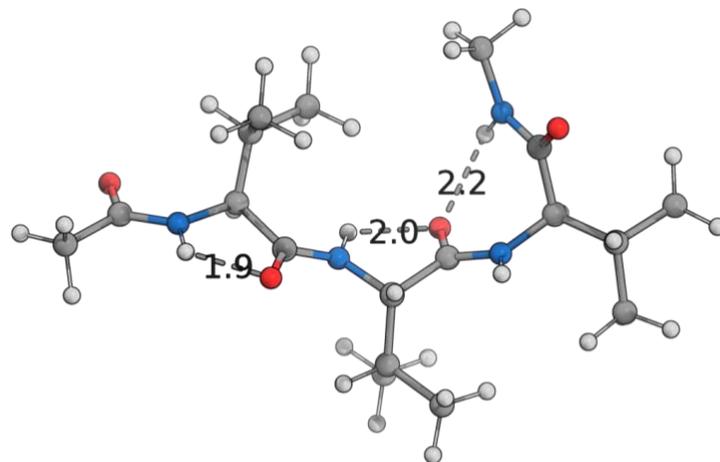


Figure 3-22 Example structure of conformational family 1 (gamma + extended)

As for cluster two, a beta-turn conformation was expected and was presented in a classical fashion where the non-bonding interaction occurs with the carbonyl oxygen and the third nearest amino hydrogen. Cluster three produced a gamma-turn conformational family with variation within. Instead of having a partially extended part akin to family one, this family contained structures with gamma-turns that occurred between amino acids one and two, two and three, or both.

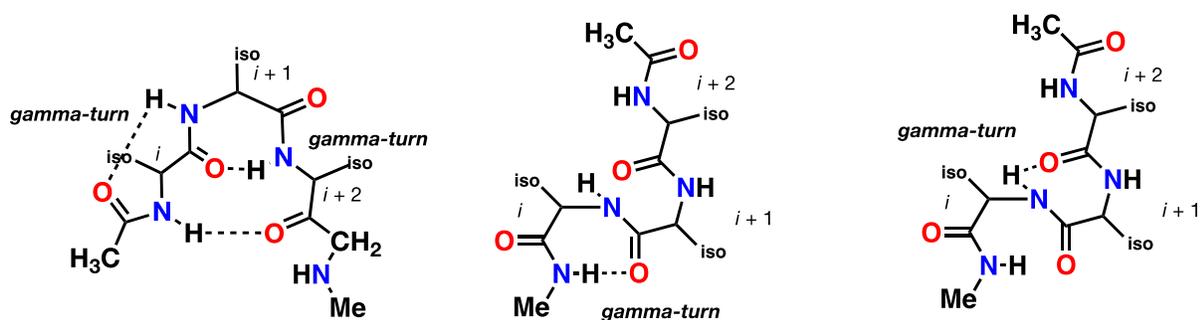


Figure 3-23 Conformational family 3 Variation (gamma-turn)

Lastly, cluster four contained extended conformations. These structures were extremely linear, as if the chain were a backbone. Overall, the separation of clusters seen on the 2D clustering graph were good, but the classification into families using all atom distances and heteroatom with

hydrogen distances were lower in accuracy than the previously mentioned 1-flouro-4-propylcyclohexane test. It is expected that results may change if the peptide chain were simplified.

3.3 Conformational Analysis of Hexacoordinated Ca^{2+} Complexes

Calcium (Ca^{2+}) is a group two alkaline earth metal that is crucial for human health. Known for its importance for bone health, Ca^{2+} is also used in the field of synthetic chemistry for stabilization purposes in chemical reactions⁶¹. The reason why it is so useful in chemistry is its ability to coordinate many other moieties around it. It is known that Ca^{2+} predominantly forms hexacoordinate species⁶². This means that it can coordinate with six other structures, which in turn gives those structures an opportunity to react with one another. In a sense, Ca^{2+} can be used as a kitchen; where spices and recipes mix and match to create something bigger and better.

Although conformational analyses of hexacoordinated Ca^{2+} complexes is not widely studied, my research lab has looked extensively into hexacoordinated Ca^{2+} complexes used to elucidate the reaction mechanism for making sulfonamides, which are compounds that make up about 27% of all sulfur-based US FDA approved drugs⁶³. Conformational families expected are just two: cis- and trans- conformations. The use of calcium as a Lewis-acid catalyst has proven to be an inexpensive, and nontoxic means for chemical transformations^{64,65}. The Lewis-acid calcium salts promote selectivity towards non redox processes and high reactivity under mild conditions.

Suitable counterions for calcium catalysis that produced the best results were weakly coordinating, non-basic anions such as triflimide (NTf_2^-)⁶⁴. Calcium(II) bis(trifluoromethanesulfonimide) (also known as $\text{Ca}(\text{NTf}_2)_2$) has been used as a catalyst/mediator for intramolecular hydroacyloxylation of unactivated alkenes⁶⁶, production of diaryl alkanes⁶⁵,

and activation of alcohols, olefins and carbonyl compounds⁶⁴. Overall, calcium salts are useful in a myriad of reactions that can be used to create drugs, synthetic materials and more.

3.3.1 Hexacoordinated Ca²⁺ Complex Test Case: [Ca(NH₃)₂(THF)₄]²⁺

Schrödinger Maestro reproduction of [Ca(NH₃)₂(THF)₄]²⁺ used Molecular-mechanics based conformational searches via OPLS 2005, mixed torsional/Low-mode as the sampling method, the maximum number of steps was set to 1000, the energy window for saving structures was 25.00 kcal/mol, and an RMSD cutoff of 0.05 Å. This produced in 226 structures. The goal of this test case is to use the angles portion of the code to find cis- and trans- conformations with respect to the position of the NH₃ groups in each structure. In this case specifically, conformational differences will come from the C-C bond torsions of the THF. Because this is a relatively large complex, the calculation of all angles for all structures took nine minutes to complete. This is a much larger time window of completion from our previous test cases, but it is also expected due to the sheer volume and complexity. The explained variance using two principal components was 31.4%. Using the “elbow method” to find the most optimal number of clusters resulted in three clusters being most optimal. Each cluster were found to be (1) cis-NH₃ groups Ca²⁺ complex (2) trans-NH₃ groups Ca²⁺ complex (3) cis-NH₃ groups Ca²⁺ complex.

Inertias for choosing best number of clusters

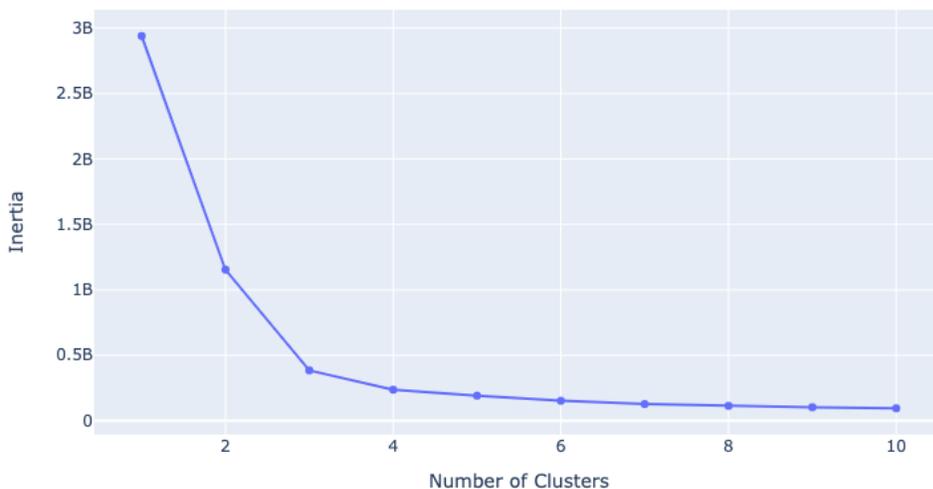


Figure 3-24 Elbow Graph of $[\text{Ca}(\text{NH}_3)_2(\text{THF})_4]^{2+}$ (all atoms; angles only)

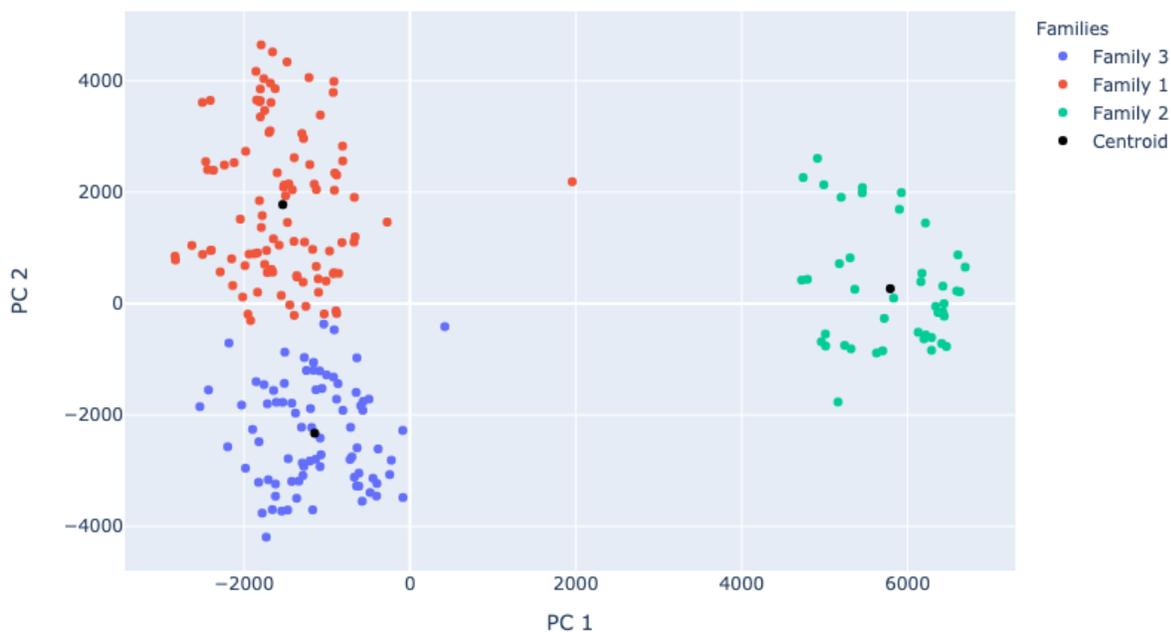


Figure 3-25 Clustering Graph of $[\text{Ca}(\text{NH}_3)_2(\text{THF})_4]^{2+}$ (all atoms; angles only)

It is clear that the program was able to decipher and distinguish between cis- and trans-complexes. This was done with complete accuracy. However, it was peculiar to see that the best

number of clusters was found to be three. This would mean that another family could be made within cis-complex conformers. However, after investigation, this is not found to be entirely true. To test this, I compared the two structures between family one and three that were furthest from each other. In theory, these structures should be the *most* different from each other.

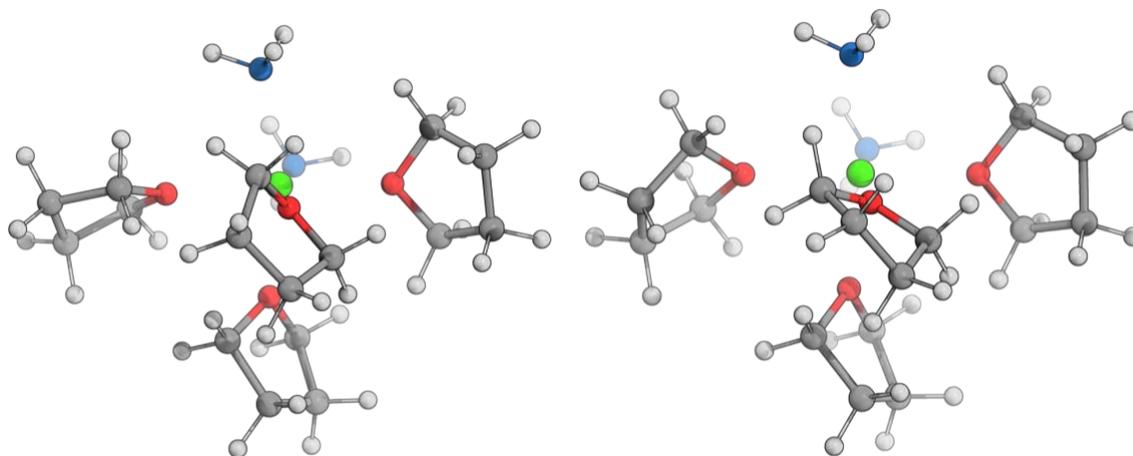


Figure 3-26 Side-by-side comparison of most different structures between cis $[\text{Ca}(\text{NH}_3)_2(\text{THF})_4]^{2+}$ families (all atoms; angles only)

It is clear that there only minute differences between the two structures, certainly not enough to describe one structure to be in a different conformational family from the other. I believe this happened due to the fact that all angles were calculated. Calculating all could be including angles that is hurting the classification process rather than helping due to the sheer amount of information. In order to test this, I calculated angles found between all atoms excluding hydrogen. The result finished in 55 seconds. Additionally, when using two principal components, increased the explained variance to 41.1%, and produced an elbow graph where the difference between choosing two and three clusters is smaller than when all atoms' angles were used.

Inertia for choosing best number of clusters

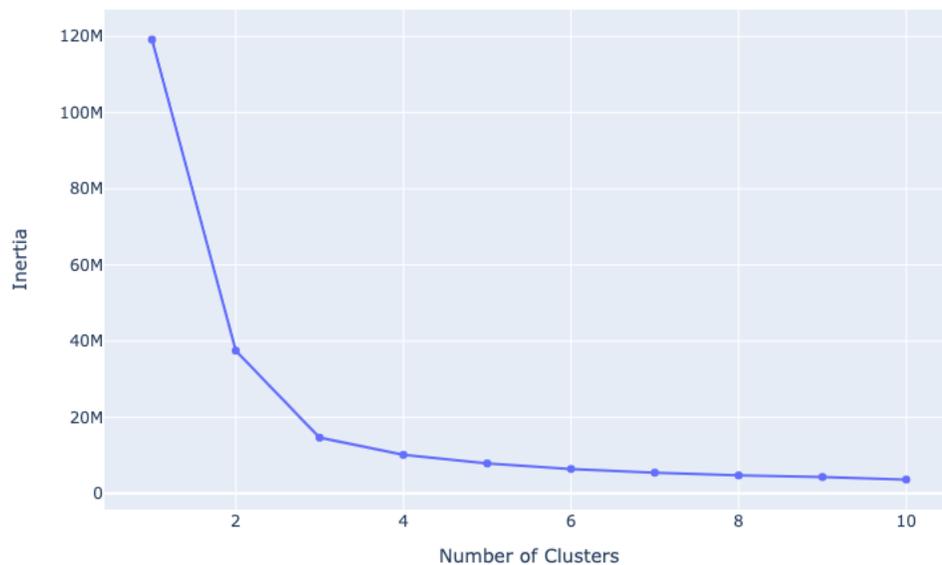


Figure 3-27 Elbow Graph of $[\text{Ca}(\text{NH}_3)_2(\text{THF})_4]^{2+}$ (O, N, C, F atoms; angles only)

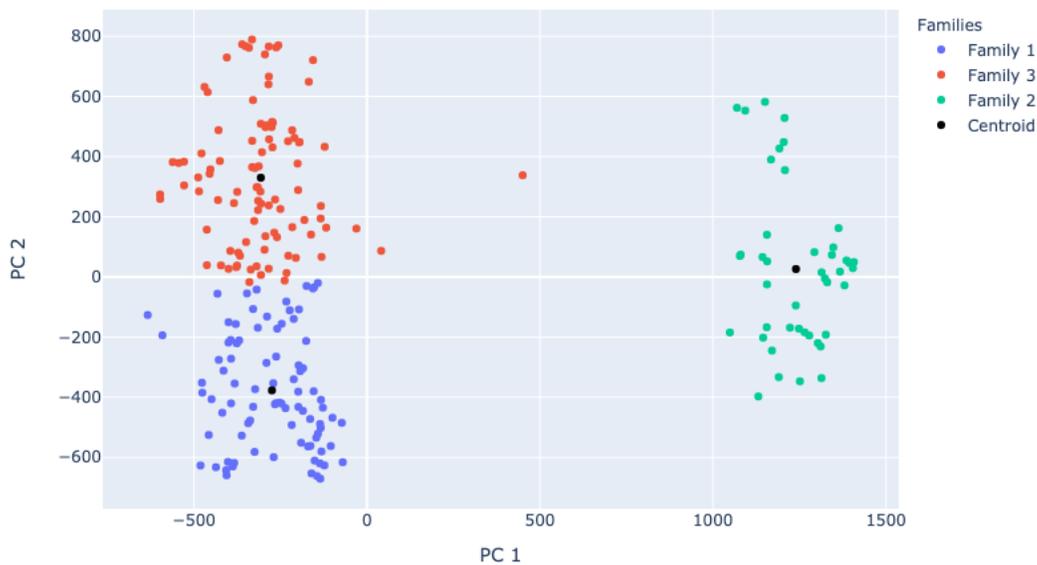


Figure 3-28 Clustering Graph of $[\text{Ca}(\text{NH}_3)_2(\text{THF})_4]^{2+}$ (O, N, C, F atoms; angles only)

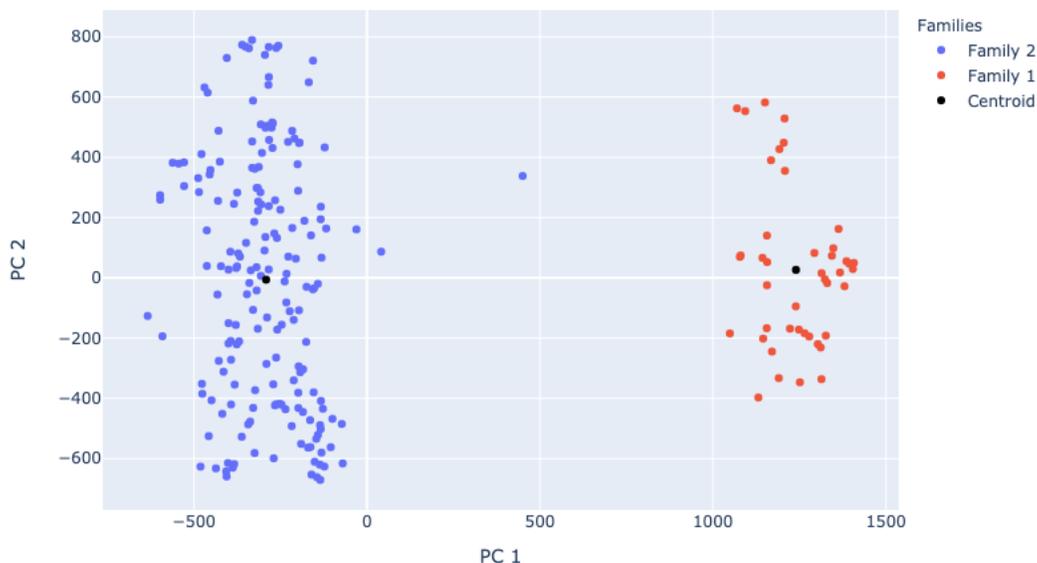


Figure 3-29 Clustering Graph of $[\text{Ca}(\text{NH}_3)_2(\text{THF})_4]^{2+}$ (O, N, C, F atoms; angles only)

This goes to show that although this software is a useful tool and has proven its ability to find and correctly classify conformational families, it is still not a chemist or biologist in its own right. Although the process of finding and sorting families can now be done expediently with this software package, the interpretation and general background knowledge behind what a scientist might see is still necessary to get the full benefit of using such a package.

4 Conclusion

4.1 Pros and Cons of Code based on Test Cases

Overall, the software does exactly what it set out to do: a software using PCA and K-Means clustering was able to find conformational families in a rapid and automated fashion. However, there are pros and cons. Each are listed below.

Pros:

(1) Speed

- a. The program runs very fast. Only one of our tests ran for over two minutes ($[\text{Ca}(\text{NH}_3)_2(\text{THF})_4]^{2+}$), and in the end, that same test was able to be shaved down to less than one minute with producing similar results. This is important for researchers because we do not want our technology to be the bottleneck in how efficient our work is. This program passes that test.

(2) Ease of use

- a. For anyone who understands how to run Python, the program is very easy to download onto one's computer, import into a python script, and subsequently use. Each step necessary to produce results has a guide to keep the user on track, and error catching capabilities to ensure that user errors do not derail the program from working.

(3) Accuracy (to an extent)

- a. In each of the cases, we have seen a range of accuracy in terms of classification. Part of this is because we chose test cases of increasing complexity. With small molecules, the accuracy is next to flawless. However, the more complex the ensemble gets, the more important it is for the user to be selecting certain atoms for calculations rather than wanting all possible calculations to be completed. In a sense, this is good because the accuracy is partially dependent on the user and their knowledge behind their system of interest. However, we understand that this could also be a fault since beginners may not know how to create inputs that are ideal enough to produce meaningful results.

(4) Visuals

- a. A part of what makes methods like PCA tough is it feels as though it is a “black box” that spits out information. In using PCA graphs to visualize what is going on, as well as clustering to help show how similar structures are spatially, it gives the power to the user to understand how similar or dissimilar each family is.

Cons:

(1) Basic Options

- a. As of now, the user can only decide whether or not to use distances, angles, or dihedrals, as well as the specific atoms in the compound to do such calculations on. While this is useful, the program may benefit if options such as polar surface area were available to the user.

(2) Dependency of dimensionality reduction

- a. The program’s ability to tease out chemical information is mostly-based on how well the dimensionality reduction of data proceeds. If meaningful information is still found within the first couple of principal components, then the program works beautifully in all its complete capacity. If that is not the case, using more than three principal components means the user is no longer able to see visually any of the graphs.

(3) File requirements

- a. The program only works with XYZ-files. In future updates, we plan to provide ability to use other file types, such as PDB.

4.2 Future Developments

In future updates, we hope to provide users more options than distances, angles, and dihedrals to find conformational families. These options could include molecular descriptors such as polar surface area and radius of gyration, dipole moments, and more. Additionally, we hope to support more file-types, and create a program that can be ran outside of python scripts. At the end of the day, the ultimate goal is to transform this program into one that incorporates a feedback processing with ML or DL algorithms. We envision this finished product to have the ability to be given a set of conformations and parameters, and through its calculations and algorithms, weigh parameters that will produce the best separation between conformers. Such an iterative process should produce improved results from our current solution.

4.3 Concluding Thoughts

Based off the test cases, it has been shown that the software package created is able to find and sort out conformational families found within ensembles. It can do this using the distances, angles and/or dihedrals found within the molecule in question, and its data can be tailored to specific atoms within the molecule as well. Using PCA and K-means clustering, conformational families found within ensembles were found with varying success. A natural rule of thumb regarding the success of family sorting involves the complexity of the family in question. The more complex the molecule and its family, the tougher it was to find the family. Nonetheless, even with its toughest test cases, classification accuracy decreases to the point where families found were just “majorities” rather than complete separation. Although not ideal, users are still able to find and understand these families with a little effort on their part. With future

improvements, we hope for this to be a useful tool for all chemical and biological researchers who are interested in conformational analysis.

References

- (1) Benfey, O. T. August Kekule and the Birth of the Structural Theory of Organic Chemistry in 1858. *J. Chem. Educ.* **1958**, *35* (1), 21. <https://doi.org/10.1021/ed035p21>.
- (2) van 't Hoff, J. H. A Suggestion Looking to the Extension into Space of the Structural Formulas at Present Used in Chemistry. And a Note Upon the Relation Between the Optical Activity and the Chemical Constitution of Organic Compounds. *Arch. Neerlandaises Sci. Exactes Nat.* **1874**, *9*, 445–454.
- (3) Grossman, R. B. Van't Hoff, Le Bel, and the Development of Stereochemistry: A Reassessment. *J. Chem. Educ.* **1989**, *66* (1), 30. <https://doi.org/10.1021/ed066p30>.
- (4) *Van't Hoff-Le Bel Centennial*; Ramsay, O. B., Ed.; ACS Symposium Series; AMERICAN CHEMICAL SOCIETY: WASHINGTON, D. C., 1975; Vol. 12.
<https://doi.org/10.1021/bk-1975-0012>.
- (5) Barton, D. H. R.; Cookson, R. C. The Principles of Conformational Analysis. *Q. Rev. Chem. Soc.* **1956**, *10* (1), 44. <https://doi.org/10.1039/qr9561000044>.
- (6) Hassel, O. Stereochemistry of Cyclohexane. *Q. Rev. Chem. Soc.* **1953**, *7* (3), 221.
<https://doi.org/10.1039/qr9530700221>.
- (7) Mandal, P. K.; Arunan, E. Hydrogen Bond Radii for the Hydrogen Halides and van Der Waals Radius of Hydrogen. *J. Chem. Phys.* **2001**, *114* (9), 3880–3882.
<https://doi.org/10.1063/1.1343905>.
- (8) Barton, D. H. R. The Conformation of the Steroid Nucleus. *Experientia* **1950**, *6* (8), 316–320. <https://doi.org/10.1007/BF02170915>.

- (9) Barton, D. H. R. The Principles of Conformational Analysis. *Science* **1970**, *169* (3945), 539–544.
- (10) Hassel, O.; Ottar, B.; Roald, B.; Linnasalmi, A.; Laukkanen, P. The Structure of Molecules Containing Cyclohexane or Pyranose Rings. *Acta Chem. Scand.* **1947**, *1*, 929–943. <https://doi.org/10.3891/acta.chem.scand.01-0929>.
- (11) Mazzanti, A.; Casarini, D. Recent Trends in Conformational Analysis: Recent Trends in Conformational Analysis. *Wiley Interdiscip. Rev. Comput. Mol. Sci.* **2012**, *2* (4), 613–641. <https://doi.org/10.1002/wcms.96>.
- (12) Hagen, K. Conformational Analysis by Gas Electron Diffraction. In *Structures and Conformations of Non-Rigid Molecules*; Laane, J., Dakkouri, M., Veken, B., Oberhammer, H., Eds.; Springer Netherlands: Dordrecht, 1993; pp 447–463. https://doi.org/10.1007/978-94-011-2074-6_22.
- (13) Berova, N.; Bari, L. D.; Pescitelli, G. Application of Electronic Circular Dichroism in Configurational and Conformational Analysis of Organic Compounds. *Chem. Soc. Rev.* **2007**, *36* (6), 914–931. <https://doi.org/10.1039/B515476F>.
- (14) Hashizume, H.; Imahori, K. Circular Dichroism and Conformation of Natural and Synthetic Polynucleotides. *J. Biochem. (Tokyo)* **1967**, *61* (6), 738–749. <https://doi.org/10.1093/oxfordjournals.jbchem.a128608>.
- (15) Purdie, N. Circular Dichroism and the Conformational Analysis of Biomolecules Edited by Gerald D. Fasman (Brandeis University). Plenum Press: New York. 1996. x + 738 Pp. \$125.00. ISBN 0-306-45142-5. *J. Am. Chem. Soc.* **1996**, *118* (50), 12871–12871. <https://doi.org/10.1021/ja965689f>.

- (16) Fasman, G. D.; Hoving, H.; Timasheff, S. N. Circular Dichroism of Polypeptide and Protein Conformations. *Film Studies. Biochemistry* **1970**, *9* (17), 3316–3324.
<https://doi.org/10.1021/bi00819a005>.
- (17) Angyal, S.; Bethell, G. Conformational Analysis in Carbohydrate Chemistry. III. The ¹³C N.M.R. Spectra of the Hexuloses. *Aust. J. Chem.* **1976**, *29* (6), 1249.
<https://doi.org/10.1071/CH9761249>.
- (18) Spiess, H. W. The Importance of NMR Spectroscopy to Macromolecular Science. *Macromolecules* **2017**, *50* (5), 1761–1777. <https://doi.org/10.1021/acs.macromol.6b02736>.
- (19) Poltev, V. Molecular Mechanics: Principles, History, and Current Status. In *Handbook of Computational Chemistry*; Leszczynski, J., Ed.; Springer Netherlands: Dordrecht, 2015; pp 1–48. https://doi.org/10.1007/978-94-007-6169-8_9-2.
- (20) Vanommeslaeghe, K.; Guvench, O.; MacKerell, A. D. Molecular Mechanics. *Curr. Pharm. Des.* **2014**, *20* (20), 3281–3292. <https://doi.org/10.2174/13816128113199990600>.
- (21) Rychnovsky, S. D.; Yang, G.; Powers, J. P. Chair and Twist-Boat Conformations of 1,3-Dioxanes: Limitations of Molecular Mechanics Force Fields. *J. Org. Chem.* **1993**, *58* (19), 5251–5255. <https://doi.org/10.1021/jo00071a040>.
- (22) Fleetwood, O.; Kasimova, M. A.; Westerlund, A. M.; Delemotte, L. Molecular Insights from Conformational Ensembles via Machine Learning. *Biophys. J.* **2020**, *118* (3), 765–780. <https://doi.org/10.1016/j.bpj.2019.12.016>.
- (23) Olivon, F.; Elie, N.; Grelier, G.; Roussi, F.; Litaudon, M.; Touboul, D. MetGem Software for the Generation of Molecular Networks Based on the T-SNE Algorithm. *Anal. Chem.* **2018**, *90* (23), 13900–13908. <https://doi.org/10.1021/acs.analchem.8b03099>.

- (24) Berg, A.; Franke, L.; Scheffner, M.; Peter, C. Machine Learning Driven Analysis of Large Scale Simulations Reveals Conformational Characteristics of Ubiquitin Chains. *J. Chem. Theory Comput.* **2020**, *16* (5), 3205–3220. <https://doi.org/10.1021/acs.jctc.0c00045>.
- (25) Jordan, S. N.; Leach, A. R.; Bradshaw, J. The Application of Neural Networks in Conformational Analysis. 1. Prediction of Minimum and Maximum Interatomic Distances. *J. Chem. Inf. Comput. Sci.* **1995**, *35* (3), 640–650. <https://doi.org/10.1021/ci00025a035>.
- (26) Meiler, J.; Meusinger, R.; Will, M. Fast Determination of ¹³C NMR Chemical Shifts Using Artificial Neural Networks. *J. Chem. Inf. Comput. Sci.* **2000**, *40* (5), 1169–1176. <https://doi.org/10.1021/ci000021c>.
- (27) Ko, G. M.; Reddy, A. S.; Kumar, S.; Bailey, B. A.; Garg, R. Computational Analysis of HIV-1 Protease Protein Binding Pockets. *J. Chem. Inf. Model.* **2010**, *50* (10), 1759–1771. <https://doi.org/10.1021/ci100200u>.
- (28) Tian, H.; Trozzi, F.; Zoltowski, B. D.; Tao, P. Deciphering the Allosteric Process of the *Phaeodactylum Tricornutum* Aureochrome 1a LOV Domain. *J. Phys. Chem. B* **2020**, *124* (41), 8960–8972. <https://doi.org/10.1021/acs.jpcc.0c05842>.
- (29) Abramyan, T. M.; Snyder, J. A.; Thyparambil, A. A.; Stuart, S. J.; Latour, R. A. Cluster Analysis of Molecular Simulation Trajectories for Systems Where Both Conformation and Orientation of the Sampled States Are Important. *J. Comput. Chem.* **2016**, *37* (21), 1973–1982. <https://doi.org/10.1002/jcc.24416>.
- (30) Peng, J.; Wang, W.; Yu, Y.; Gu, H.; Huang, X. Clustering Algorithms to Analyze Molecular Dynamics Simulation Trajectories for Complex Chemical and Biological Systems. *Chin. J. Chem. Phys.* **2018**, *31* (4), 404–420. <https://doi.org/10.1063/1674-0068/31/cjcp1806147>.

- (31) Ivanov, P. M. Computational Studies on the Conformations of Some Large-Ring Cyclodextrins (CDn, n = 20, 21, 22, 23). *Chirality* **2011**, *23* (8), 628–637. <https://doi.org/10.1002/chir.20995>.
- (32) Das, G.; Gentile, F.; Coluccio, M. L.; Perri, A. M.; Nicastrì, A.; Mecarini, F.; Cojoc, G.; Candeloro, P.; Liberale, C.; De Angelis, F.; Di Fabrizio, E. Principal Component Analysis Based Methodology to Distinguish Protein SERS Spectra. *J. Mol. Struct.* **2011**, *993* (1–3), 500–505. <https://doi.org/10.1016/j.molstruc.2010.12.044>.
- (33) Buslaev, P.; Gordeliy, V.; Grudin, S.; Gushchin, I. Principal Component Analysis of Lipid Molecule Conformational Changes in Molecular Dynamics Simulations. *J. Chem. Theory Comput.* **2016**, *12* (3), 1019–1028. <https://doi.org/10.1021/acs.jctc.5b01106>.
- (34) Wolf, A.; Kirschner, K. N. Principal Component and Clustering Analysis on Molecular Dynamics Data of the Ribosomal L11·23S Subdomain. *J. Mol. Model.* **2013**, *19* (2), 539–549. <https://doi.org/10.1007/s00894-012-1563-4>.
- (35) Toyao, T.; Maeno, Z.; Takakusagi, S.; Kamachi, T.; Takigawa, I.; Shimizu, K. Machine Learning for Catalysis Informatics: Recent Applications and Prospects. *ACS Catal.* **2020**, *10* (3), 2260–2297. <https://doi.org/10.1021/acscatal.9b04186>.
- (36) Gao, H.; Struble, T. J.; Coley, C. W.; Wang, Y.; Green, W. H.; Jensen, K. F. Using Machine Learning To Predict Suitable Conditions for Organic Reactions. *ACS Cent. Sci.* **2018**, *4* (11), 1465–1476. <https://doi.org/10.1021/acscentsci.8b00357>.
- (37) Gómez-Bombarelli, R.; Wei, J. N.; Duvenaud, D.; Hernández-Lobato, J. M.; Sánchez-Lengeling, B.; Sheberla, D.; Aguilera-Iparraguirre, J.; Hirzel, T. D.; Adams, R. P.; Aspuru-Guzik, A. Automatic Chemical Design Using a Data-Driven Continuous

- Representation of Molecules. *ACS Cent. Sci.* **2018**, *4* (2), 268–276.
<https://doi.org/10.1021/acscentsci.7b00572>.
- (38) Struble, T. J.; Alvarez, J. C.; Brown, S. P.; Chytil, M.; Cisar, J.; DesJarlais, R. L.; Engkvist, O.; Frank, S. A.; Greve, D. R.; Griffin, D. J.; Hou, X.; Johannes, J. W.; Kreatsoulas, C.; Lahue, B.; Mathea, M.; Mogk, G.; Nicolaou, C. A.; Palmer, A. D.; Price, D. J.; Robinson, R. I.; Salentin, S.; Xing, L.; Jaakkola, T.; Green, William. H.; Barzilay, R.; Coley, C. W.; Jensen, K. F. Current and Future Roles of Artificial Intelligence in Medicinal Chemistry Synthesis. *J. Med. Chem.* **2020**, *63* (16), 8667–8682.
<https://doi.org/10.1021/acs.jmedchem.9b02120>.
- (39) Mater, A. C.; Coote, M. L. Deep Learning in Chemistry. *J. Chem. Inf. Model.* **2019**, *59* (6), 2545–2559. <https://doi.org/10.1021/acs.jcim.9b00266>.
- (40) Grambow, C. A.; Pattanaik, L.; Green, W. H. Deep Learning of Activation Energies. *J. Phys. Chem. Lett.* **2020**, *11* (8), 2992–2997. <https://doi.org/10.1021/acs.jpcllett.0c00500>.
- (41) Meyer, R.; Schmuck, K. S.; Hauser, A. W. Machine Learning in Computational Chemistry: An Evaluation of Method Performance for Nudged Elastic Band Calculations. *J. Chem. Theory Comput.* **2019**, *15* (11), 6513–6523.
<https://doi.org/10.1021/acs.jctc.9b00708>.
- (42) Allison, J. R. Computational Methods for Exploring Protein Conformations. *Biochem. Soc. Trans.* **2020**, *48* (4), 1707–1724. <https://doi.org/10.1042/BST20200193>.
- (43) Ding, X.; Zhang, B. DeepBAR: A Fast and Exact Method for Binding Free Energy Computation. *J. Phys. Chem. Lett.* **2021**, *12* (10), 2509–2515.
<https://doi.org/10.1021/acs.jpcllett.1c00189>.

- (44) Rupp, M.; Bauer, M. R.; Wilcken, R.; Lange, A.; Reutlinger, M.; Boeckler, F. M.; Schneider, G. Machine Learning Estimates of Natural Product Conformational Energies. *PLoS Comput. Biol.* **2014**, *10* (1), e1003400. <https://doi.org/10.1371/journal.pcbi.1003400>.
- (45) Oblinsky, D. G.; VanSchouwen, B. M. B.; Gordon, H. L.; Rothstein, S. M. Procrustean Rotation in Concert with Principal Component Analysis of Molecular Dynamics Trajectories: Quantifying Global and Local Differences between Conformational Samples. *J. Chem. Phys.* **2009**, *131* (22), 225102. <https://doi.org/10.1063/1.3268625>.
- (46) Ahmad, M.; Helms, V.; Kalinina, O. V.; Lengauer, T. Relative Principal Components Analysis: Application to Analyzing Biomolecular Conformational Changes. *J. Chem. Theory Comput.* **2019**, *15* (4), 2166–2178. <https://doi.org/10.1021/acs.jctc.8b01074>.
- (47) Papaleo, E.; Mereghetti, P.; Fantucci, P.; Grandori, R.; De Gioia, L. Free-Energy Landscape, Principal Component Analysis, and Structural Clustering to Identify Representative Conformations from Molecular Dynamics Simulations: The Myoglobin Case. *J. Mol. Graph. Model.* **2009**, *27* (8), 889–899. <https://doi.org/10.1016/j.jmgm.2009.01.006>.
- (48) Rossum, G. V.; Drake, F. L. *The Python Language Reference Manual*; Network Theory Limited, 2011.
- (49) Reback, J.; McKinney, W.; Jbrockmendel; Bossche, J. V. D.; Augspurger, T.; Cloud, P.; Gfyoung; Hawkins, S.; Sinhrks; Roeschke, M.; Klein, A.; Terji Petersen; Tratner, J.; She, C.; Ayd, W.; Naveh, S.; Garcia, M.; Patrick; Schendel, J.; Hayden, A.; Saxton, D.; Jancauskas, V.; Shadrach, R.; Gorelli, M.; McMaster, A.; Battiston, P.; Skipper Seabold; Kaiqi Dong; Chris-B1; H-Vetinari. *Pandas-Dev/Pandas: Pandas 1.2.3*; Zenodo, 2021. <https://doi.org/10.5281/ZENODO.4572994>.

- (50) Plotly Technologies Inc. *Collaborative Data Science*; Plotly Technologies Inc.: Montreal, QC, 2015.
- (51) Harris, C. R.; Millman, K. J.; van der Walt, S. J.; Gommers, R.; Virtanen, P.; Cournapeau, D.; Wieser, E.; Taylor, J.; Berg, S.; Smith, N. J.; Kern, R.; Picus, M.; Hoyer, S.; van Kerkwijk, M. H.; Brett, M.; Haldane, A.; del Río, J. F.; Wiebe, M.; Peterson, P.; Gérard-Marchant, P.; Sheppard, K.; Reddy, T.; Weckesser, W.; Abbasi, H.; Gohlke, C.; Oliphant, T. E. Array Programming with NumPy. *Nature* **2020**, *585* (7825), 357–362.
<https://doi.org/10.1038/s41586-020-2649-2>.
- (52) Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; Duchesnay, É. Scikit-Learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12* (85), 2825–2830.
- (53) *PyMol*; The PyMol Molecular Graphics System; Schrödinger, LLC.
- (54) The Cyclohexane Molecule <https://www.worldofmolecules.com/solvents/cyclohexane.htm> (accessed Apr 20, 2021).
- (55) Eliel, E. L.; Wilen, S. H.; Mander, L. N. *Stereochemistry of Organic Compounds*; Wiley: New York, 1994.
- (56) Loudon, G. M. *Organic Chemistry*, 5th ed.; Roberts and Co: Greenwood Village, Colo, 2009.
- (57) Gillespie, P.; Cicariello, J.; Olson, G. L. Conformational Analysis of Dipeptide Mimetics. *PeptideScience* **1997**, *43* (3), 191–217.
- (58) Balducci, D.; Bottoni, A.; Calvaresi, M.; Porzi, G.; Sandri, S. Synthesis and Conformational Preferences of Unnatural Tetrapeptides Containing L-Valine Units.

- Tetrahedron Asymmetry* **2006**, *17* (23), 3273–3281.
<https://doi.org/10.1016/j.tetasy.2006.12.006>.
- (59) Ooi, T.; Scott, R. A.; Vanderkooi, G.; Scheraga, H. A. Conformational Analysis of Macromolecules. IV. Helical Structures of Poly-L-Alanine, Poly-L-Valine, Poly- β -Methyl-L-Aspartate, Poly- γ -Methyl-L-Glutamate, and Poly-L-Tyrosine. *J. Chem. Phys.* **1967**, *46* (11), 4410–4426. <https://doi.org/10.1063/1.1840561>.
- (60) Tobias, D. J.; Brooks, C. L. Thermodynamics and Mechanism of α -Helix Initiation in Alanine and Valine Peptides. *Biochemistry* **1991**, *30* (24), 6059–6070.
<https://doi.org/10.1021/bi00238a033>.
- (61) Mukherjee, P.; Woroch, C. P.; Cleary, L.; Rusznak, M.; Franzese, R. W.; Reese, M. R.; Tucker, J. W.; Humphrey, J. M.; Etuk, S. M.; Kwan, S. C.; am Ende, C. W.; Ball, N. D. Sulfonamide Synthesis via Calcium Triflimide Activation of Sulfonyl Fluorides. *Org. Lett.* **2018**, *20* (13), 3943–3947. <https://doi.org/10.1021/acs.orglett.8b01520>.
- (62) Katz, A. K.; Glusker, J. P.; Beebe, S. A.; Bock, C. W. Calcium Ion Coordination: A Comparison with That of Beryllium, Magnesium, and Zinc. *J. Am. Chem. Soc.* **1996**, *118* (24), 5752–5763. <https://doi.org/10.1021/ja953943i>.
- (63) Scott, K.; Njardarson, J. Analysis of US FDA-Approved Drugs Containing Sulfur Atoms. *Top. Curr. Chem.* **2018**, *376*. <https://doi.org/10.1007/s41061-018-0184-5>.
- (64) Begouin, J.-M.; Niggemann, M. Calcium-Based Lewis Acid Catalysts. *Chem. – Eur. J.* **2013**, *19* (25), 8030–8041. <https://doi.org/10.1002/chem.201203496>.
- (65) Qi, C.; Gandon, V.; Lebœuf, D. Calcium(II)-Catalyzed Intermolecular Hydroarylation of Deactivated Styrenes in Hexafluoroisopropanol. *Angew. Chem. Int. Ed Engl.* **2018**, *57* (43), 14245–14249. <https://doi.org/10.1002/anie.201809470>.

- (66) Qi, C.; Yang, S.; Gandon, V.; Lebcœuf, D. Calcium(II)- and Triflimide-Catalyzed Intramolecular Hydroacyloxylation of Unactivated Alkenes in Hexafluoroisopropanol. *Org. Lett.* **2019**, *21* (18), 7405–7409. <https://doi.org/10.1021/acs.orglett.9b02705>.

Appendix A. Additional PCA Graphs + Data

A.1 Cis-1-Flouro-4-Propylcyclohexane

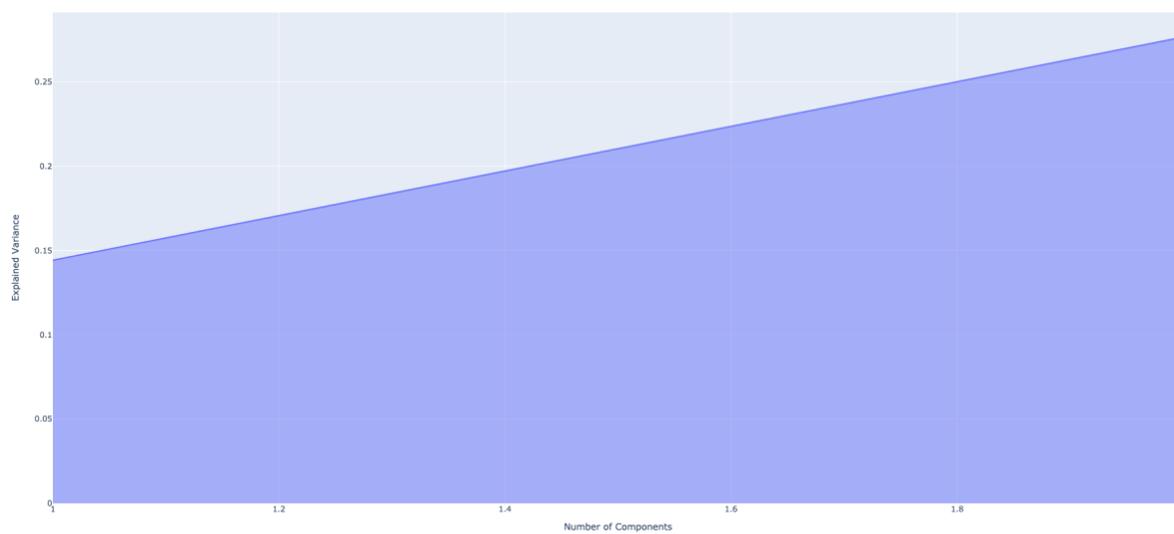


Figure A-1 Explained Variance of Cis-1-Flouro-4-Propylcyclohexane (Dihedral data only)

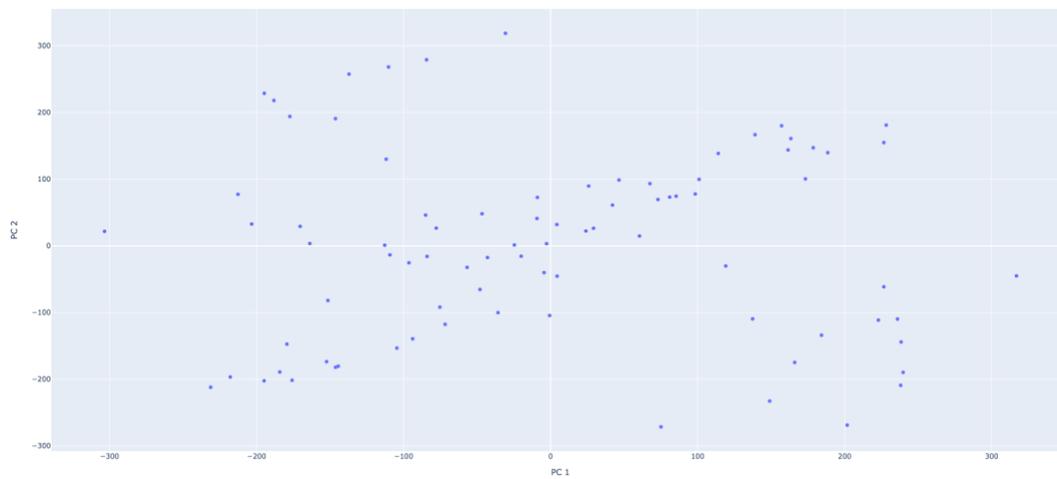


Figure A-2 PCA of Cis-1-Flouro-4-Propylcyclohexane (Dihedral data only)

Top 10 Features and their given magnitude
and direction **(PC 1)**

11 H to 3 C to 4 C to 14 H -0.280831

2 C to 3 C to 4 C to 5 C -0.275485

12 F to 3 C to 4 C to 13 H -0.272610

9 H to 2 C to 3 C to 11 H 0.228605

1 C to 2 C to 3 C to 4 C 0.224279

10 H to 2 C to 3 C to 12 F 0.221936

2 C to 3 C to 4 C to 13 H 0.193290

14 H to 4 C to 5 C to 16 H 0.190066

3 C to 4 C to 5 C to 6 C 0.187832

13 H to 4 C to 5 C to 15 H 0.187580

Top 10 Features and their given magnitude
and direction **(PC 2)**

1 C to 6 C to 18 C to 20 H -0.286943

7 H to 1 C to 6 C to 18 C -0.264472

8 H to 1 C to 6 C to 17 H -0.257414

2 C to 1 C to 6 C to 5 C -0.253506

5 C to 6 C to 18 C to 21 C -0.244885

5 C to 6 C to 18 C to 19 H 0.234255

17 H to 6 C to 18 C to 19 H -0.224756

15 H to 5 C to 6 C to 18 C 0.213718

1 C to 6 C to 18 C to 21 C 0.207048

16 H to 5 C to 6 C to 17 H 0.206262

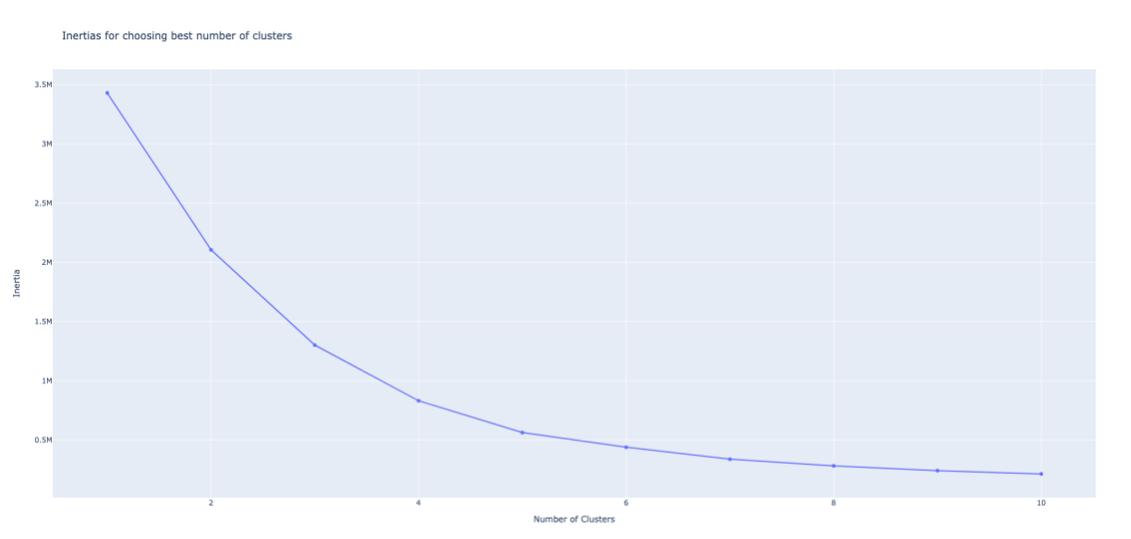


Figure A-3 Inertia v. # Clusters of Cis-1-Flouro-4-Propylcyclohexane (Dihedral data only)

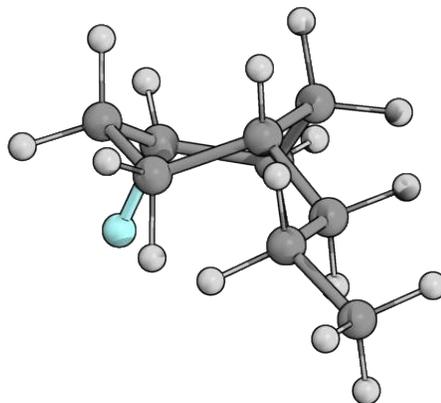


Figure A-4 Structure 29 of Cis-1-Flouro-4-Propylcyclohexane (Dihedral data only)

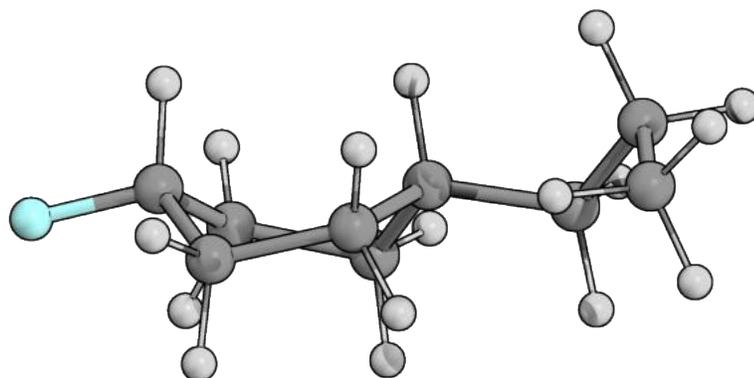


Figure A-5 Structure 50 of Cis-1-Flouoro-4-Propylcyclohexane (Dihedral data only)

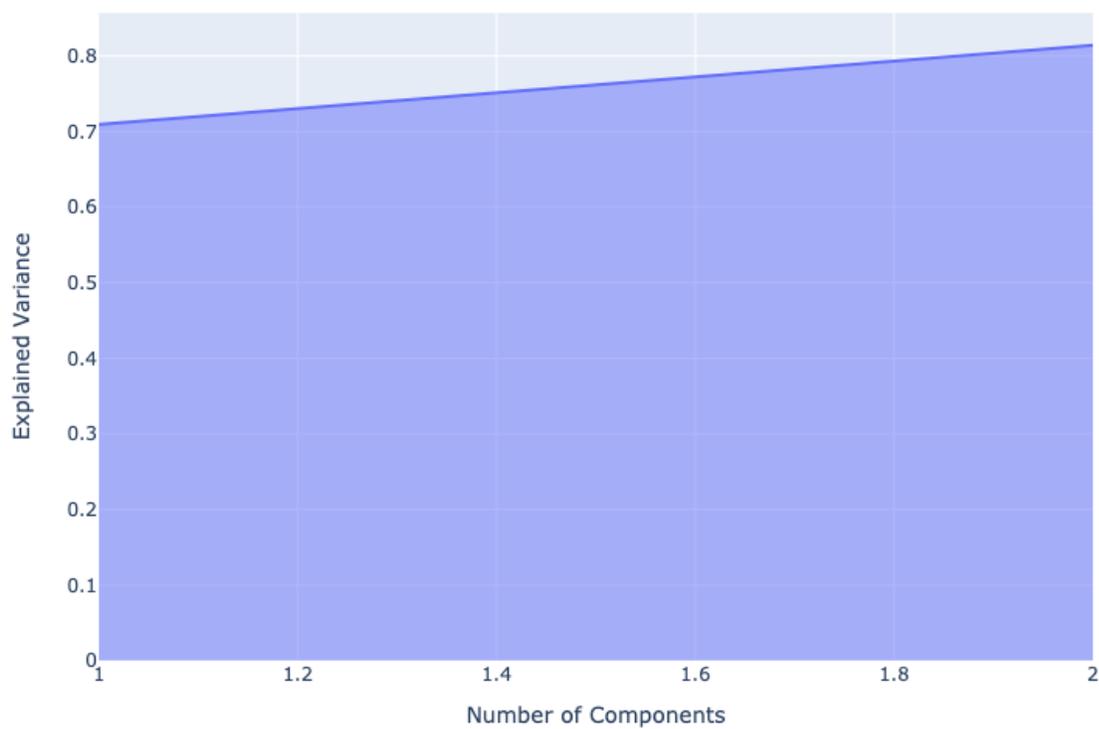


Figure A-6 Explained Variance of Ac-(Gly)₃-NHMe (Distances data only)

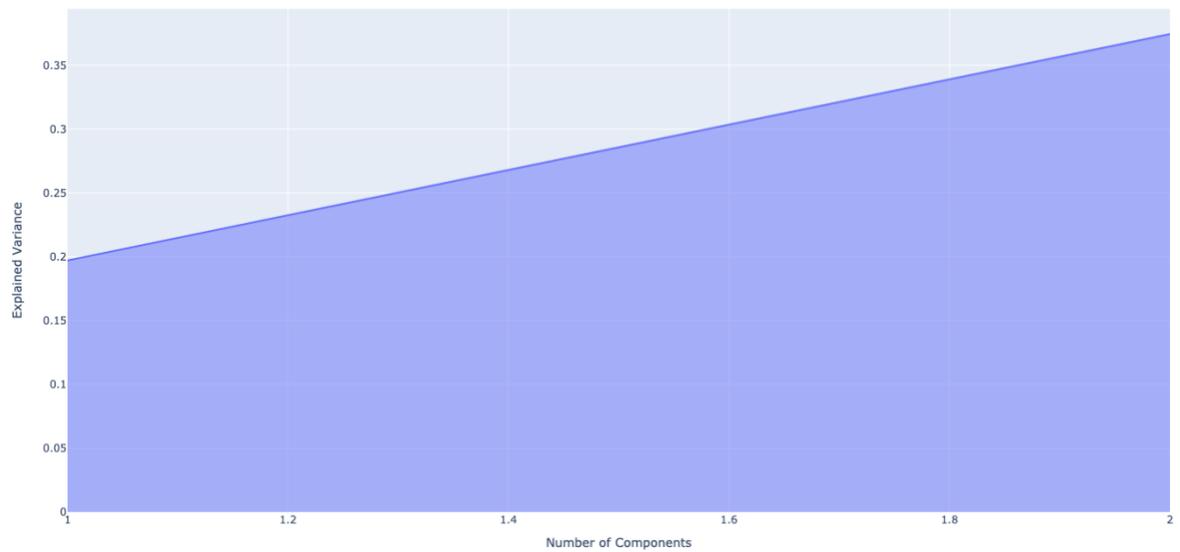


Figure A-7 Explained Variance of Ac-(Val)₃-NHMe (Distances data only)

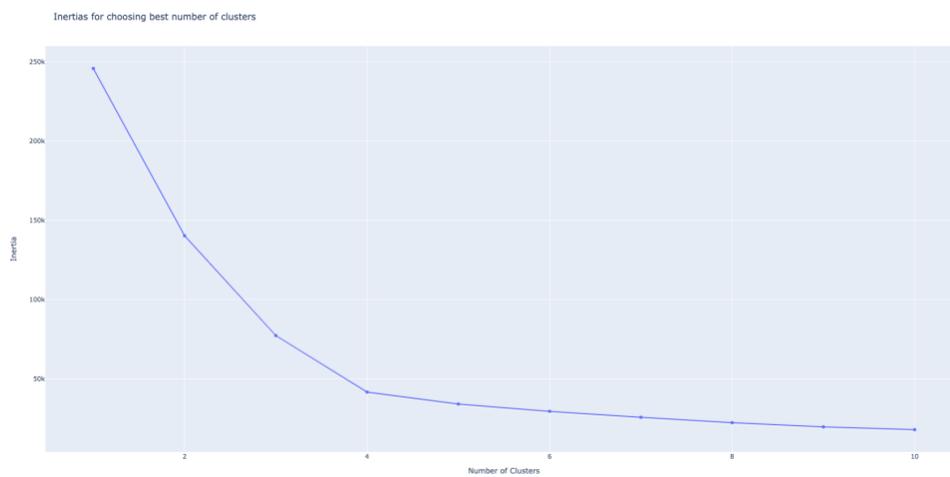


Figure A-8 Inertia v. # Clusters of Ac-(Val)₃-NHMe (Distances data only)

Appendix B. Atomic Min/Max Distance

Settings

Element	Minimum Bond Distance (Å)	Maximum Bond Distance	Minimum Number of Bonds	Maximum Number of Bonds
Carbon	1.01 Å	2.13 Å	4	4
Oxygen	0.90 Å	1.94 Å	2	2
Hydrogen	0.70 Å	1.64 Å	1	1
Nitrogen	0.90 Å	1.54 Å	1	4
Fluorine	0.90 Å	1.54 Å	1	4