Software Project Development

Assignment 3: Applications

Ioannis Z. Emiris

emiris@di.uoa.gr

December 2016

Outline

This project is comprised of two parts.

1. Clustering of proteins: dRMSD vector, cRMSD metric (Emiris)
2. Recommendation: vector (cosine-LSH), metric space (Chamodrakas)

Outline

1. Clustering of proteins

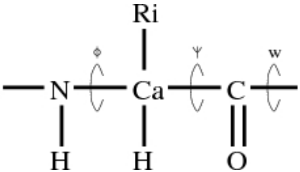
Importance of proteins

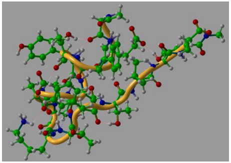
Dogma (or principle) of molecular biology:

aminoacid	$\longrightarrow$	3-dim	$\longrightarrow$	chemical
sequence	folding	structure	``surface"	function

Aminoacid (residue) sequence determines 3D (tertiary) structure (almost).

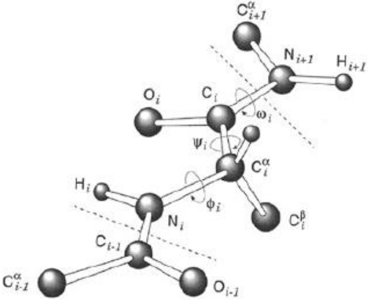
Proteins





Chains of aminoacids, from 20 choices. Each aminoacid consists of: backbone N-Cα-C, residue Ri attached to Cα, $i \in \{1, \ldots, 20\}$, Ri starts at C_i^{β}.

Structure determined by 3d coordinates of backbone atoms, basically Cα.



distance Root Mean Squared Deviation

Assume r distances $d_i, i = 1, \ldots, r$ are known between point-pairs in X and the corresponding pairs in Y, denoted by d'_i, $i = 1, \ldots, r$; clearly $r \leq \binom{n}{2}$.

Definition (d-RMSD)

There is a distance metric, namely d-RMSD, where

$$\text{distance-RMSD} = \sqrt{\frac{1}{r}\sum_{i=1}^{r}(d_i - d'_i)^2},$$

for r corresponding distances, $r \leq \binom{n}{2}$.

- Advantage: d-RMSD invariant under rigid transforms (incl. translation, rotation).

Vector of distances

Equivalent formulation

Let

$$v(X) = (d_1, \ldots, d_r), v(Y) = (d'_1, \ldots, d'_r) \in \mathbb{R}^r$$

be the vectors of distances in X, Y respectively. Then their Euclidean distance is

$$|v(X) - v(Y)|_2 = \sqrt{r} \cdot \text{ d-RMSD}(X, Y).$$

Subset of distances

- Use $r \leq \binom{n}{2}$ distances.
- Must correspond to the same pairs of points in all conformations.
- Typical choice 1: r uniformly selected pairs among $\binom{n}{2}$.
- Typical choice 2: smallest or largest distances, in one conformation.

coordinate Root Mean Square Deviation

Definition (c-RMSD)

Two sets of n corresponding points $x_i, y_i \in \mathbb{R}^3$, $i = 1, \dots, n$, expressing the backbone (C_α) atom coordinates in SAME coordinate frame. Then,

$$\text{c-RMSD} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} |x_i - y_i|^2}.$$

Equivalently: Let $X = [x_1, \dots, x_n]^T$, $Y = [y_1, \dots, y_n]^T \in \mathbb{R}^{n\times 3}$, then

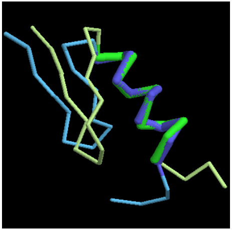
$$\text{c-RMSD}(X, Y) = \frac{1}{\sqrt{n}}|X - Y|_F, \quad \text{where} \quad |M|_F^2 = \sum_{i,j} M_{ij}^2 = \operatorname{tr}(M^T M)$$

is the Frobenius norm of M, $\operatorname{tr}(A) = \sum_i A_{ii}$ is the trace of square matrix A.

Optimal 3D Alignment

Definition (Problem)

Find (1) translation and (2) rotation minimizing c-RMSD.



1. Translate to common origin by subtracting the centroid from all $x_i \in X$:

$$x_c = \frac{1}{n} \sum_{i=1}^{n} x_i,$$

and by subtracting centroid y_c from all points y_i in "set" Y.

2. Rotate to optimal alignment by 3×3 rotation matrix Q.
By definition, $Q^T Q = I$, $\det Q = |Q| = 1$.

Recall rotated vector is $v^T Q$ or Qv, for column vector $v \in \mathbb{R}^3$.
Counter-clockwise rotation in the plane about the origin by θ:

$$Q = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}, \qquad Q \begin{bmatrix} x \\ y \end{bmatrix} = \text{rotated vector},$$

where $$Q^T Q = I, \det Q = |Q| = 1.$$

Rotation on 3D sphere by θ, α:

$$\begin{bmatrix} \cos\theta & -\sin\theta \cdot \cos\alpha & \sin\theta \cdot \sin\alpha \\ \sin\theta & \cos\theta \cdot \cos\alpha & -\cos\theta \cdot \sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}$$

Optimal rotation

Assume common centroid = 0, for pointsets $X, Y \in \mathbb{R}^{n\times 3}$:

$$\text{c-RMSD}(X, Y) = \min_Q |Y - XQ|_F,$$

for rotation matrix Q.

Lemma

Optimizing rotation $Q \in \mathbb{R}^{3\times 3}$ reduces to finding optimum

$$\max_Q tr(Q^T X^T Y), \qquad Q^T Q = I_3, \det Q = 1,$$

where we compute rotation matrix Q.

Proof. Linear algebra calculations.

Rotation by matrices

SVD (Singular value decomposition): $X^T Y = U \Sigma V^T$, where

$$U^T U = V^T V = I, \;\; \Sigma = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix} : \sigma_1 \geq \sigma_2 \geq \sigma_3,$$

where U, V, Σ are 3×3, singular values $\sigma_i = |e_i| \geq 0$, e_i eigenvalues of $X^T Y$.

We search for rotation Q maximizing $\mathrm{tr}(Q^T U \Sigma \cdot V^T) = \mathrm{tr}(V^T \cdot Q^T U \Sigma) \;\leq\; \mathrm{tr}(\Sigma)$.

Theorem

Maximum occurs at $V^T Q^T U = I \Leftrightarrow Q = U V^T$, for rotation matrix Q.

If $\det(UV^T) \simeq -1$, then negate 3rd column of U to define matrix W, return rotation $Q = WV^T$ (right-handed system).

Algorithm

Input: pointsets $X, Y \in \mathbb{R}^{n \times 3}$ of n corresponding points.

Output: minimum c-RMSD of translated and rotated sets.

$x_c \leftarrow \sum_{i=1}^{n} x_i / n, \;\; y_c \leftarrow \sum_{i=1}^{n} y_i / n.$

$X \leftarrow \{x - x_c : x \in X\}, \;\; Y \leftarrow \{y - y_c : y \in Y\}.$

SVD: $X^T * Y = U \Sigma V^T$.

Check: Confirm $\sigma_3 > 0$, where $\Sigma = \mathrm{diag}[\sigma_1, \sigma_2, \sigma_3]$.

$Q \leftarrow U * V^T$.

if $\det Q < 0$ **then** $Q \leftarrow [U_1, U_2, -U_3] * V^T$

end if // U_i : ith column

Return $|X * Q - Y|_F / \sqrt{n}$ // = $\sqrt{\sum_{i=1}^{n} |Q x_i - y_i|^2 / n}$

Implementation

- LAPACKE: (high-level) C Interface to LAPACK, `www.netlib.org/lapack/lapacke.html`.
  - `lapacke.h`: 2D arrays passed as pointers (to 1D array), and `int` $\in$ { `LAPACK_ROW_MAJOR, LAPACK_COL_MAJOR` }
  - Routines: `LAPACKE_xbase: x` $\in$ { `s,d` } for single, double precision, `base = gesvd` for SVD, `getrf` for LU decomposition (for det).
  - BLAS Support with `cblas.h: cblas_xgemm` computes $\alpha\, op(A) op(B) + \beta C$, $op(A)$ can be A or A^T.

Implementation (cont'd)

- GNU Scientific Library (GSL)
  - Vectors and Matrices: containers, `gsl_matrix_add`, `gsl_matrix_sub`.
  - BLAS Support: `gsl_blas_xgemm`
  - Linear Algebra: `gsl_linalg_LU_det`, `gsl_linalg_SV_decomp`
- EIGEN C++ library