CS612 - Algorithms in Bioinformatics

Dimensionality Reduction Methods

May 8, 2023

Dimensionality Reduction – Motivation

- The computations required to simulate protein motion in silico are computationally expensive and involve non-trivial energy calculations.
- These simulations provide us with the (x, y, z) positions of all the atoms in the molecule.
- For a molecule with N atoms, this amounts to 3N numbers per conformation.
- For large molecules (such as proteins) the number of atoms is large and thus the dimensionality of the obtained data is extremely high.
- That is, a conformation sample for a protein with N atoms can be thought of as a 3N-dimensional point.

Dimensionality Reduction – Motivation

- However, many biological processes are known to be very structured at the molecular level, since the atoms self-organize to achieve their bio-chemical goal.
- An example of such a process is protein folding.
- To study such processes based on data gathered through simulations, there is a need to "summarize" the high-dimensional conformational data.
- Simply visualizing the time-series of a moving protein as produced by simulation packages does not provide a lot of insight into the process itself.
- One way is to turn conformations into a low-dimensional representation, such as a vector with very few components, that somehow give the "highlights" of the process.
- This data analysis process is called *dimensionality reduction*.

Dimensionality Reduction – Motivation

- When molecular motion is sampled, there is a need to simplify the high-dimensional (albeit redundant) representation of a molecule given as a 3N-dimensional point.
- It is believed that the actual degrees of freedom of the process are much less, as explained before.
- The resulting simplified representation are used to classify the different conformations along one or more "directions" or "axes" that provide enough discrimination between them.

Dimensionality Reduction – Motivation

- Dimensionality Reduction techniques aim at analyzing a set of points, given as input, and producing the corresponding low-dimensional representation for each.
- The goal is to discover the **true** dimensionality of a data set that is only apparently high-dimensional.
- There exist mathematical tools to perform automatic dimensionality reduction, based on arbitrary input data in the form of high-dimensional points (not just molecules).

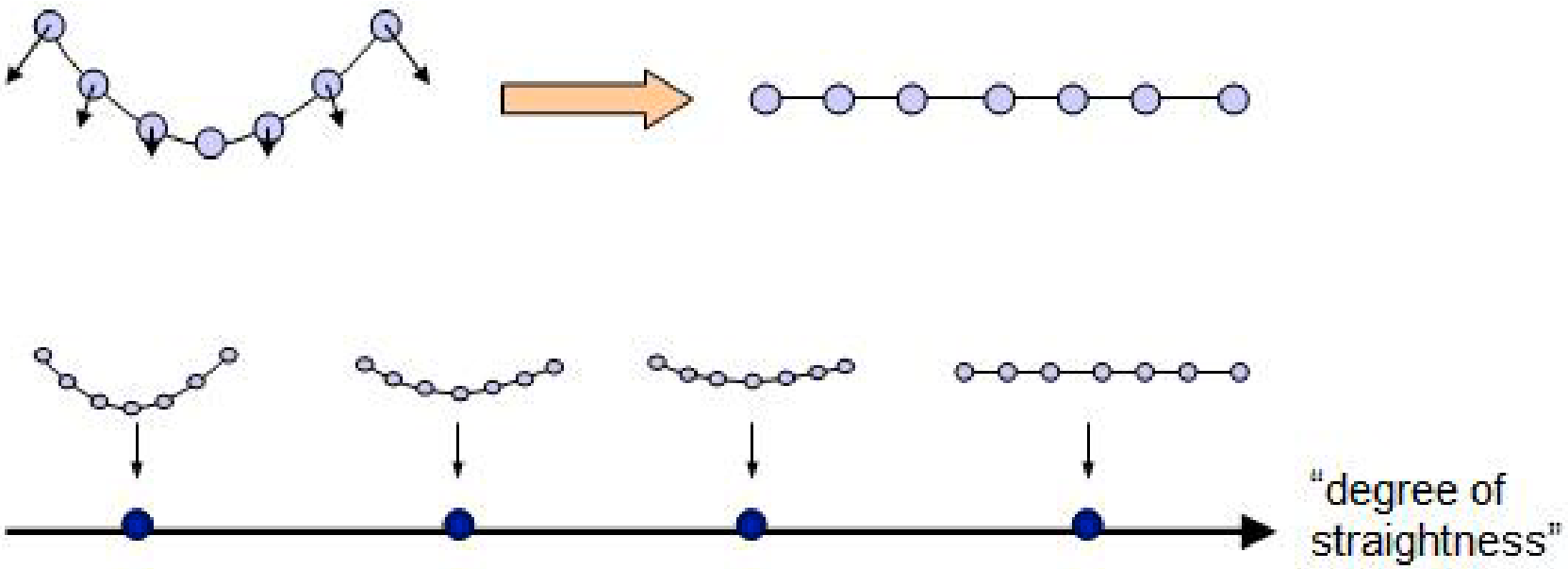
Dimensionality Reduction

- Although different techniques achieve their goals in different ways, and have both advantages and disadvantages, the most general definition for dimensionality reduction could be stated as follows:
  - INPUT: A set of M-dimensional points.
  - OUTPUT: A set of d-dimensional points, one for each of the input points, where $d << M$.
- Some dimensionality reduction methods can also produce other useful information, such as a "direction vector" that can be used to interpolate atomic positions continuously along the main motions (like in PCA, which will be discussed later).

A Simple Example

- As a simple example of dimensionality reduction, consider the case of a bending string of beads.
- The input data has $3x7 = 21$ dimensions (if given as the (x, y, z) coordinates of each bead) but the beads always move **collectively** from the "bent" to the "straight" arrangement.
- Under this simpified view, the process can be considered as one-dimensional, and a meaningful axis for it would represent the "degree of straightness" of the system.
- Using this axis, each string of beads can be substituted by one single number, its "coordinate" along the proposed axis.
- Thus, the location of a shape along this axis can quickly indicate in what stage of the bending process it is.

A Simple Example



Dimensionality Reduction in Molecules

- When dimensionality reduction methods are applied to molecular motion data, the goal is to find the main "directions" or "axes" collectively followed by the atoms, and the placement of each input conformation along these axes.
- The meaning of such axes can be intuitive or abstract, depending on the technique used and how complex the system is. We can reword the definition of dimensionality reduction when working with molecular motion samples as:
  - INPUT: A set of molecular conformations sampled from some physical process, given as the (x, y, z) coordinates for each atom. These are 3N-dimensional points for a molecule with N atoms.
  - OUTPUT: A set of d coordinates for each input conformation, such that $d << 3N$. These d coordinates should help classify the conformations throughout the main stages of the studied process.

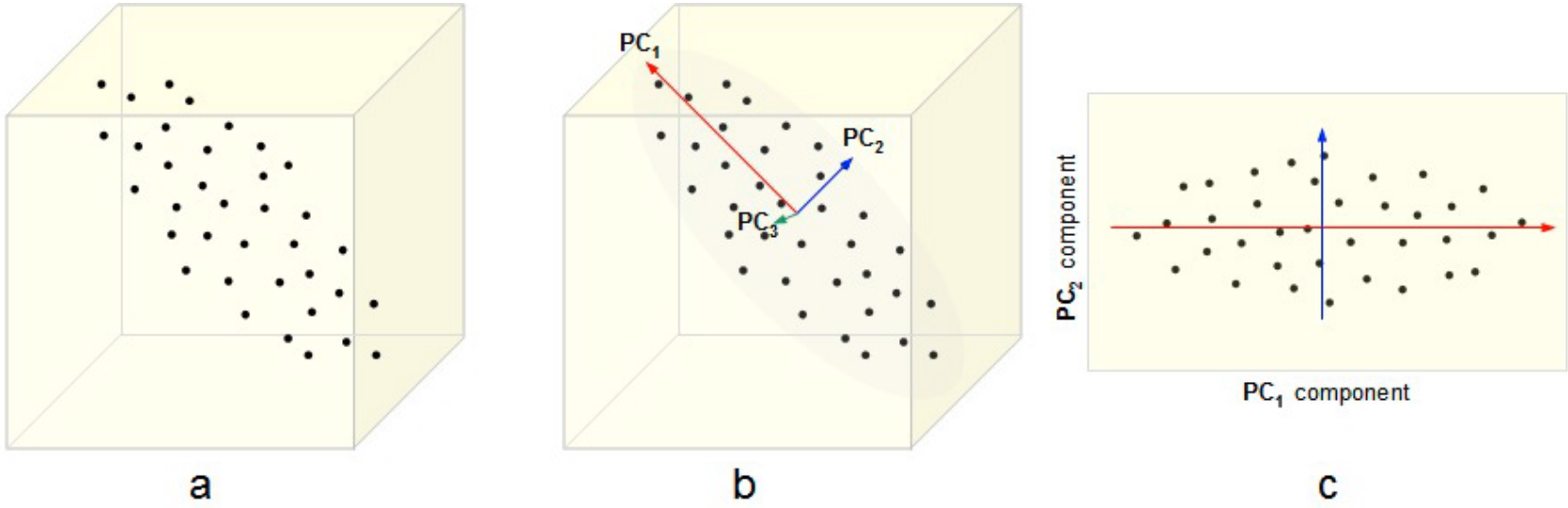
Dimensionality Reduction in Molecules

- Dimensionality reduction methods can be either linear or non-linear.
- Linear methods typically compute the low-dimensional representation of each input point by a series of mathematical operations involving linear combinations and/or linear matrix operations.
- Non-linear methods use either non-linear mathematics or modify linear methods with algorithmic techniques that encode the data's "curvature" (such as Isomap, explained later).
- Both categories of methods have advantages and disadvantages, which will become clear through the rest of this lecture.

Principal Component Analysis (PCA)

- Each point in this simple data set is given as a 3-dimensional vector (x, y, z)
- The discussion will later be turned to the molecular motion domain, and the interpretation of such data.
- In the example below, even though this data set is given as 3-dimensional points, it is obvious that the data points are distributed mostly on a two-dimensional surface.
- Our objective is then to find the inherent, 2-dimensional parameterization of this data set.
- For the simple example, the reader can agree that the last direction of maximum variance, the 3rd in this case, accounts for little or no data variability.
- Therefore we discard the components that do not add a significant contribution to the data variance.

PCA Example



PCA Overview

- For data points of dimensionality M, the goal of PCA is to compute M so-called Principal Components (PCs), which are M-dimensional vectors that are aligned with the directions of maximum variance (in the mathematical sense) of the data. These PCs have the following properties:
  - The PCs are ordered by data variance. In other words, the first PC is aligned with the direction of maximum variance, the second PC in the next direction contributing to the most variance, and so on.
  - The PCs form an orthonormal basis, that is, they are all mutually perpendicular and have unit length. This gives PCs the useful property of being uncorrelated.

PCA Main Uses

- Project the input data onto the PCs: The dot product of an input data point with any PC returns the scalar value of the projection of the point onto the PC.
- Since the PCs have unit length, this projection serves as the coordinate of the input point along the PC in question.
- In principle, M-dimensional input data can be projected onto its M PCs, but typically we using just the first d PCs as a basis and computing the projections onto them to get the best d-dimensional representation for each point.
- This is the actual dimensionality reduction.

PCA Main Uses

- Interpolate or synthesize new points: The PCs themselves point in the direction of maximum variance.
- For this reason, PC_i can be used as a direction vector along which new points can be synthesized by choosing parameter values a_i and then producing artificial M-dimensional points by doing the linear combination $a_1 PC_1 + a_2 PC_2 + \ldots$.
- Points synthesized in this way would lie approximately on the low-dimensional hyperplane spanned by the original data set.
- The projections of the original points correspond to particular values for these new "coordinates" a_i.
- Being able to interpolate other points not in the original data set is a useful property that other dimensionality reduction methods do not have.

Summary of PCA Stages

1. Let the input data consist of n observations x_i, each of dimensionality M. Construct an $n \times M$ matrix X of centered observations by subtracting the data mean from each point, so that $X_{ij} = x_{ij} - < x >_j$
2. Construct the covariance matrix $C = XX^T$
3. Compute the top d eigenvalues and corresponding eigenvectors of C, for example by performing an SVD of C.
4. The first d PCs of the data are given by the eigenvectors, which can be placed in a $d \times M$ matrix P. The residual variance can be computed from the eigenvalues as explained above.
5. To project the original (centered) points into the optimal d-dimensional hyperplane, compute the dot product of each point with the PCs to obtain the projections y_i. This can be written as $Y = P^T X$.

PCA Outline

- The data has to be centered in every direction first.
- If new points need to be synthesized using the PCs, the centroid can be added to place the newly synthesized points in the correct region of space.
- To compute the principal components, let X be an $n \times M$ matrix that contains n M-dimensional data points in its columns, centered at the origin
- **Goal:** find P, an $M \times M$ orthonormal transformation containing the PCs, such that:
  - $Y = P^T X$, where the columns of Y are the projections onto the PCs.
  - $PP^T = I$, that is, P is orthonormal.
  - $YY^T = D$, the covariance matrix of the projected points Y, is a diagonal matrix, so that the resulting projections are uncorrelated.

PCA Outline

- The resulting covariance matrix YY^T can be re-written as:

$$YY^T = (P^T X)(P^T X)^T = P^T(XX^T)P$$

- We want YY^T to be a diagonal matrix D so we can write: $YY^T = P^T(XX^T)P = D$
- Multiplying by P to the left and P^T to the right we get: $XX^T = PDP^T$
- Remember that since P is orthonormal, $PP^T = I$
- Applying SVD on XX^T yields: $XX^T = VSW^T$
- Where V and W are the left and right eigenvectors of XX^T, and S is a diagonal matrix with the eigenvalues.
- In this case the left and right eigenvalues coincide since XX^T is a symmetric matrix by construction, so we can write: $PDP^T = VSV^T$.

PCA Outline (cont.)

- The above, plus the fact that P and V are orthonormal, means that $P = V$ and $D = S$ (because both D and S are diagonal)
- So, the PCs are given by the eigenvectors of the centered covariance matrix XX^T.
- Moreover, the diagonal matrix of eigenvalues, S, is equal to the matrix D, which is the covariance of the projected points Y.
- Since it is a diagonal matrix, the diagonal contains the variance of the projected data set along each dimension.
- SVD usually sorts the eigenvectors by decreasing order of variance.

The Meaning of the PCs

- The eigenvalues eigenvalues actually correspond to the variance along each PC.
- By computing the ratio of each eigenvalue s_i to the total sum, one can obtain the fraction of total variance explained by each PC when the data is projected onto them.
- Subtracting the sum of variance fraction for the first d PCs from 1, we can obtain the *residual variance* r_d – the amount of variance in the original data left unexplained by discarding the PCs corresponding to the lower M-d eigenvalues:

$$r_d = 1 - \sum_{i=1}^{d} \frac{s_i}{\sum_{j=1}^{M} s_j}$$

- Which is a typical measure of the error made by approximating the data set using only the first d principal components

PCA of Conformational Data

- We can apply PCA to a set of molecular conformations, which will serve as our high-dimensional points.
- The input dimensionality of each point is $3N$, where N is the number of atoms in the molecule.
- We have n such conformations, that have been gathered through sampling (for example MD simulations), and we want to reduce the dimensionality of each "point" (conformation) for analysis purposes.
- The data used as input for PCA is in the form of several atomic position vectors corresponding to different structural conformations which together constitute a vector set.
- Each vector in the conformational vector set has dimension 3N and is of the form $[x_1, y_1, z_1, x_2, y_2, z_2, \ldots, x_N, y_N, z_N]$, where $[x_i, y_i, z_i]$ corresponds to the Cartesian coordinates of the i^{th} atom.

PCA of Conformational Data

- First, all conformations need to be aligned with a reference structure.
- It is important because usually molecular conformations that result from simulations are similar in shape but translated/rotated with respect to each other.
- Aligning all conformations to the same reference structure yields comparable results in general.
- The PCA procedure can then be used exactly as detailed above.
- The first step is to determine the average conformation that contains the average for all 3N dimensions of the data set.

PCA of Conformational Data

- It is important to note that this "average conformation" is not a physically feasible molecule.
- It is used to center the data.
- The PCs can be used to obtain a low-dimensional representation of each point and to synthesize (or interpolate) new conformations by following the PCs.
- The PCs represent the "main directions" followed collectively by the atoms.
- Interpolating along each PC makes each atom follow a linear trajectory, that corresponds to the direction of motion that explains the most data variance.

PCA of Conformational Data

- For this reason, the PCs are often called Main or Collective Modes of Motion.
- Interpolating along the first few PCs has the effect of removing atomic "vibrations" that correspond to the least important modes of motion.
- It is now possible to define a lower-dimensional subspace of protein motion spanned by the first few PCs and project the initial high dimensional data onto this subspace.
- Since the PCs are displacements (and not positions), in order to interpolate conformations along the main modes of motion one has to start from one of the known structures and add a multiple of the PCs as perturbations.

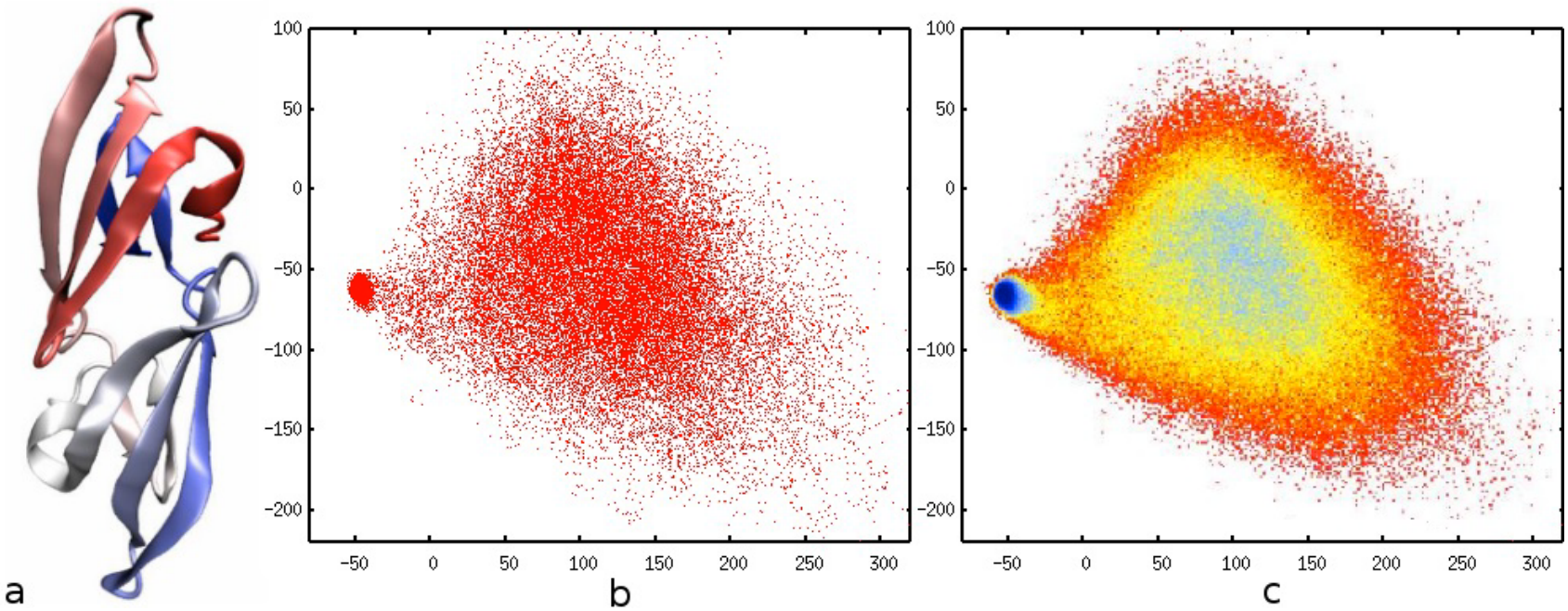
PCA of Conformational Data

- in order to produce conformations interpolated along the first PC one can compute $c + \lambda_1 PC_1$ where c is a conformation from the (aligned) data set and λ_1 adds a deviation from the structure c along the main direction of motion.
- The parameter can be either positive or negative.
- However, large values of the interpolating parameter will start stretching the molecule beyond physically acceptable shapes, since the PCs make all atoms follow *straight lines* and will fairly quickly destroy the molecule's topology.
- A typical way of improving the physical feasibility of interpolated conformations is to subject them to a few iterations of energy minimization.

PCA of Conformational Data

- PCA is typically used to determine the smallest number of uncorrelated principal components that explain a large percentage of the total variation in the data, as quantified by the residual variance.
- The exact number of principal components chosen is application-dependent and constitutes a truncated basis of representation.
- The example shows Cyanovirin-N (CV-N) protein (2EZM).
- This protein only has 101 amino acids and using $C\alpha$ based simulation yields 303 degrees of freedom.
- Folding/unfolding simulations starting from the native PDB structure produce abundant conformation samples of CV-N along the folding reaction.

PCA Example


a
100
50
0
-50
-100
-150
-200
-50
0
50
100
150
200
250
300
b
100
50
0
-50
-100
-150
-200
-50
0
50
100
150
200
250
300
c

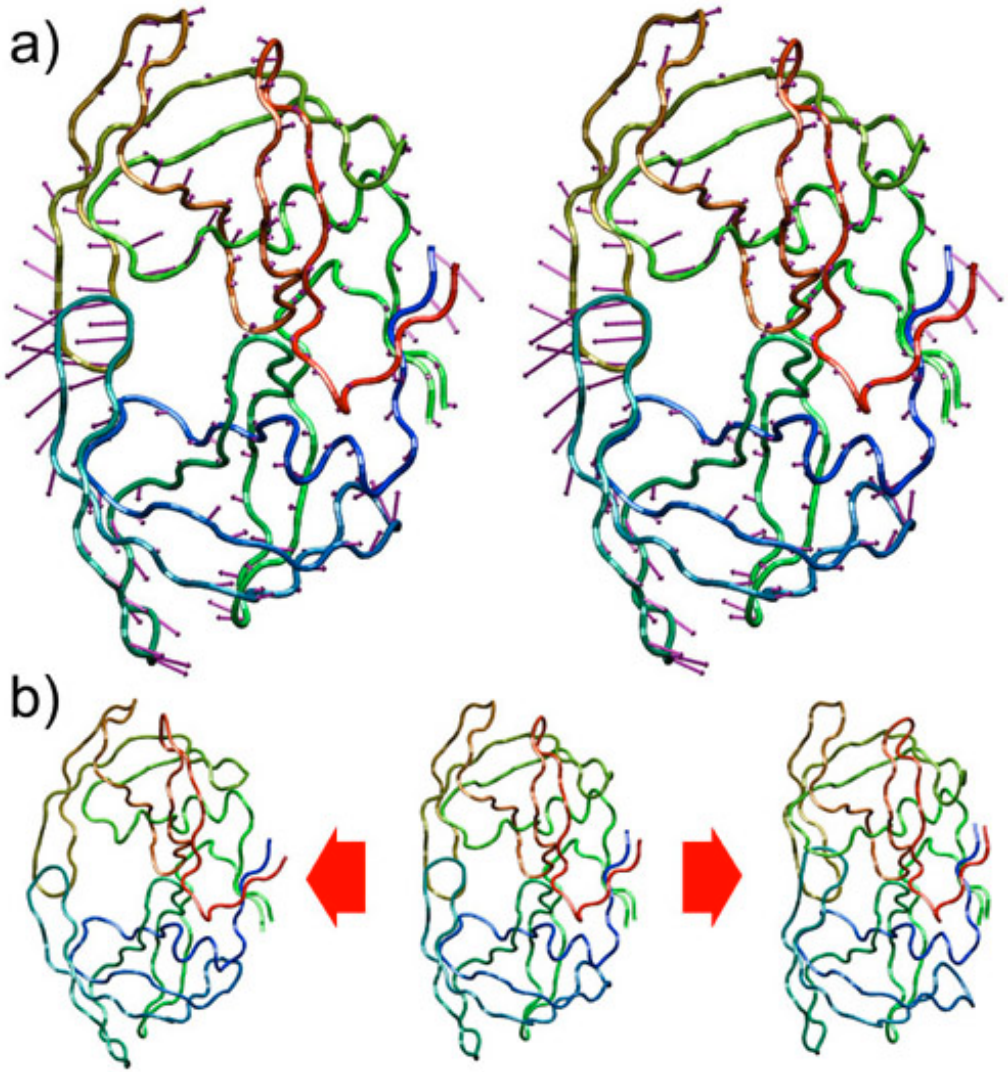
Example – HIV Protease

- The HIV-1 protease plays a vital role in the maturation of the HIV-1 virus by targeting amino acid sequences in the gag and gag-pol polyproteins.
- The active site of HIV-1 protease is formed by the homodimer interface and is capped by two identical beta-hairpin loops from each monomer, which are usually referred to as "flaps".
- The active site structure for the bound form is significantly different from the structure of the unbound conformation.
- In the bound state the flaps adopt a closed conformation acting as clamps on the bound inhibitors or substrates, whereas in the unbound conformation the flaps are more open.
- A backbone-only representation of HIV-1 has 594 atoms, which amounts to 1,782 degrees of freedom for each conformation.
- This is the input to the simulation.

Example – HIV Protease

- Applying PCA to a set of HIV-1 samples from simulation produces 1,782-dimensional principal components.
- Since the physical interpretation of the PCs is quite intuitive in this case, the PC coordinates can be split in groups of 3 to obtain the (x, y, z) components for each of the 594 atoms.
- These components are 3-dimensional vectors that point in the direction each atom would follow along the first PC.
- Note that the first mode of motion corresponds mostly to the "opening" and "closing" of the flaps
- Thus, interpolating in the direction of the first PC produces an approximation of this motion, but using only one degree of freedom.
- This way, the complex dynamics of the system and the 1,782 apparent degrees of freedom have been approximated by just one, effectively reducing the dimensionality of the representation.

PCA Example

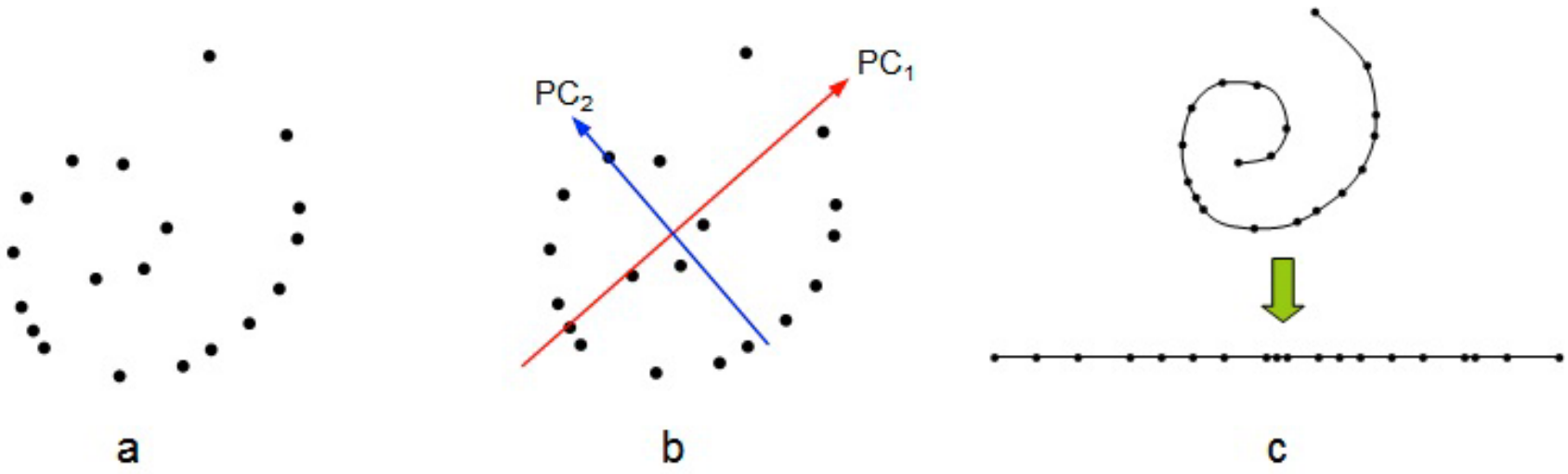


Non-Linear Dimensionality Reduction

- PCA is a linear method, since the PCs are computed as a series of linear operations on the input coordinates.
- Linear methods work well only when the collective atom motions are small (or linear), which is hardly the case for most interesting biological processes.
- Non-linear dimensionality reduction methods are normally much more computationally expensive and have other disadvantages as well.
- However, they are much more effective in describing complex processes using much fewer parameters.

Non-Linear Example

- The data in the following example is apparently two-dimensional, and naively considering the data variance in this way leads to two "important" principal components.
- However, the data has been sampled from a one-dimensional, but nonlinear, process.



Non-Linear Dimensionality Reduction

- Most interesting molecular processes are low-dimensional but highly nonlinear in nature.
- For example, in protein folding the atom positions follow very complicated, curved paths to achieve the folded shape.
- However, the process can often still be thought of as mainly one-dimensional.
- Linear methods such as PCA would fail to correctly identify collective modes of motion that do not destroy the protein when followed.
- Several non-linear dimensionality reduction techniques exist, that can be classified as either parametric or non-parametric.

Parametric vs. Non-parametric Method

- **Parametric methods** need to be given a model to try to fit the data, in the form of a mathematical function called a kernel.
- Kernel PCA is a variant of PCA that projects the points onto a mathematical hypersurface provided as input together with the points.
- When using parametric methods, the data is forced to lie on a supplied surface, so in general this does not work well with molecular motion data.
- **Non-parametric methods** use the data itself in an attempt to infer the non-linearity from it
- The most popular methods are Isometric Feature Mapping (Isomap) Locally Linear Embedding (LLE).

Isometric Feature Mapping (Isomap)

- is based on an improvement over a previous technique known as MultiDimensional Scaling (MDS).
- Isomap aims at capturing the non-linearity of a data set by computing relationships between neighboring points.
- **MDS** is a technique that produces a low-dimensional representation for an input set of n points, where the dimensions are ordered by variance, so it is similar to PCA in this respect.
- However, MDS requires as input a data set of points, and a *distance measure* $d(i, j)$ between any pair of points x_i and x_j .

Multidimensional Scaling (MDS)

- MDS first computes all the pairwise distances between the input points, and creating a matrix D so that $D_{ij} = d(i, j)$.
- The goal is to produce, for every point, a set of n Euclidean coordinates such that the Euclidean distance between all pairs match as close as possible the original pairwise distances D_{ij}.
- If the distance measure between points has the metric properties, then n Euclidean coordinates can always be found such that the Euclidean distance between them matches the original distances.
- Such coordinates are assumed to be centered around 0.

Multidimensional Scaling (MDS)

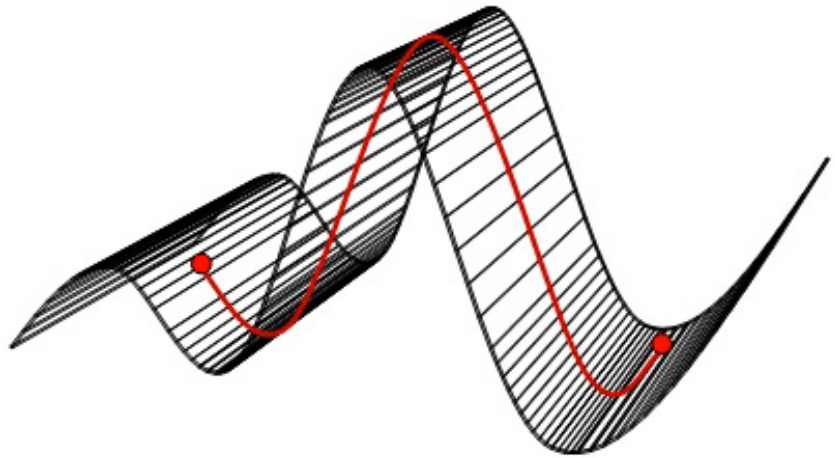
- The matrix of pairwise distances D can then be converted into a matrix of dot products by squaring the distances and performing a "double centering" on them to produce the matrix B of pairwise dot products $B = -\frac{1}{2} H_n D^2 H_n$
- $H_n = I_n - \frac{1}{n} 11^T$ is a centering matrix (11^T is a $n \times n$ matrix of all 1's).
- Let's assume that n Euclidean coordinates exist for each data point and that such coordinates can be placed in a matrix X.
- Then, multiplying X by its transpose should equal the matrix B of dot products computed before.

Multidimensional Scaling (MDS)

- Finally, in order to retrieve the coordinates in the unknown matrix X, we can perform an SVD of B which can be expressed as: $B = XX^T = Q\Lambda Q^T$
- The left and right singular vectors coincide because B is a symmetric matrix.
- The diagonal matrix of eigenvalues can be split into two identical matrices, each having the square root of the eigenvalues, and a solution for X can be found as $X = Q\Lambda^{1/2}$.
- Just like PCA, the coordinates are ordered by variance.

Geodesic Distances

- The notion of "geodesic" distance was originally defined as shortest path between two points on the surface of the Earth
- The concept can be generalized to any mathematical surface, and defined as "the length of the shortest path between two points that lie on a surface, when the path is constrained to lie on the surface.



The Isomap Algorithm Outline

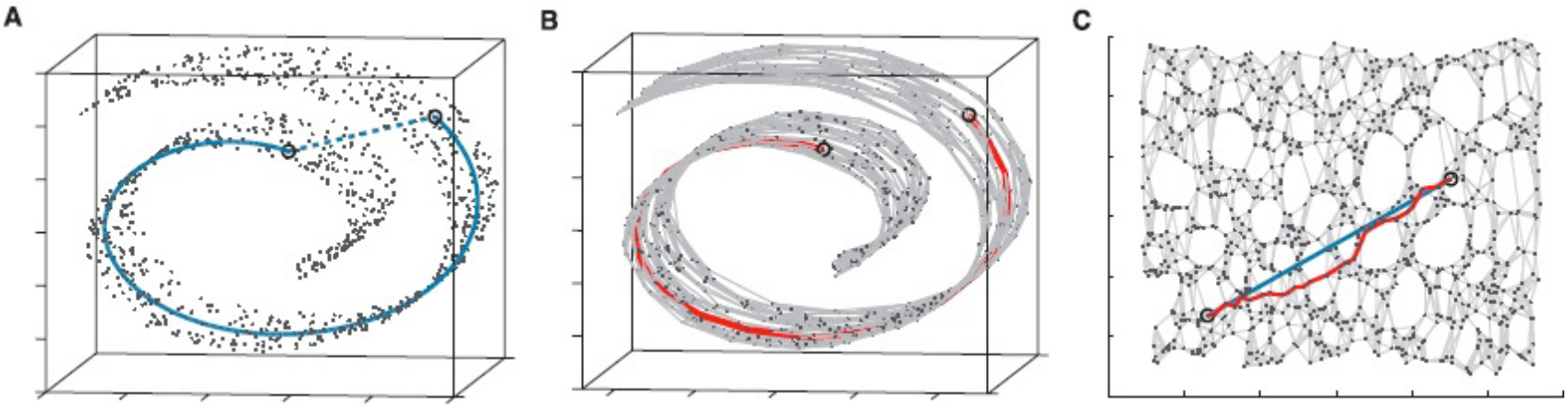
1. Build the neighborhood graph. Take all input points and connect each point to the closest ones according to the distance measure used. Different criteria can be used to select the closest neighbors, such as the k closest or all points within some threshold distance.
2. Compute all pairwise geodesic distances. These are computed as the shortest paths on the neighborhood graph. Efficient algorithms for computing all-pairs-shortest-paths exist, such as Dijkstra's algorithm. All geodesic distances can be put in a matrix D.
3. Perform MDS on geodesic distances. Take the matrix D and apply MDS to it. That is, apply the double-centering formula explained above to produce B, and compute the low-dimensional coordinates by computing B's eigenvectors and eigenvalues.

The Isomap Algorithm

- The Isomap method augments classical MDS with the notion of geodesic distance, in order to capture the non-linearity of a data set.
- Isomap uses MDS to compute few Euclidean coordinates that best preserve pairwise geodesic distances, rather than direct distances.
- Since the coordinates computed by MDS are Euclidean, these can be plotted on a Cartesian set of axes.
- The effect of Isomap is similar to "unrolling" the non-linear surface into a natural parameterization.
- Isomap approximates the geodesic distances from the data itself, by first building a neighborhood graph for the data.
- A neighborhood graph consists of the original set of points, together with a connection between "neighboring" points.

The Isomap Algorithm

- After the neighborhood graph has been built, it can be used to approximate the geodesic distance between all pairs of points as the shortest path distance along the graph.
- Naturally, the sampling of the data set has to be enough to capture the inherent topology of the non-linear space for this approximation to work.
- MDS then takes these geodesic distances and produces the Euclidean coordinates for the set.



Advantages and Disadvantages

- The Isomap algorithm captures the non-linearity of the data set automatically, from the data itself. It returns a low-dimensional projection for each point;
- These projections can be used to understand the underlying data distribution better.
- However, Isomap does not return "modes of motion" like PCA does, along which other points can be interpolated.
- Also, Isomap is much more expensive than PCA, since building a neighborhood graph and computing all-pairs-shortest-paths can have quadratic complexity on the number of input points.
- Plus, there is the hidden cost of the distance measure itself, which can also be quite expensive.

Applications to Bioinformatics

- In order to apply Isomap to a set of molecular conformations all that is needed is a distance measure between two conformations, for example IRMSD.
- There is no need to pre-align all the conformations in this case, since IRMSD already includes pairwise alignment.
- Thus, the Isomap algorithm as described above can be directly applied to molecular conformations.
- Choosing an appropriate value for a neighborhood parameter (such as k) may require experience, though, and it may depend on the data.
- It should be noted that we do not know, a priori, what the surface looks like.
- But we know that the process should be low-dimensional and highly non-linear in nature.
- The distance measure has an impact on the final coordinates as well.

CVN Example

- There is a clear difference between PCA and Isomap.
- Isomap identified an intermediate and folding route where PCA did not.
- However, Isomap is much more expensive.