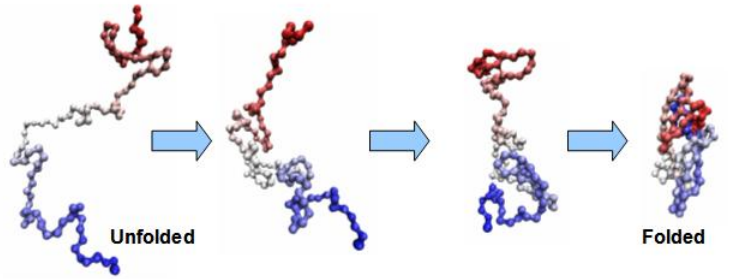
CS612 - Algorithms in Bioinformatics

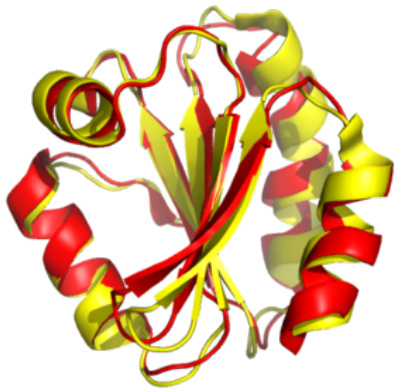
Structural Alignment

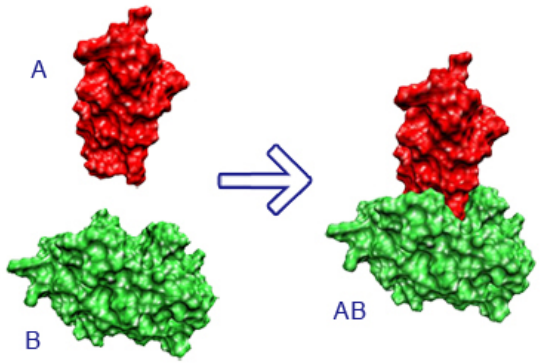
April 12, 2023

Problems in Structural Bioinformatics

- Protein folding.
- Protein structural alignment and motif discovery.
- Protein-protein docking.
- Protein-drug interaction.
- ...







Rapid Structural Analysis Methods – Structural Alignment

- Emergence of large structural databases which do not allow manual (visual) analysis and require efficient 3-D search and classification methods.
- Structure is much better preserved than sequence – proteins may have similar structures but dissimilar sequences.
- Structural motifs may predict similar biological function
- Getting insight into protein folding. Recovering the limited (?) number of protein folds.
- Comparing proteins of not necessarily the same family.

Structural Alignment

- Least RMSD calculation requires a correspondence.
- Major task in structural comparison – the correspondence between two matching proteins.
- When two conformations of the same protein – measure distance, otherwise – difficult.
- How do you compare two different proteins?
- In other words – find the optimal (sub)structural alignment of two proteins.

Protein alignment – Problem Definition

- Given two configurations of points in the three dimensional space, find those rotations and translations of one of the point sets which produce "large" superimpositions of corresponding 3-D points.
- Not necessarily looking for the largest match, but perhaps the most meaningful one.

Sequence Order Dependence

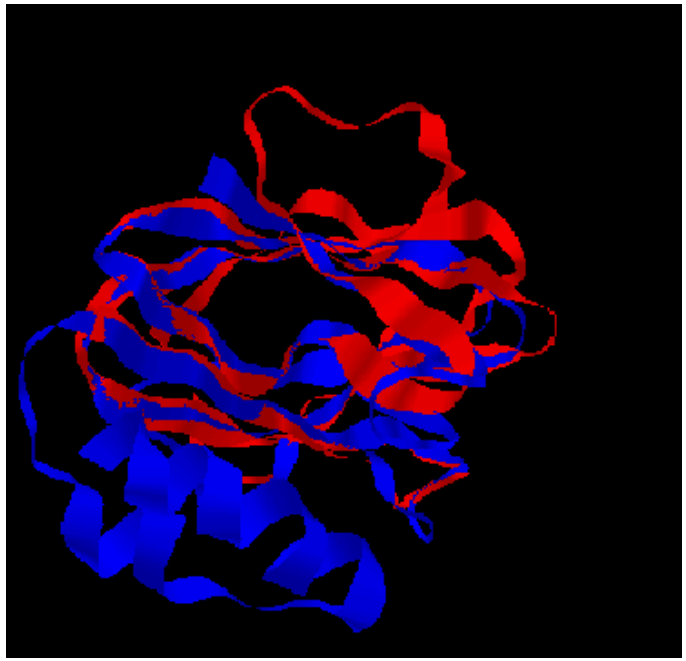
- Sequence order dependent alignment = 3-D curve matching – an inherently 1-D task.
- Sequence order independent alignment - a "real" 3-D task.
- Enables detection of non-sequential motifs in proteins, e.g. molecular surface motifs, especially, similar binding sites.
- Allows search of structural databases with only partial and disconnected structural information.
- Same algorithm applies to other molecular structures, e.g. drugs.

Sequence Order Dependence

- Motifs are small (ca. 3–20 aa) patterns that may have biological or functional significance.
- Motifs preserving sequence order might be biologically more meaningful than similar size non-sequential motifs.
- The computational task becomes much more complex, when sequence order is not exploited.

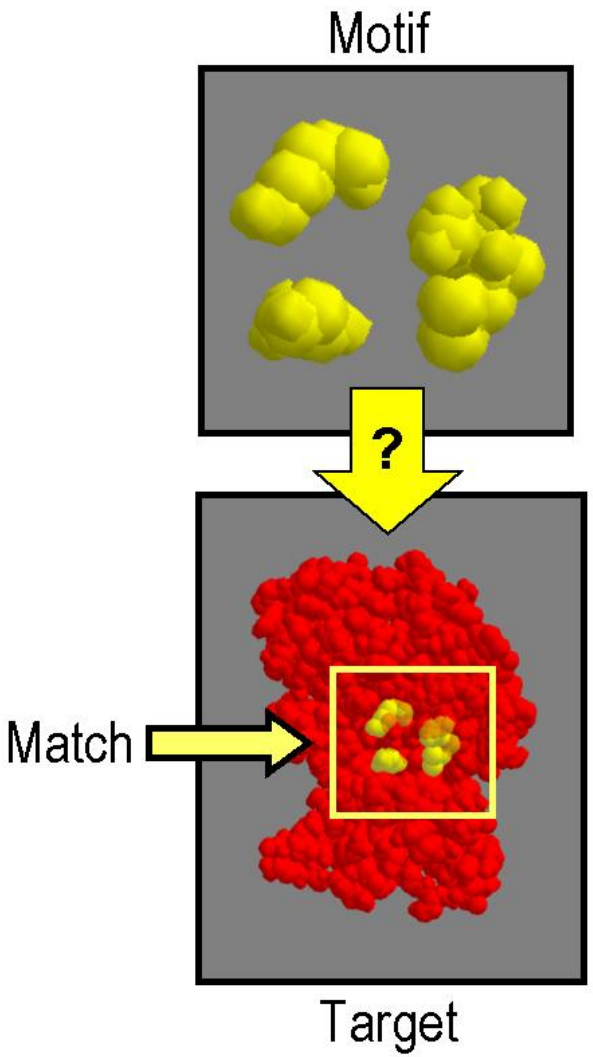
Global Structural Alignment

- Find the best overall alignment (good for protein classification).
- Dali (based on the alignment of pairwise contact matrices).
- LOCK (based on geometric hashing + dynamic programming).
- SSAP (sequential structure alignment program).
- ...



Local Structural Alignment

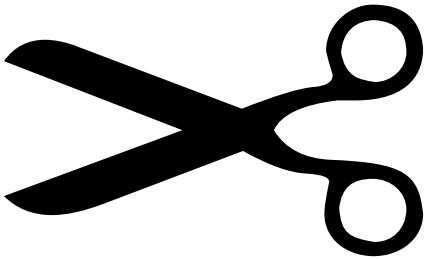
- Selecting a correspondence between a motif, a substructure of a protein, often between 3 and 20 amino acids, and a target, a full protein structure.
- Once a correspondence has been established, the "distance" of the motif to the identified part of the target is measured using IRMSD.

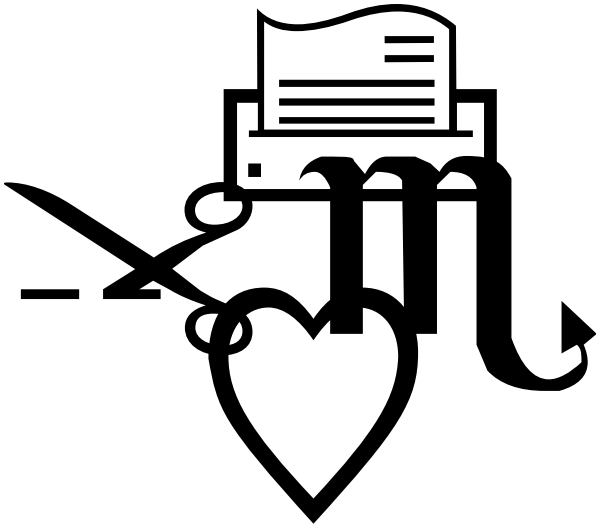


Geometric Hashing

Originally taken from computer vision – pattern matching.
Two stages:

1. Preprocessing stage: learn a model pattern.
2. Recognition stage: the system is exposed to a new pattern of points, the Target, from which it is to identify a subset of reasonable geometric similarity to the model.

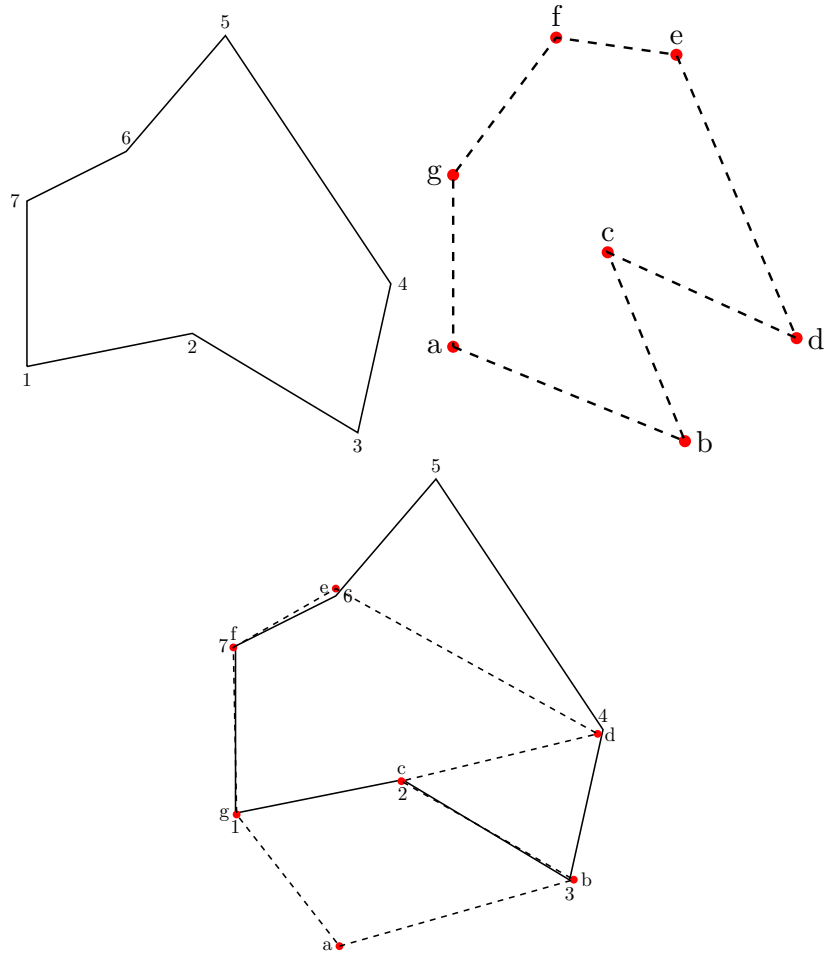




Lamdan, Wolfson 1988. Molecular biology adaptation – Wolfson and Nussinov, 1989.

A 2D Example

- Given a query (top) and an image (middle) we search for a (possibly transformed) copy of the set of points that has maximum overlap with the image.
- By overlap we mean – given two points, a model point a and a query point b, the distance between a and b is below a certain threshold.
- It can be seen that six out of the seven pairs of points overlap – the pairs are $(1, g), (2, c), (3, b), (4, d), (6, e), (7, f)$.



A 2D Reference Frame

- Two points uniquely define a rigid transformation in 2D.
- These points are the *basis* to the transformation. Such a transformation is also called a *reference frame*, since the rest of the system can be positioned (rotated and translated) with respect to it.
- Given two points a and b, build the reference frame as follows:
  - The origin lies on point a, so the translation vector is the coordinates of a.
  - The x direction lies on the line between a and b.
- That's all! The x axis is the normalized vector $||b - a||$.

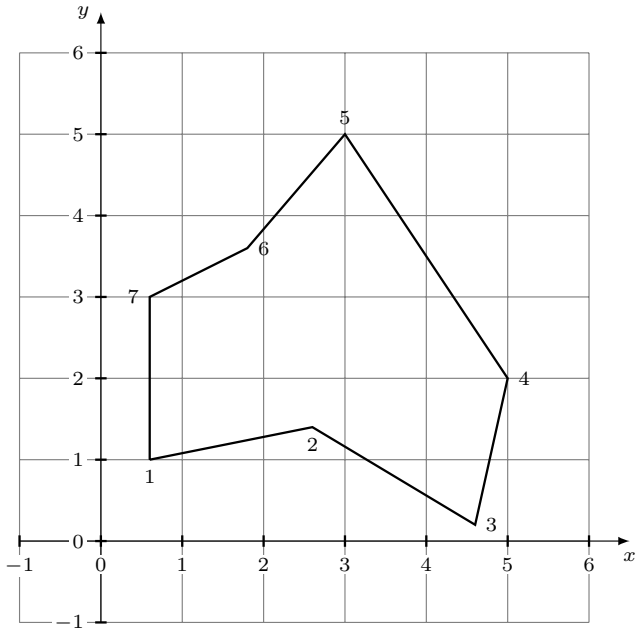
A 2D Reference Frame

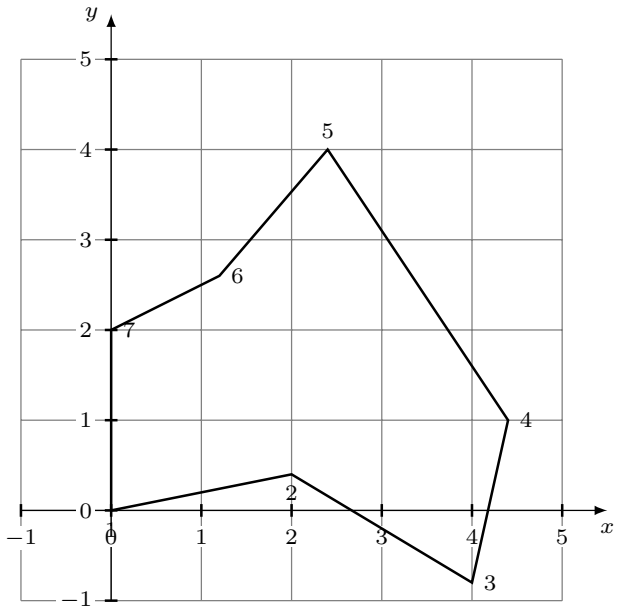
- The y axis is perpendicular to the x axis at the counter-clockwise direction. We can calculate its value using the fact that the two vectors constituting a 2D rotation matrix must be of the form:

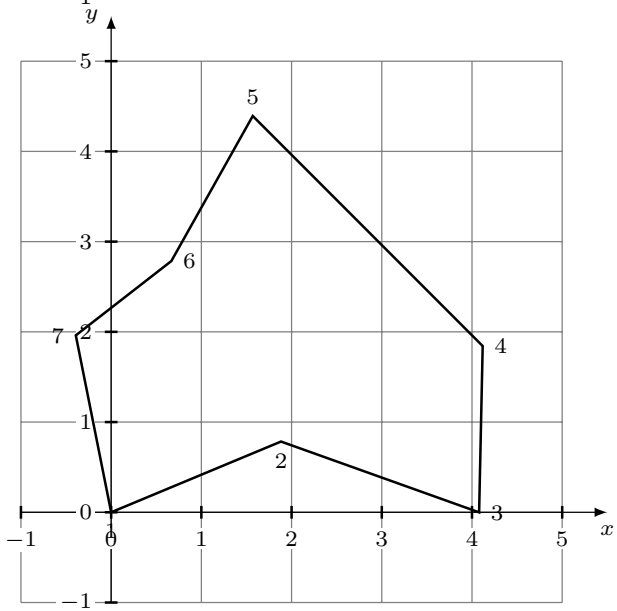
$$\begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \end{bmatrix}$$

- We can extract the magnitude of θ as $arcsin(x_2)$ or $arccos(x_1)$.
- The sign of θ is positive if x_2 is positive, negative if x_2 is negative and 0 if $x_2 = 0$.
- Based on that we can calculate y (We don't even need this. θ is enough).

A 2D Reference Frame







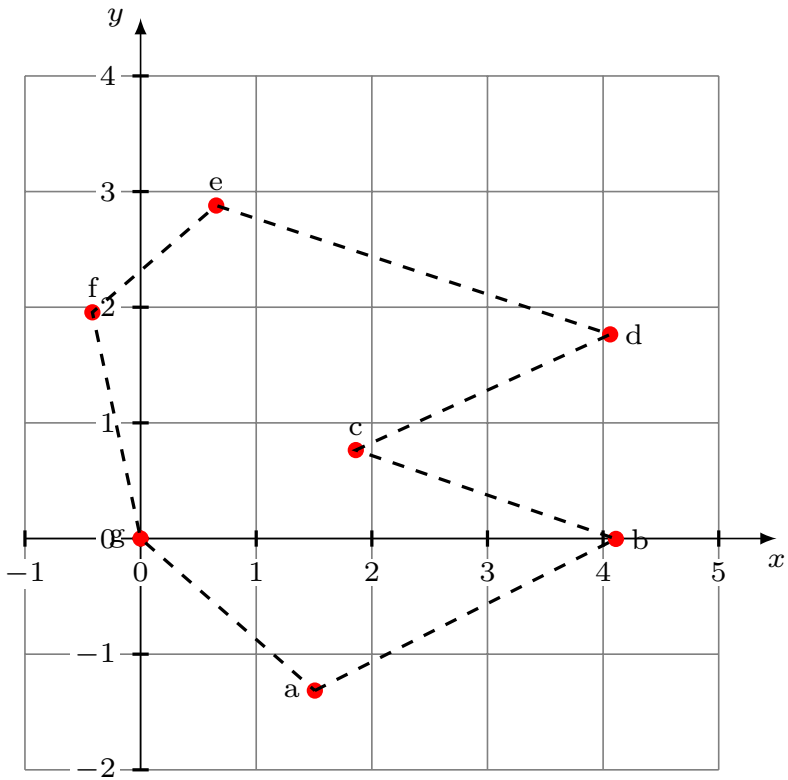
Transforming a Frame – Example

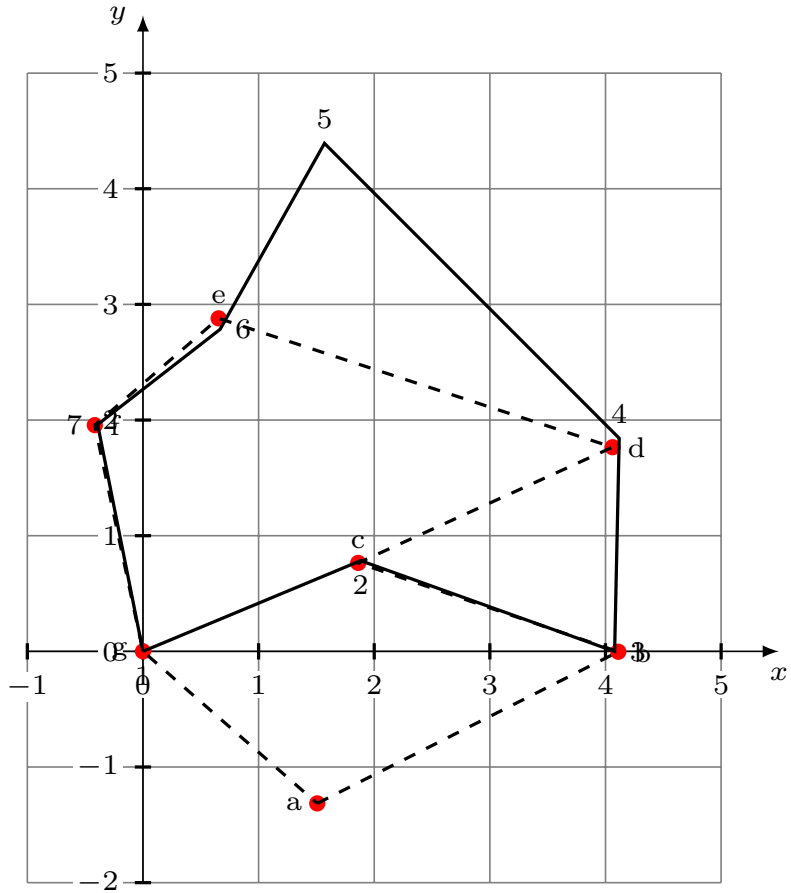
- The direction of the x axis is parallel to the vector $||b - a||$.
- The direction of the vector is $(b - a) = (20, -4)$. To normalize the vector, divide it by its magnitude which is $\sqrt{20^2 + 4^2} = \sqrt{416} = 20.4$.
- The normalized vector is $(0.981, -0.196)$.
- The rotation angle θ is $arcsin(-0.196) = -11.3°$.
- Notice that this is the angle the vector between points 1 and 3 makes with the x axis.
- In order to align this vector with the x axis, we have to rotate the shape $+11.3$ degrees to "rotate it back" into the x axis.

Ranking a Frame – Example

- To compare the model and query systems, we apply a transformation to the query.
- Let us select points g and b as our reference frame, calling the frame (g, b).
- The original coordinates for the two points are $(10, 20)$ for g and $(23.5, 4.5)$ for b. The translation vector is therefore $(-10, -20)$.
- The direction of the x axis is $(13.5, -15.5)$. The magnitude is 22.55 (normalized to $(0.657, -0.754)$).
- The rotation angle is $-48.9°$.
- Applying the same transformation to query, five pairs of points coincide out of the possible seven – $(1, g), (2, c), (3, b), (4, d), (7, f)$.

Ranking a Frame – Example





How Many Unique Transformations Should we Compare?

- The maximum number of transformations $n * (n-1)$ for the model and $m * (m-1)$ for the query.
- Comparing all against all results in $n * (n-1) * m * (m-1)$ possible transformations.
- Notice, however, that redundancies may happen,
- For example, let (a_i, a_k) and (b_j, b_l) be selected as base pairs from model a and query b, respectively.
- Say (a_r, b_u) and (a_s, b_v) both coincide, then it is likely that similar coincidence sets will be found if (a_r, a_s) and (b_u, b_v) are selected as base pairs.

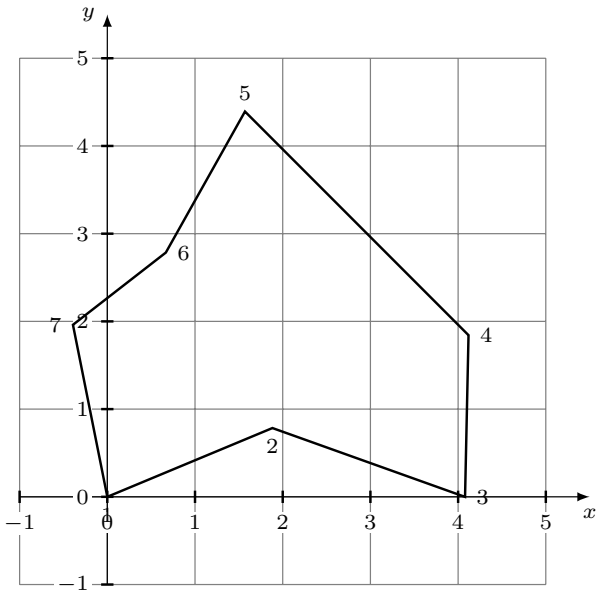
Hashing Transformations

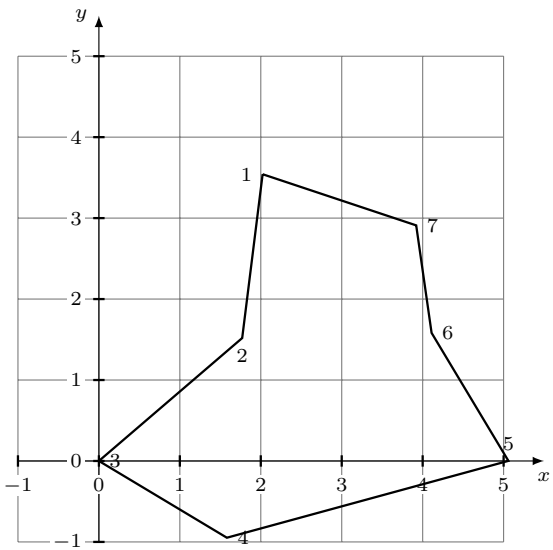
- Now that we have a way to transform the model and query shapes, we can utilize the power of hashing to conduct an efficient search for the transformation(s) that maximize the number of coinciding points.
- We construct a hash table that stores the geometric locations of the transformed points (hence the name Geometric Hashing).
- This allows us to simultaneously compare a query frame system to all the model frame systems.
- Geometric hashing has two stages, preprocessing and recognition

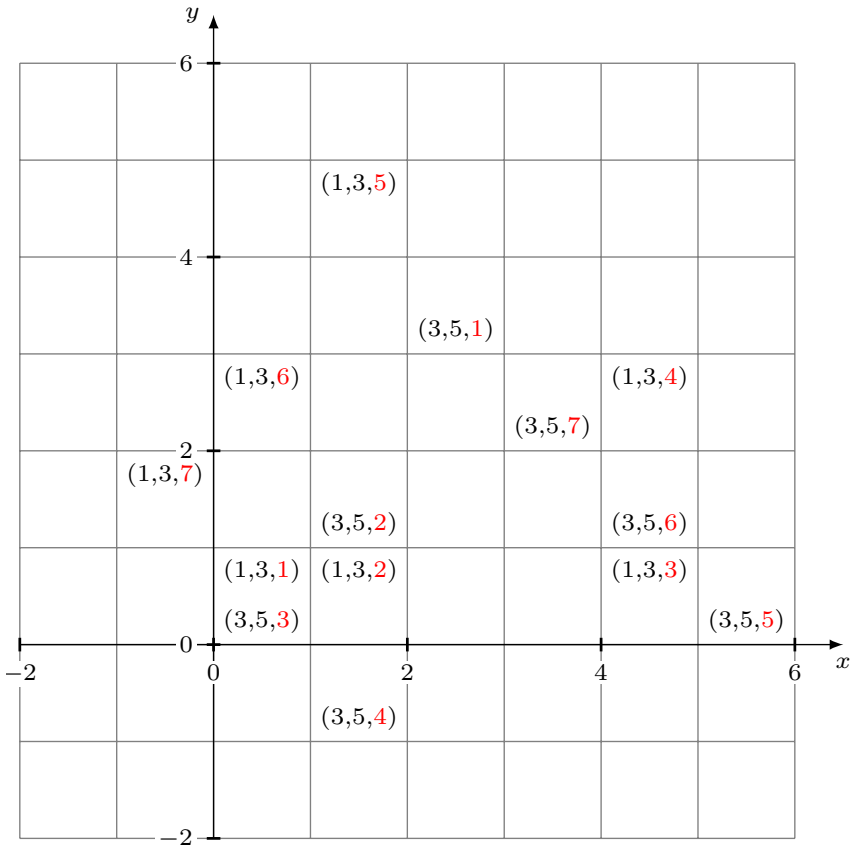
Preprocessing

- Build a hash table H, which has a bin for each cell in the frame system.
- The dimension of H is determined by the number of points that define a frame, two dimensional in our cases.
- You may think of H as a two dimensional coordinate system divided into a grid at a specified resolution.
- If there is a point in the cell (p, q) in the frame system with basis (a_i, a_k), then (a_i, a_k) is placed in the bin $H(p, q)$.
- Every entry in H contains the frame identifier and the point identifier.
- We calculate the reference frame in a similar way for each model basis and transform the model in each reference frame, inserting the transformed copies into the table.
- After this stage, H contains multiple transformed copies of the model. H may take up a lot of space.

Preprocessing



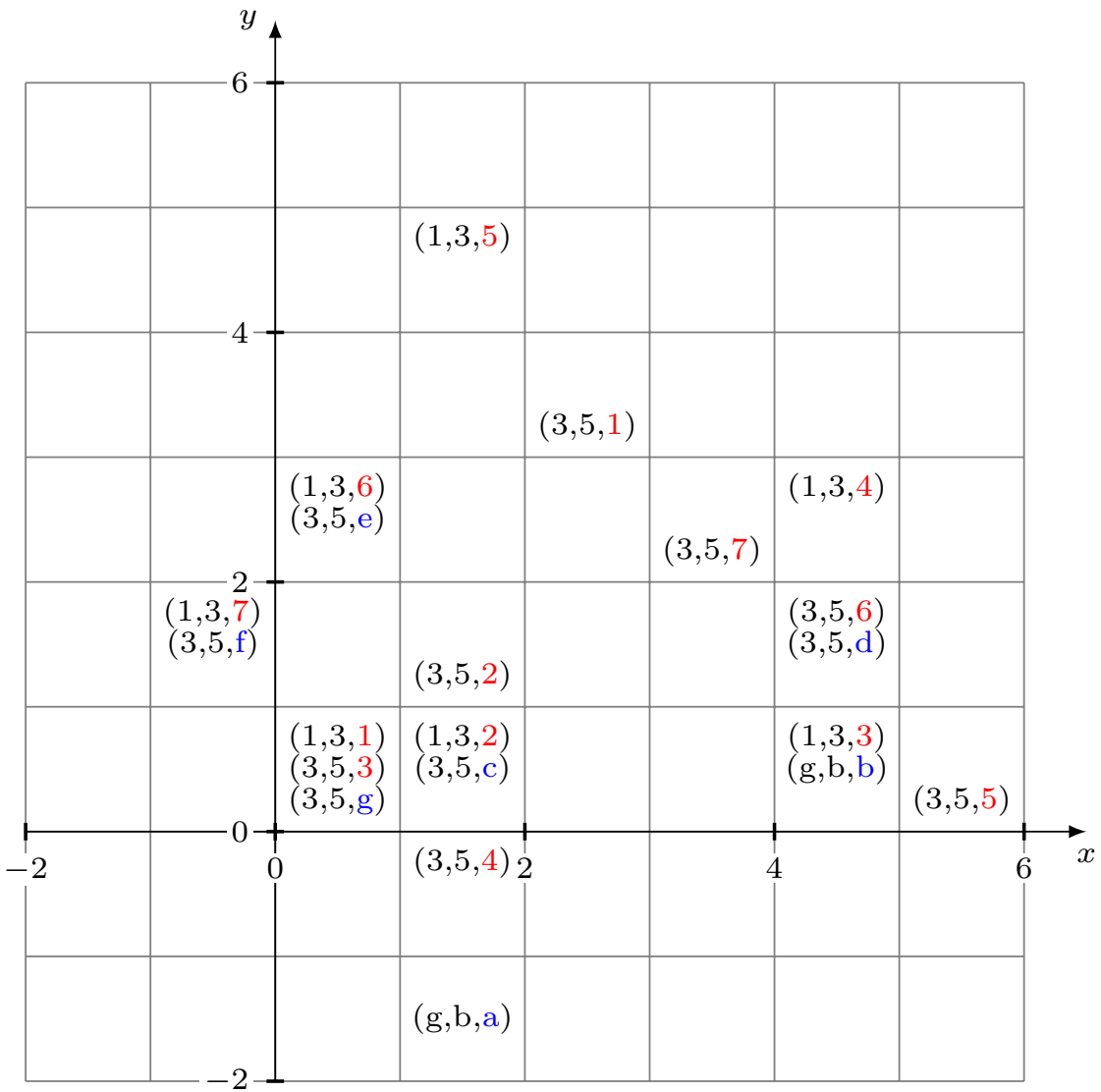




Recognition

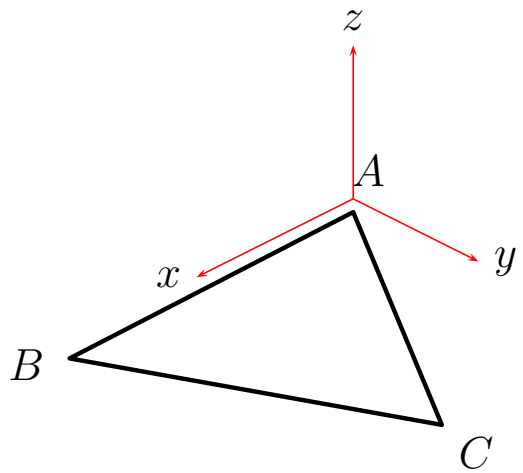
- In this stage the coordinates of the query are calculated according to some basis.
- The query points are then used as indices to H.
- Each query point is indexed into a cell containing transformed model points with similar coordinates.
- For each cell being index, a "vote" is given to the basis pair(s) with points found in the cell.
- The number of votes for a model basis pair is the number of coinciding points to the query (using the specified query basis pair).
- In the example the query points are marked in blue. Model $(1, 3)$ got five votes and model $(3, 5)$ received two votes.
- In the end, the basis with the most votes is output.

Recognition



3D Reference Frame

- An ordered triplet of non-collinear points uniquely defines a reference frame or a rigid transformation (translation + rotation in 3D).
- It can be thought of as an axis system.
- The side lengths are invariant to rigid transformation (rotation + translation) and can be the key to store the information in a hash table.

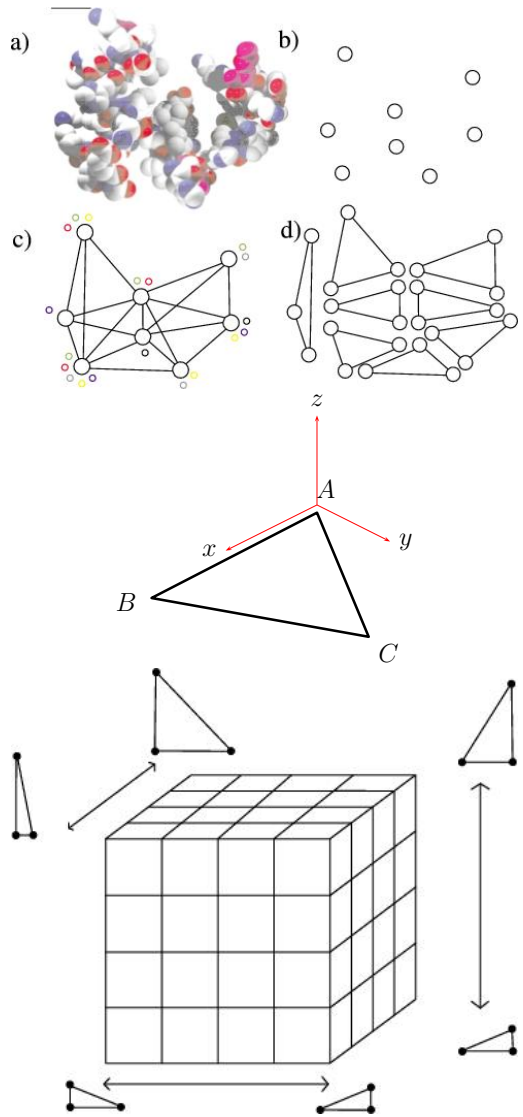


Structural Alignment – Outline

- Define local neighborhoods of residues (in practice an annulus defined by min and max radii).
- Using Geometric Hashing detect seed matches defined by a transformation and a match-list.
- Cluster seed matches and merge match-lists.
- Extend the seed matches and detect best RMSD transformations.
- Iterate last step.

Search Through Geometric Hashing – Preprocessing

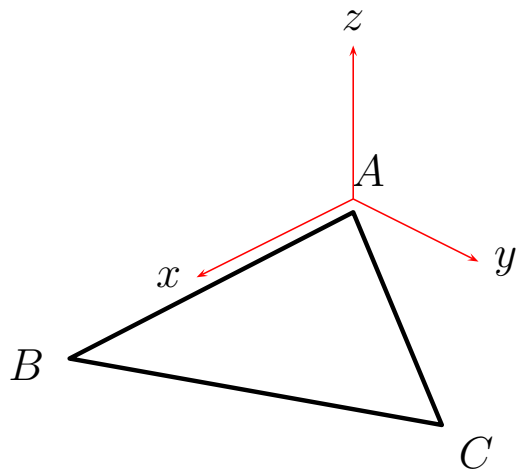
- An ordered triplet of non-collinear points uniquely defines a reference frame (rigid transformation) – translation + rotation in 3D.
- For each reference frame:
- Compute the coordinates of all the other points (in a pre-specified neighborhood) in this reference frame.
- Use each coordinate as an address to the hash (look-up) table.



Wolfson, H.J. & Rigoutsos, I (1997). Geometric Hashing: An Overview. IEEE Computational Science and Engineering, 4(4), 10-21.

Search Through Geometric Hashing – Preprocessing

- Calculating a coordinate system:
- Origin at the centroid.
- X-axis parallel to a-b (how do we calculate it?).
- Z-axis perpendicular to the plane defined by a-b and b-c (cross-product of the two vectors).
- Y-axis perpendicular to X and Z axes above (again, using cross-product).
- Hash key – lengths of the sides or the length of vector a-c and the coordinates of b w.r.t BC, or any other invariant.
- Two congruent triangles can be aligned by aligning their coordinate systems.



Search Through Geometric Hashing – Recognition

- For the target protein do :
- Pick a reference frame satisfying pre-specified constraints.
- Compute the coordinates of all other points in the current reference frame .
- Use each coordinate to access the hash table to retrieve all the records (prot., r.f., shape sign., pt.).
- For records with matching shape sign. "vote" for the (protein, r.f.).
- Compute the transformations of the "high scoring" hypotheses.
- Repeat the above steps for each r.f.

Seed Match Retrieval

- An alignment of more than three points clearly involves more than one pair of matching triplets.
- Since the source motif is rigid, then the transformations aligning these 3-plets must be (almost) identical.
- Therefore, the largest alignment has the most pairs of similar triplets.
- The largest clusters of similar transformations mark the region where the transformation generating the largest alignment can be found.

Seed Match Retrieval

- A clustering algorithm generates clusters of similar transformations, and then generates a representative optimum alignment from each cluster.
- In this stage we take the set of representative alignments and align the entire source and target.
- These "best" alignments are then submitted as output after augmentation.

Match Augmentation Methods

- Iterative best least IRMSD:
- Calculate best transformation for a seed match.
- Extend match with other pairs within a threshold.
- Continue until convergence.

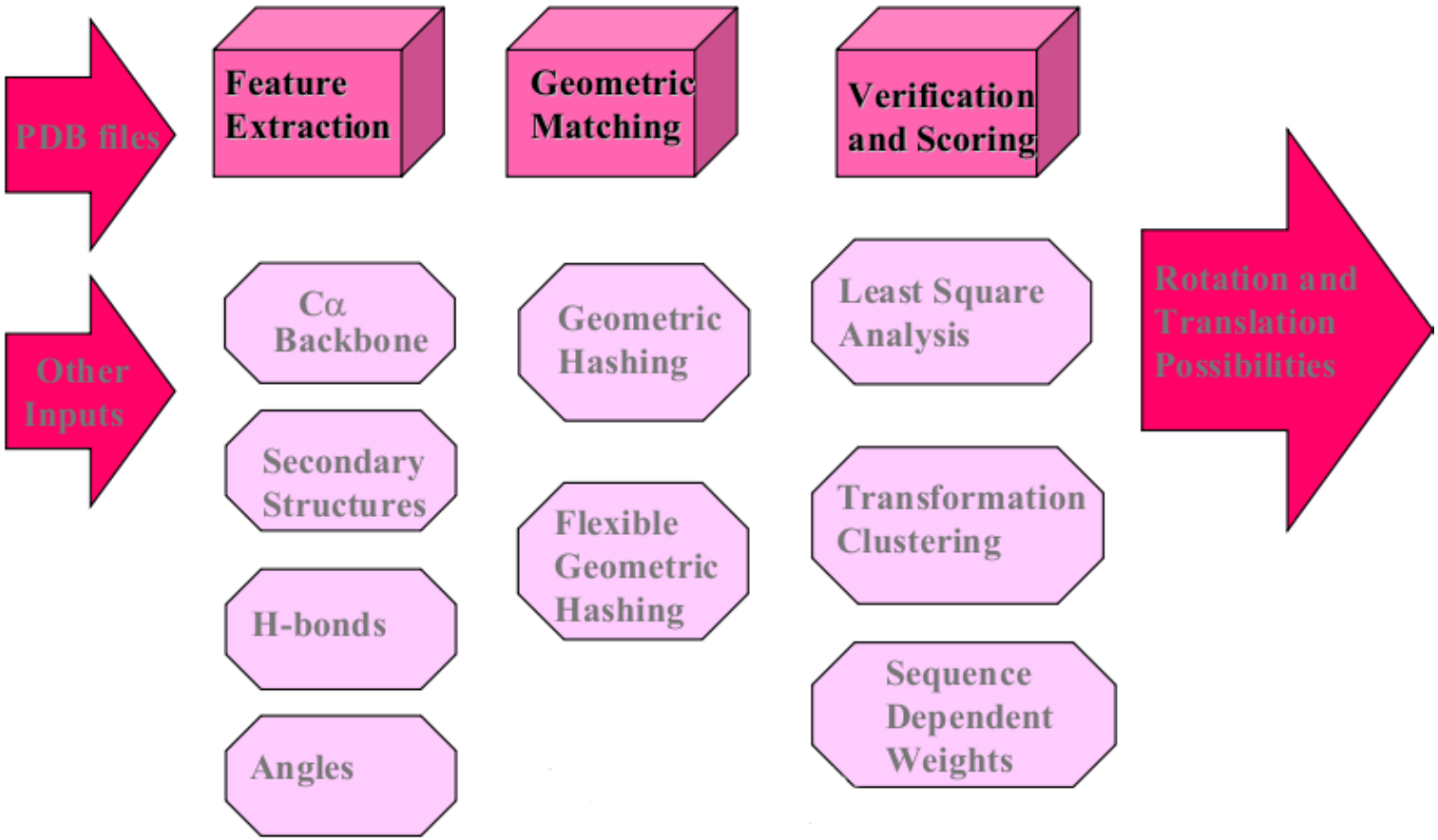
Complexity of Geometric Hashing

- N – number of structures (proteins).
- $O(n)$ – no. of "features" in a structure (e.g. atoms or amino acids).
- R – no. of reference frames (bases).
- Typically, $R = n$, n^2, or n^3.
- Preprocessing: $O(N * R * n)$.
- Match Detection/Recognition : $O(R * n * s)$.
- s – size of a hash-table entry. Can be kept low by not processing "fat" entries.
- These entries are known in advance after Preprocessing.

Applications

- $C\alpha$ backbone matching.
- Secondary structure configuration matching.
- Structural comparison of protein-protein interfaces.
- A representative set of the PDB monomers and interfaces.
- Amino acid substitution matrices based on structural comparison statistics.
- Molecular surface motifs.
- Multiple Structure Alignment.
- Flexible (Hinge - based) structural alignment.

Illustration



Motif Finding using LabelHash

- The algorithm consists of two phases – preprocessing and augmentation.
- The structural motif is defined by the $C\alpha$ coordinates of a number of residues which are encoded as labels.
- In the preprocessing stage any motif can then be matched against the same preprocessed data.
- The aim is to find possible candidate partial matches. This is done by finding all n-tuples of residues (referred to as reference sets) that satisfy certain geometric constraints.
- All valid reference sets for all targets are stored in a hash table.
- A key is a sorted n-tuple of residue labels, and the value is a list of reference sets that contain residues with these labels in any order.
- Unlike geometric hashing, LabelHash does not store transformed copies of the targets for each reference set, which allows to store many more reference sets in the same amount of memory.

Motif Finding

- Match augmentation is an application of depth first search that begins with the list of seed matches.
- Start out with matches retrieved from the hash table.
- Find correspondences for the unmatched motif points within the target.
- Interpret the list of seed matches as a stack of partially complete matches. Pop out the first match, and considering the IRMSD alignment of this match, plot the position p of the next unmatched motif point s_i target points relative to the aligned orientation of the motif.
- In the spherical region V around p, identify all target points ti compatible with s_i.
- Compute the IRMSD alignment of all correlated points, include the new correlation (s_i, t_i).

Motif Finding

- If the new alignment contains no more unmatched motif points and it is long enough, put it into a heap which maintains the match with smallest IRMSD.
- If there are more unmatched motif points, put this partial match back onto the stack.
- Continue to test correlations in this manner, until V contains no more target points that satisfy our criteria.
- Then, return to the stack, and begin again by popping off the first match on the stack, repeating this process until the stack is empty.

Other Approaches– DISCO

- DISCO - approach towards finding the Largest Common Point set.
- DISCO operates on two ligand structures (small molecules) by generating a graph along the following definition:
- Graph Nodes: For a node a, all pairs of points a_1, a_2, with a1 from ligand 1, a_2 from ligand 2.
- Graph Edges: An edge (a,b) exists if the pairs (a_1, a_2) and (b_1, b_2) can be aligned simultaneously. (i.e. the distance between a_1 and b_1, and a_2 and b_2 is very similar)
- This means that for a DISCO graph G, finding a clique implies finding a set of reasonably congruent points common to both ligand structures.

Other Approaches – DISCO

- Finding the largest clique is a well known NP complete problem.
- This is a significant problem, but because ligand structures often have very few points, computation of the largest clique in G is often tractable.
- To this end, a standard clique detection algorithm due to Bron and Kerbosch can be applied to detect cliques in G.
- In addition, if multiple ligands are available then one can be chosen as a reference, and the rest compared to it.

Other Approaches – TM-Align

- Generating optimized residue-to-residue alignment based on structural similarity using dynamic programming iterations.
- Initial alignment tries to match secondary structures.
- A heuristic attempts to extend the alignment iteratively.
- TM-score is returned (see distance measurements)

Other Approaches – MatchMaker and Match→ Align(Chimera)

- The MatchMaker extension of Chimera constructs pairwise sequence alignments and uses them to superimpose the structures.
- The fit can be improved iteratively by pruning residue pairs far apart in space
- Given a superimposed set of two or more protein structures, Match → Align constructs a corresponding sequence alignment.
- Residue types are not used, only the spatial proximities of C-α. The user specifies a cutoff distance and a column inclusion criterion.
- In the pairwise case, a dynamic programming algorithm is used to determine the sequence alignment that best represents the structural alignment. Otherwise, heuristics are used.