UT Southwestern Medical Center
Lyda Hill Department of Bioinformatics

BioHPC

Dimensionality Reduction

[email] biohpc-help@utsouthwestern.edu
[register] portal.biohpc.swmed.edu/accounts/register

Outline

1. Introduction to dimensionality reduction methods
2. PCA mathematical background
3. Important features of PCA
4. Some PCA examples in literature
5. Demonstration of PCA examples

What is Dimensionality Reduction?

Dimensionality reduction is the transformation of data from high dimensional space to low dimensional space, which allows us to better interpret the progression of data.

Applications:

- Signal processing
- Speech recognition
- Neuroinformatics
- Bioinformatics
- Molecular dynamics
- Stock prediction

Dimensionality Reduction

There are multiple dimensionality reduction techniques, but they are mostly categorized into two classes: Linear and Non-linear types.

Linear:

- **PCA**
- Linear discriminant analysis

Non-linear:

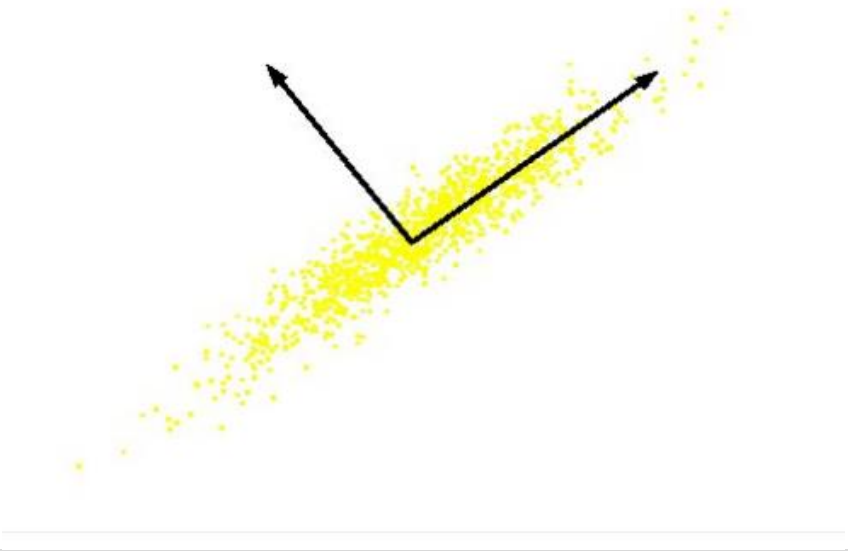
- Generalized discriminant analysis
- t-SNE
- UMAP

What is Principal Component Analysis?

Why is it linear?

Principal components are a sequence of *p* orthogonal unit vectors, where the i^{th} vector is the direction of a line best fitting the data.

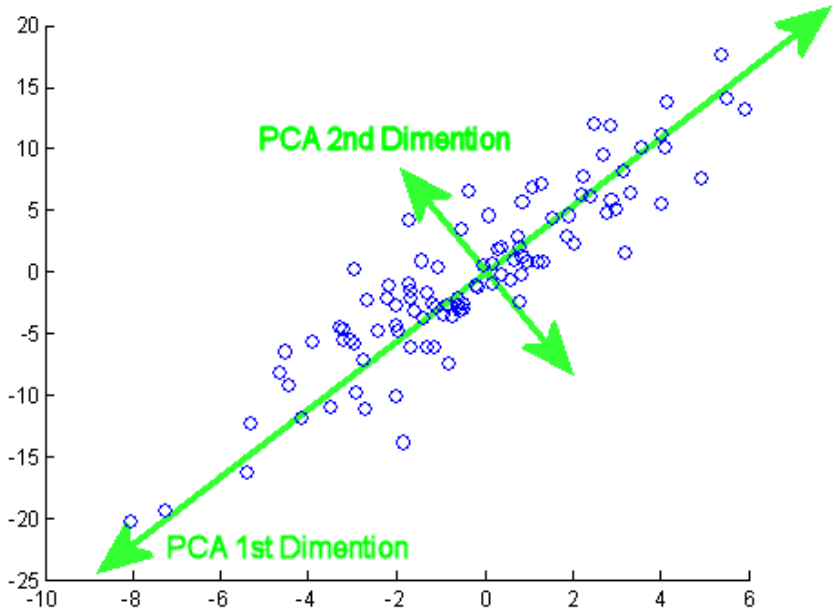
Principal component analysis is the process computing the principal components and using them to perform a change of basis on data.



Why Principal Component Analysis?

In principle, PCA allows us to look at the meaningful part sophisticated data.

The transformation of full data into principal components, or collective variables, is done via a multi-dimensional **ROTATION**.



Example: 2 meaningful components can be extracted. (source: medium.com)

How Does It Work?

Basically, PCA algorithm is considered an unsupervised machine learning method.

PCA captures the most variance of data onto the 1st PC, the second most variance onto the 2nd PC…

It maps each row vector ($x_{(1)}$, …, $x_{(n)}$) of X (representing data that needs to be transformed) to new vectors $t_{(i)}$ (PCs) using a matrix W = ($w_{(1)}$, …, $w_{(p)}$):

$$t_{k(i)} = x_{(i)} \bullet w_{(k)}$$

Capturing Variance for the PCs

$$\mathbf{w}_{(1)} = \arg\max_{\|\mathbf{w}\|=1} \left\{ \sum_i (t_1)^2_{(i)} \right\} = \arg\max_{\|\mathbf{w}\|=1} \left\{ \sum_i \left(\mathbf{x}_{(i)} \cdot \mathbf{w} \right)^2 \right\}$$

$$\mathbf{w}_{(1)} = \arg\max \left\{ \frac{\mathbf{w}^T \mathbf{X}^\mathbf{T} \mathbf{X} \mathbf{w}}{\mathbf{w}^T \mathbf{w}} \right\}$$

Capturing Variance for the PCs

For the remaining PCs:

$$\hat{\mathbf{X}}_k = \mathbf{X} - \sum_{s=1}^{k-1} \mathbf{X}\mathbf{w}_{(s)}\mathbf{w}_{(s)}^{\mathrm{T}}$$

Since we're done with k-1 PCs, we need to subtract them out.

Then moving on to do the same process for the k[th] component:

$$\mathbf{w}_{(k)} = \arg\max_{\|\mathbf{w}\|=1} \left\{ \|\hat{\mathbf{X}}_k \mathbf{w}\|^2 \right\} = \arg\max \left\{ \frac{\mathbf{w}^T \hat{\mathbf{X}}_k^T \hat{\mathbf{X}}_k \mathbf{w}}{\mathbf{w}^T \mathbf{w}} \right\}$$

Covariance Matrix

For forward (raw → PC) and backward (PC → raw) processes, transformation can be done with an orthonormal matrix U_C.

Denote X as a matrix of $\mathrm{n} \times N$ elements (n samples and N dimensions), we can obtain matrix Q $(\mathrm{n} \times N)$ of principle components by:

$$Q = U_C X$$

U_C is obtained by diagonalizing C, the Hessian covariance matrix:

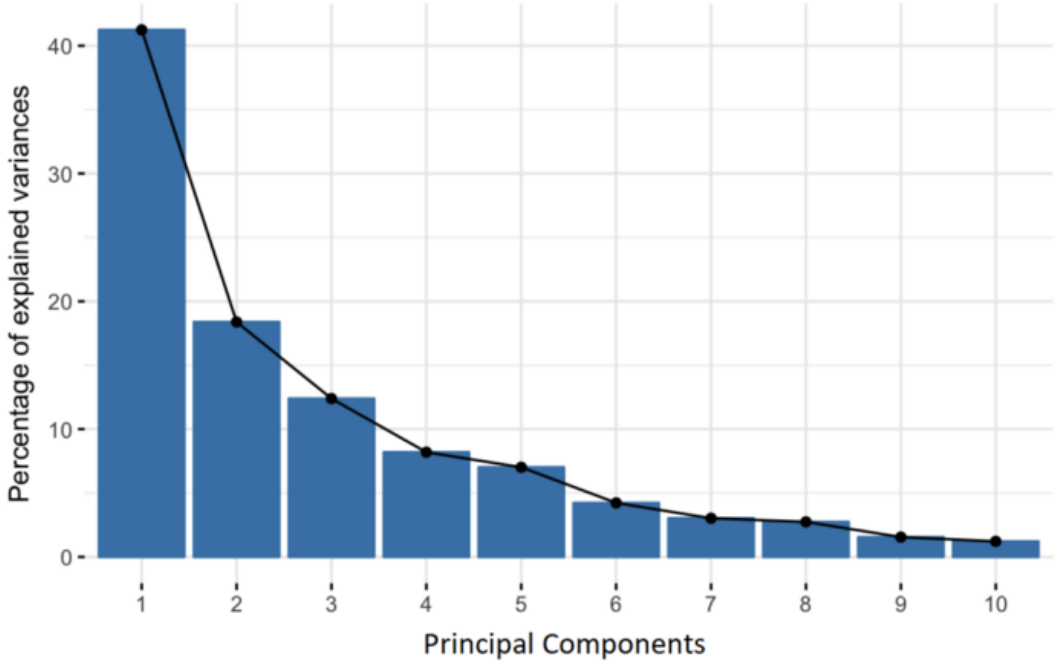
$$\Lambda = U_C C U_C^T$$

where U_C is the eigenvector matrix, Λ is the eigenvalue matrix.
See also: Hare, Bratholm, Glowacki, Carpenter, Chem. Sci. (2019).

How to Evaluate Significance of PCs?

PC with eigenvalue > 1: variance of more than one variable.

Largest eigenvalues: capture most variance from the MD data.
(Anal. Methods 2014, 6 2812)



Source: https://builtin.com/data-science/step-step-explanation-principal-component-analysis

11

Applications

Protein folding (HP35): PCA of 300 microseconds trajectory (JCP 141, 014111, 2014).

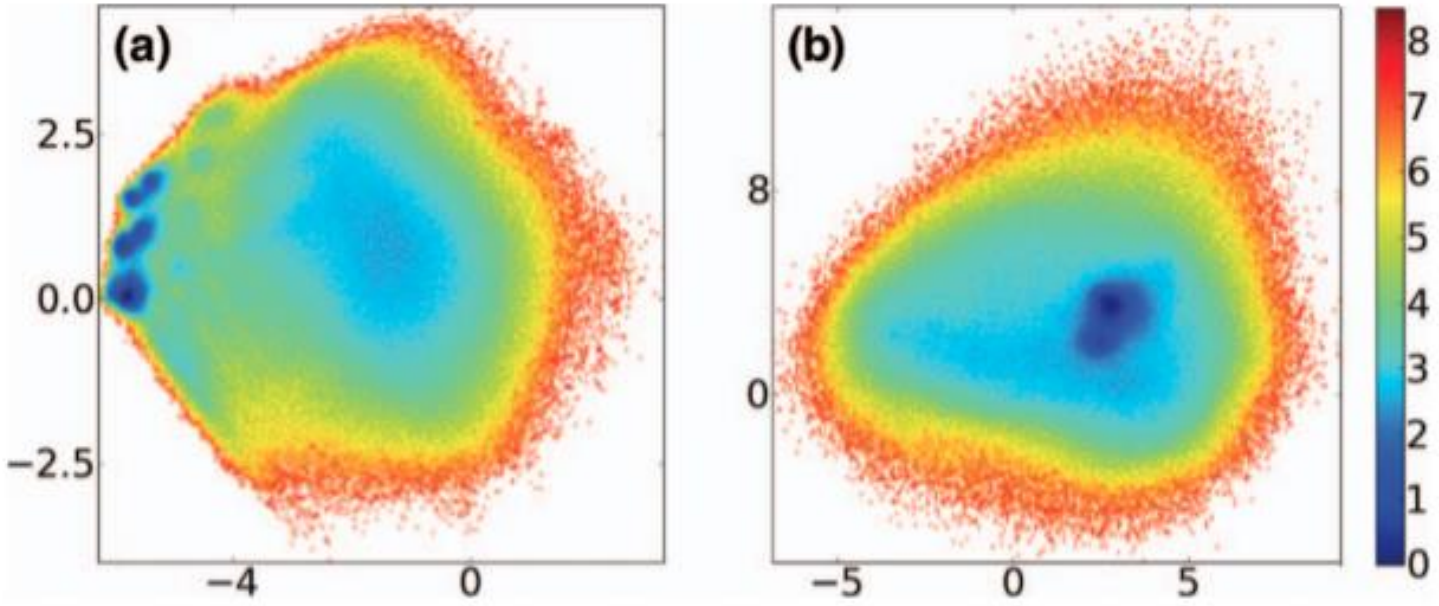


FIG. 1. Free energy landscape $\Delta G(V_1, V_2)$ (in units of $k_B T$) of the folding of HP35 obtained from a PCA using (a) the backbone dihedral angles of the protein (dPCA) and (b) the Cartesian coordinates of all backbone atoms (cPCA), respectively.

Applications

Folding of penta-alanine: dihedral angle PCA of 100 ns trajectory (PROTEINS: Structure, Function, and Bioinformatics 58:45-52, 2005).

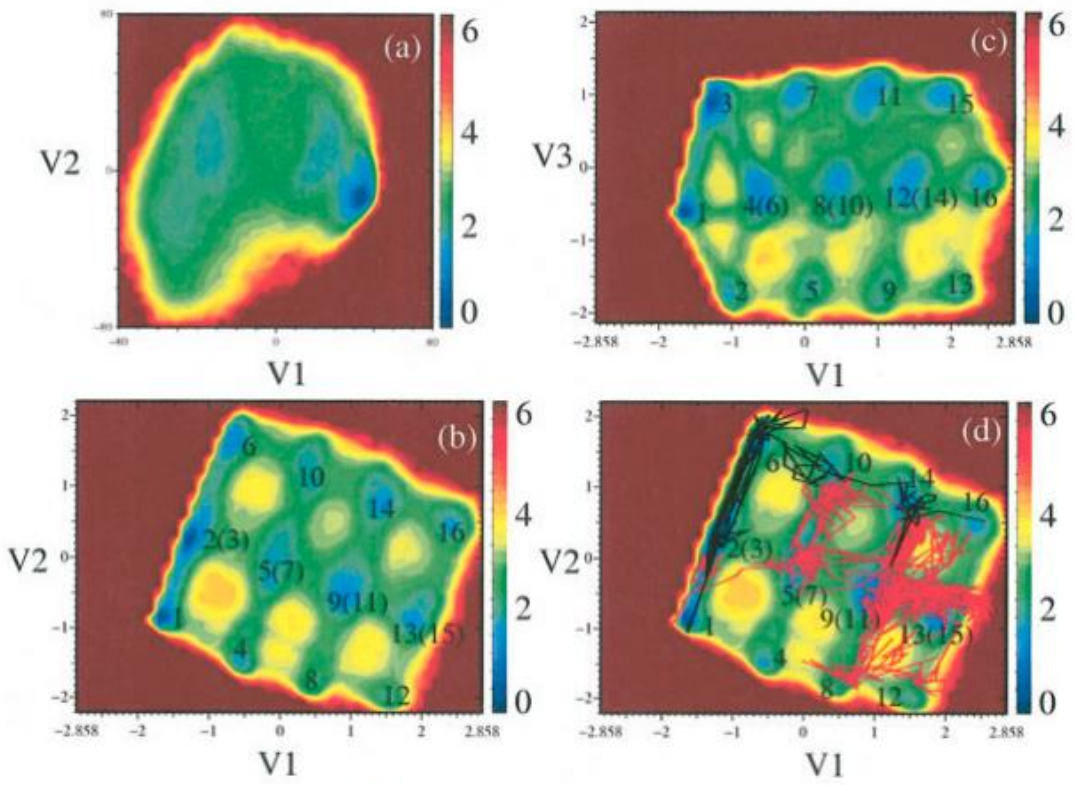


Fig. 2. Free energy landscape (in kcal/mol) of penta-alanine as obtained from various PCAs of the 100 ns MD simulation. Compared are the results of (**a**) the standard PCA using Cartesian coordinates and (**b**–**d**) the dPCA using dihedral angles. In (**a**), (**b**), and (**d**) the energy is plotted in the plane of the first two PCA eigenvectors v_1 and v_2, in (**c**) the first and third PCA eigenvectors v_1 and v_3 are used. The red and black lines in (**d**) reflect two representative folding pathways of the peptide.

13

Applications

Reaction sites surrounded by solvents.

Rescaling mass of solvent: lessening the effect of solvent.

JPCB 123, 5483, 2019.

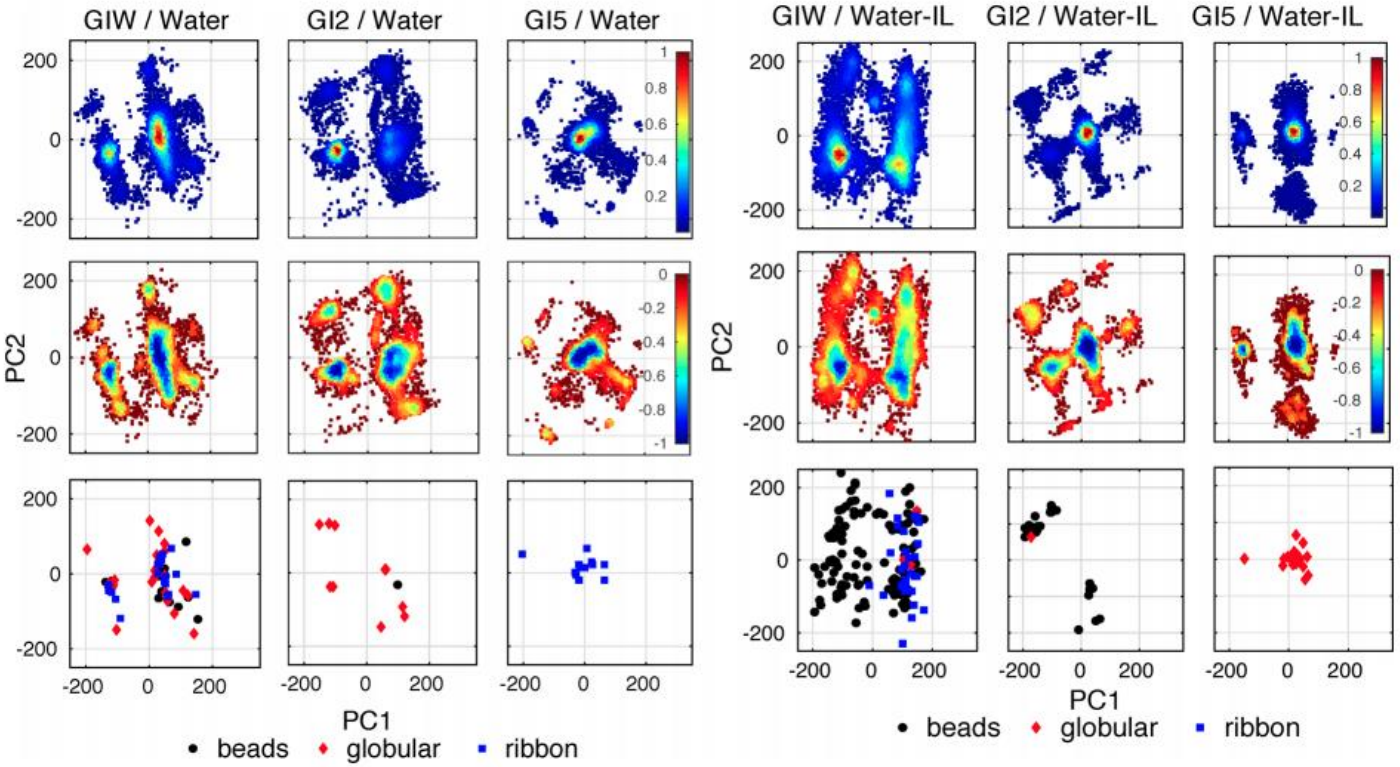


Figure 1. (i) Top: Scores of the REMD snapshots in the dynamical space spanned by the first and second principal components, PC1 and PC2, respectively, for the three peptides in two solvents. (ii) Middle: Corresponding normalized total internal energies (peptide plus solvent) projected on the PC space for all the cases. (iii) Bottom: Snapshots satisfying the S−S distance cutoff criteria for any of the three disulfide bond isoforms and their corresponding locations in the PC1−PC2 dynamical space.

14

Applications

PCA of gas phase reactions.
Chem. Sci. 10, 9954, 2019.
PathReducer provides interesting pre-processing and visualization features.

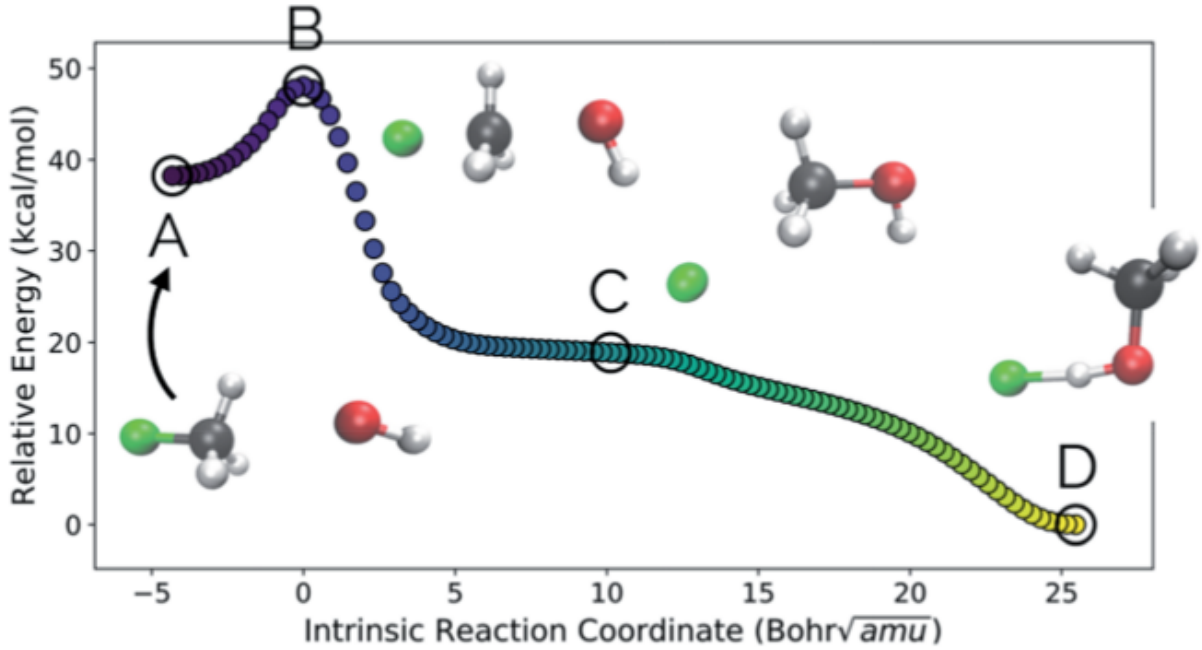


Fig. 5 The IRC for the S_N2 system. Structures A, B, and D represent the reactant, transition state, and product structures (after the fluoride ion orbits the system to hydrogen bond with the hydroxyl group), respectively. Structure C represents the structure where PC2 is at a minimum (Fig. 6), which can be thought of as the structure of the system when the fluoride ion has fully dissociated from the methanol, but has not yet come back to hydrogen bond with the hydroxyl group.

15

Using PCA on BioHPC

PCA is a widely used feature in the sklearn.decomposition.

It can be run easily in a Python 3 environment.

Through the demonstrated examples, we present the utilization of PCA on BioHPC:

- A conda environment (*pca-test*) is created.
- We provide a jupyter notebook that consists of the example code.
- We also present some data pre-processing method that are useful for PCA.