3D Shape Matching

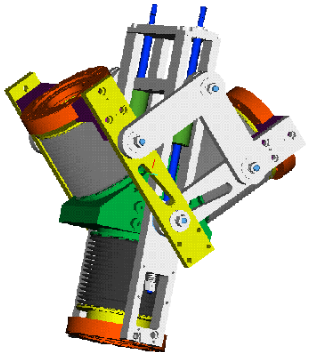
Thomas Funkhouser

COS 429

Motivation

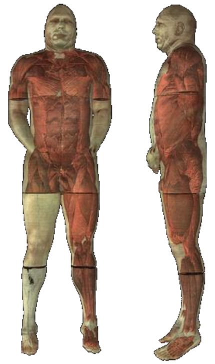
Large repositories of 3D data are available

Computer Graphics



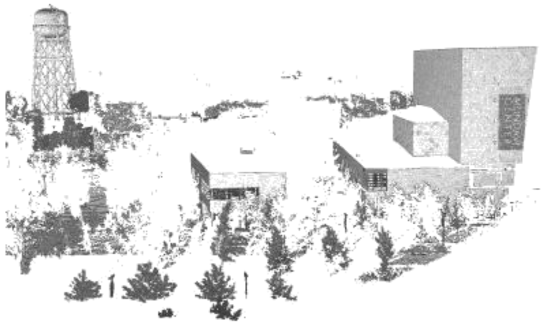
Mechanical CAD

Anthropometry



Medicine

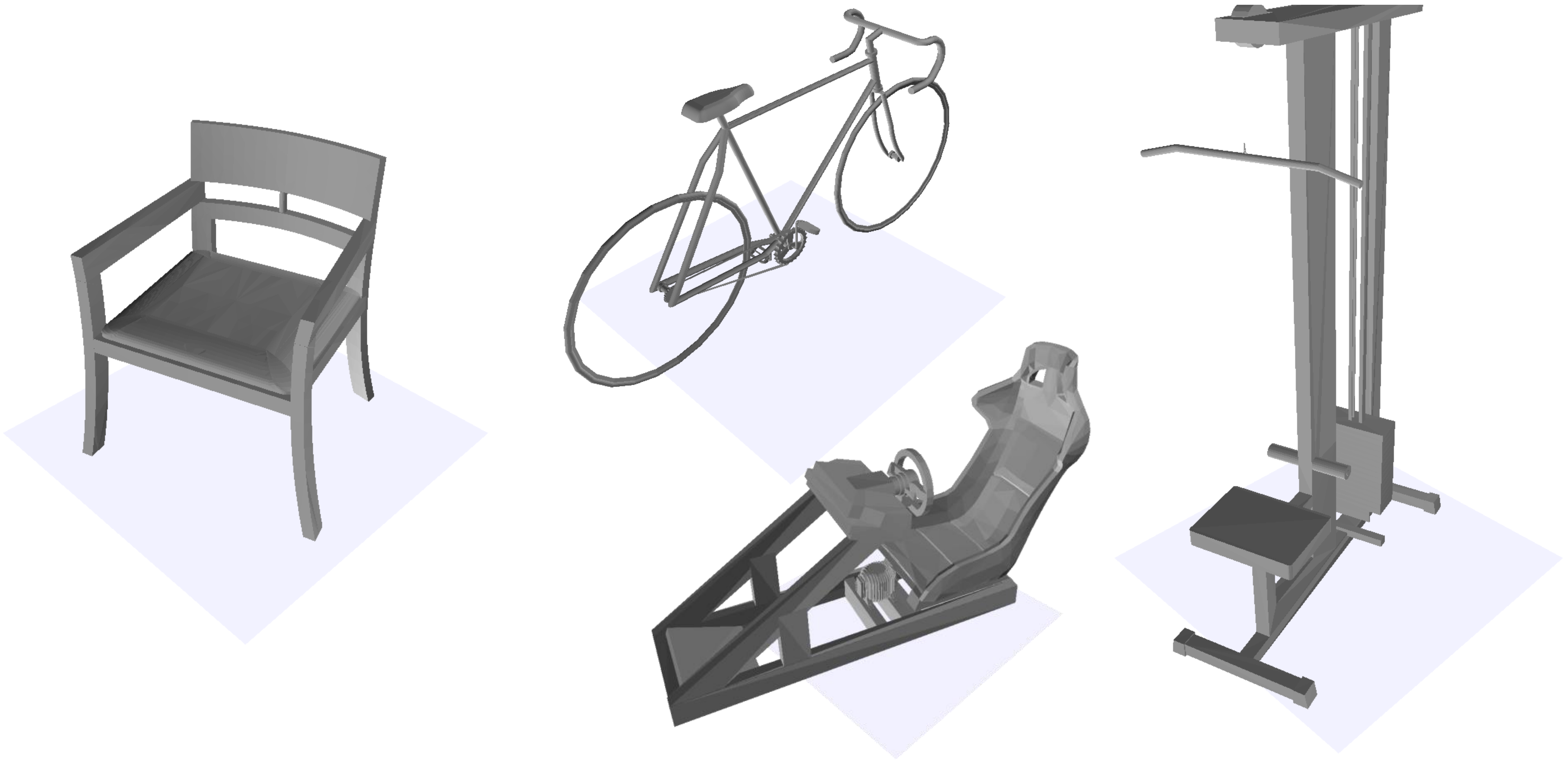
Cultural Heritage



Site Monitoring

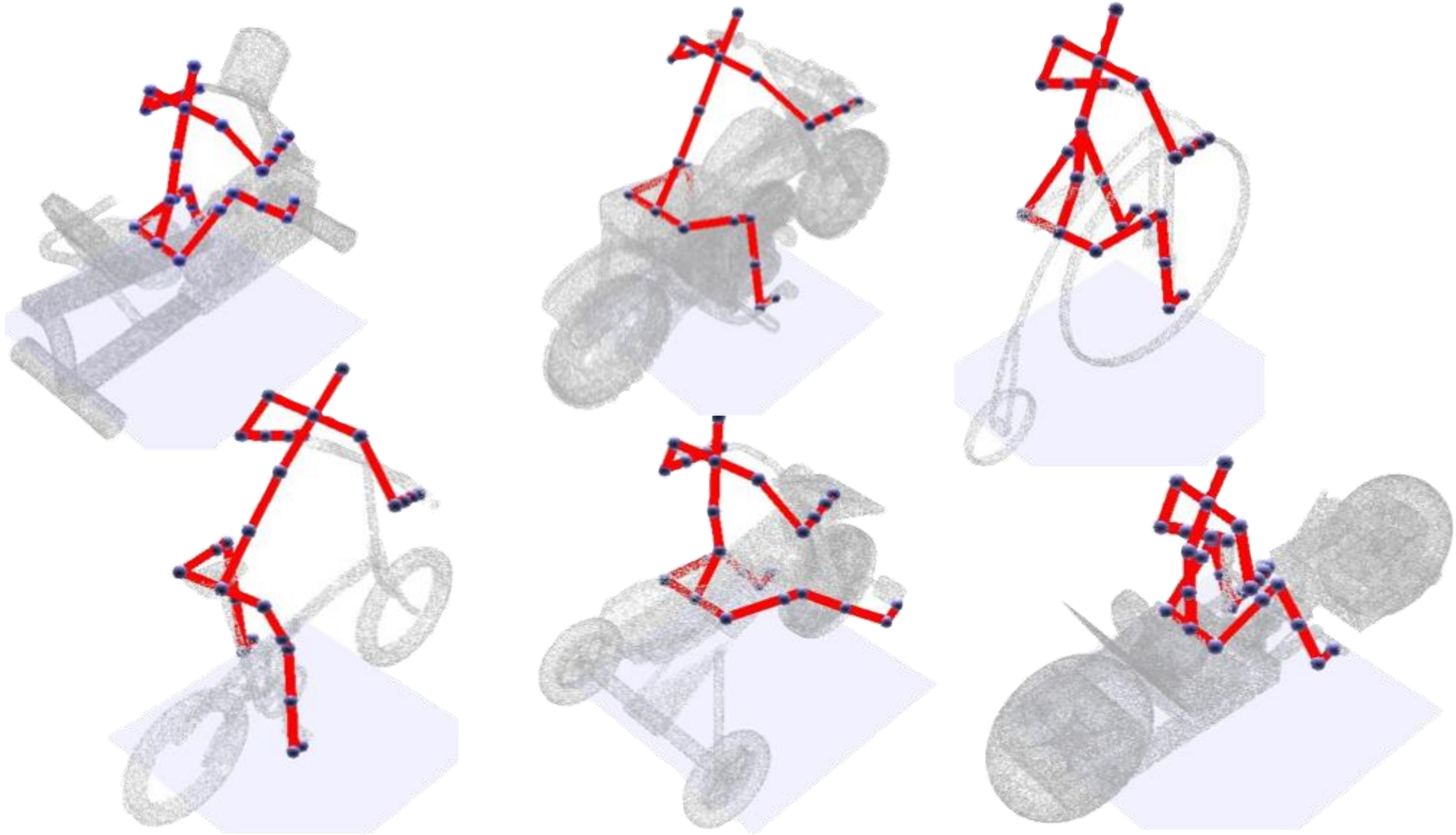
Problem

Most 3D data lacks structural, semantic, and functional annotations



Goal

Infer structures, labels, functions, and relationships automatically from 3D data



Shape Matching

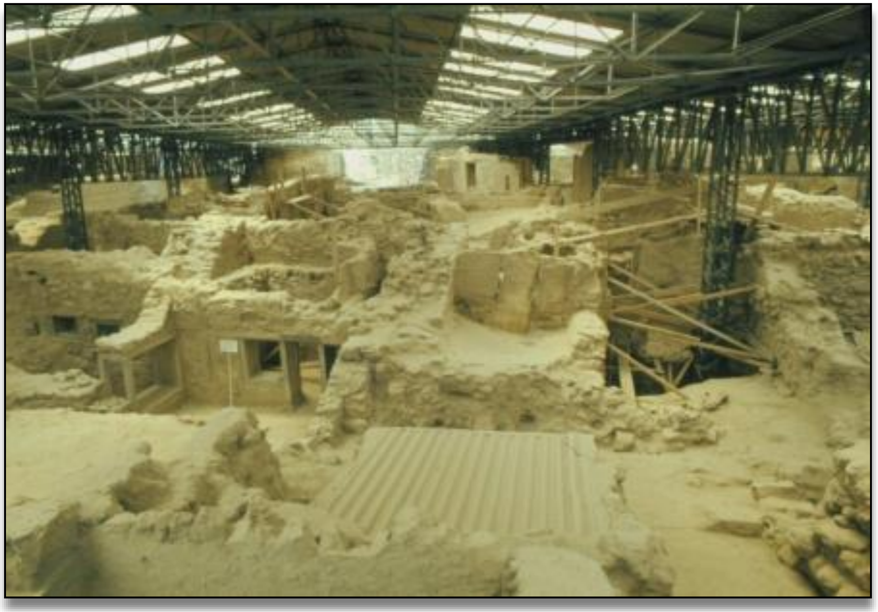
Example applications:

- Archaeology
- Molecular biology
- Paleontology
- Neuroscience
- Urban planning
- Numismatics
- Geometric modeling
- Medicine
- Art
- etc.

Archaeology:
Matching Fresco Fragments

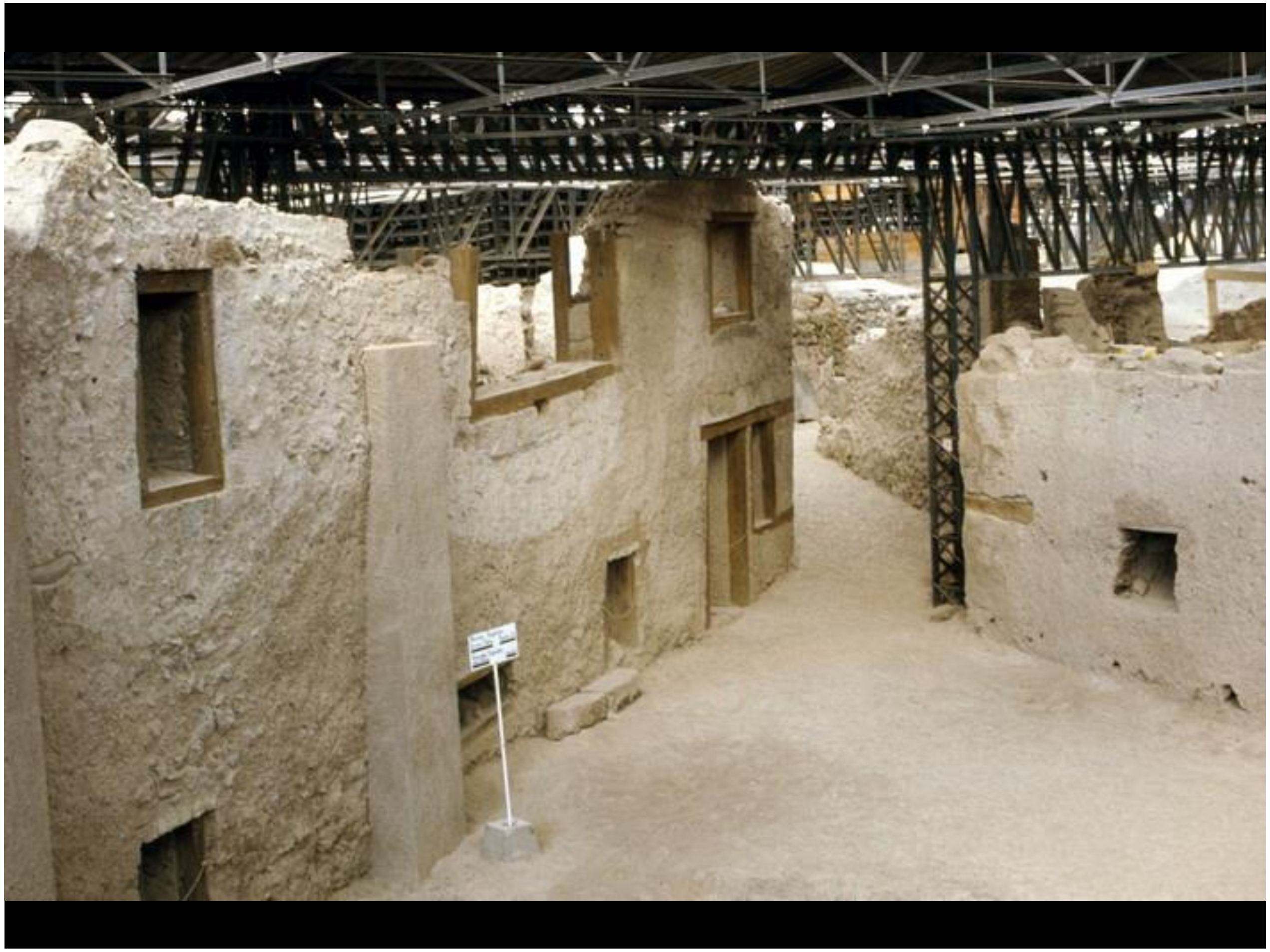
Akrotiri

Buried city discovered in 1967



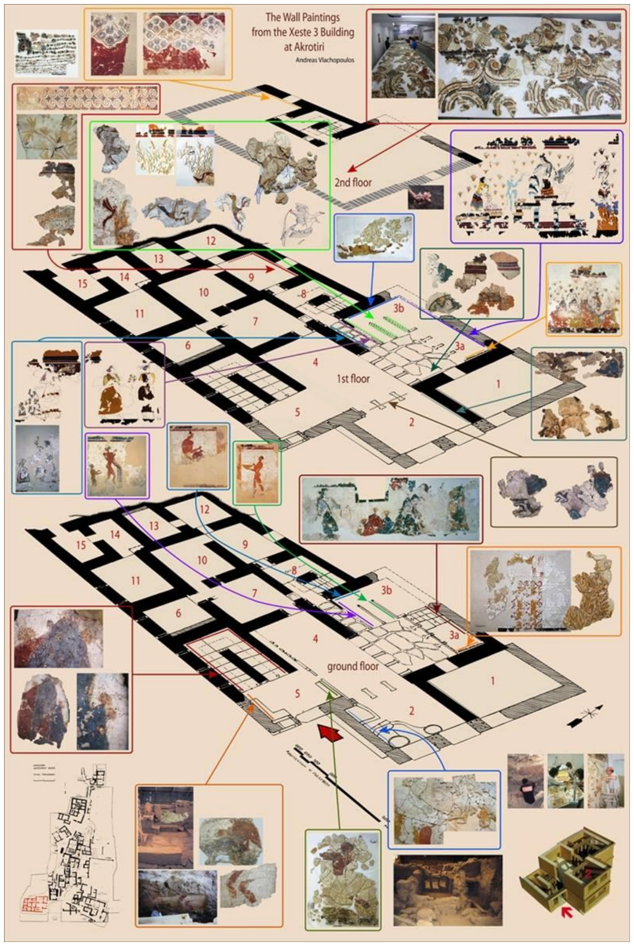
Akrotiri

Buried city discovered in 1967



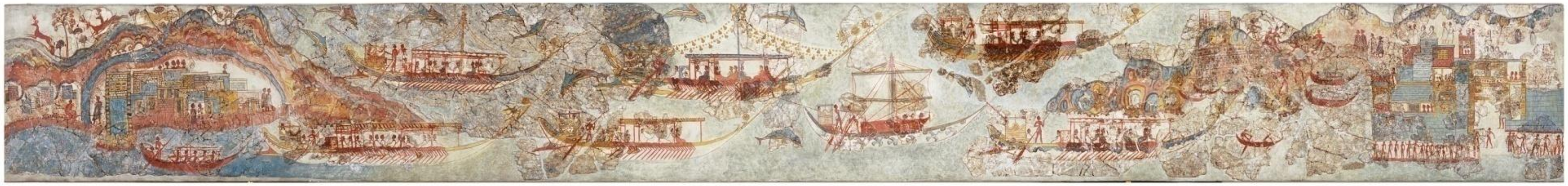
Akrotiri

Many walls were decorated with wall paintings



Akrotiri

Many walls were decorated with wall paintings

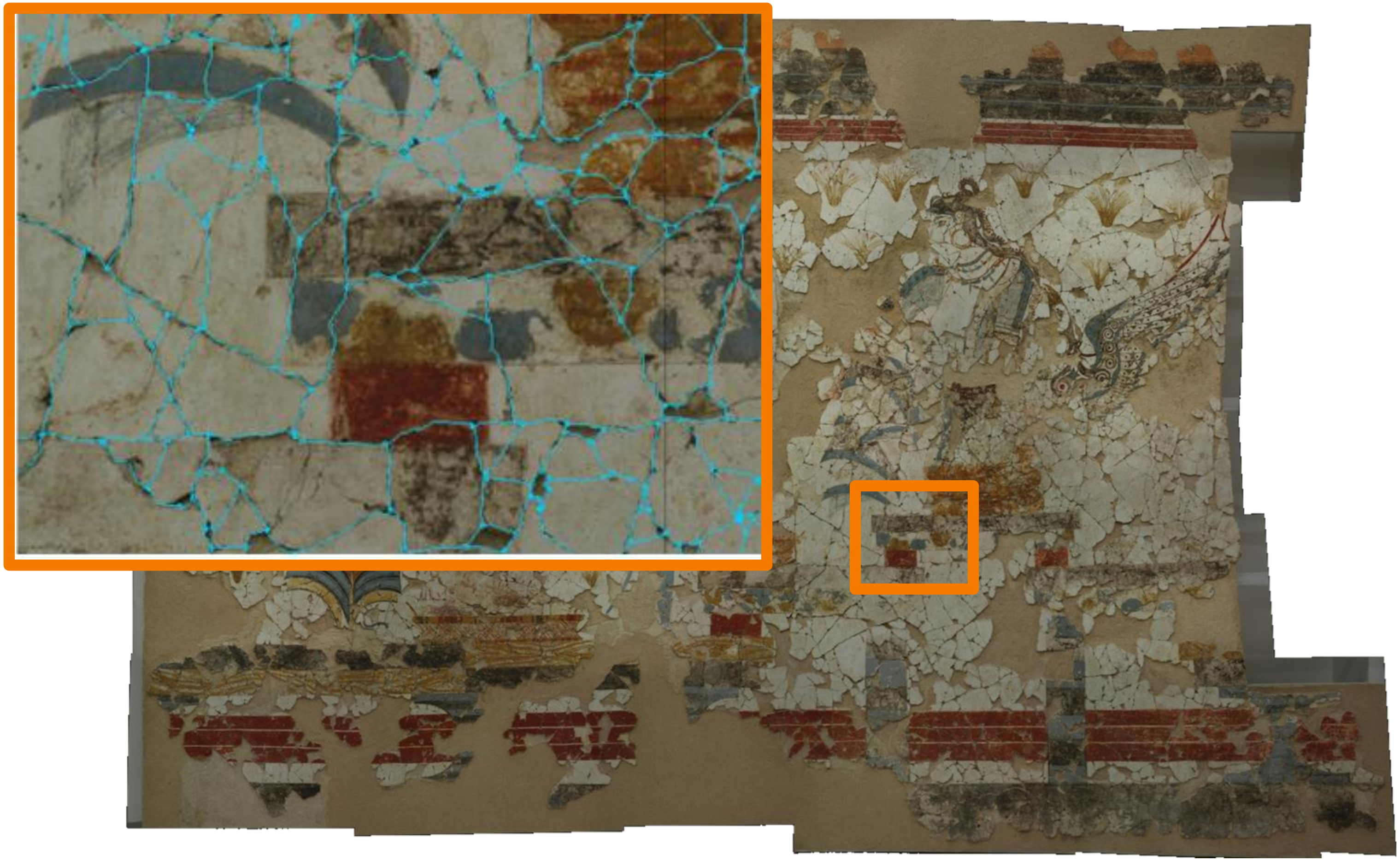


Akrotiri

… but most walls are shattered into fragments

Akrotiri

… but most walls are shattered into fragments



Challenge

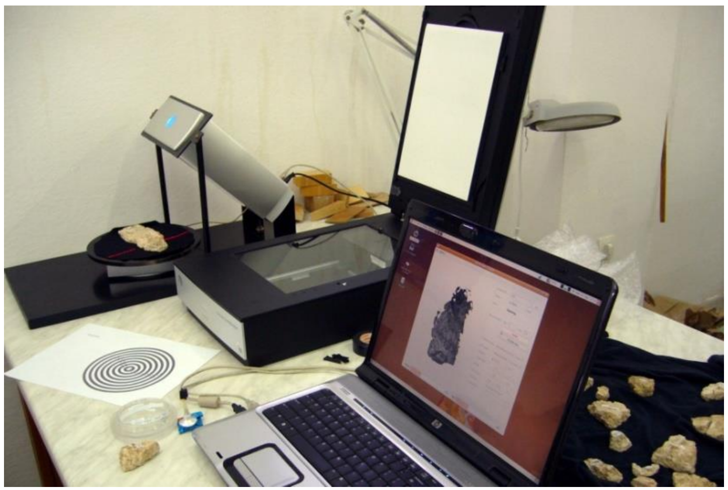
… and re-assembling the fragments is difficult

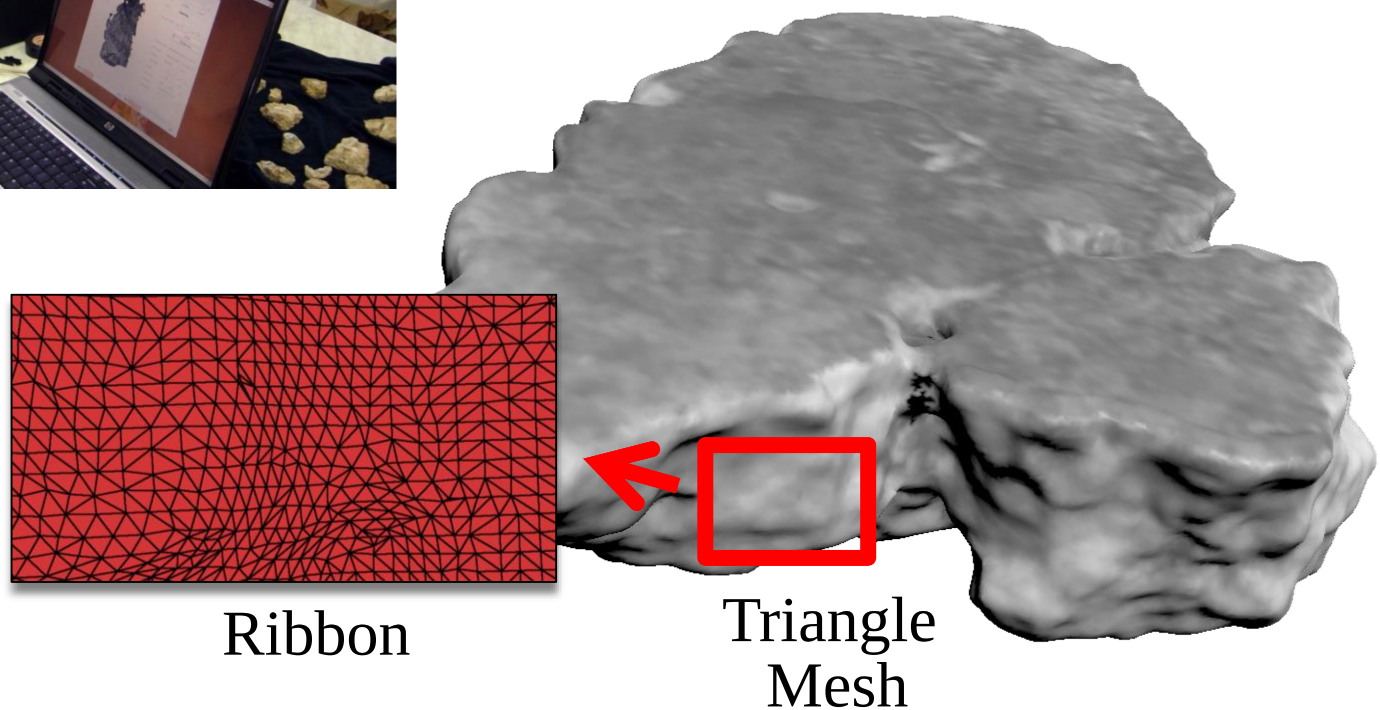
Challenge

… and re-assembling the fragments is difficult

Computer-Assisted Reconstruction

1) Scan digital representations of fragments





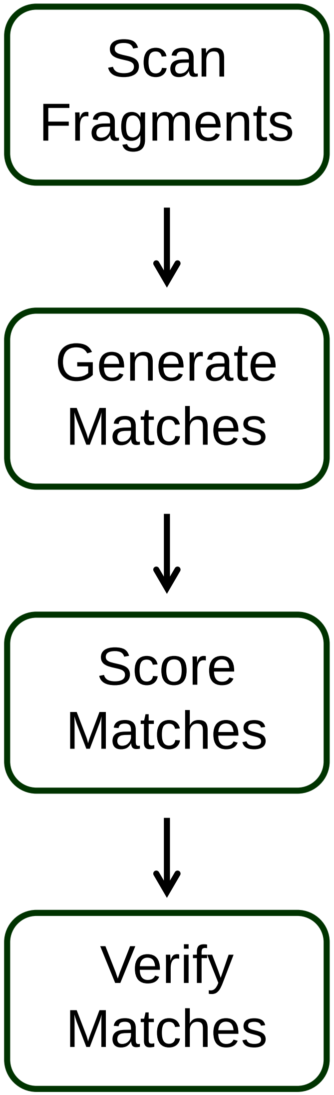
Computer-Assisted Reconstruction

2) Reconstruct frescoes with computer algorithms

Scanned Fragments

Reconstructed Fresco

Computer-Assisted Reconstruction



Computer-Assisted Matching

Scan Fragments

↓

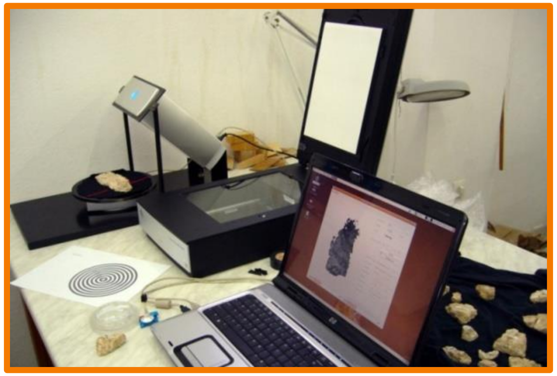
Generate Matches

↓

Score Matches

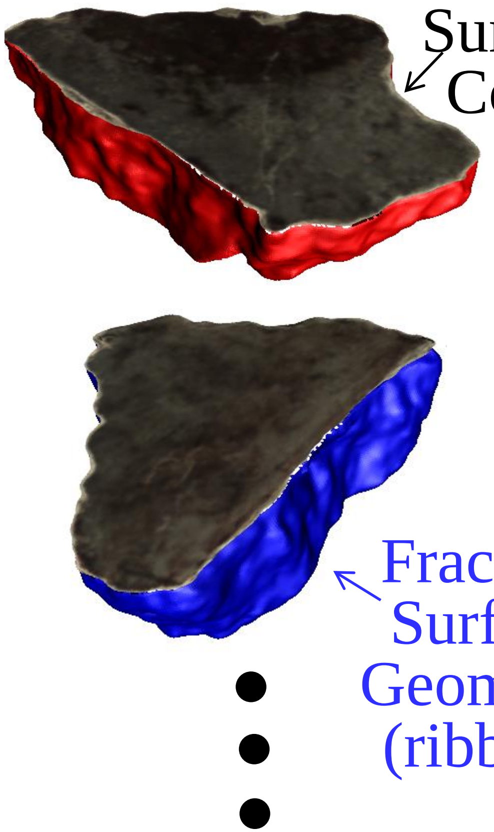
↓

Verify Matches



Scanning System

Brown et al., SIGGRAPH 2008



Scanned Fragments

Computer-Assisted Matching

Scan Fragments

↓

Generate Matches

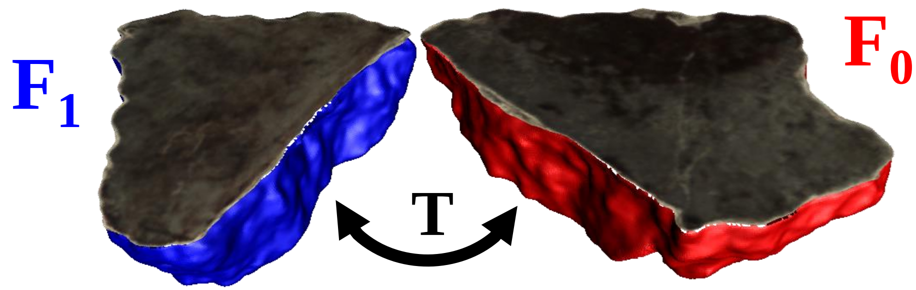
↓

Score Matches

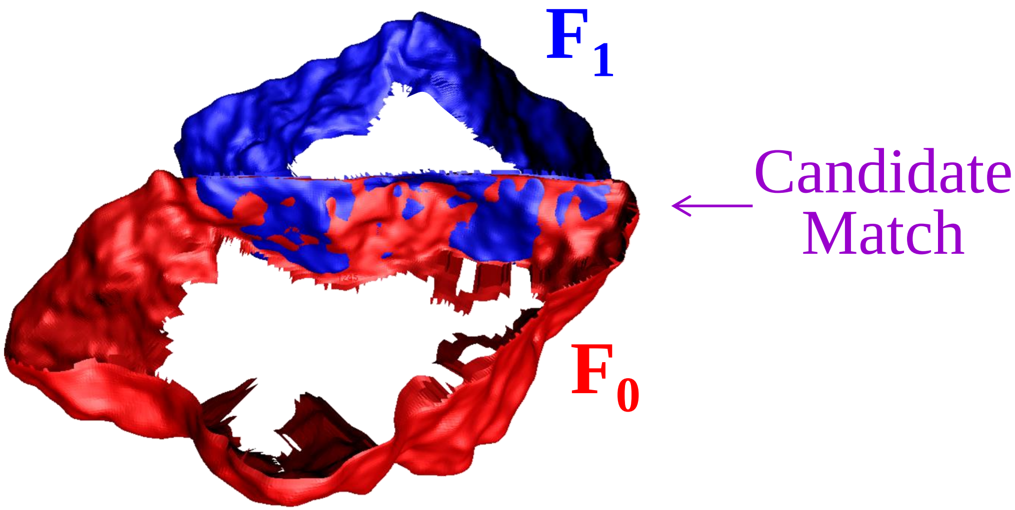
↓

Verify Matches

For every pair of fragments F_0 and F_1 …



Generate candidate match for every plausible aligning transformation T



Computer-Assisted Matching

Scan Fragments

↓

Generate Matches

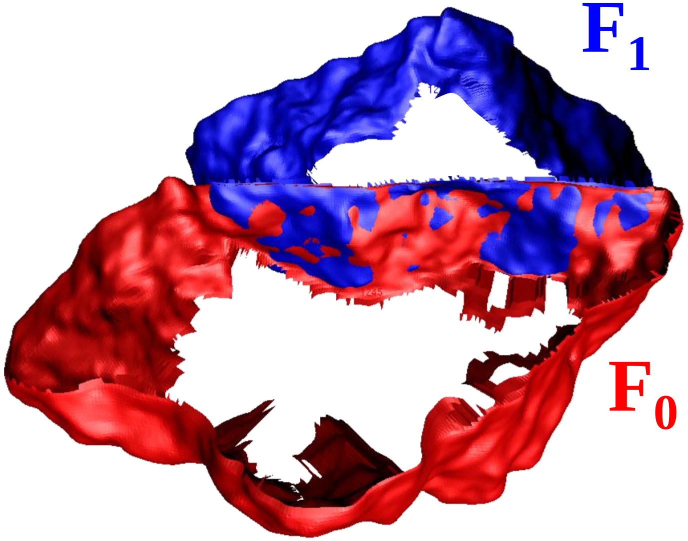
↓

Score Matches

↓

Verify Matches

For every candidate match, compute a score representing “how good it is”



Candidate Match

$$S(F_0,F_1,T)$$

Score

Computer-Assisted Matching

Scan Fragments

↓

Generate Matches

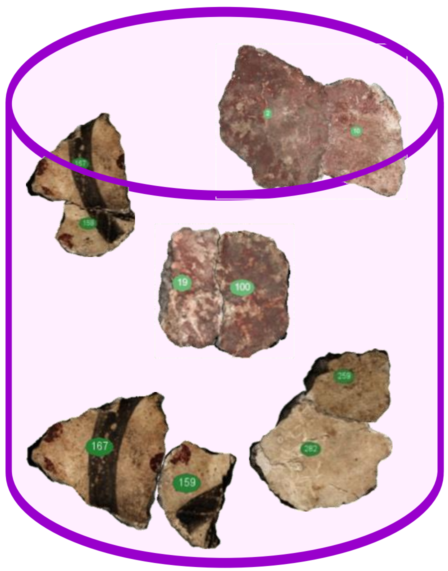
↓

Score Matches

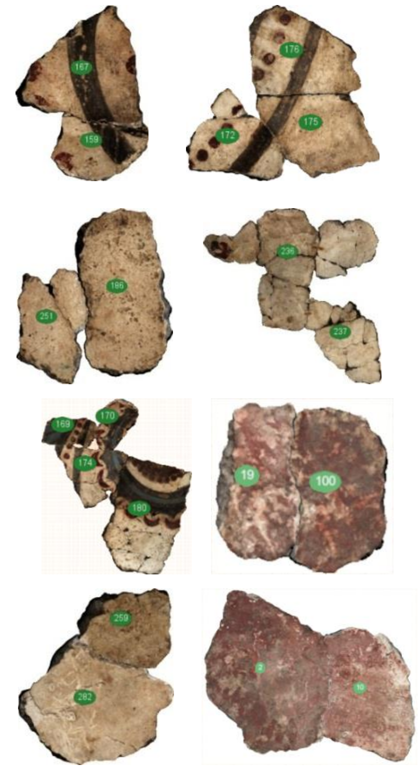
↓

Verify Matches

Sort the candidate matches by score, and check top ones to see if they are correct

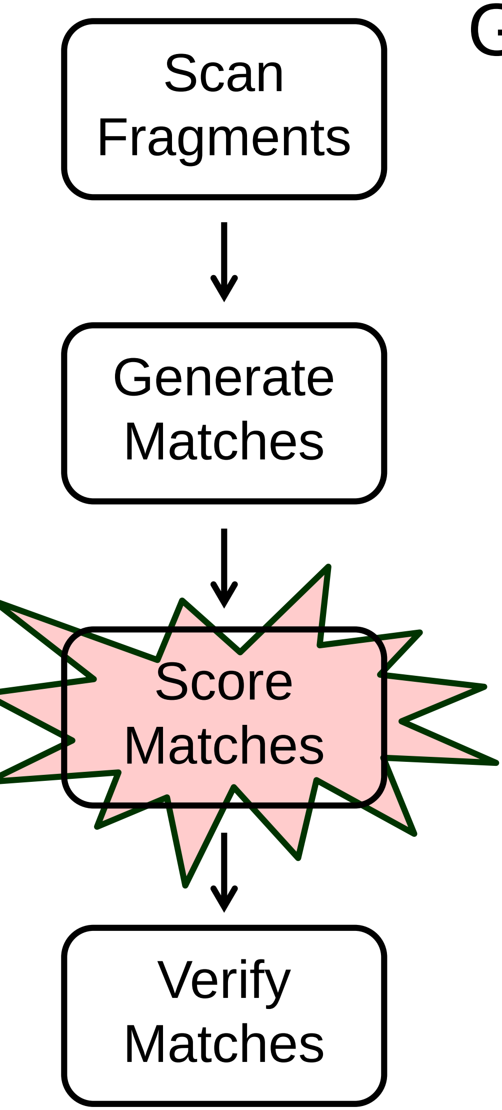


Candidate Matches (millions)



Verified Correct Matches (tens or hundreds)

Focus of This Talk



Goal: Develop a scoring method that accurately estimates the probability that a candidate match is correct

Previous Methods

Most prior systems scored matches using functions combining a few match properties with weights

- McBride et al., 2003

$$\lambda_1 \cdot C_{\text{distance}} + \lambda_2 \cdot \sqrt{C_{\text{length}}} + \lambda_3 \cdot \sqrt{C_{\text{diagnostic}}}$$

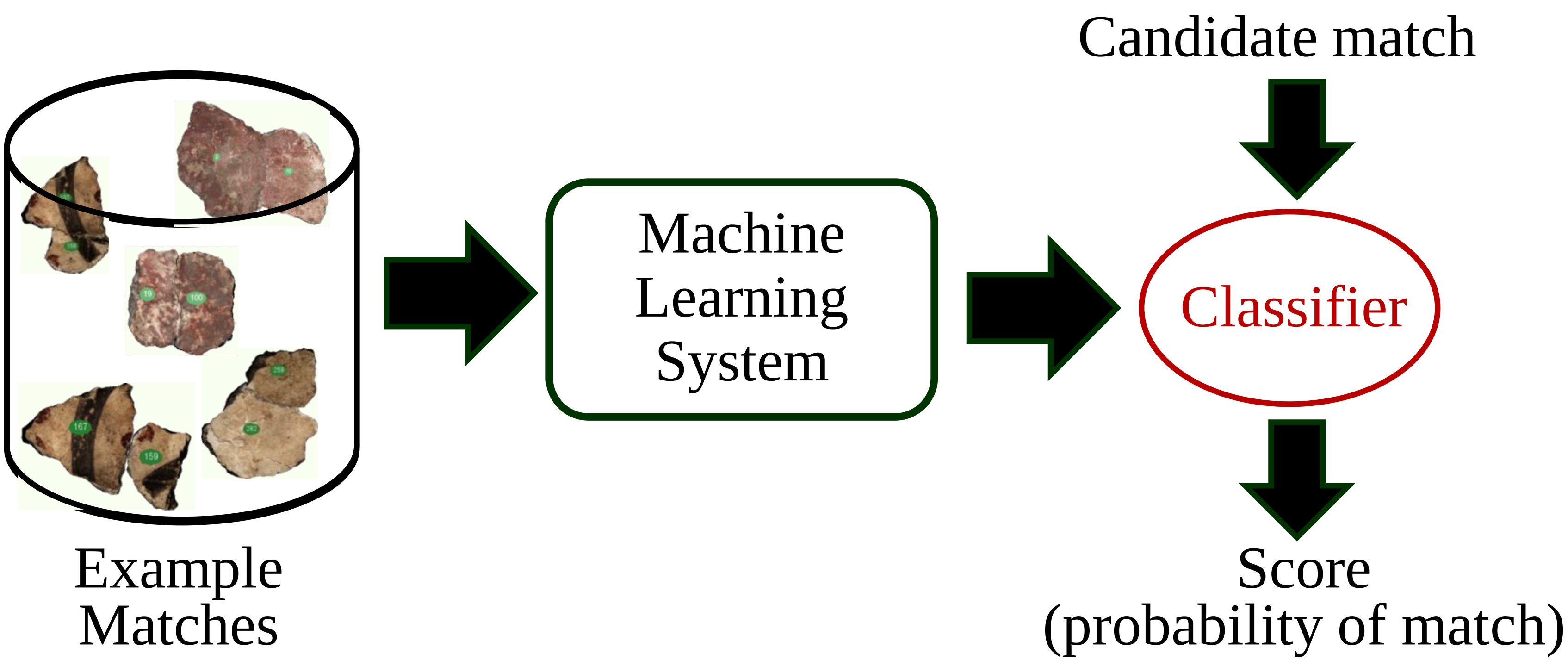
- Brown et al., 2008 (Ribbonmatcher Error)

$$\lambda_1 \cdot C_{\text{WindowRMSD}} + \lambda_2 \cdot C_{\text{Thickness}}$$

Our Approach

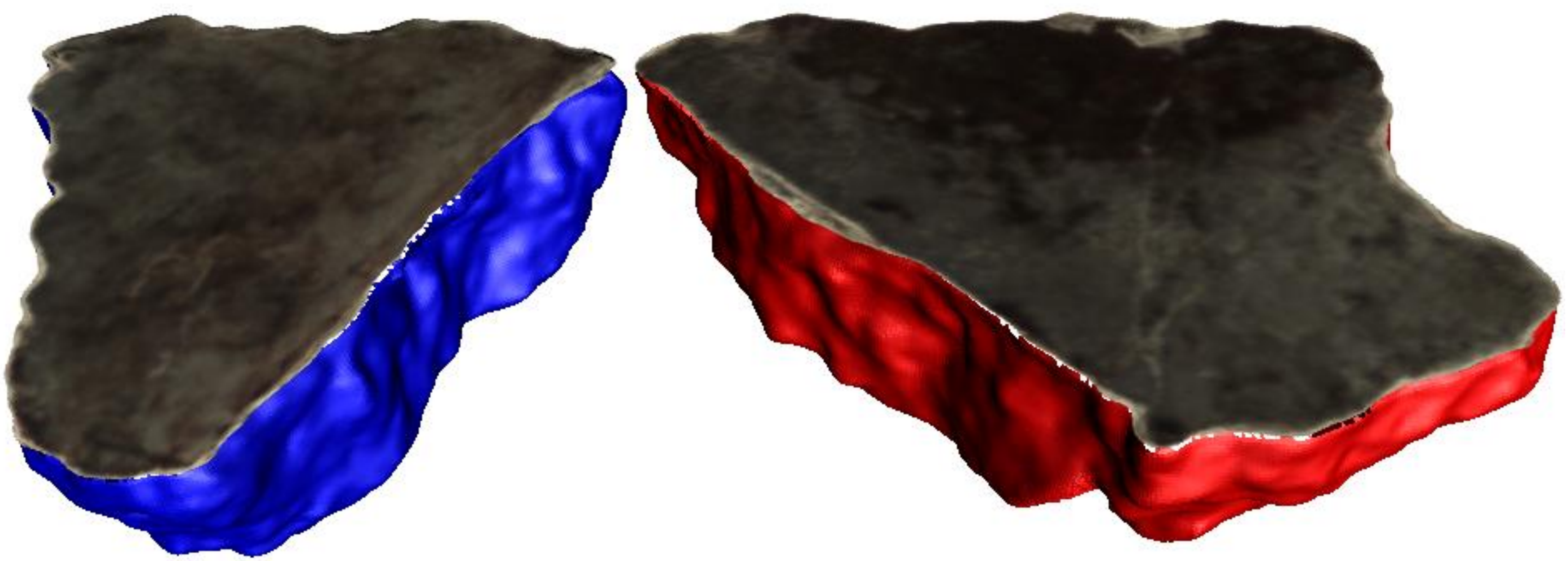
Machine learning

- User provides example correct and incorrect matches
- System learns classifier to predict correctness of new candidate matches based on their properties



Computing Match Properties

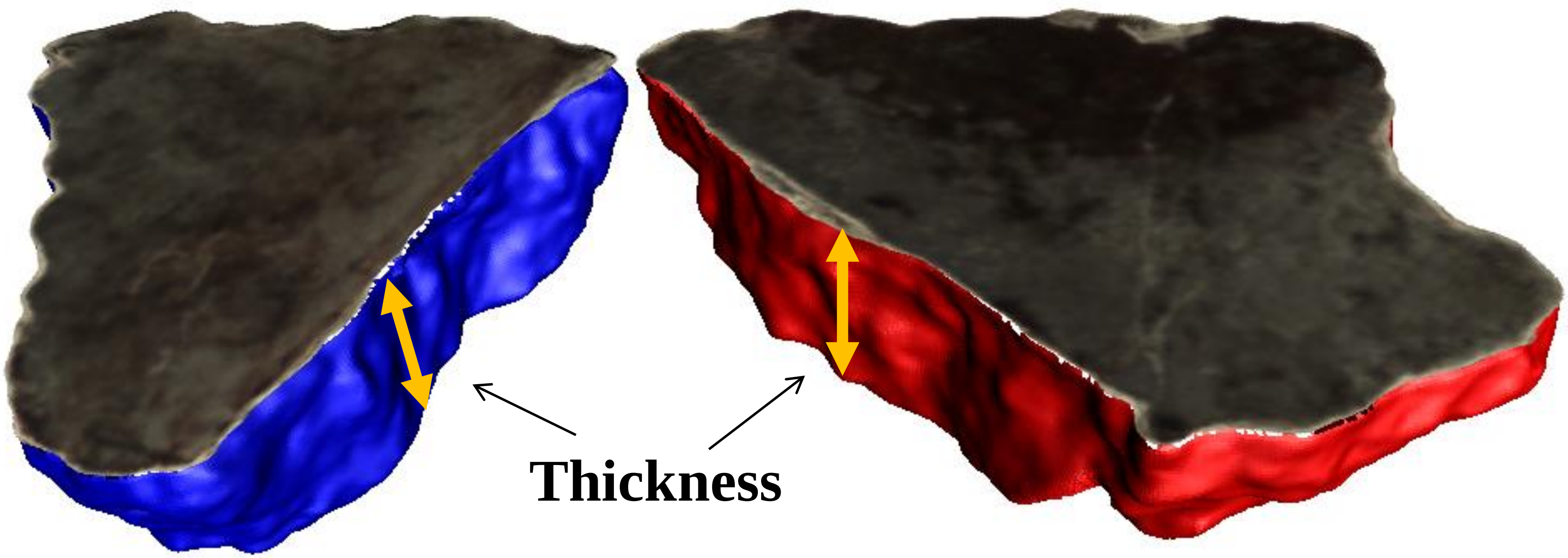
Measure compatibility of fragments



Computing Match Properties

Measure compatibility of fragments

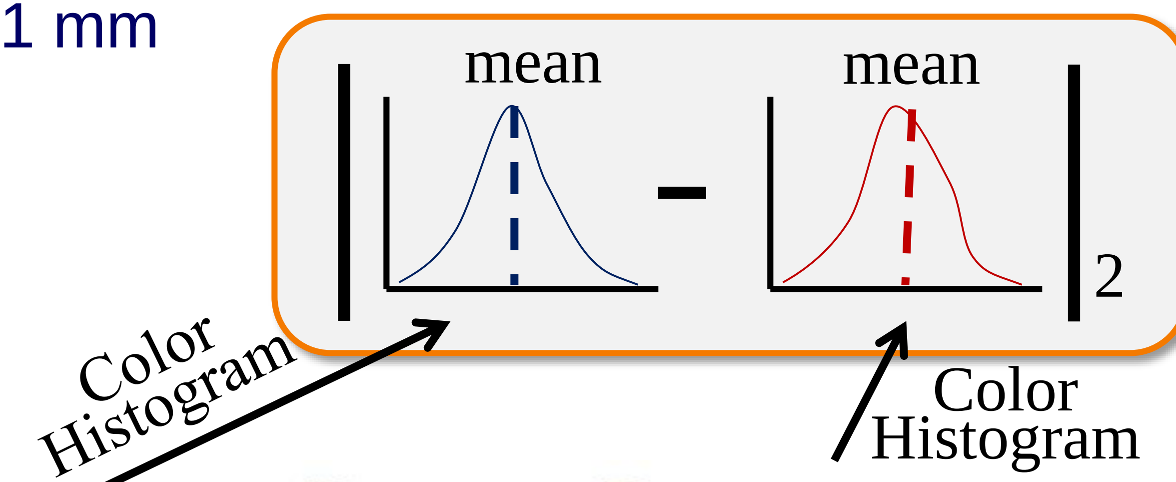
- ➢ΔThickness = 0.1 mm

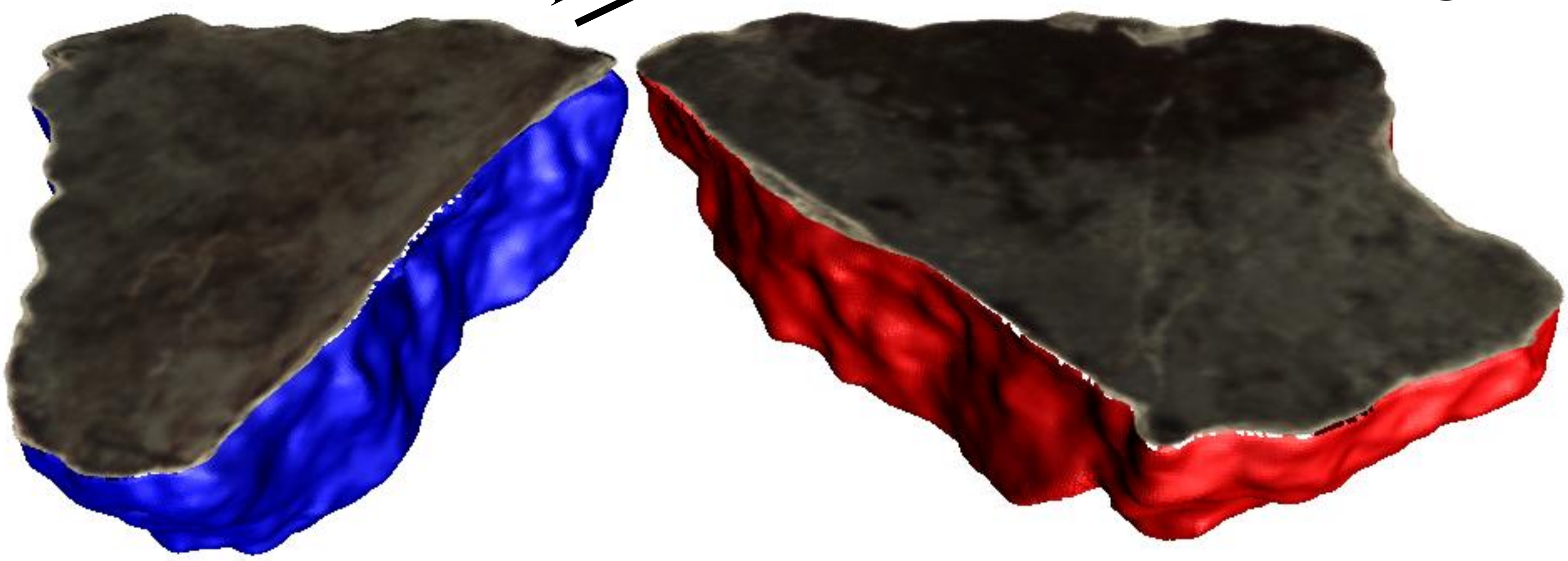


Computing Match Properties

Measure compatibility of fragments

- ΔThickness = 0.1 mm
- ➢ΔColor = 0.002

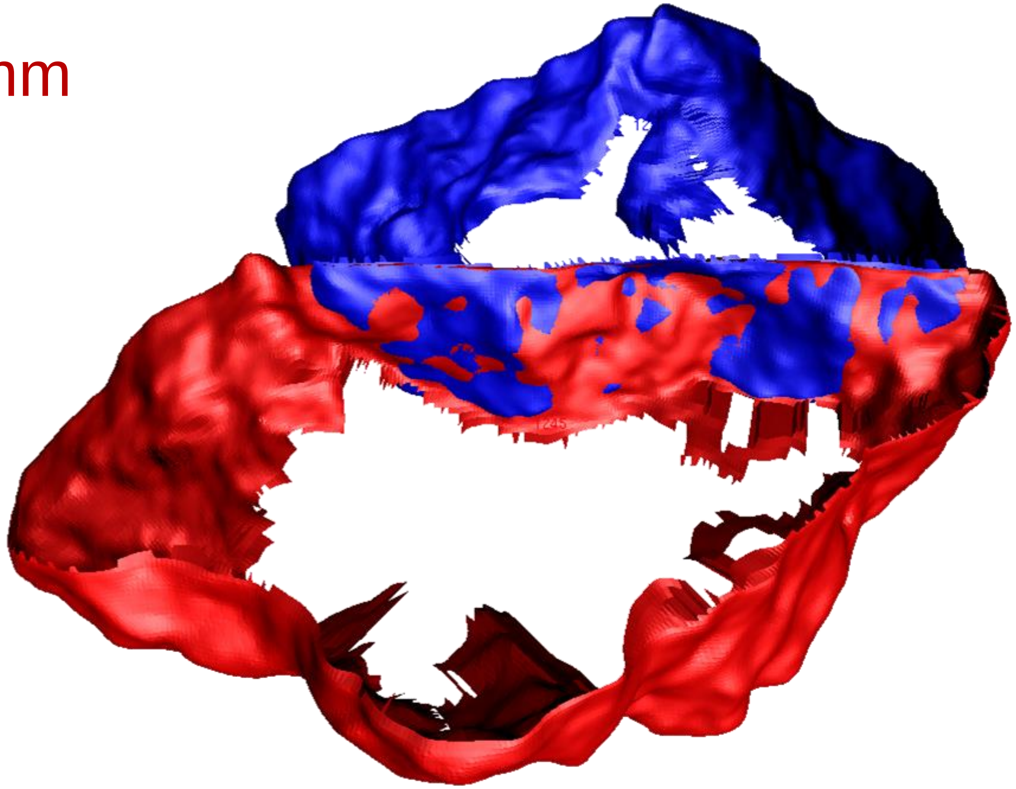




Computing Match Properties

Measure alignment of fragments

- ΔThickness = 0.1 mm
- ΔColor = 0.002
- ➢ΔAlignment = 0.24 mm

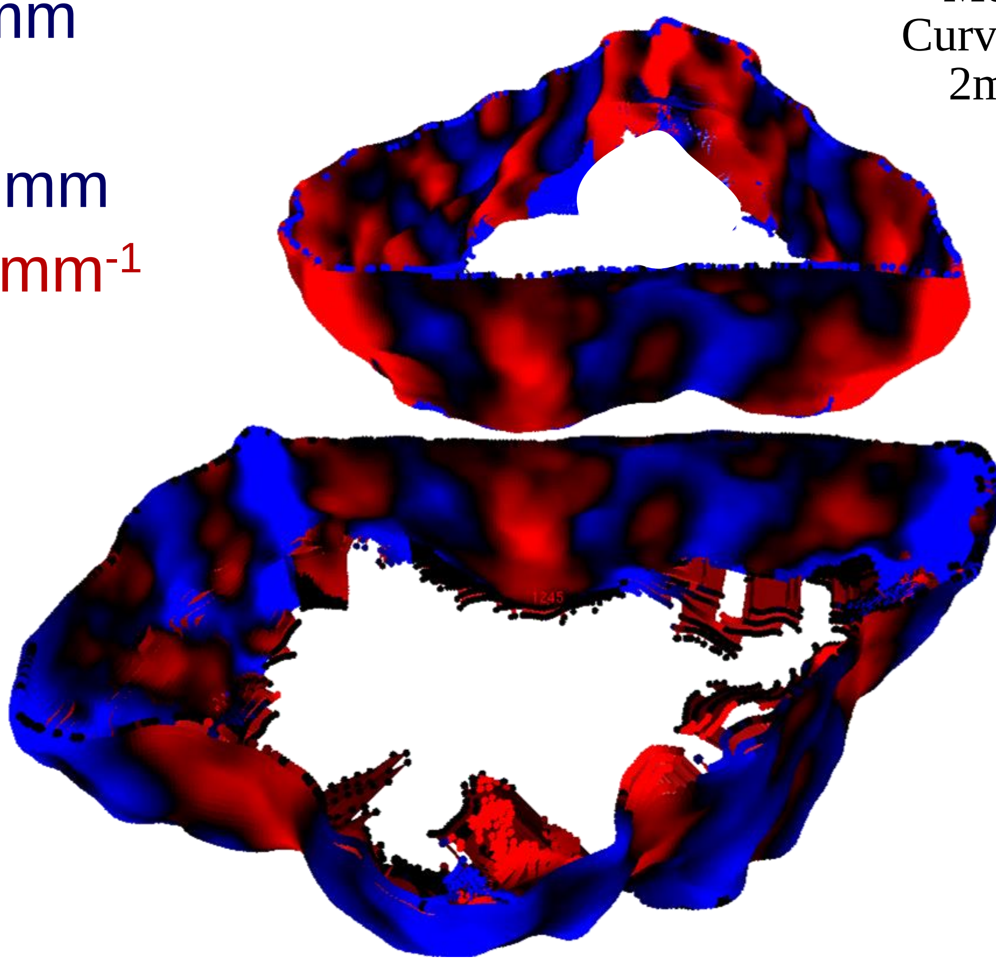


Computing Match Properties

Measure alignment of fragments

- ΔThickness = 0.1 mm
- ΔColor = 0.002
- ΔAlignment = 0.24 mm
- ➢ ΔCurvature = 0.06 mm^{-1}

Mean
Curvature
2mm

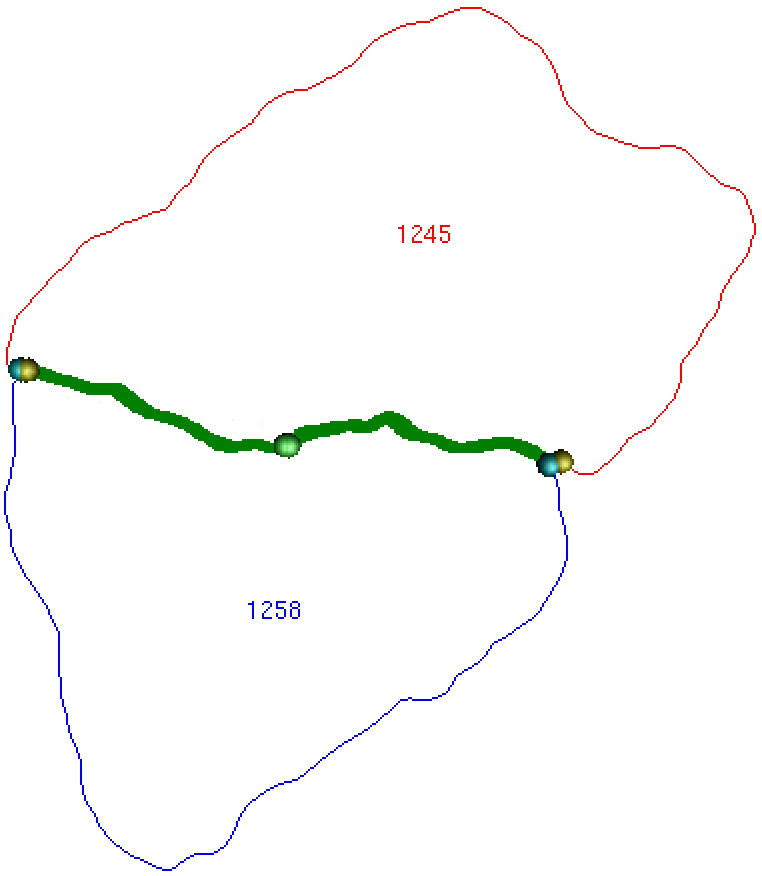


Negative curvature
Positive curvature

Computing Match Properties

Measure alignment of fragments

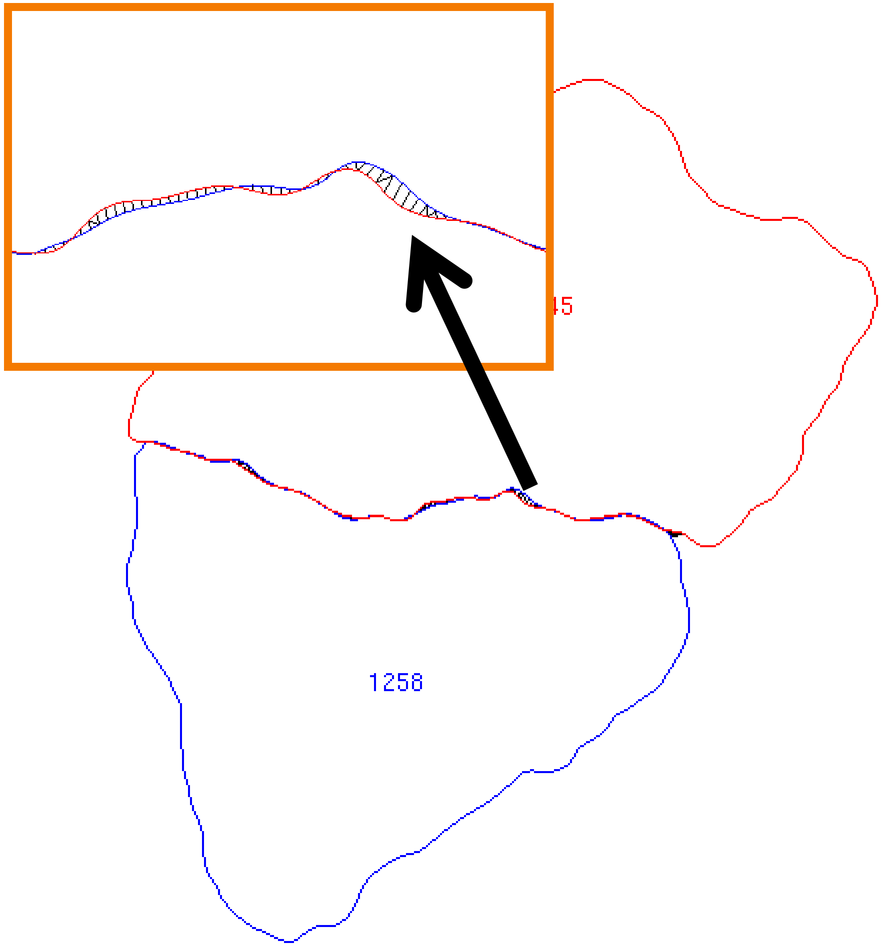
- ΔThickness = 0.1 mm
- ΔColor = 0.002
- ΔAlignment = 0.24 mm
- ΔCurvature = 0.06 mm^{-1}
- ➢ Length = 43.6 mm
- Overlap = 0.7 mm
- Min int. angle = 88 °
- Max ext. angle = 191 °
- Etc.



Computing Match Properties

Measure alignment of fragments

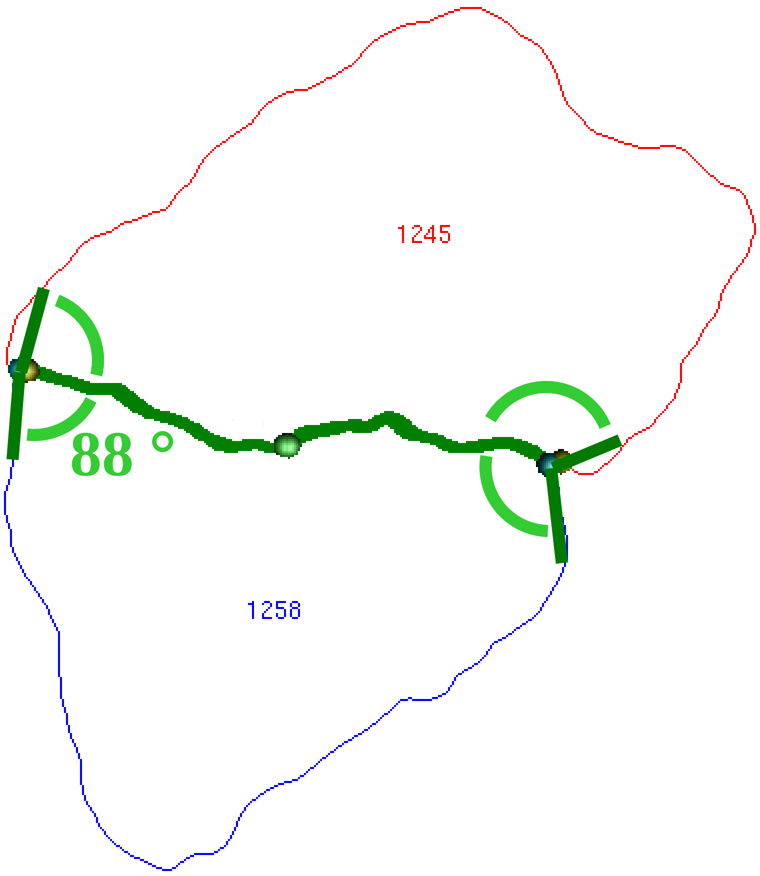
- ΔThickness = 0.1 mm
- ΔColor = 0.002
- ΔAlignment = 0.24 mm
- ΔCurvature = 0.06 mm^{-1}
- Length = 43.6 mm
- Overlap = 0.7 mm
- Min int. angle = 88 °
- Max ext. angle = 191 °
- Etc.



Computing Match Properties

Measure alignment of fragments

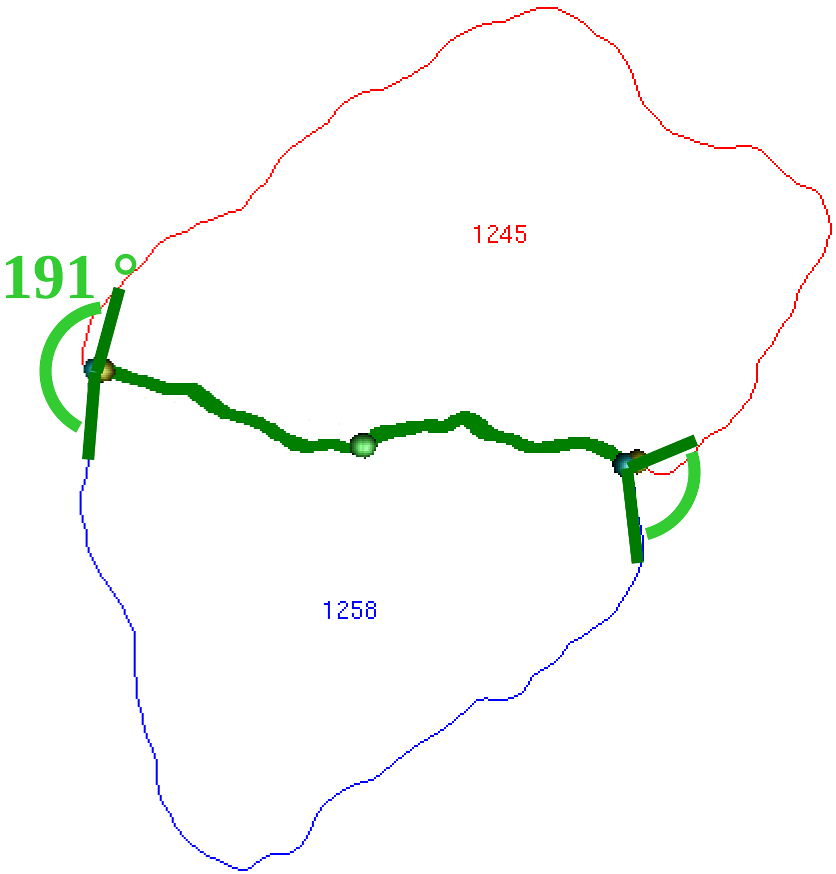
- ◦ ΔThickness = 0.1 mm
- ◦ ΔColor = 0.002
- ◦ ΔAlignment = 0.24 mm
- ◦ ΔCurvature = 0.06 mm^{-1}
- • Length = 43.6 mm
- ◦ Overlap = 0.7 mm
- ➢ Min int. angle = 88 °
- ◦ Max ext. angle = 191 °
- ◦ Etc.



Computing Match Properties

Measure alignment of fragments

- ΔThickness = 0.1 mm
- ΔColor = 0.002
- ΔAlignment = 0.24 mm
- ΔCurvature = 0.06 mm^{-1}
- Length = 43.6 mm
- Overlap = 0.7 mm
- Min int. angle = 88 °
- ➢ Max ext. angle = 191 °
- Etc.



Computing Match Properties

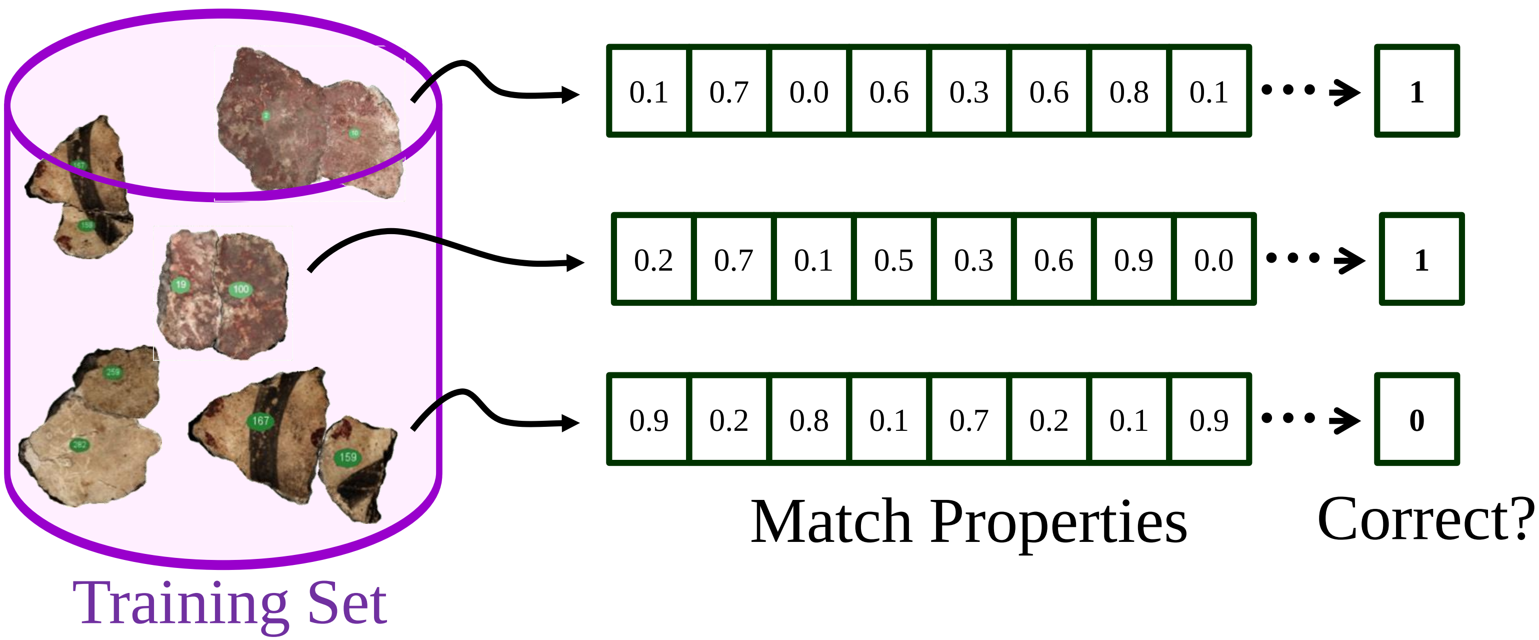
In all, 64 properties per match

ContourContactLength	$0.5 \cdot (\lvert CR_0\rvert + \lvert CR_1\rvert)$
ContourContactDensity	$0.5 \cdot (\lvert CC_0\rvert/\lvert CR_0\rvert + \lvert CC_1\rvert/\lvert CR_1\rvert)$
ContourContactRMSD	$\sqrt{\Sigma_{i,j}(C_{i,i}[j] - C_{i,1-i}[j])^2}$, where $(C_{i,i}[j], C_{i,1-i}[j]) \in CC_i, i \in \{0,1\}, j \in \{0,..,\lvert CC_i\rvert\}$
ContourContactLinearity	$\sqrt{\Sigma_{i,j}(C_{i,i}[j] - L_i)^2}$, where $C_{i,i}[j] \in CC_i, i \in \{0,1\}, j \in \{0,..,\lvert CC_i\rvert\}$, and L_i is the minimizing line
ContourContactCurvL2 (4 properties)	$\sqrt{\Sigma_{i,j}(\mathrm{Curv}(C_{i,i}[j],t,s) - \mathrm{Curv}(C_{i,1-i}[j],t,s))^2}$, where $(C_{,ii}[j], C_{i,1-i}[j]) \in CC_i, i \in \{0,1\}, j \in \{0,..,\lvert CC_i\rvert\}$, $t \in$ { Horizontal }, and $s \in$ { 1mm, 2mm, 4mm, 8mm }
ContourContactLengthFraction (4 properties)	Stat($\lvert CR_i\rvert$)/Measurement(C_i)), where Stat $\in$ { Min, Max }, and Measurement $\in$ { Perimeter, $\sqrt{Area}$ }
ContourWindowRMSD (3 properties)	$\sqrt{\Sigma_{i,j}(C_{i,i}[j] - C_{i,1-i}[j])^2}$, where $(C_{i,i}[j], C_{i,1-i}[j]) \in CW(s), j \in \{0,..,\lvert CW(s)\rvert\}$, and $s \in$ { 4mm, 8mm, 16mm }
ContourMergeConvexity	Convexity($C_0 \cup C_1$)
ContourMergeConvexityFraction (2 properties)	Stat(Convexity(C_0) / Convexity($C_0 \cup C_1$), Convexity(C_1) / Convexity($C_0 \cup C_1$)), where Stat $\in$ { Min, Max }
ContourOverlapArea	$\lvert C_0 \cap C_1\rvert$
ContourOverlapDepth (2 properties)	Stat(Depth($C_{i,i}[j]$)), where $C_{i,i}[j] \in CC_i, i \in \{0,1\}, j \in \{0,..,\lvert CC_i\rvert\}$, and Stat $\in$ { Avg, Max }
ContourJunctionAngle (4 properties)	Stat(Angle(CJ_i, t)), where Stat $\in$ { Min, Max }, and $t \in$ { Exterior, Interior }

RibbonContactArea	$0.5 \cdot (\lvert RR_0\rvert + \lvert RR_1\rvert)$
RibbonContactDensity	$0.5 \cdot (\lvert RC_0\rvert/\lvert RR_0\rvert + \lvert RC_1\rvert/\lvert RR_1\rvert)$
RibbonContactLength	$0.5 \cdot ((\lvert RR_0 \rightarrow C_0\rvert + \lvert RR_1 \rightarrow C_1\rvert)$, where $RR_i \rightarrow C_i$ is the projection of RR_i onto C_i
RibbonContactRMSD	$\sqrt{\Sigma_{i,j}(R_{i,i}[j] - R_{i,1-i}[j])^2}$, where $(R_{i,i}[j], R_{i,1-i}[j]) \in RC_i, i \in \{0,1\}, j \in \{0,..,\lvert RC_i\rvert\}$
RibbonContactPlanarity	$\sqrt{\Sigma_{i,j}(R_{i,i}[j] - P_i)^2}$, where $R_{i,i}[j] \in RC_i, i \in \{0,1\}, j \in \{0,..,\lvert RC_i\rvert\}$, and P_i is the minimizing vertical plane
RibbonContactHCurvL2 (4 properties)	$\sqrt{\Sigma_{i,j}(\mathrm{Curv}(R_{i,i}[j],t,s) - \mathrm{Curv}(R_{i,1-i}[j],t,s))^2}$, where $(R_{i,i}[j], R_{i,1-i}[j]) \in RC_i, i \in \{0,1\}, j \in \{0,..,\lvert RC_i\rvert\}$, $t \in$ { Horizontal }, and $s \in$ { 1mm, 2mm, 4mm, 8mm }
RibbonContactCurvL2 (4 properties)	$\sqrt{\Sigma_{i,j}(\mathrm{Curv}(R_{i,i}[j],t,s) - \mathrm{Curv}(R_{i,1-i}[j],t,s))^2}$, where $(R_{i,i}[j], R_{i,1-i}[j]) \in RC_i, i \in \{0,1\}, j \in \{0,..,\lvert RC_i\rvert\}$, $t \in$ { Vertical, Mean }, and $s \in$ { 1mm, 2mm }
RibbonWindowRMSD (3 properties)	$\sqrt{\Sigma_{i,j}(R_{i,i}[j] - R_{i,1-i}[j])^2}$, where $(R_{i,i}[j], R_{i,1-i}[j]) \in RW(s), j \in \{0,..,\lvert RW(s)\rvert\}$, and $s \in$ { 4mm, 8mm, 16mm }
RibbonJunctionAngle (4 properties)	Stat(Angle(RJ_i, t)), where Stat $\in$ { Min, Max }, and $t \in$ { Exterior, Interior }
FragmentThicknessL2	(Thickness(F_0) - Thickness(F_1))^2, where Thickness(F_i) is the average number of columns with scanned vertex positions in each row of R_i
FragmentFrontColorL2 (12 properties)	(Stat(I_0, c) - Stat(I_1, c))^2, where Stat $\in$ { Mean, Median, Variance }, and $c \in$ { Red, Green, Blue, Luminance }
FragmentAreaFraction	min($\lvert C_0\rvert/\lvert C_1\rvert, \lvert C_1\rvert/\lvert C_0\rvert$)

Learning a Scoring Function

Learn a classifier that predicts the probability that a match is correct based on its properties



Learning a Scoring Function

Classifier

- Decision Tree
  - Each branch checks the value of a property
  - Each leaf has linear regression model
  - Produces score “roughly” modeling probability
  - Selects good features automatically

```
RibbonContactRMSD <= 0.429 :
  RibbonContactRMSD <= 0.375 :
    RibbonContactPlanarity <= 0.517 :
      ContourContactRMSD <= 0.286 :
        ContourContact4mmHorizCurvL2 <= 0.009 : LM1 (29)
        ContourContact4mmHorizCurvL2 > 0.009 : LM2 (112)
      ContourContactRMSD > 0.286 : LM3 (560)
    RibbonContactPlanarity > 0.517 :
      RibbonContactArea <= 446.36 :
        RibbonContactRMSD <= 0.36 :
          RibbonJunctionMinInteriorAngle <= 2.232 :
            ContourContactRMSD <= 0.217 : LM4 (17)
            ContourContactRMSD > 0.217 :
              ContourContactMinLenAreaFract <= 0.309 : LM5 (20)
              ContourContactMinLenAreaFract > 0.309 :
                RibbonContactRMSD <= 0.331 : LM6 (12)
                RibbonContactRMSD > 0.331 : LM7 (20)
          RibbonJunctionMinInteriorAngle > 2.232 : LM8 (29)
        RibbonContactRMSD > 0.36 : LM9 (91)
      RibbonContactArea > 446.36 : LM10 (53)
  RibbonContactRMSD > 0.375 :
    RibbonContactArea <= 235.969 : LM11 (3015)
    RibbonContactArea > 235.969 :
      RibbonContact1mmMeanCurvL2 <= 0.121 : LM12 (603)
      RibbonContact1mmMeanCurvL2 > 0.121 : LM13 (151)
RibbonContactRMSD > 0.429 : LM14 (7416)
```

Decision tree
learned on Synthetic Fresco

Learning a Scoring Function

Classifier

- Decision Tree
  - Each branch checks the value of a property
  - Each leaf has linear regression model

```
Truth =
      -0.0013 * RibbonContactRMSD
      + 0 * RibbonContactArea
      + 0.0001 * RibbonContactPlanarity
      + 0.0005 * RibbonContact1mmMeanCurvatureL2
      + 0 * RibbonJointMinInteriorAngle
      + 0 * RibbonJointMaxExteriorAngle
      - 0.0001 * ContourContactRMSD
      - 0.0007 * ContourContact4mmHorizontalCurvatureL2
      + 0.0002
```

"Matches with large ContactRMSD are unlikely"
(score is near zero)

```
RibbonContactRMSD <= 0.429 :
  RibbonContactRMSD <= 0.375 :
    RibbonContactPlanarity <= 0.517 :
      ContourContactRMSD <= 0.286 :
        ContourContact4mmHorizCurvL2 <= 0.009 : LM1 (29)
        ContourContact4mmHorizCurvL2 > 0.009 : LM2 (112)
      ContourContactRMSD > 0.286 : LM3 (560)
    RibbonContactPlanarity > 0.517 :
      RibbonContactArea <= 446.36 :
        RibbonContactRMSD <= 0.36 :
          RibbonJunctionMinInteriorAngle <= 2.232 :
            ContourContactRMSD <= 0.217 : LM4 (17)
            ContourContactRMSD > 0.217 :
              ContourContactMinLenAreaFract <= 0.309 : LM5 (20)
              ContourContactMinLenAreaFract > 0.309 :
                RibbonContactRMSD <= 0.331 : LM6 (12)
                RibbonContactRMSD > 0.331 : LM7 (20)
          RibbonJunctionMinInteriorAngle > 2.232 : LM8 (29)
        RibbonContactRMSD > 0.36 : LM9 (91)
      RibbonContactArea > 446.36 : LM10 (53)
  RibbonContactRMSD > 0.375 :
    RibbonContactArea <= 235.969 : LM11 (3015)
    RibbonContactArea > 235.969 :
      RibbonContact1mmMeanCurvL2 <= 0.121 : LM12 (603)
      RibbonContact1mmMeanCurvL2 > 0.121 : LM13 (151)
RibbonContactRMSD > 0.429 : LM14 (7416)
```

Decision tree
learned on Synthetic Fresco

Learning a Scoring Function

Classifier

- Decision Tree
  - Each branch checks the value of a property
  - Each leaf has linear regression model

Truth =
-5.1265 * RibbonContactRMSD
+ 0 * RibbonContactArea
+ 0.0138 * RibbonContactPlanarity
+ 0.012 * RibbonContact1mmMeanCurvatureL2
- 0.0286 * RibbonJointMinInteriorAngle
+ 0.0006 * RibbonJointMaxExteriorAngle
- 0.0011 * ContourContactRMSD
- 0.1677 * ContourContact4mmHorizontalCurvatureL2
+ 0.6273 * ContourContactMinLengthAreaFraction
+ 1.9331

```
RibbonContactRMSD <= 0.429 :
  RibbonContactRMSD <= 0.375 :
    RibbonContactPlanarity <= 0.517 :
      ContourContactRMSD <= 0.286 :
        ContourContact4mmHorizCurvL2 <= 0.009 : LM1 (29)
        ContourContact4mmHorizCurvL2 > 0.009 : LM2 (112)
      ContourContactRMSD > 0.286 : LM3 (560)
    RibbonContactPlanarity > 0.517 :
      RibbonContactArea <= 446.36 :
        RibbonContactRMSD <= 0.36 :
          RibbonJunctionMinInteriorAngle <= 2.232 :
            ContourContactRMSD <= 0.217 : LM4 (17)
            ContourContactRMSD > 0.217 :
              ContourContactMinLenAreaFract <= 0.309 : LM5 (20)
              ContourContactMinLenAreaFract > 0.309 :
                RibbonContactRMSD <= 0.331 : LM6 (12)
                RibbonContactRMSD > 0.331 : LM7 (20)
          RibbonJunctionMinInteriorAngle > 2.232 : LM8 (29)
        RibbonContactRMSD > 0.36 : LM9 (91)
      RibbonContactArea > 446.36 : LM10 (53)
  RibbonContactRMSD > 0.375 :
    RibbonContactArea <= 235.969 : LM11 (3015)
    RibbonContactArea > 235.969 :
      RibbonContact1mmMeanCurvL2 <= 0.121 : LM12 (603)
      RibbonContact1mmMeanCurvL2 > 0.121 : LM13 (151)
RibbonContactRMSD > 0.429 : LM14 (7416)
```

"Matches with small ContactRMSD,
high Planarity, a small interior angle at least at one junction,
and a large relative contact length are likely to be correct"
(score is large)

Decision tree
learned on Synthetic Fresco

Experimental Data Sets

Synthetic Fresco

- Made specifically for this project
- Made in the style of Akrotiri wall paintings
- Destroyed purposely in 2007 A.D.

Tongeren Vrijthof

- Tongeren, Belgium
- Roman building
- Destroyed by fire between 1 A.D. – 300 A.D.

Akrotiri

- Thera (Santorini, Greece)
- Late Bronze Age settlement
- Destroyed by earthquake around 1650 B.C.

Experiment Design

Train on Fresco X

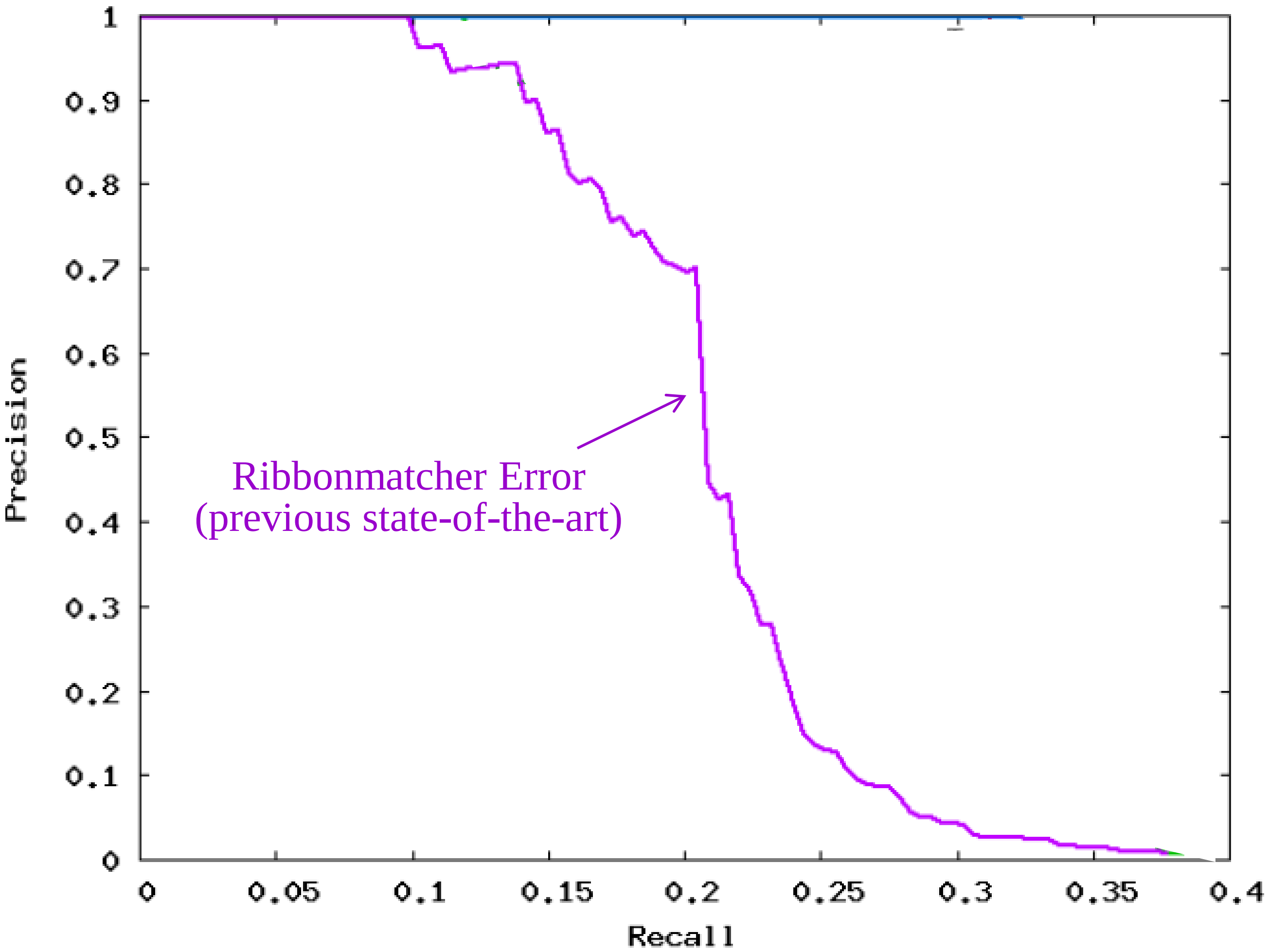
- Use ribbonmatcher to generate candidate matches
- Compute properties of candidate matches
- Mark candidate matches that are correct
- Learn classifier to predict correctness of matches

Test on Fresco Y

- Use ribbonmatcher to generate candidate matches
- Compute properties of candidate matches
- Mark candidate matches that are correct
- Apply classifier to predict correctness of (score) matches
- Sort matches by score, and plot precision vs. recall

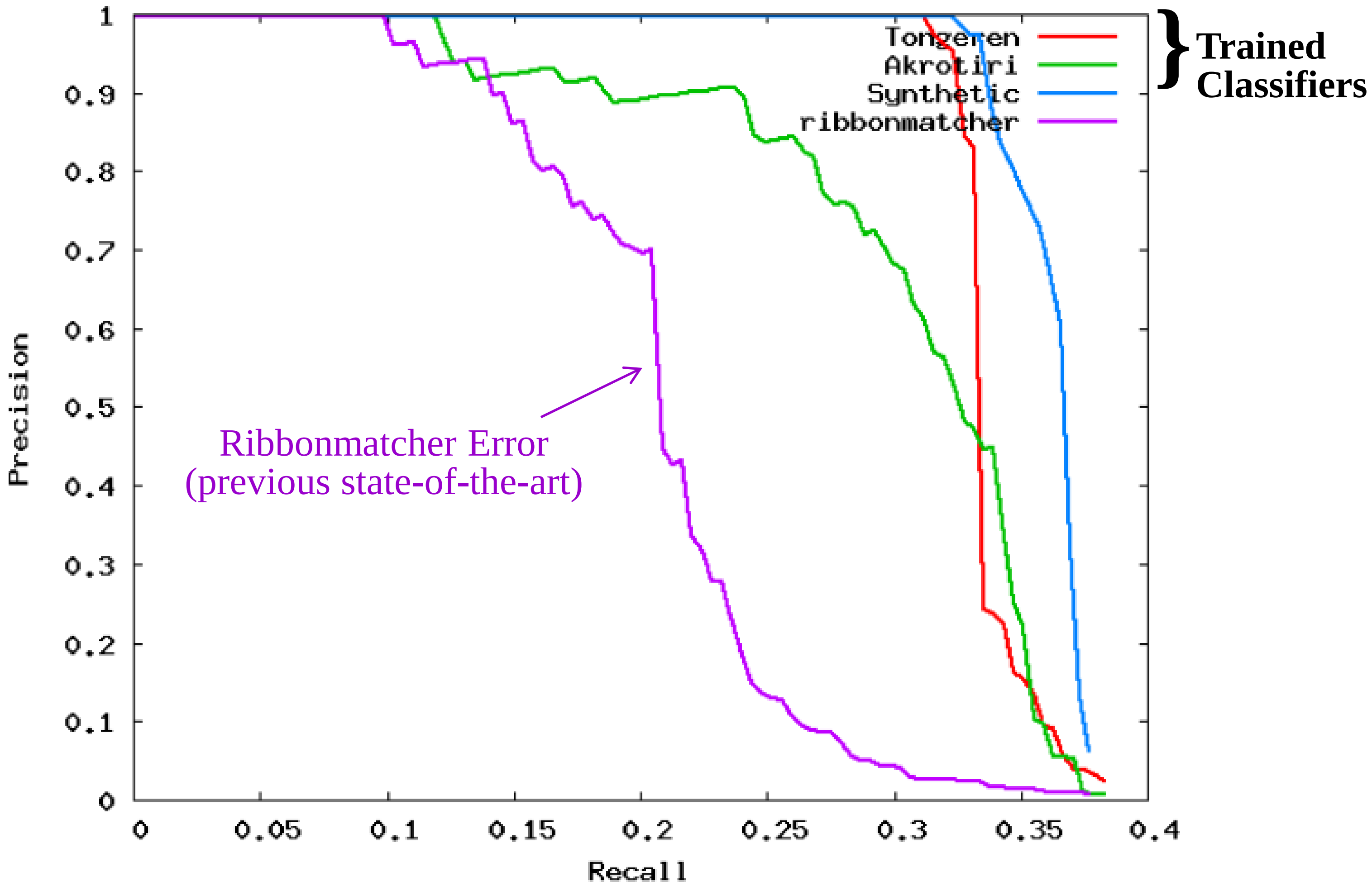
Experiment Results

Testing on Synthetic Fresco:



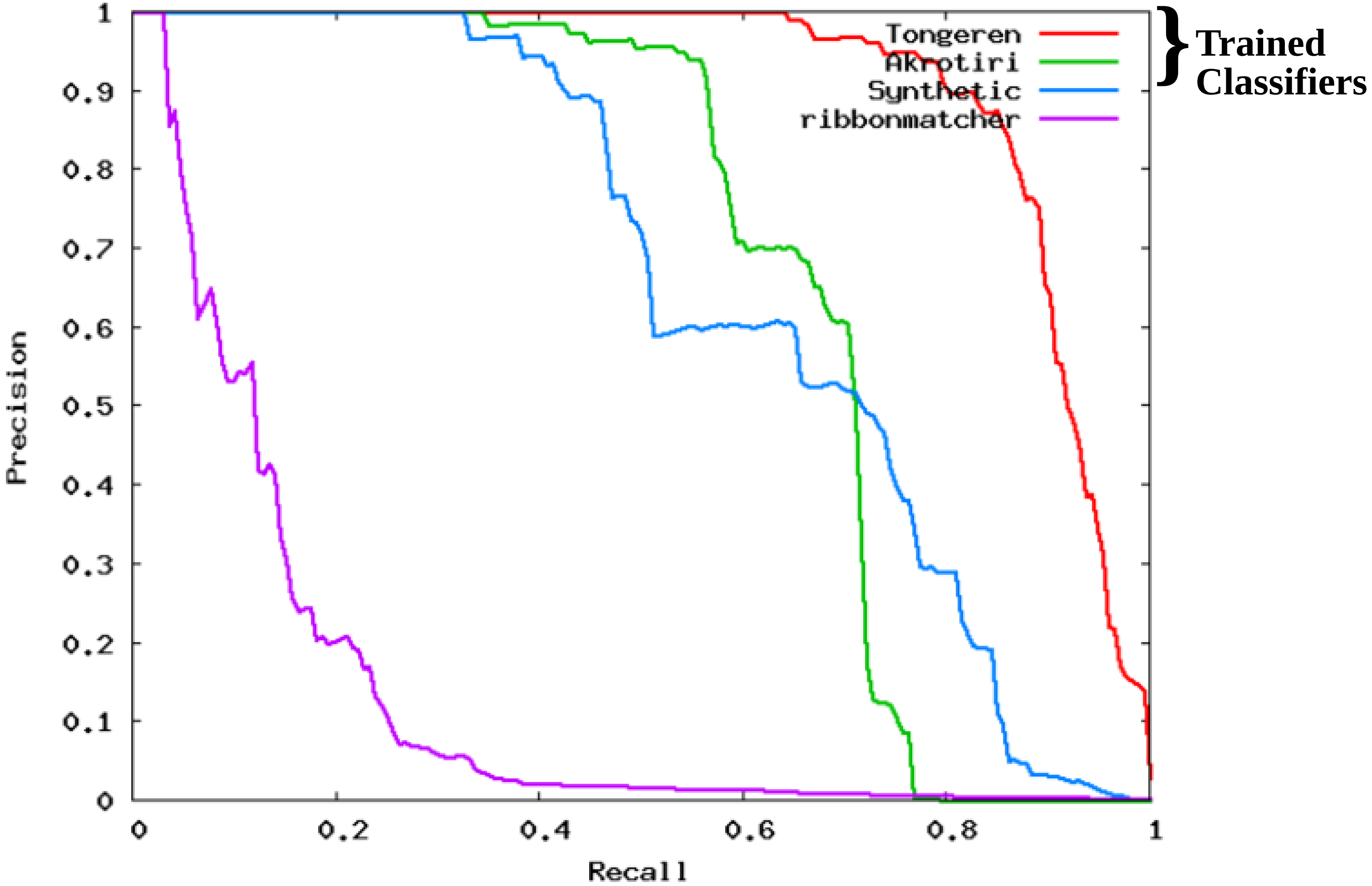
Experiment Results

Testing on Synthetic Fresco:



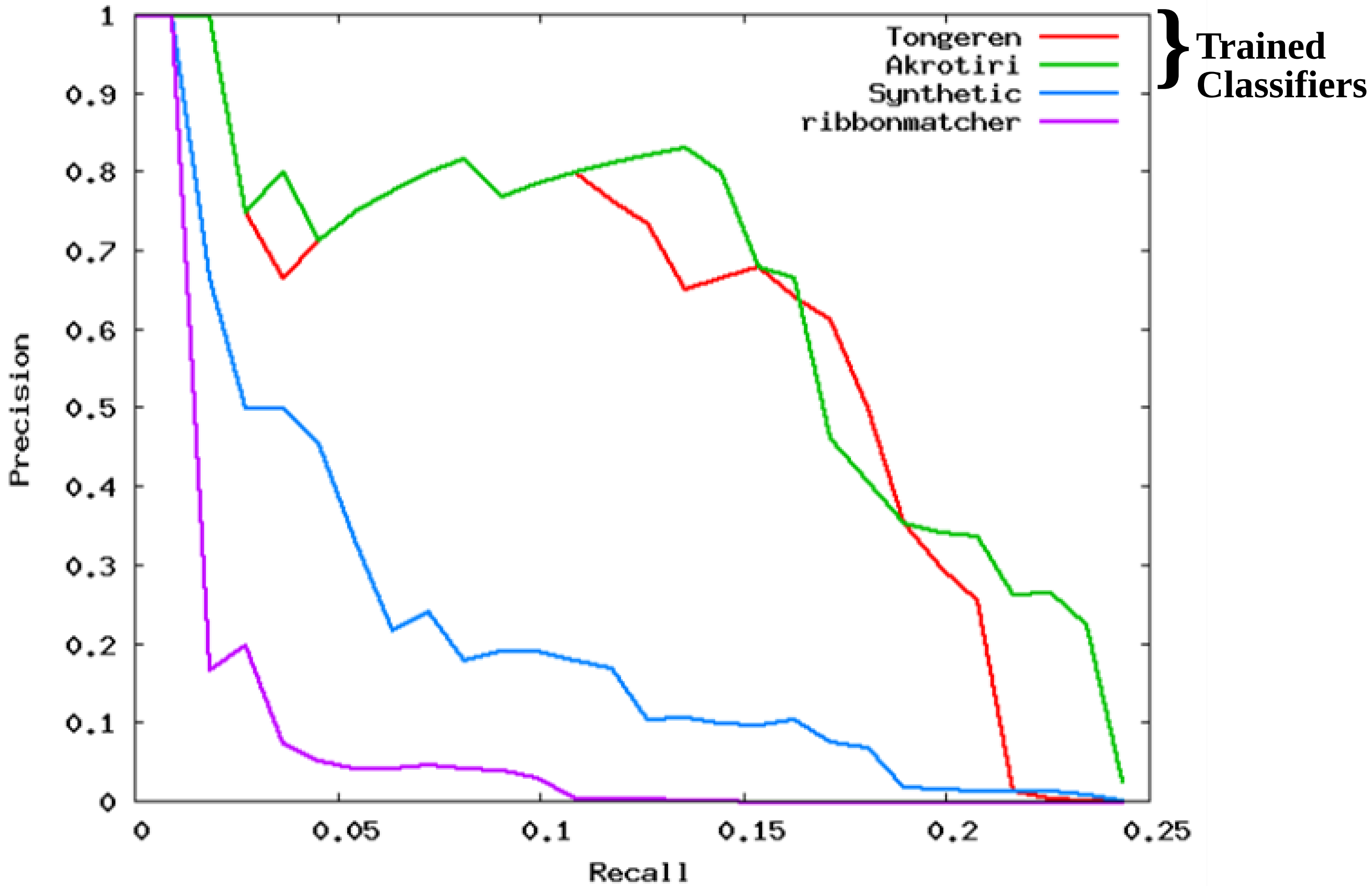
Experiment Results

Testing on Tongeren Vrijthof Fresco:



Experiment Results

Testing on Akrotiri Fresco:



Results of Predictions

Totals of all predictions:

- Likely: 48 correct, 1 incorrect, 1 uncheckable
- Probable: 7 correct, 0 incorrect
- Possible: 25 correct, 19 incorrect, 1 uncheckable
- Maybe: 5 correct, 10 incorrect
- Remote: 2 correct, 15 incorrect
- Longshot: 0 incorrect, 14 incorrect

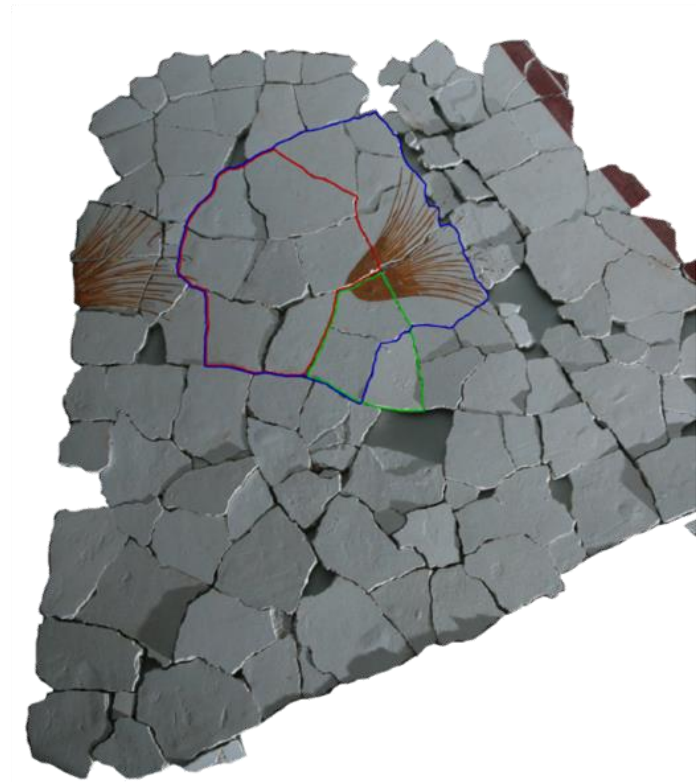
Summary:

- 87 correct matches
- 36 missed (found by conservators)
- 43 new (not found by conservators)

Follow up work …

Assembling matches into full reconstruction

Candidate Matches



Reconstructed Fresco

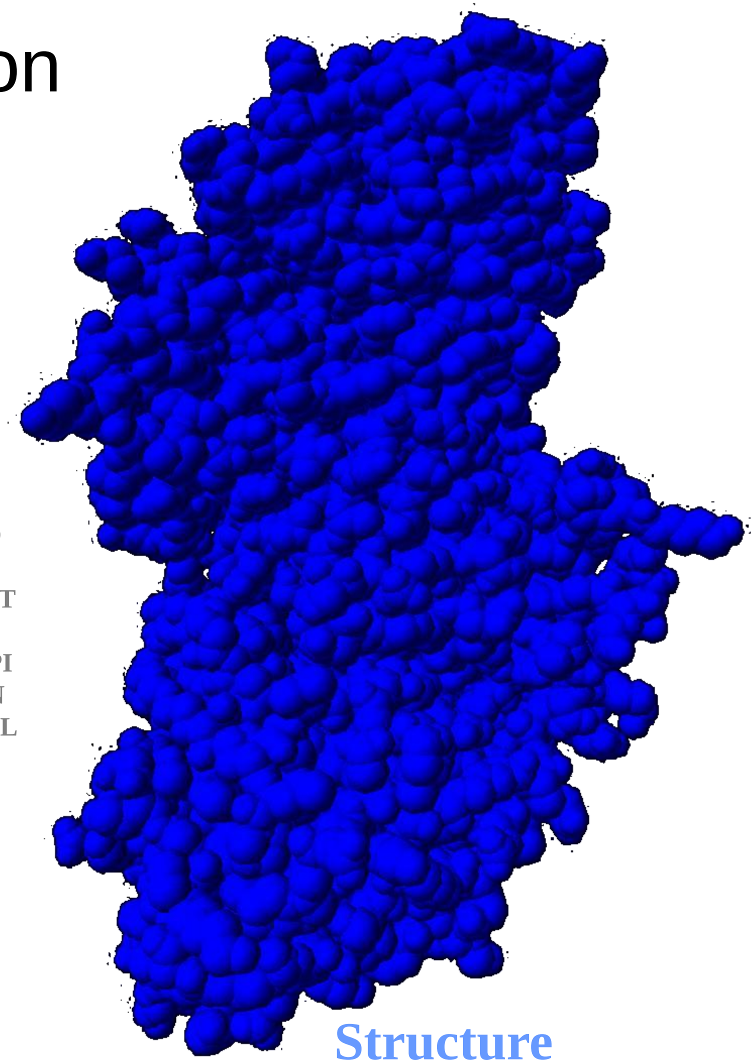
Molecular Biology:
Matching Protein Structures

Goal

Given a protein structure,
predict its molecular function

STAGKVIKCKAAVLWEEKKPFSIEEVEVAPPKAHEVRIKMVATGICRSDD
HVVSGTLVTPLPVIAGHEAAGIVESIGEGVTTVRPGDKVIPLFTPQCGKC
RVCKHPEGNFCLKNDLSMPRGTMQDGTSRFTCRGKPIHHFLGTSTFSQYT
VVDEISVAKIDAASPLEKVCLIGCGFSTGYGSAVKVAKVTQGSTCAVFGL
GGVGLSVIMGCKAAGAARIIGVDINKDKFAKAKEVGATECVNPQDYKKPI
QEVLTEMSNGGVDFSFEVIGRLDTMVTALSCCQEAYGVSVIVGVPPDSQN
LSMNPMLLLSGRTWKGAIFGGFKSKDSVPKLVADFMAKKFALDPLITHVL
PFEKINEGFDLLRSGESIRTILTF

Sequence



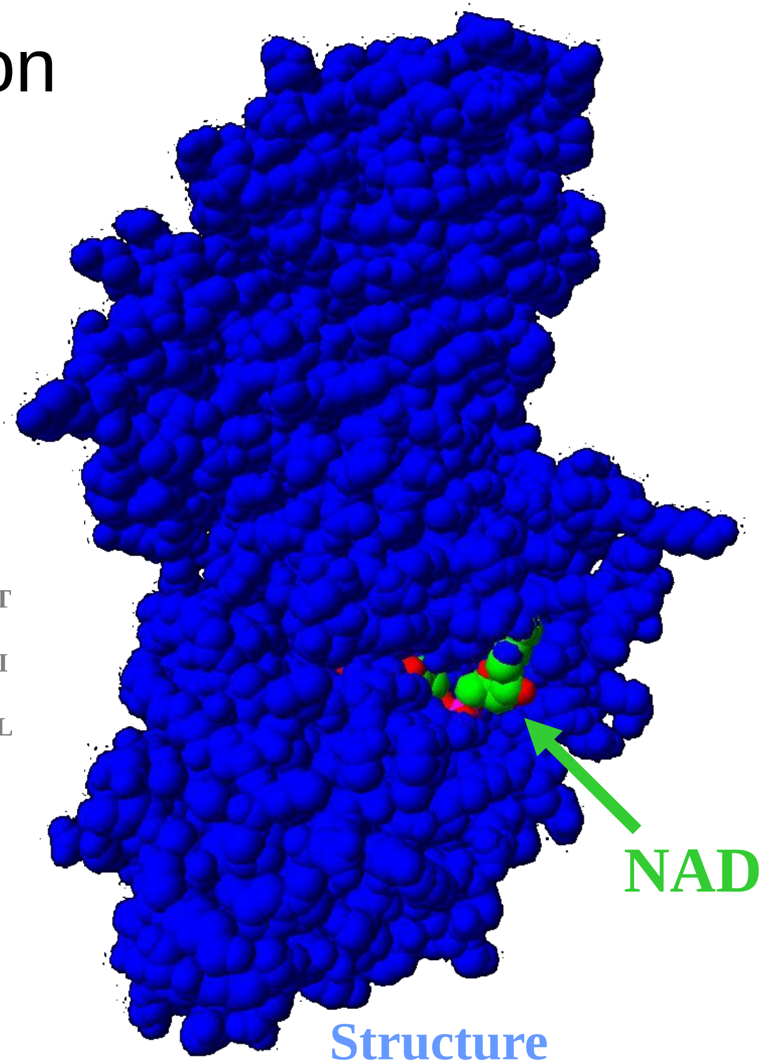
Structure

Goal

Given a protein structure, predict its molecular function (e.g., bound ligand type)

STAGKVIKCKAAVLWEEKKPFSIEEVEVAPPKAHEVRIKMVATGICRSDD
HVVSGTLVTPLPVIAGHEAAGIVESIGEGVTTVRPGDKVIPLFTPQCGKC
RVCKHPEGNFCLKNDLSMPRGTMQDGTSRFTCRGKPIHHFLGTSTFSQYT
VVDEISVAKIDAASPLEKVCLIGCGFSTGYGSAVKVAKVTQGSTCAVFGL
GGVGLSVIMGCKAAGAARIIGVDINKDKFAKAKEVGATECVNPQDYKKPI
QEVLTEMSNGGVDFSFEVIGRLDTMVTALSCCQEAYGVSVIVGVPPDSQN
LSMNPMLLLSGRTWKGAIFGGFKSKDSVPKLVADFMAKKFALDPLITHVL
PFEKINEGFDLLRSGESIRTILTF

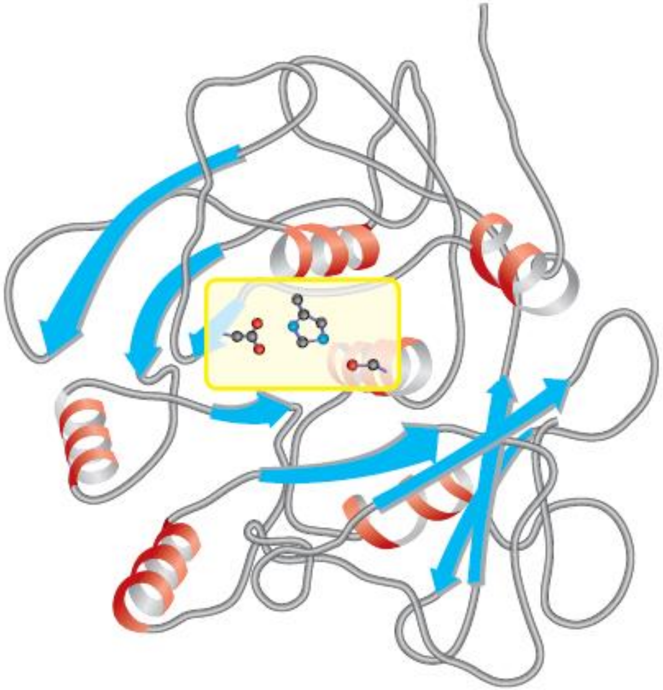
Sequence



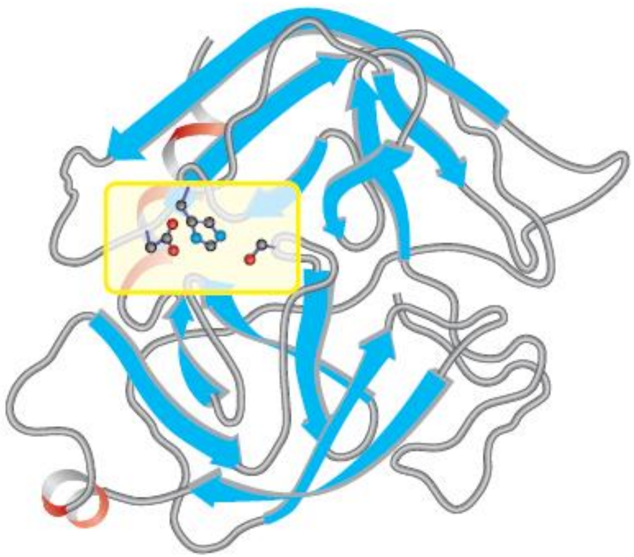
Structure

Observation

Similarities of 3D structures in binding sites can reveal functional similarities



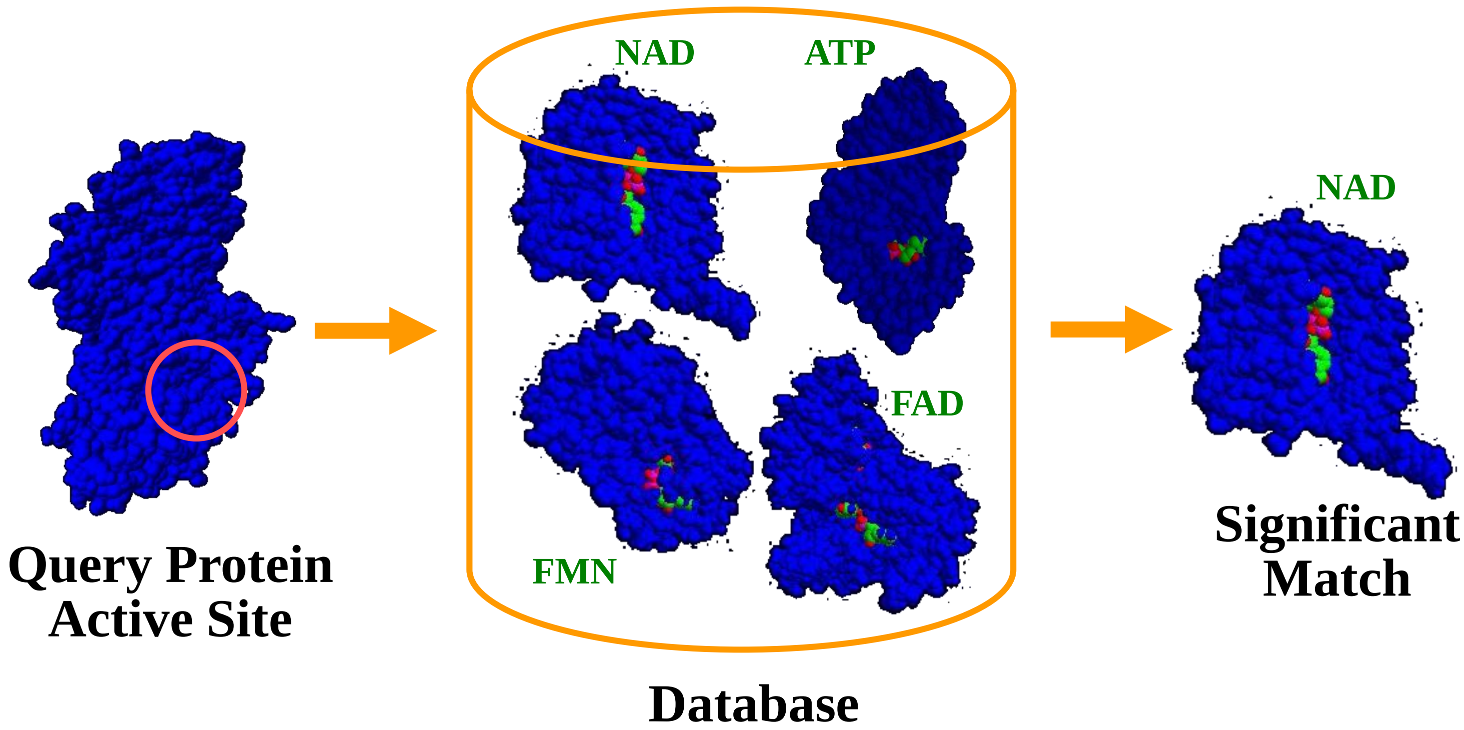
Subtilisin
(bacterial serine protease)



Chymotrypsin
(mammalian serine protease)

General Strategy

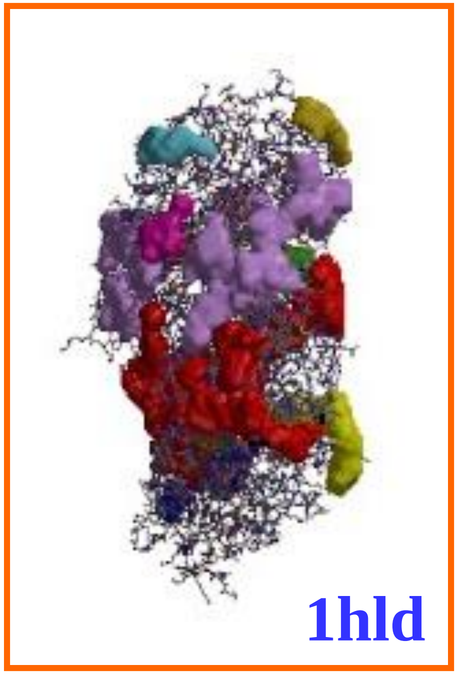
Match structure of query protein active site to others with known functions, and transfer annotations



Previous Work

Predicting protein-ligand binding site locations

- SURFNET [Laskowski95]
- LIGSITE [Hendlich97]
- Pocketfinder [An04]
- CAST [Liang98]
- PASS [Brady00]
- FEATURE [Wei03, Yoon07]
- Q-SiteFinder [Laurie05]
- Solvent mapping [Silberstein03]
- Conservation (e.g., [Lichtarge02])
- etc.

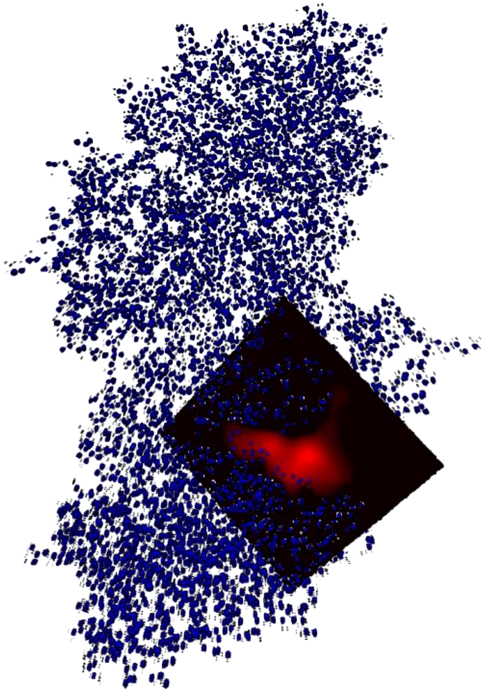


SURFNET [Laskowski96]

Previous Work

Geometric models of protein-ligand binding sites

- Grids [Goodford85]
- Templates [Wallace97]
- Shells [Wei98]
- Alpha shapes [Liang98]
- Pseudo-centers [Schmitt02]
- Surfaces [Kinoshita03]
- Radial extents [Morris05]
- etc.

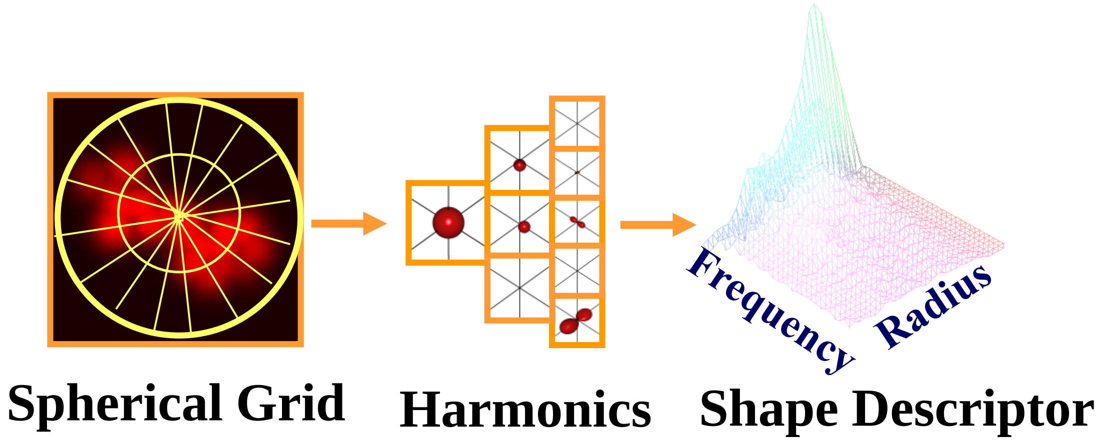


Grid-based Model of Binding Site

Previous Work

Efficient algorithms for matching site models

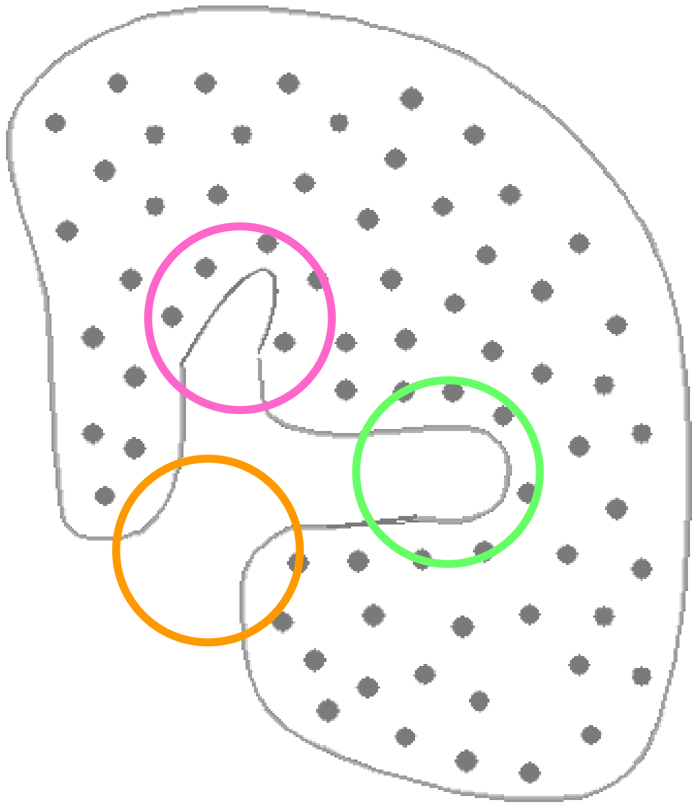
- Fast Fourier Transform [Katchalski-Katzir92]
- Association graphs [Artymiuk94]
- Geometric hashing [Wolfson97]
- Fast rotational matching [Kovacs02]
- Combinatorial expansion [Ferre04]
- ➢Shape descriptors (e.g., [Funkhouser06])
- etc.



Problem

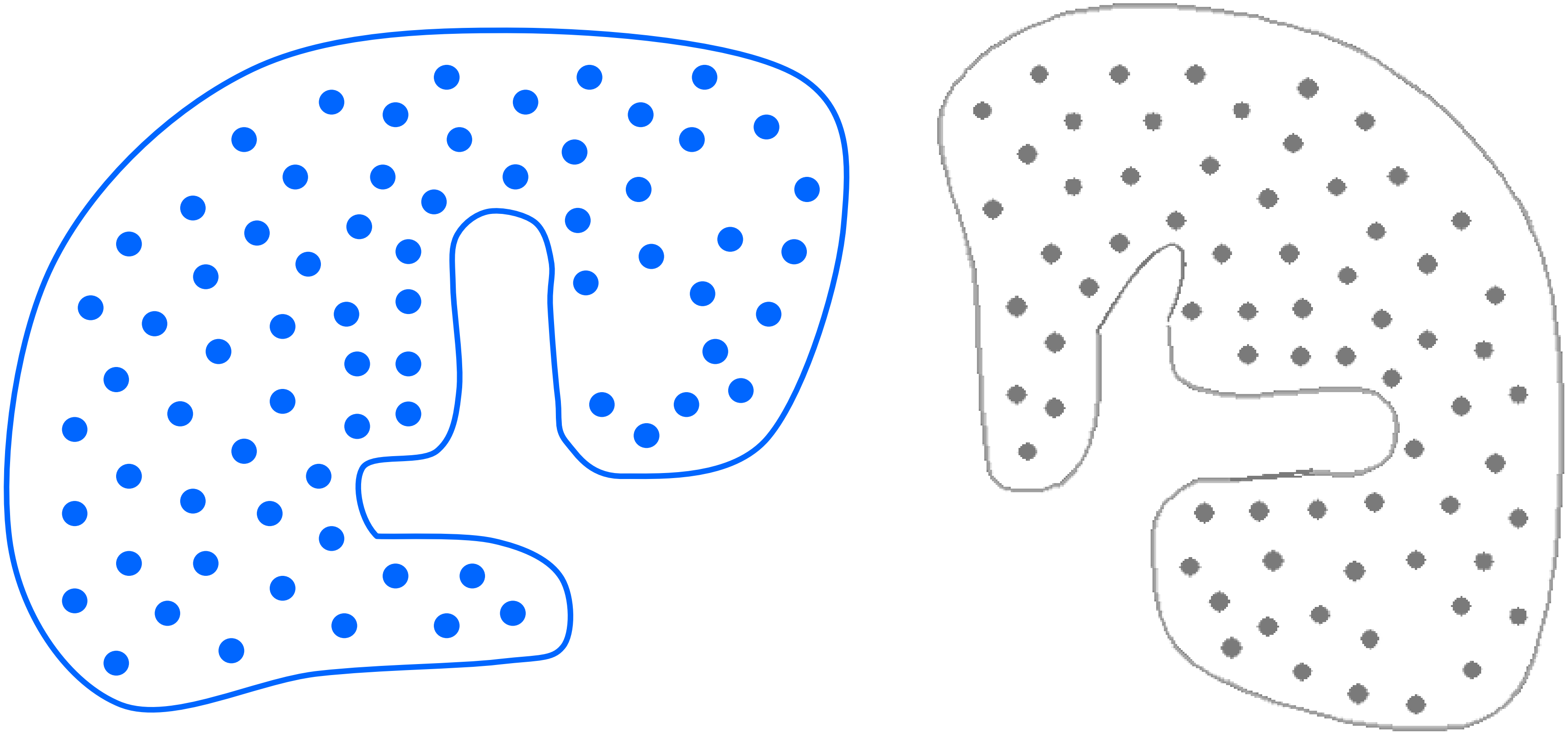
How find distinctive, partial regions of cavities?

How use them to match proteins?



Overview

Given query protein, and labeled target

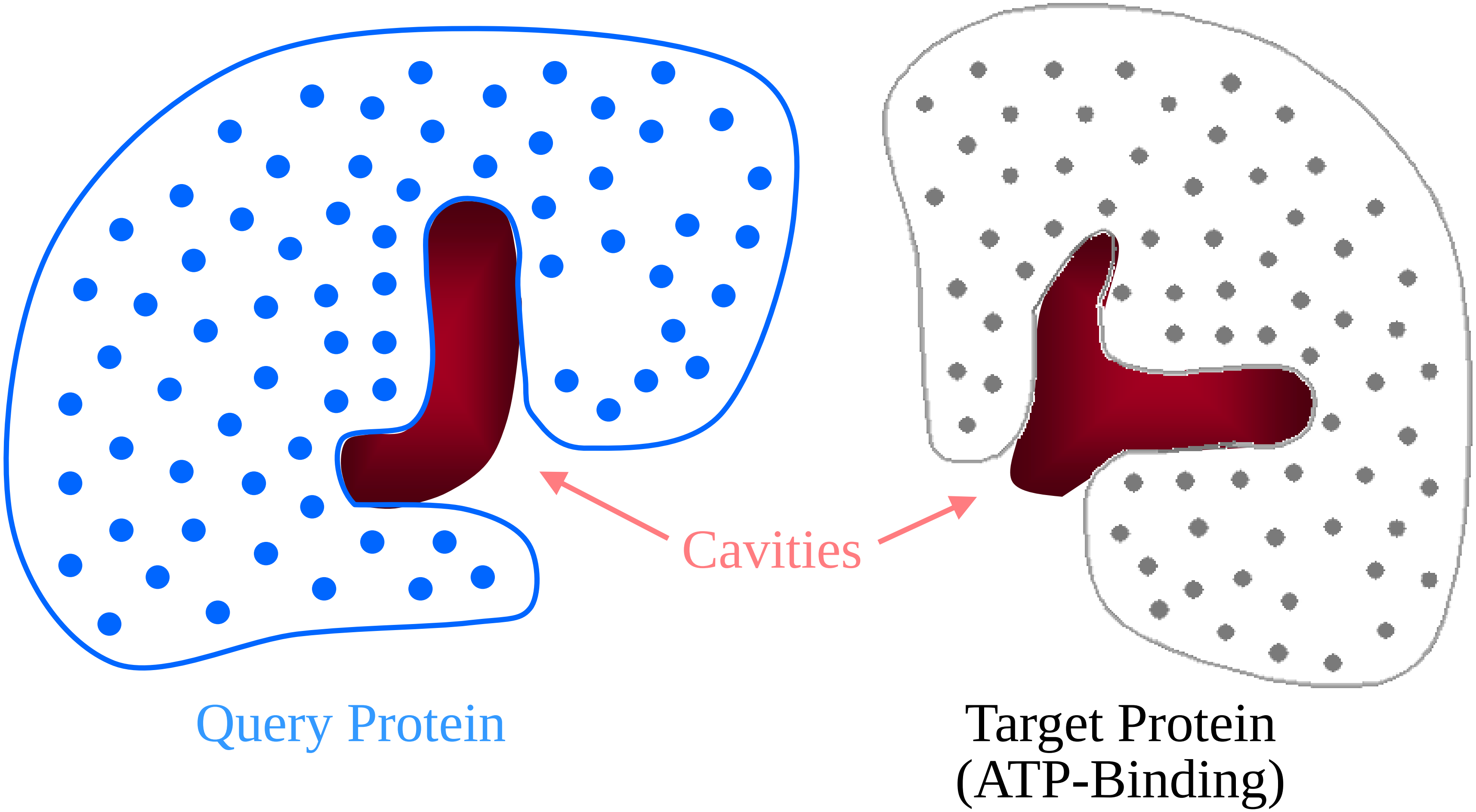


Query Protein

Target Protein
(ATP-Binding)

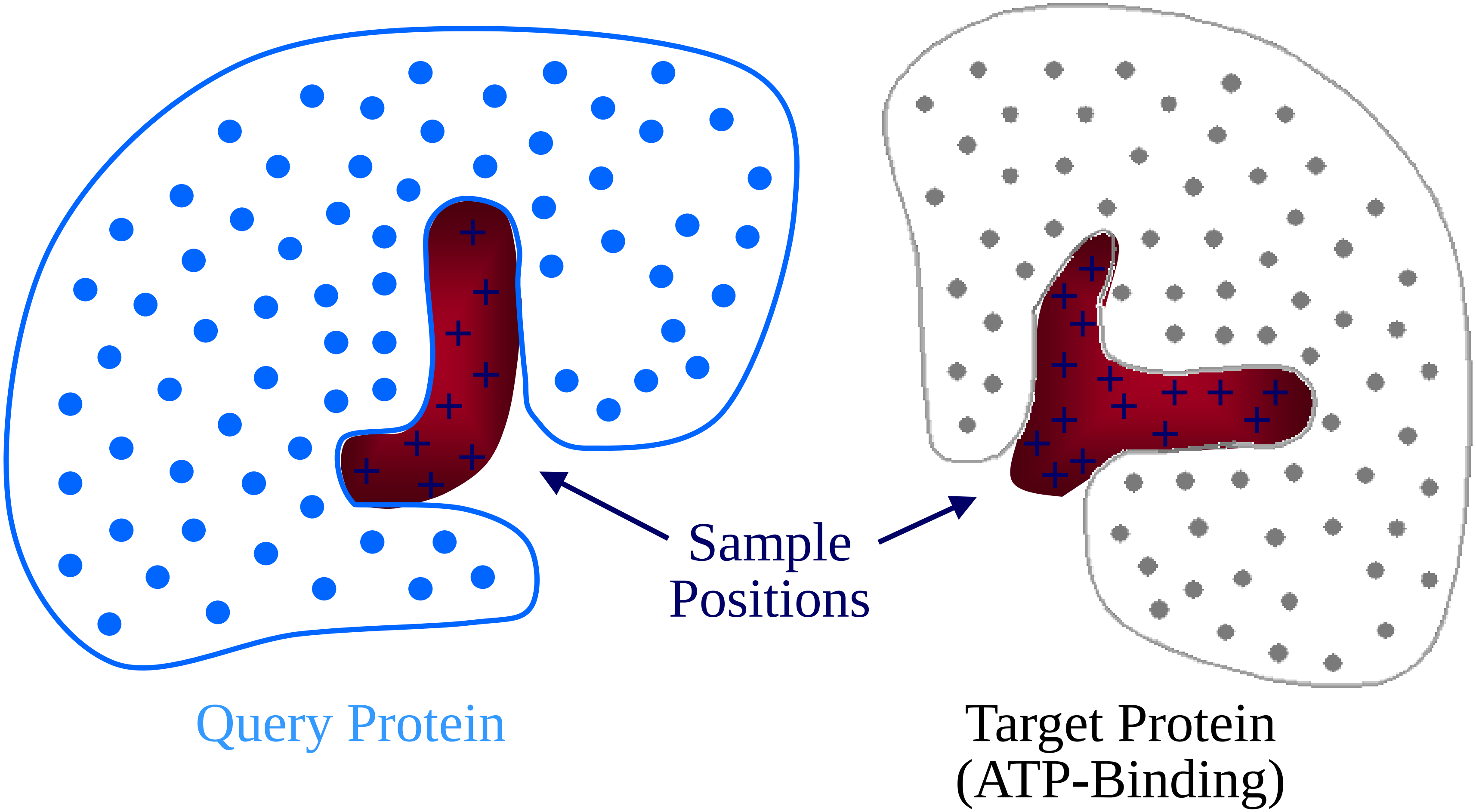
Overview

Step 1: Locate cavities



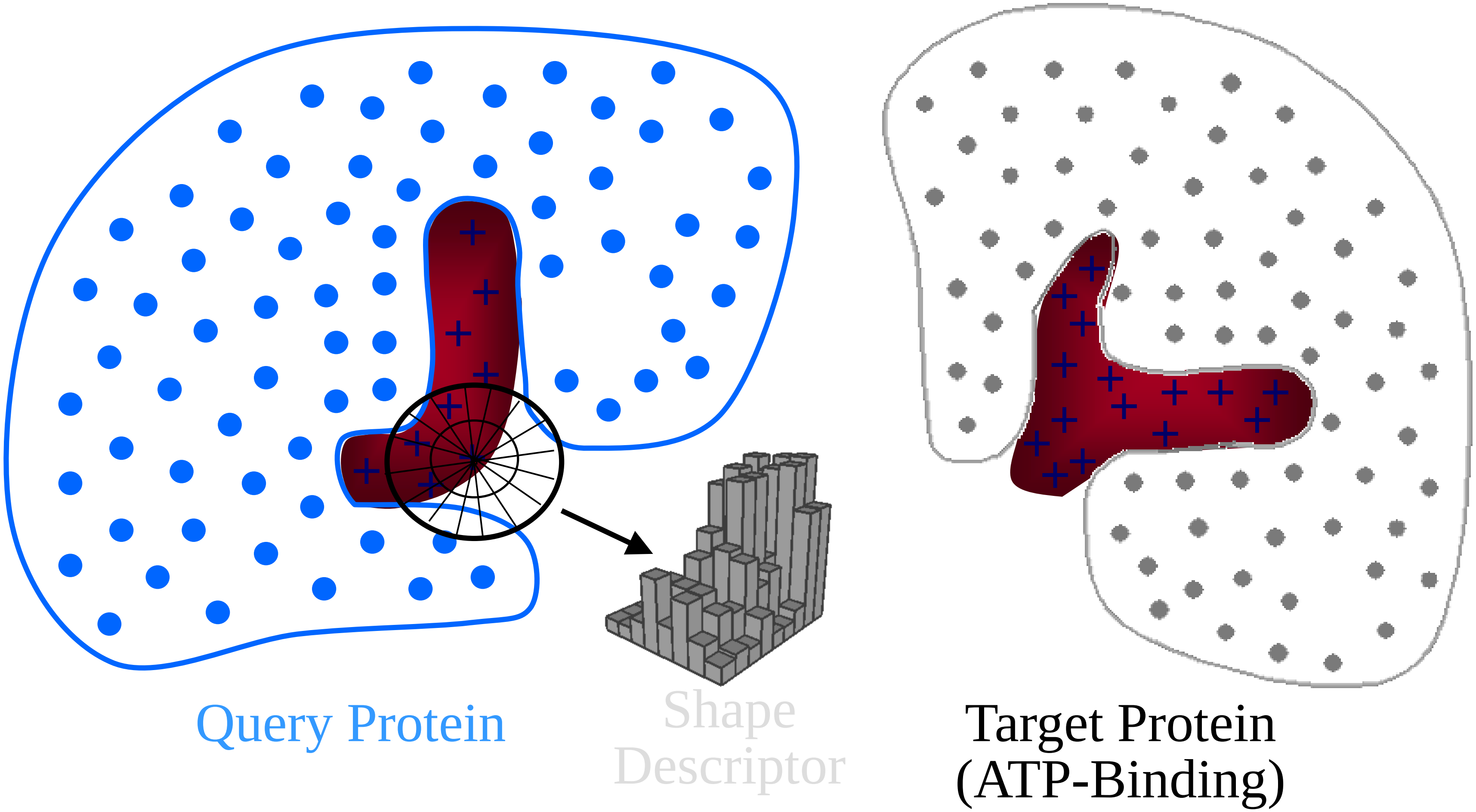
Overview

Step 2: Sample positions inside cavity



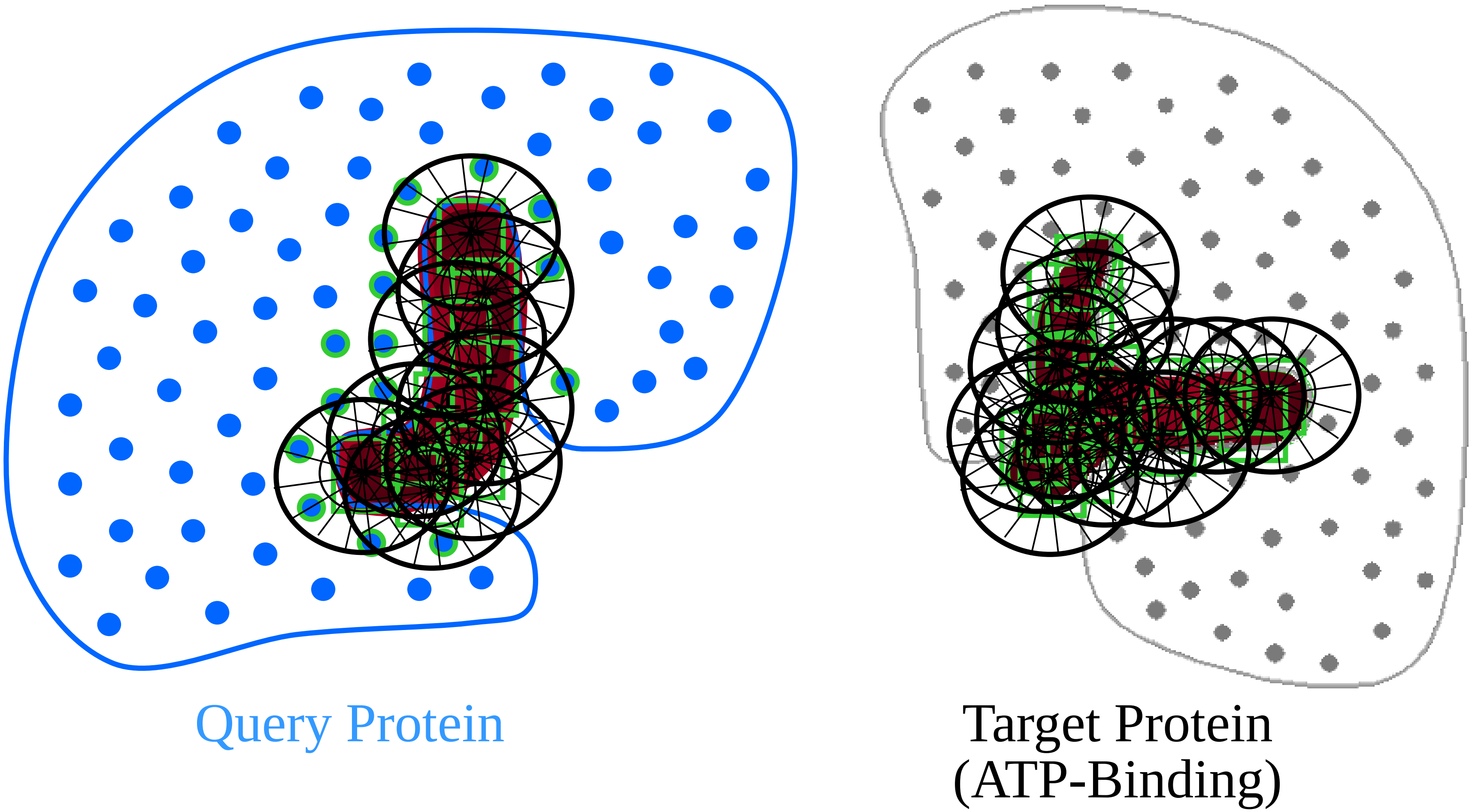
Overview

Step 3: Compute shape descriptors at every sample



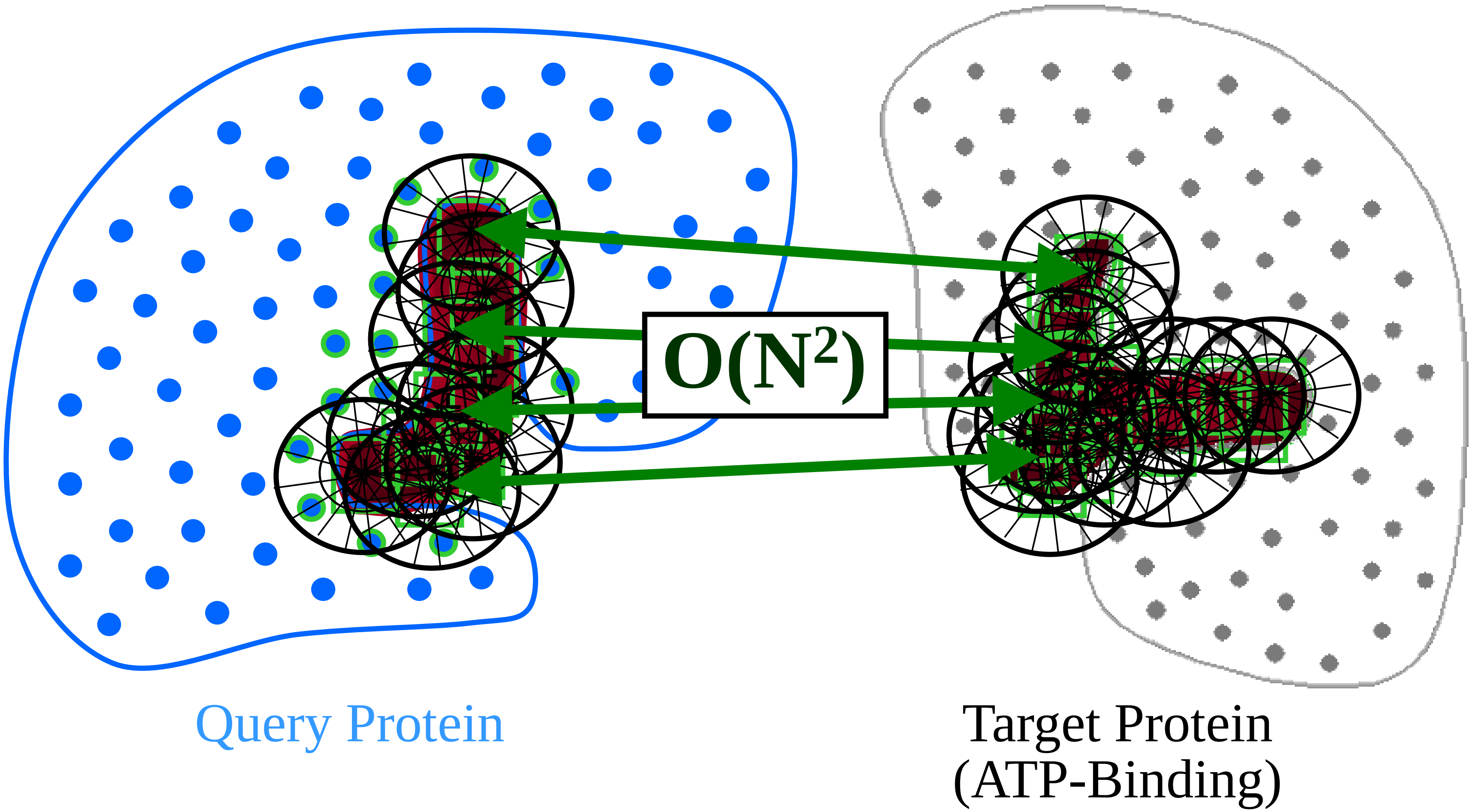
Overview

Step 3: Compute shape descriptors at every sample



Overview

Step 4: Match all pairs of shape descriptors



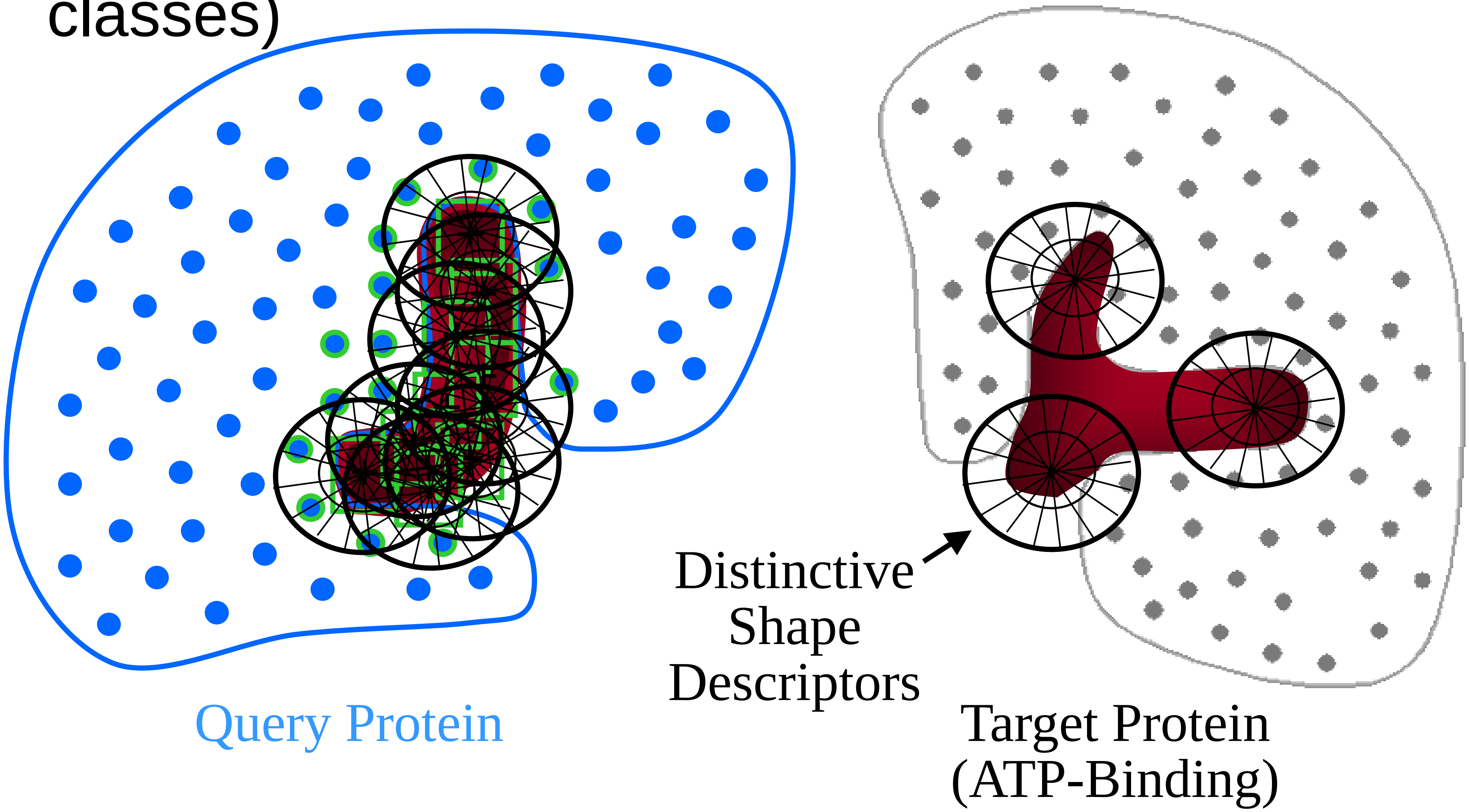
Overview

Step 4: Match all pairs of shape descriptors

NOT!

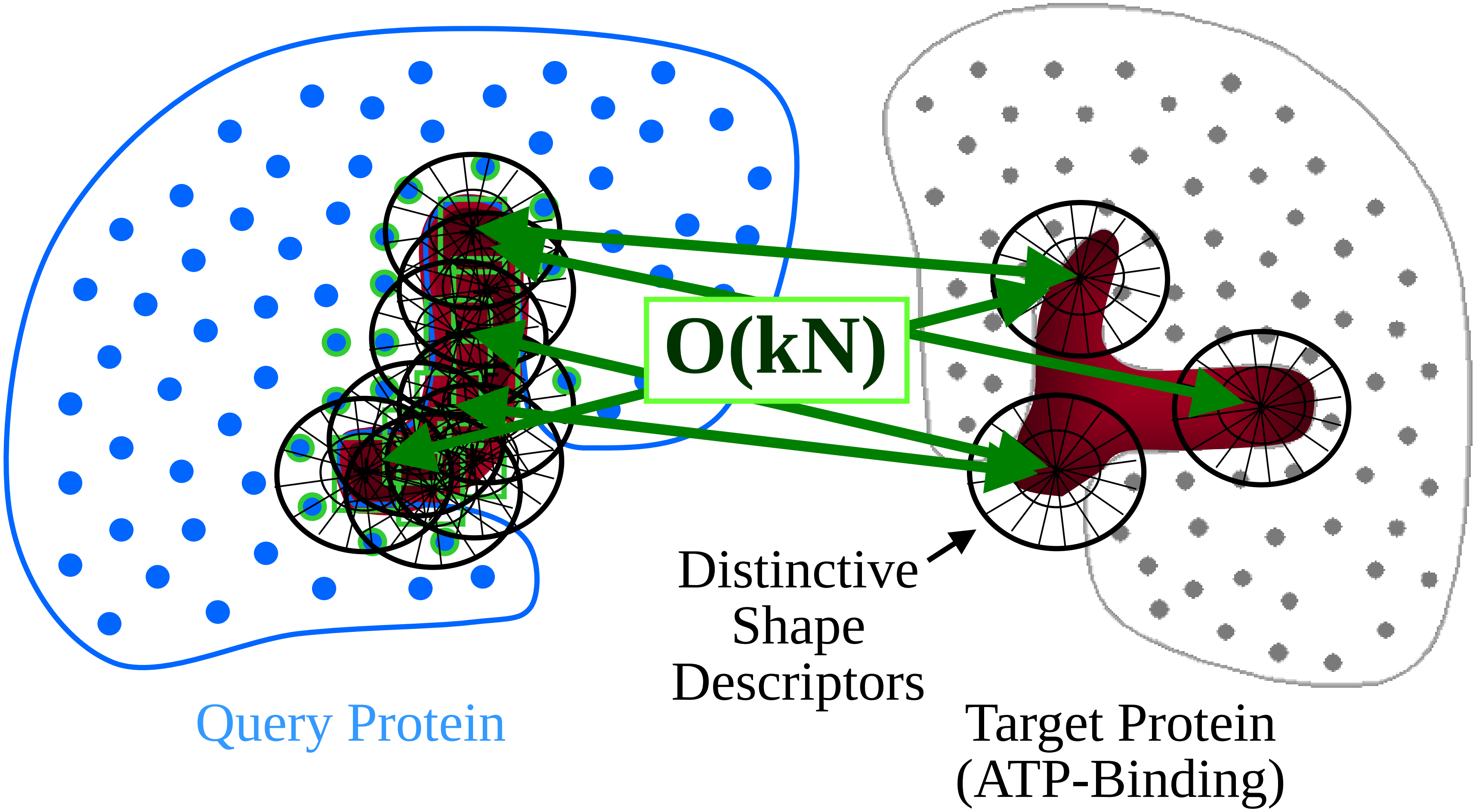
Overview

Step 4: Select *distinctive* shape descriptors for target (ones learned to discriminate functional classes)



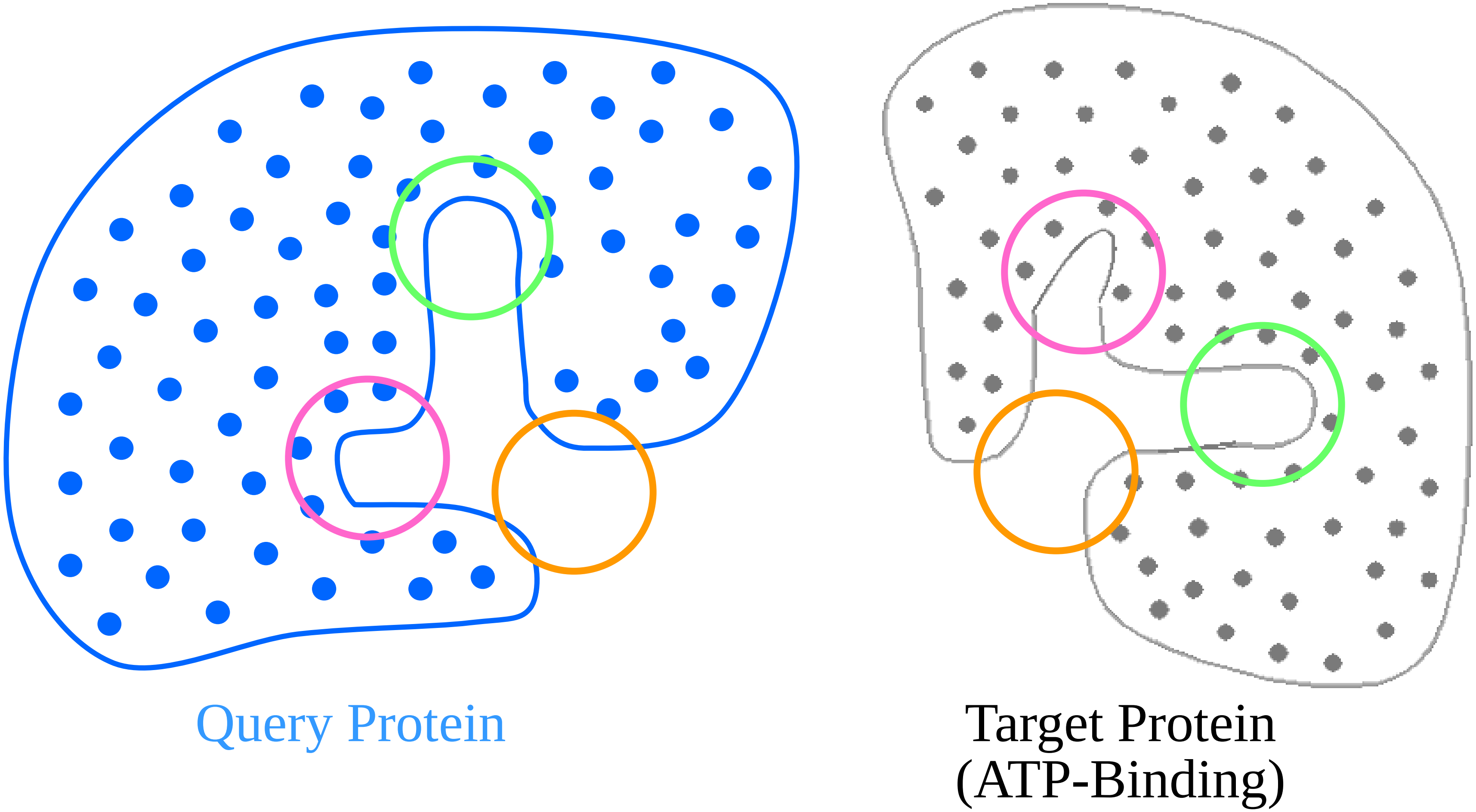
Overview

Step 5: Match query samples only to distinctive ones



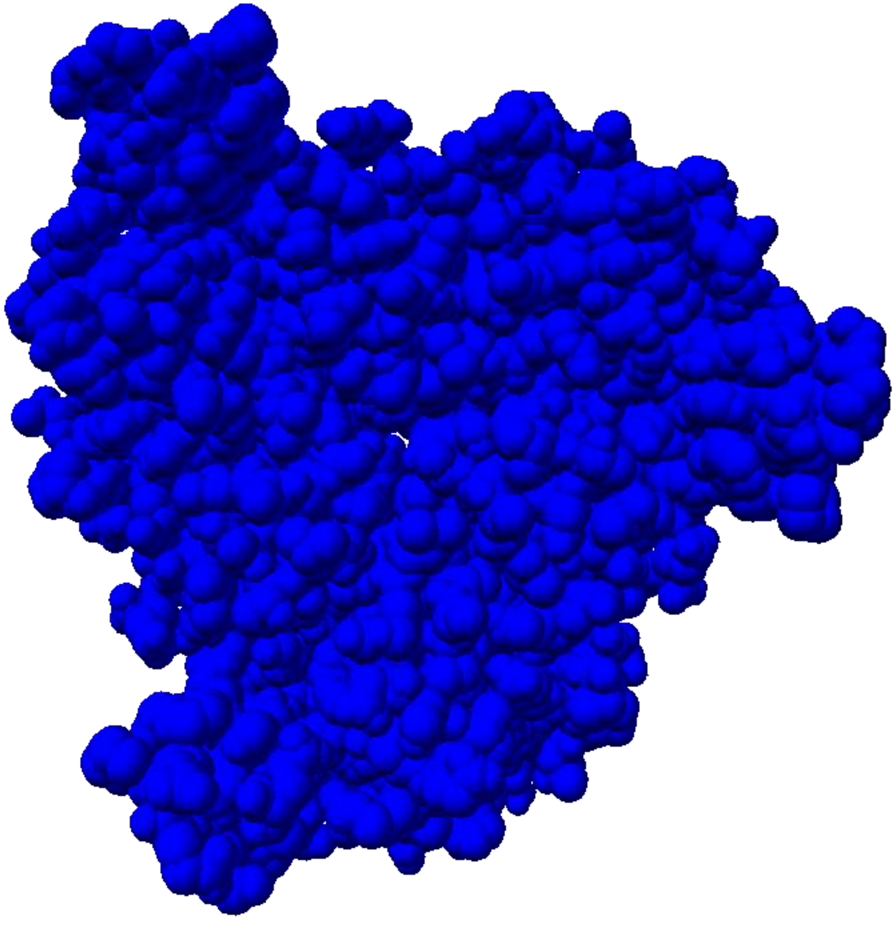
Overview

Step 6: Report the best match(es)



Methods

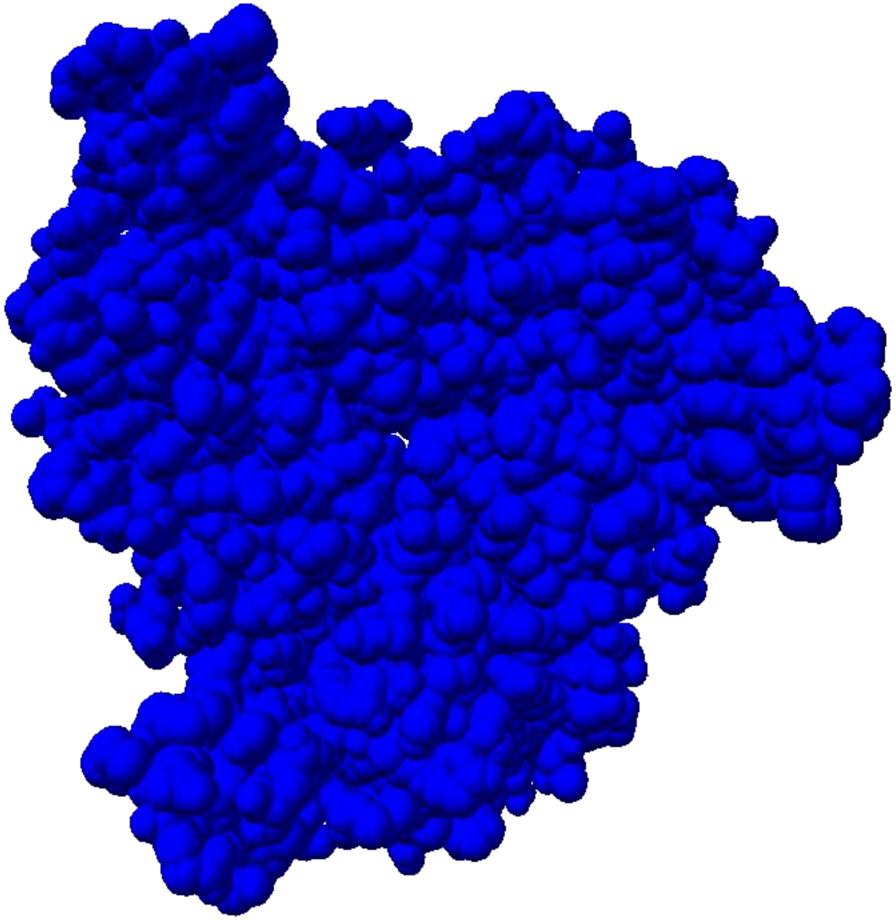
Input: protein structure



3D Atomic Coordinates

Methods

1) Locate cavities

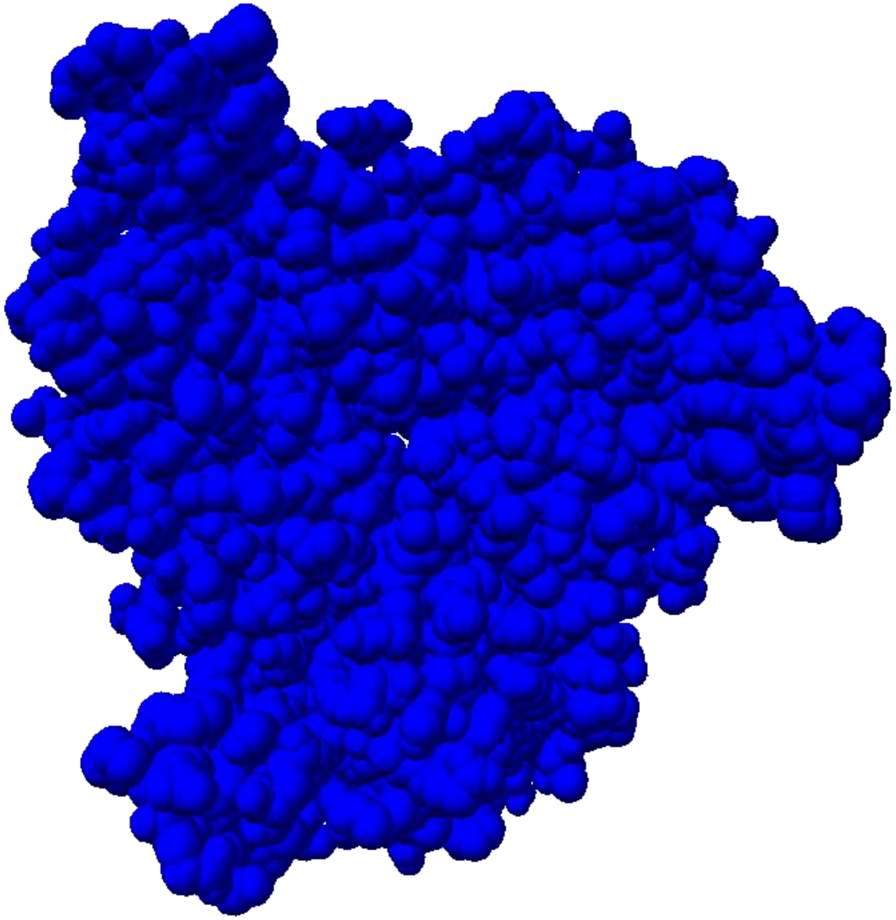


3D Atomic Coordinates

Methods

1) Locate cavities

- Form feature vector for every grid point
- Learn classifier [weka] to recognize ligands
- Use classifier to estimate probability of finding ligand at any grid point
- Ensure spatial coherence and consistency

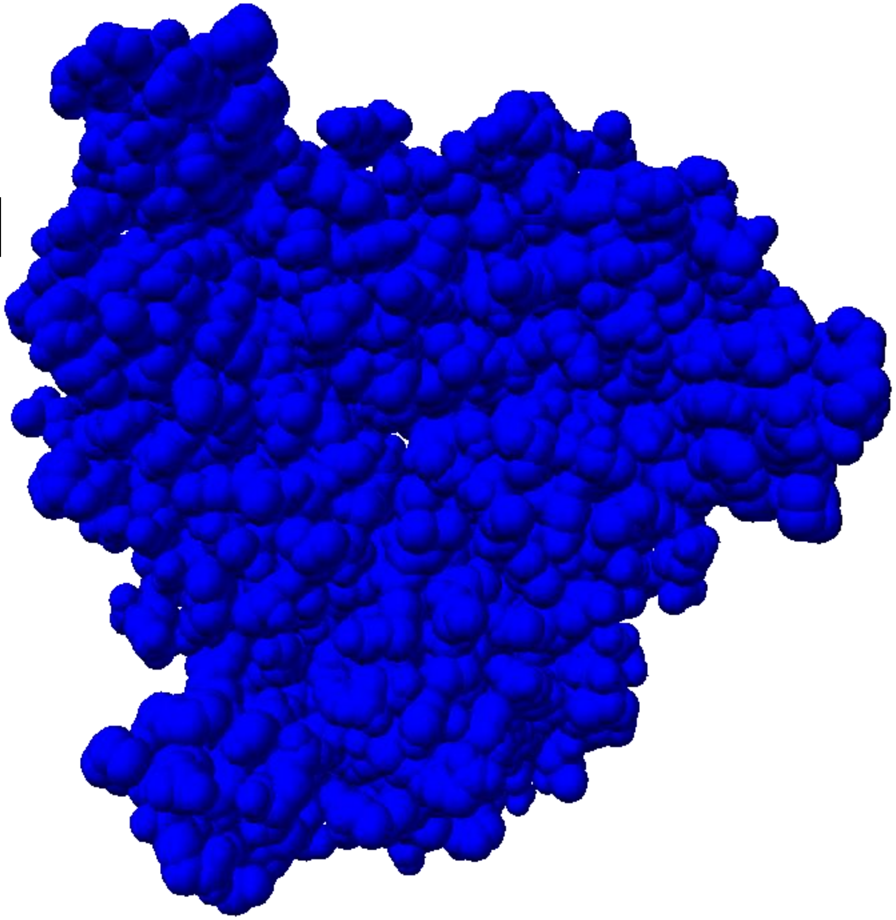


3D Atomic Coordinates

Methods

1) Locate cavities

- Form feature vector for every grid point
  - Ligsite [Hendlich97]
  - Surfnet* [Laskowski95]
  - Pocketfinder [An04]
  - Distance from surface
  - Multichain distances
  - Cavity size and rank

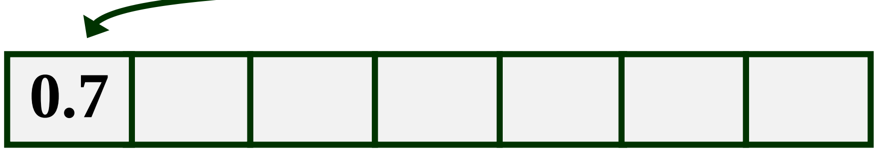


3D Atomic Coordinates

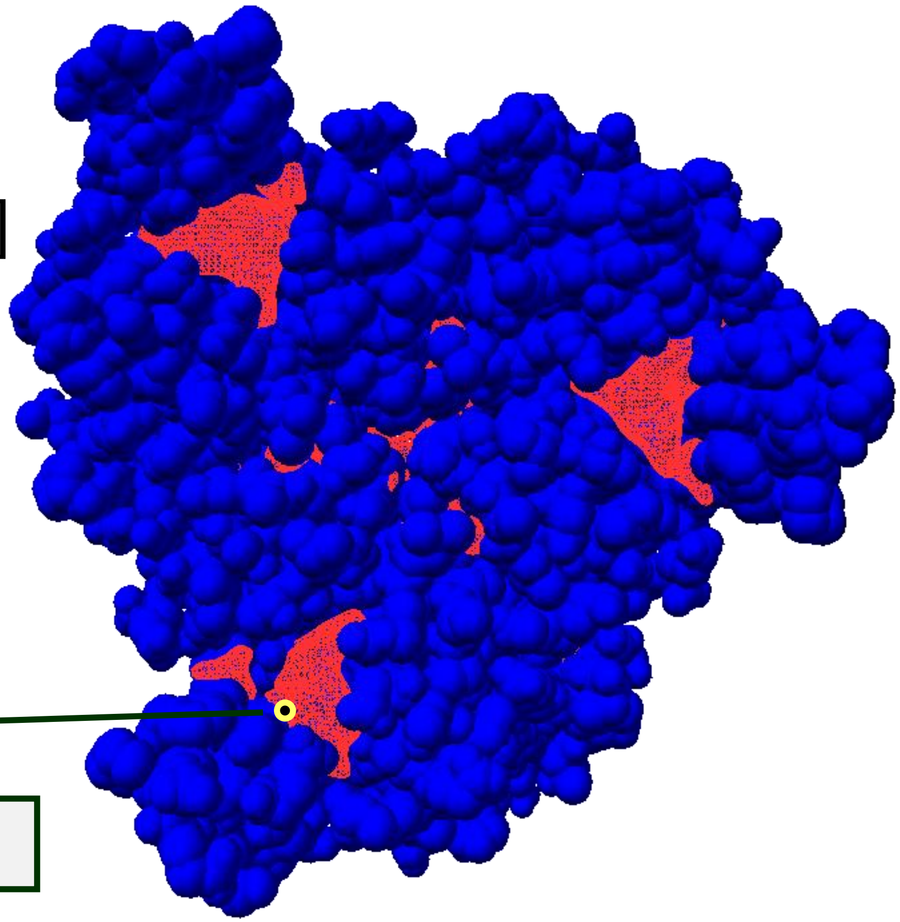
Methods

1) Locate cavities

- Form feature vector for every grid point
  - Ligsite [Hendlich97]
    - Surfnet* [Laskowski95]
    - Pocketfinder [An04]
    - Distance from surface
    - Multichain distances
    - Cavity size and rank



Feature Vector



Ligsite [Hendlich97]

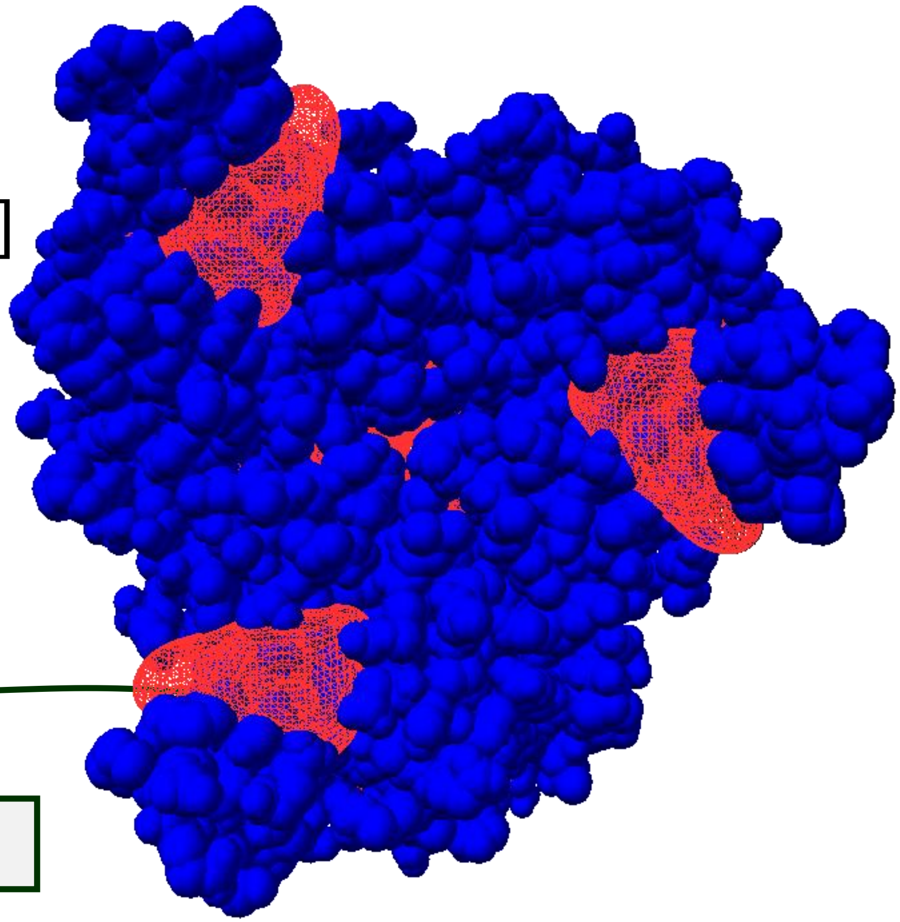
Methods

1) Locate cavities

- Form feature vector for every grid point
  - Ligsite [Hendlich97]
  - Surfnet* [Laskowski95]
  - Pocketfinder [An04]
  - Distance from surface
  - Multichain distances
  - Cavity size and rank

0.7	**0.6**					

Feature Vector

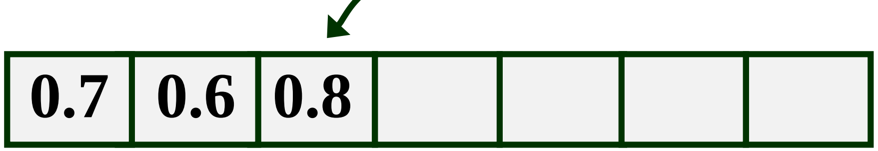


Surfnet* [Laskowski95]

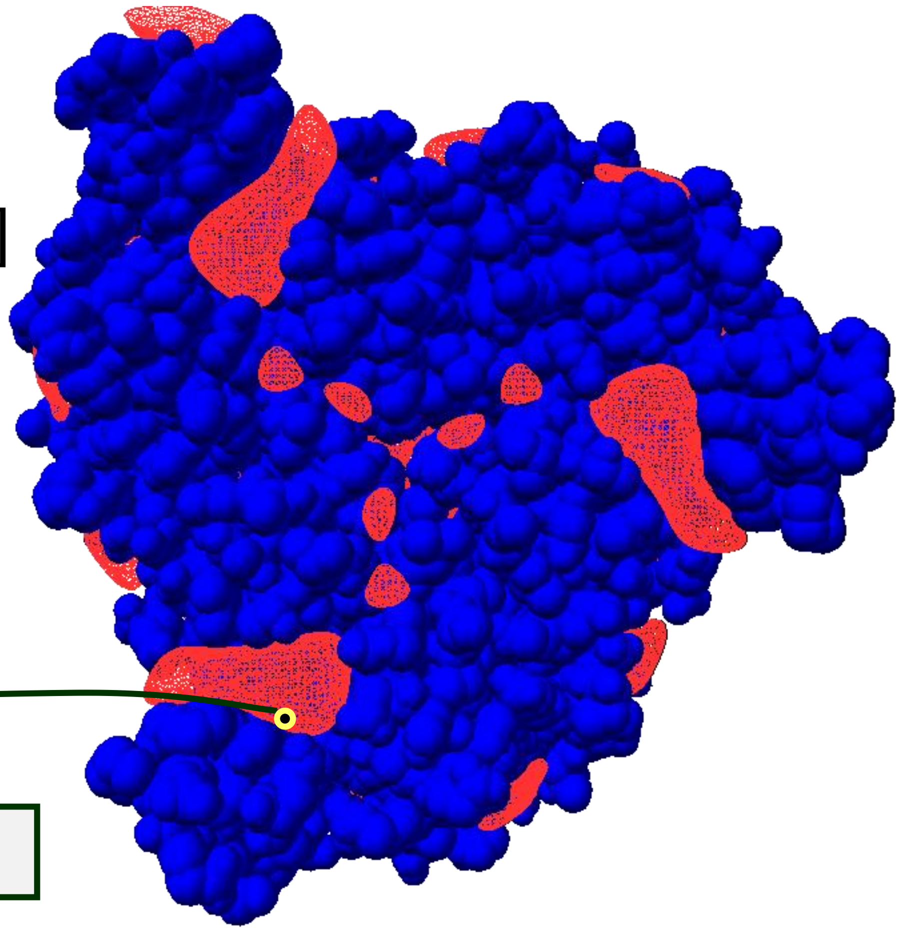
Methods

1) Locate cavities

- ➢Form feature vector for every grid point
  - Ligsite [Hendlich97]
  - Surfnet* [Laskowski95]
  - ➢Pocketfinder [An04]
  - Distance from surface
  - Multichain distances
  - Cavity size and rank



Feature Vector

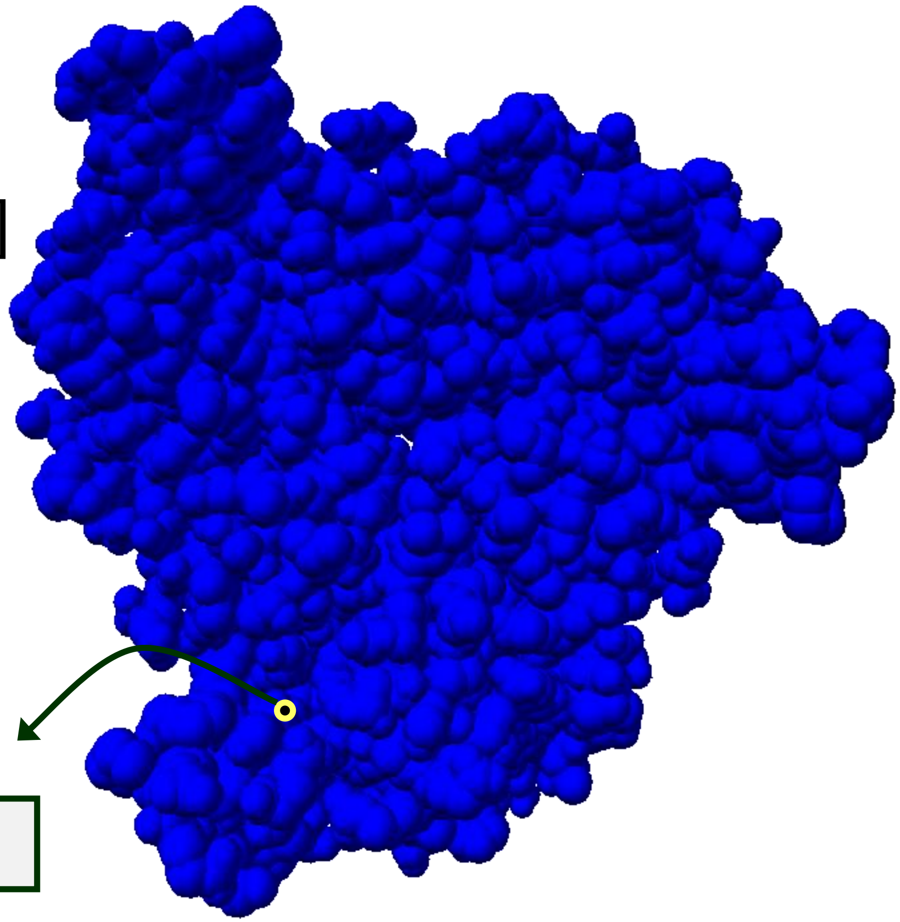


Pocketfinder [An04]

Methods

1) Locate cavities

- Form feature vector for every grid point
  - Ligsite [Hendlich97]
  - Surfnet* [Laskowski95]
  - Pocketfinder [An04]
  - Distance from surface
  - Multichain distances
  - Cavity size and rank



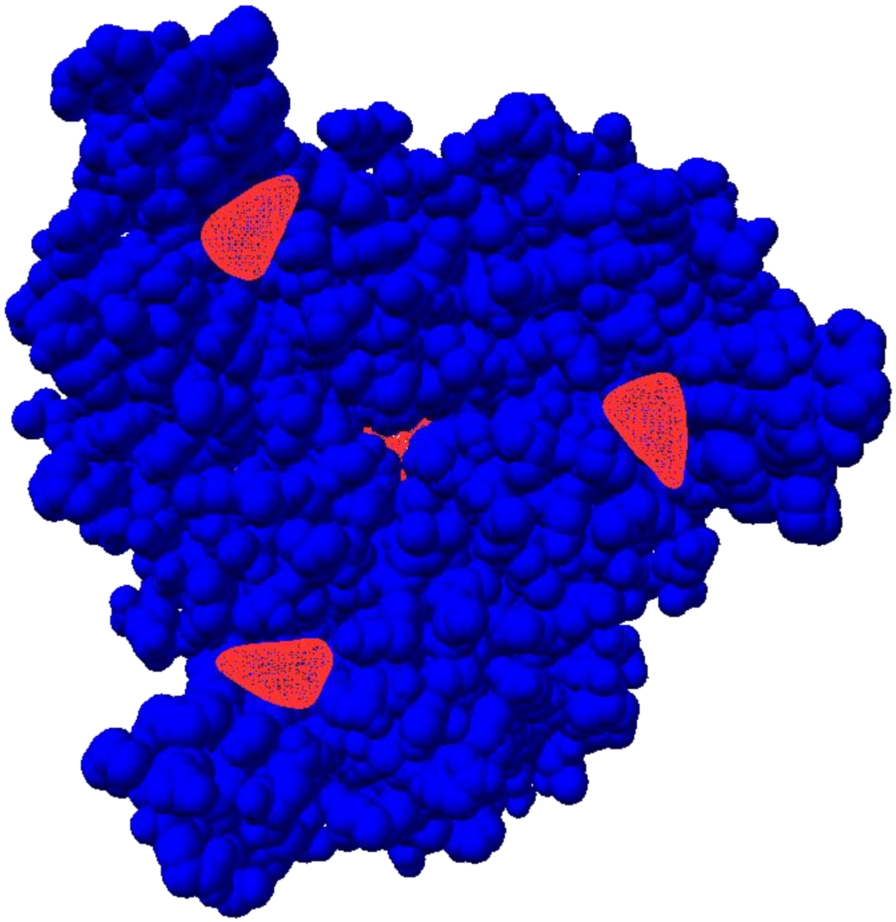
0.7	0.6	0.8	2.1	0	23	1

Feature Vector

Methods

1) Locate cavities

- Form feature vector for every grid point
- Learn classifier [weka] to recognize ligands
- Use classifier to estimate probability of finding ligand at any grid point
- Ensure spatial coherence and consistency

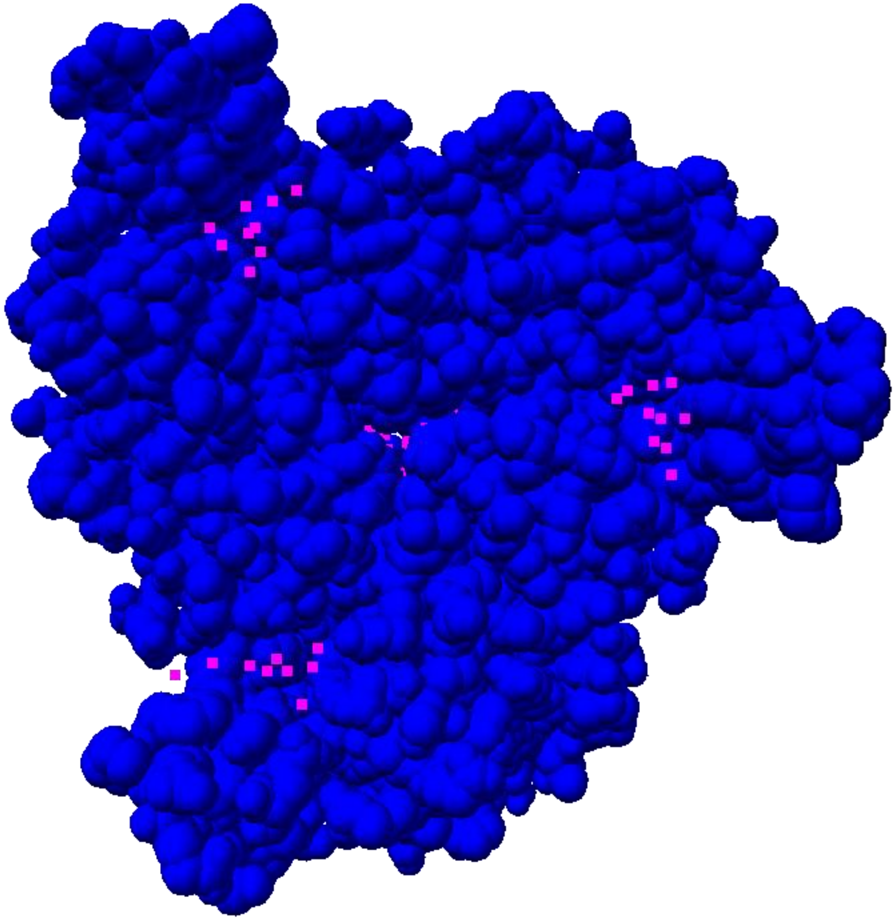


Ligand Binding Probability Map

Methods

2) Sample most probable site locations

- 128 samples

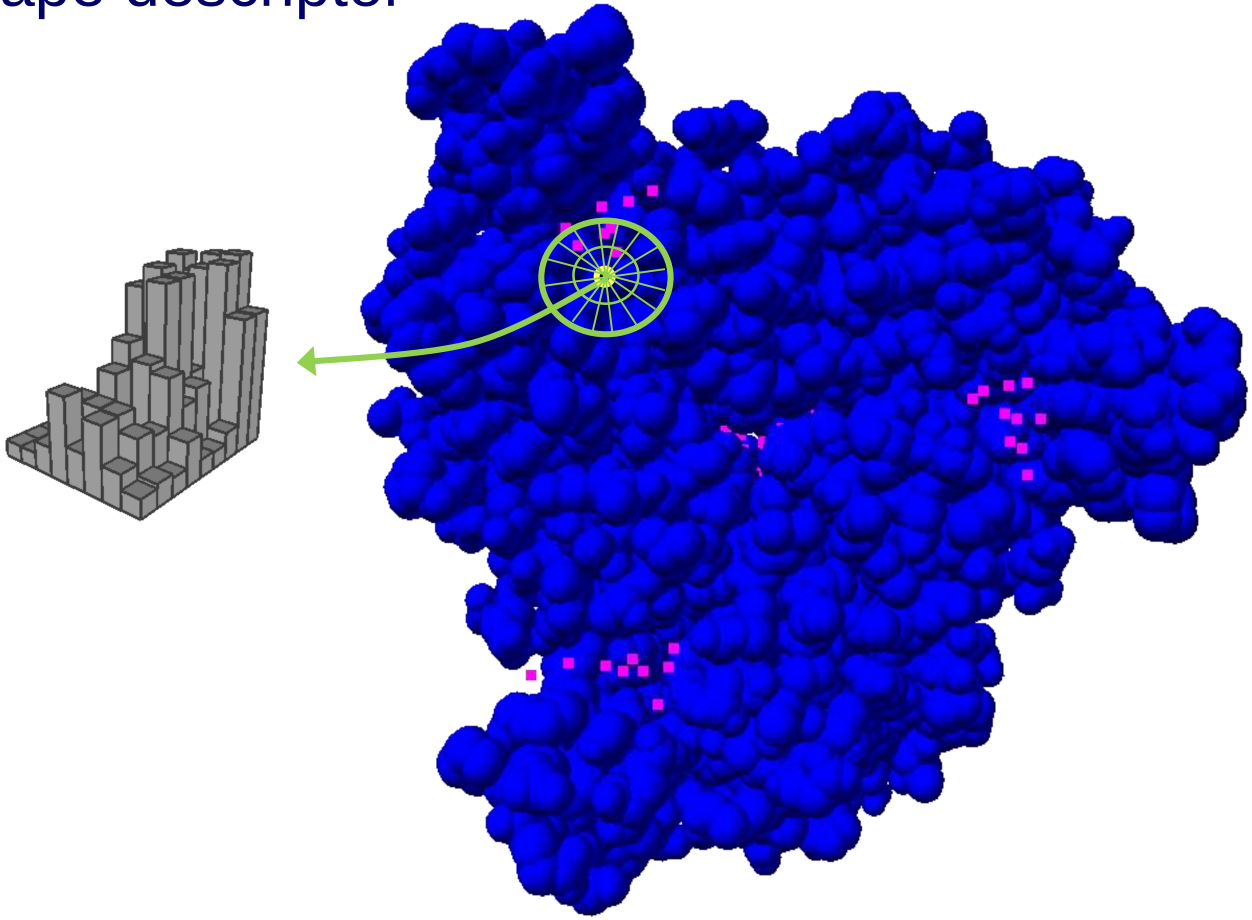


Sampled Ligand Binding Site Locations

Methods

3) Build shape descriptor for every sample location

- Harmonic shape descriptor

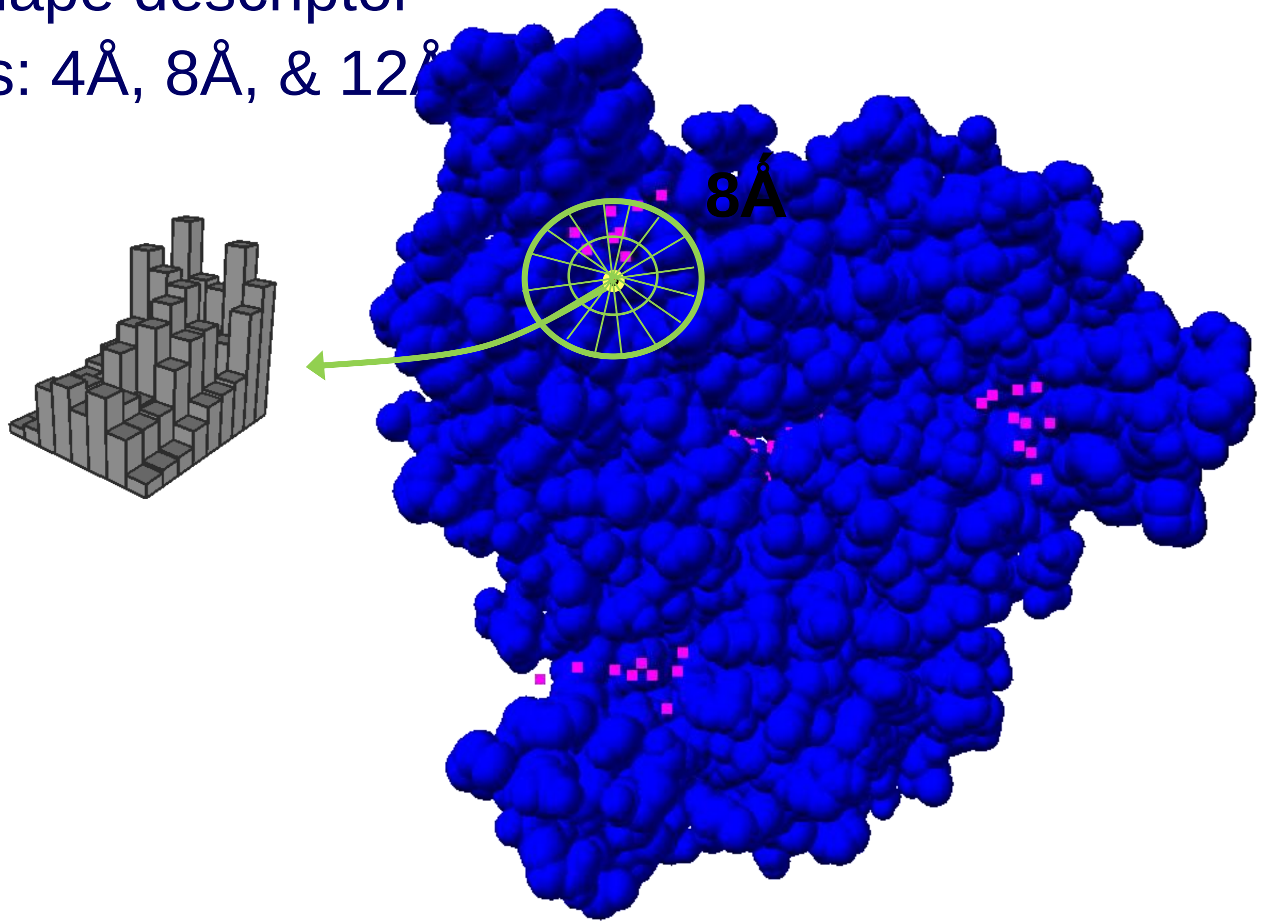


Spherical Harmonic Shape Descriptor

Methods

3) Build shape descriptor for every sample location

- Harmonic shape descriptor
- Three scales: 4Å, 8Å, & 12Å

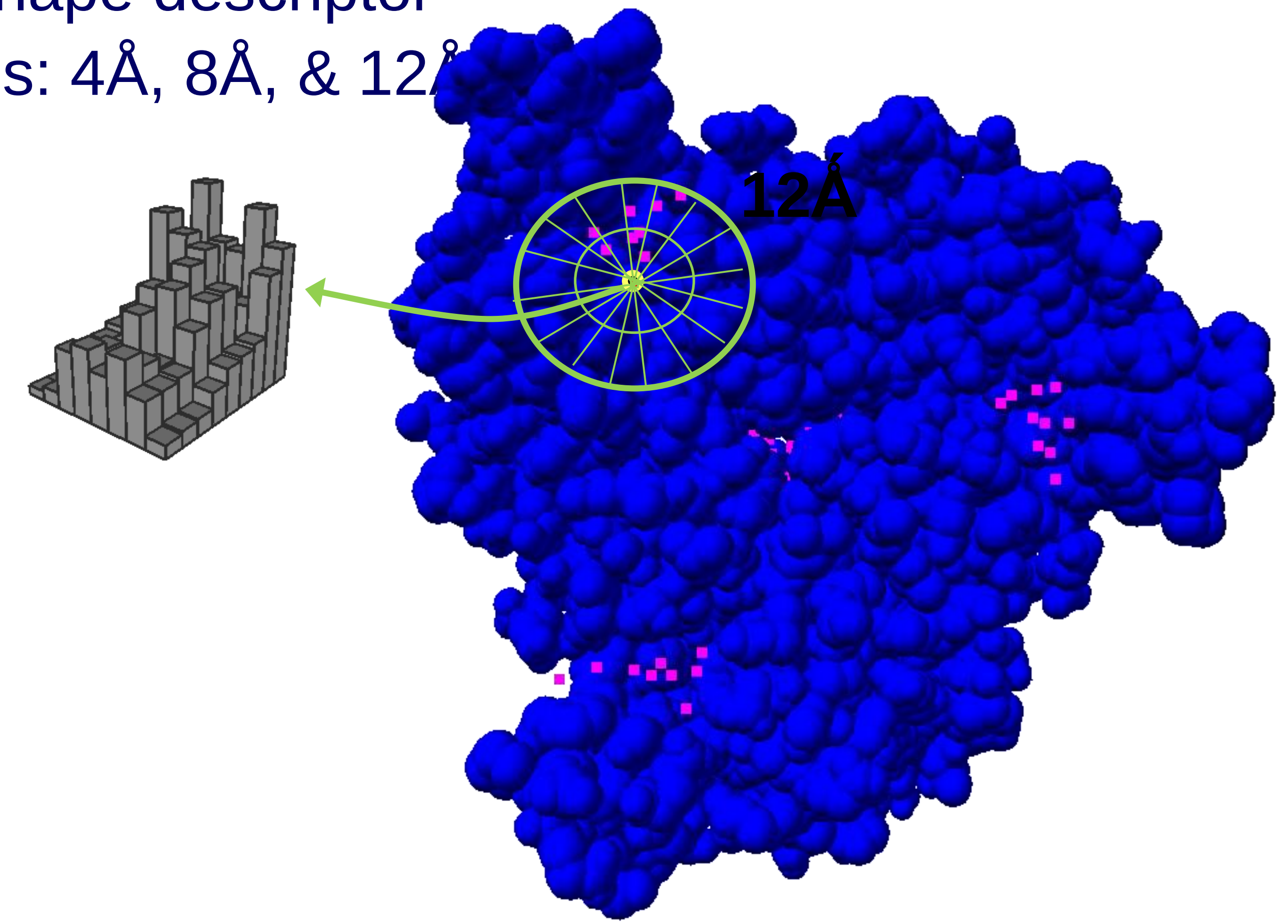


Spherical Harmonic Shape Descriptor

Methods

3) Build shape descriptor for every sample location

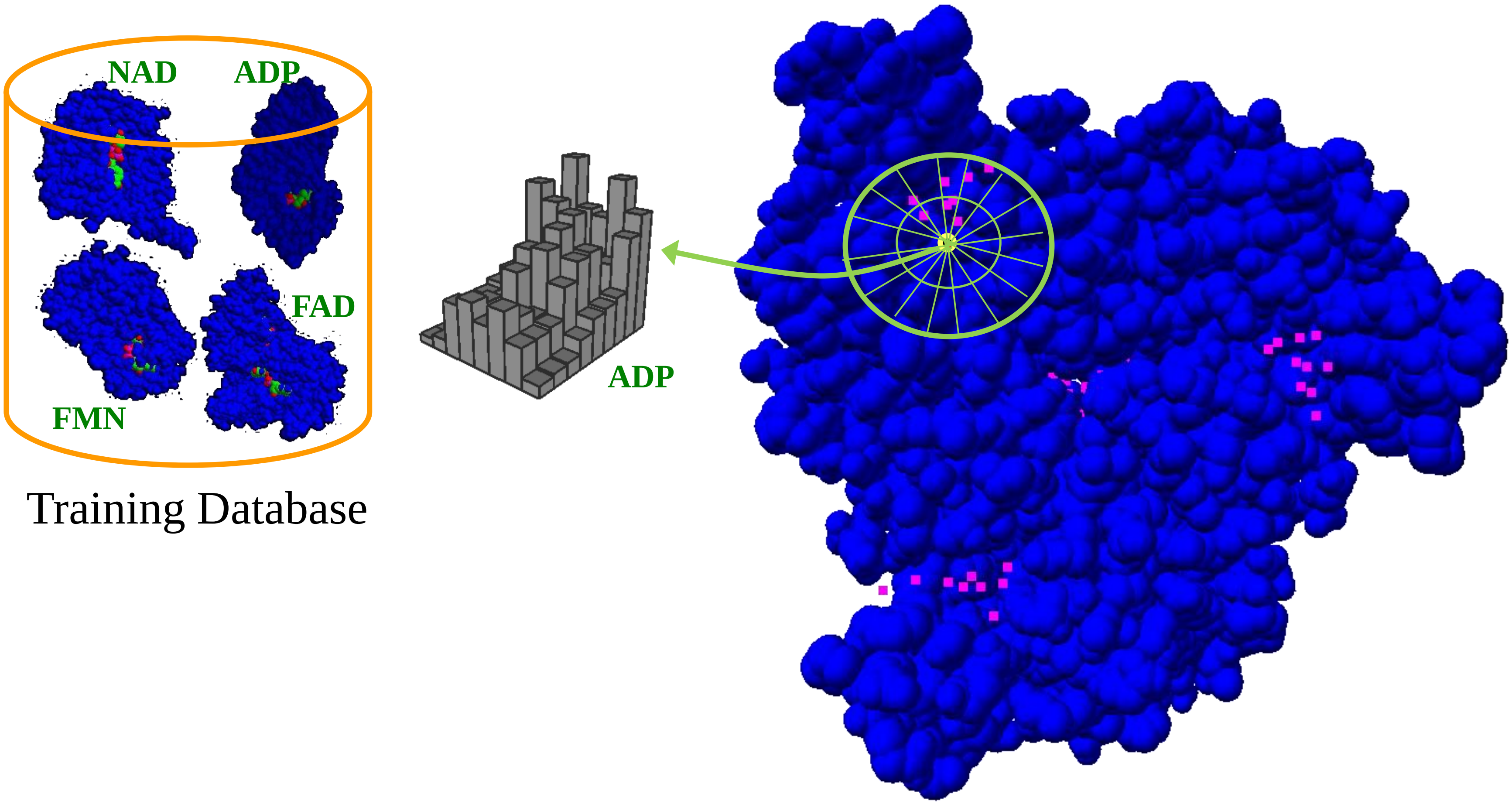
- Harmonic shape descriptor
- Three scales: 4Å, 8Å, & 12Å



Spherical Harmonic Shape Descriptor

Methods

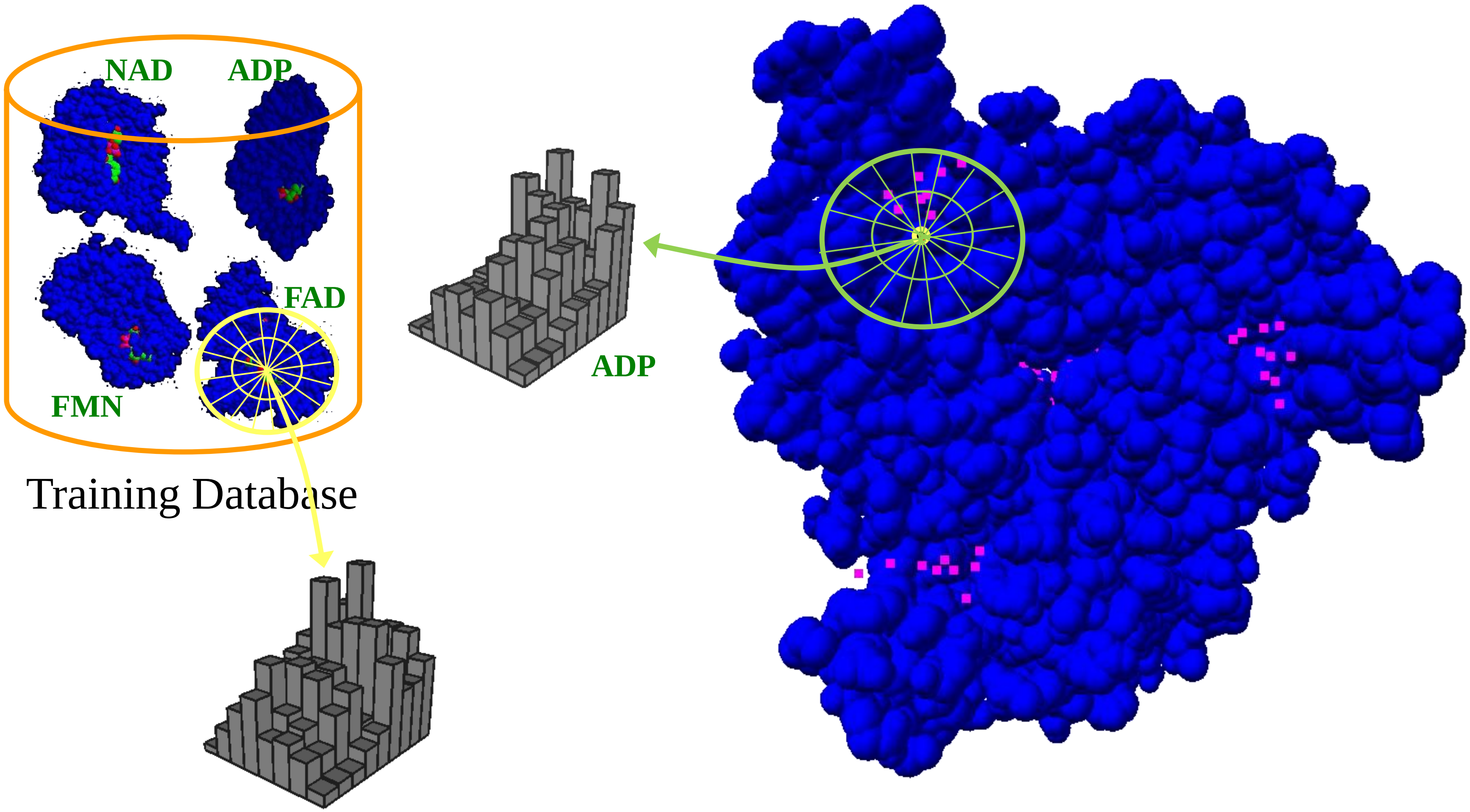
4) Compute "distinction" of every shape descriptor



Training Database

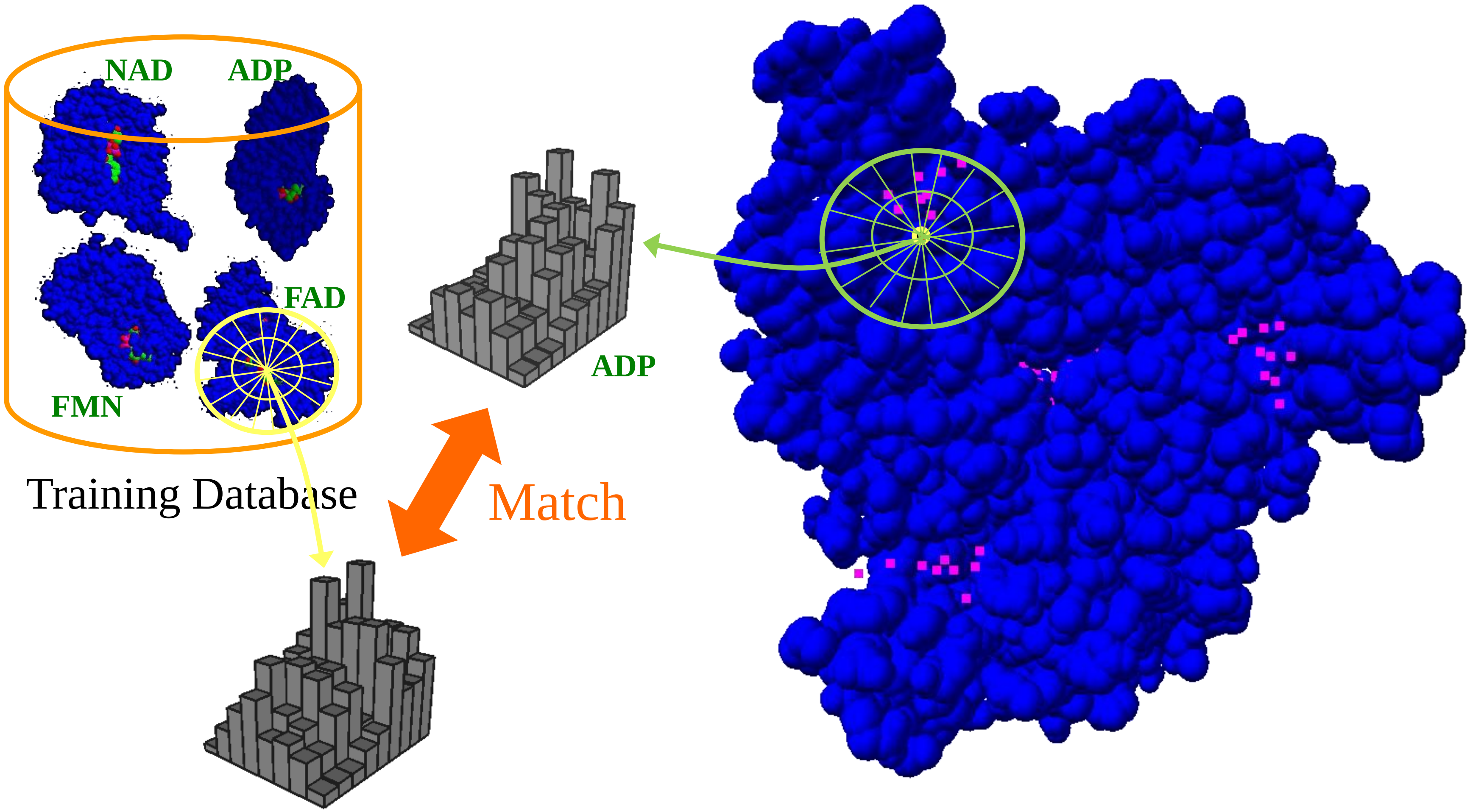
Methods

4) Compute “distinction” of every shape descriptor



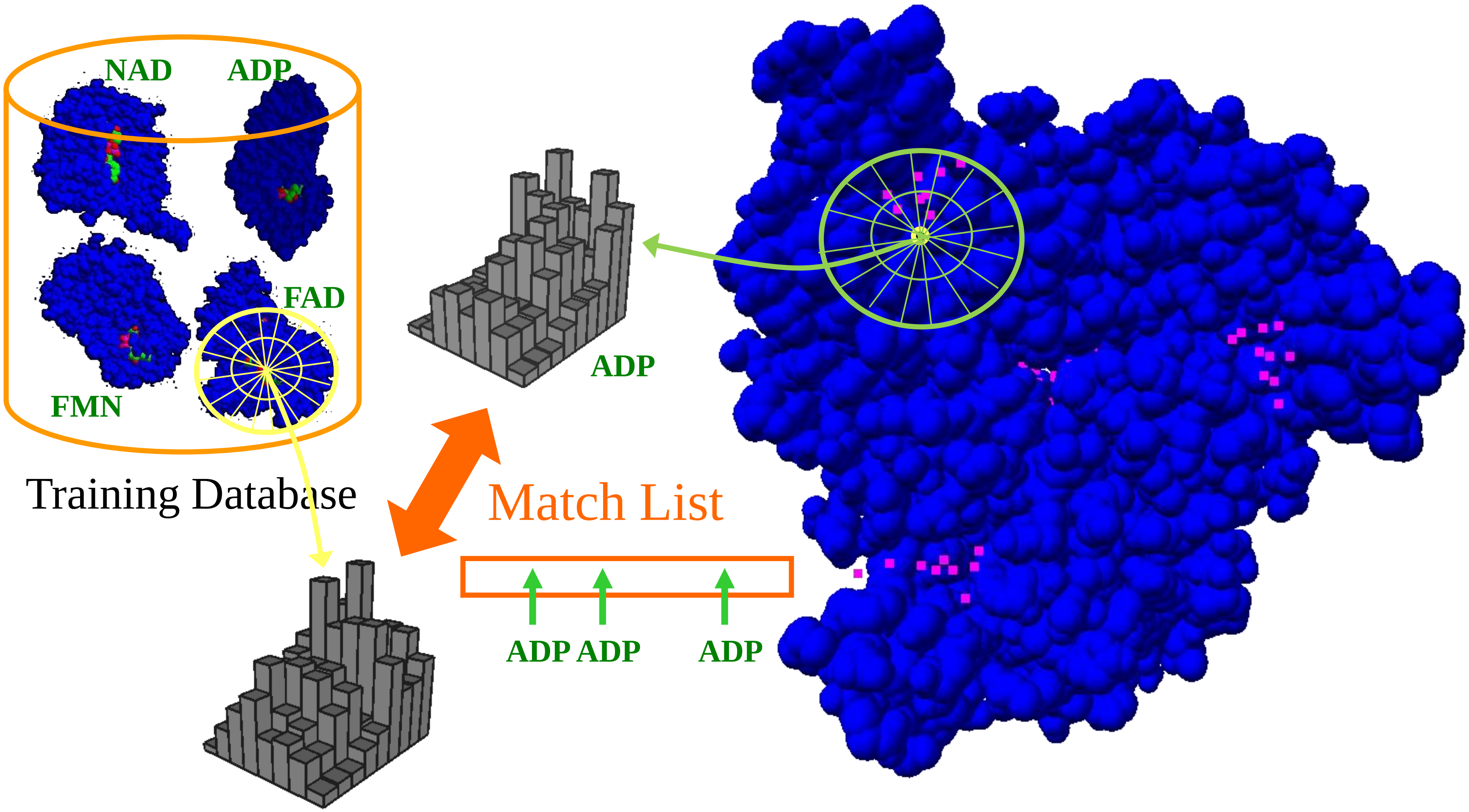
Methods

4) Compute “distinction” of every shape descriptor



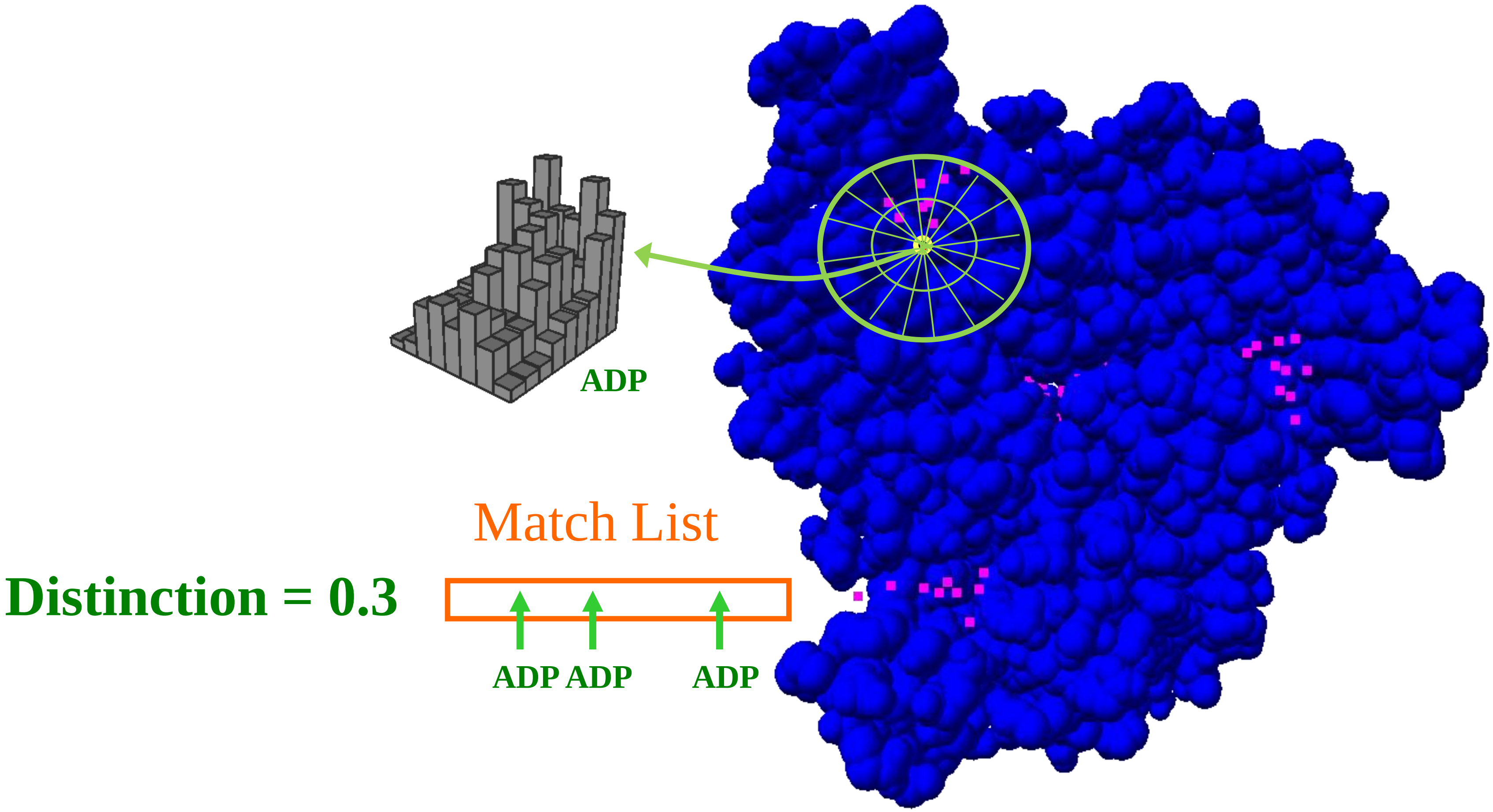
Methods

4) Compute “distinction” of every shape descriptor



Methods

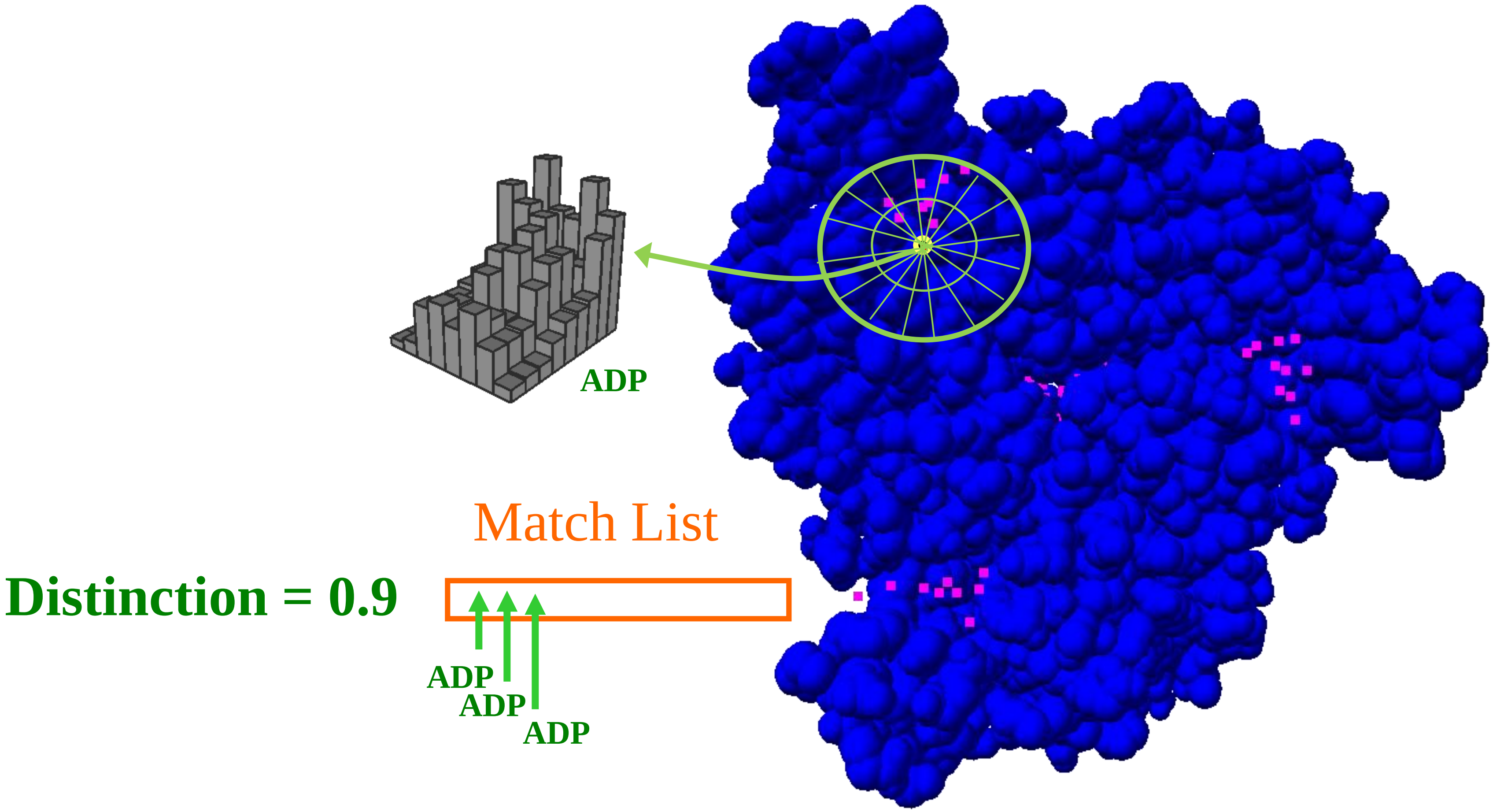
4) Compute "distinction" of every shape descriptor



Distinction is measured by information retrieval metric

Methods

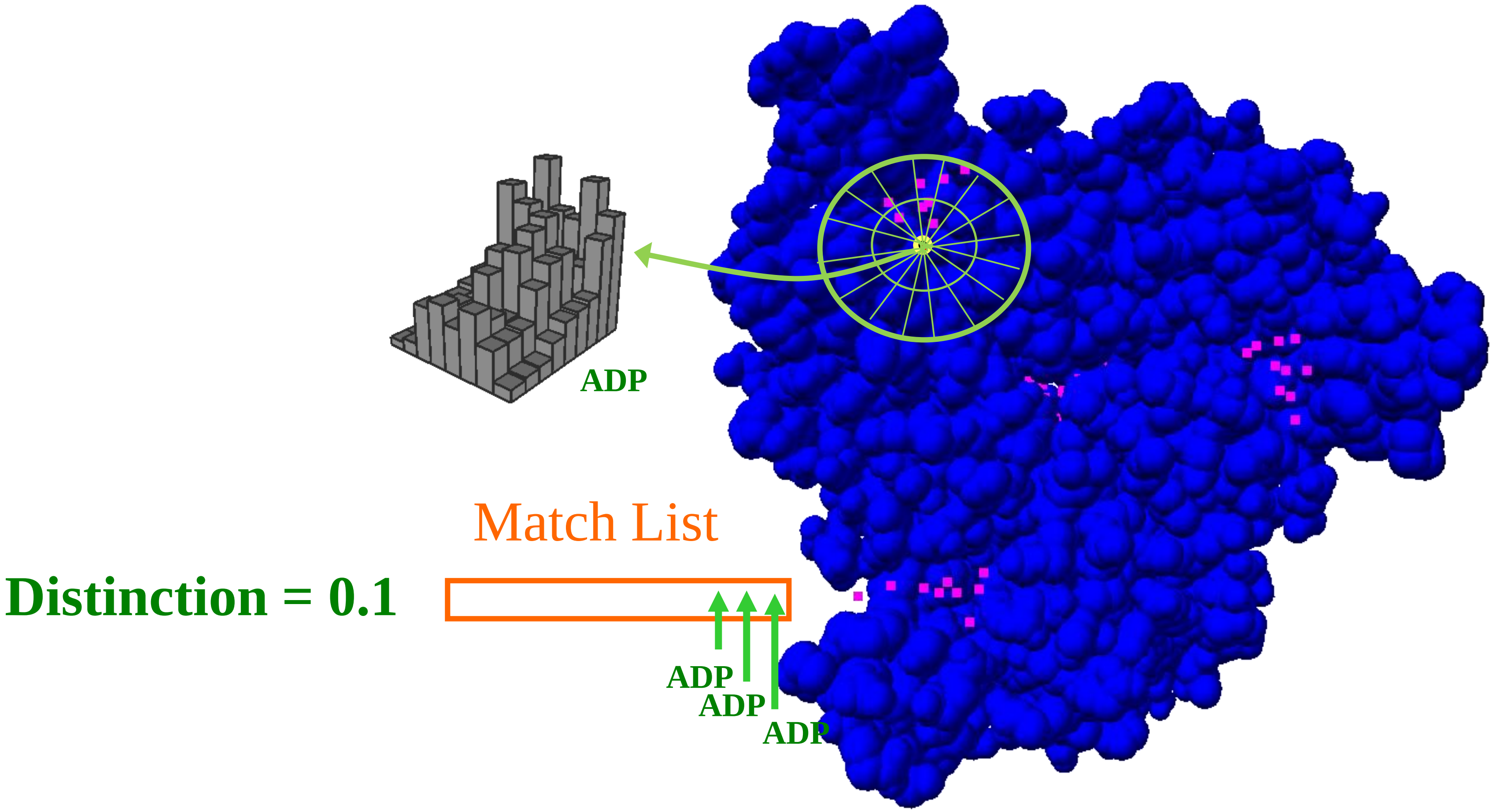
4) Compute "distinction" of every shape descriptor



Distinction is measured by information retrieval metric

Methods

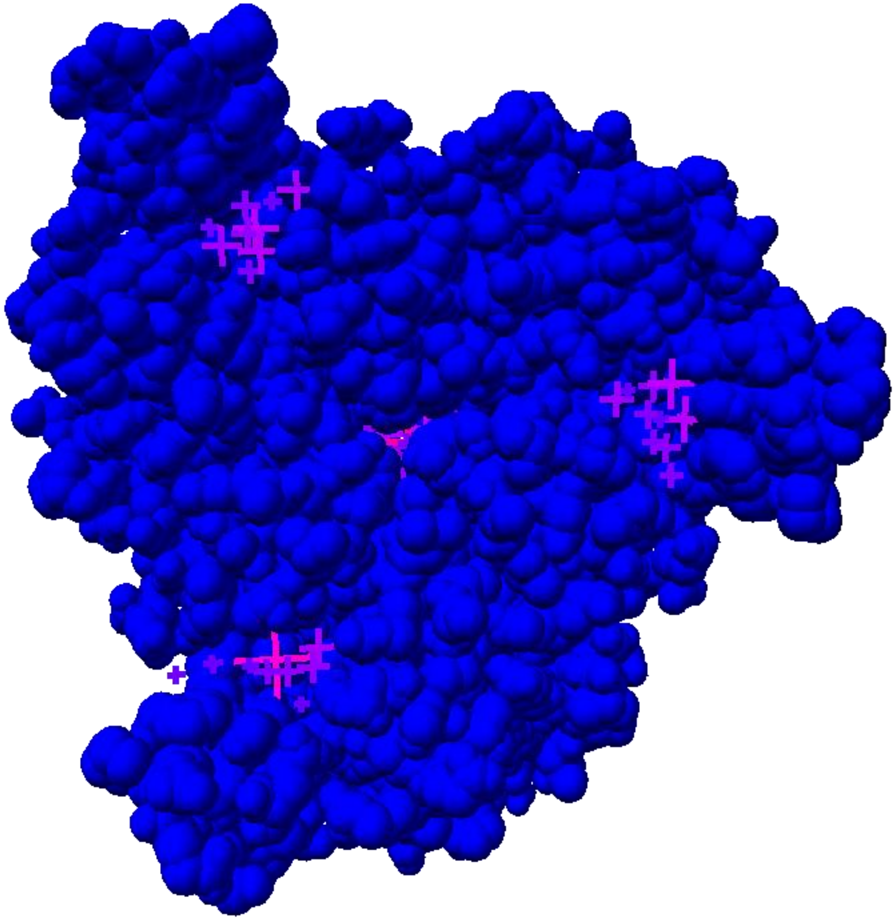
4) Compute "distinction" of every shape descriptor



Distinction is measured by information retrieval metric

Methods

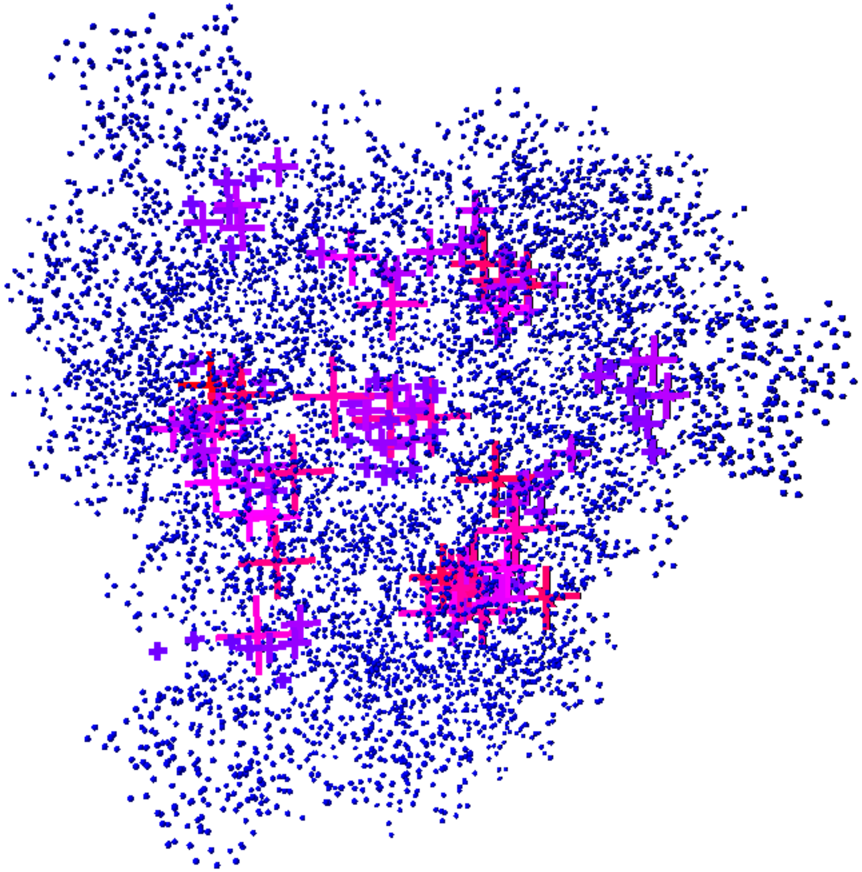
4) Compute "distinction" of every shape descriptor



Points shown red/big are most distinctive

Methods

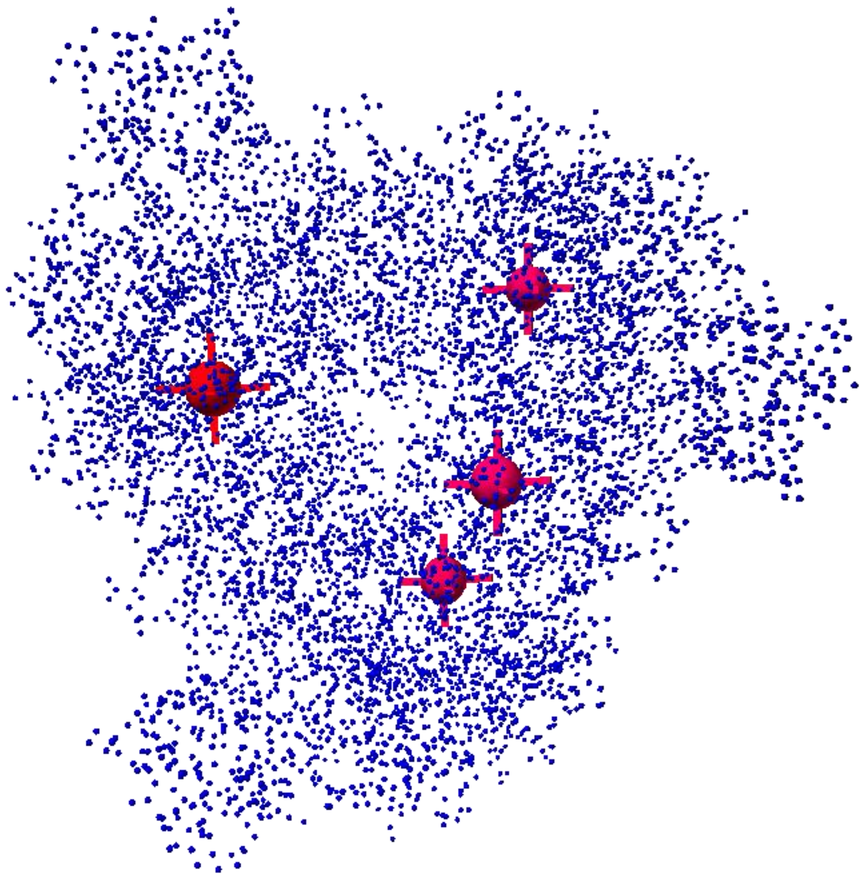
4) Compute “distinction” of every shape descriptor



Points shown red/big are most distinctive

Methods

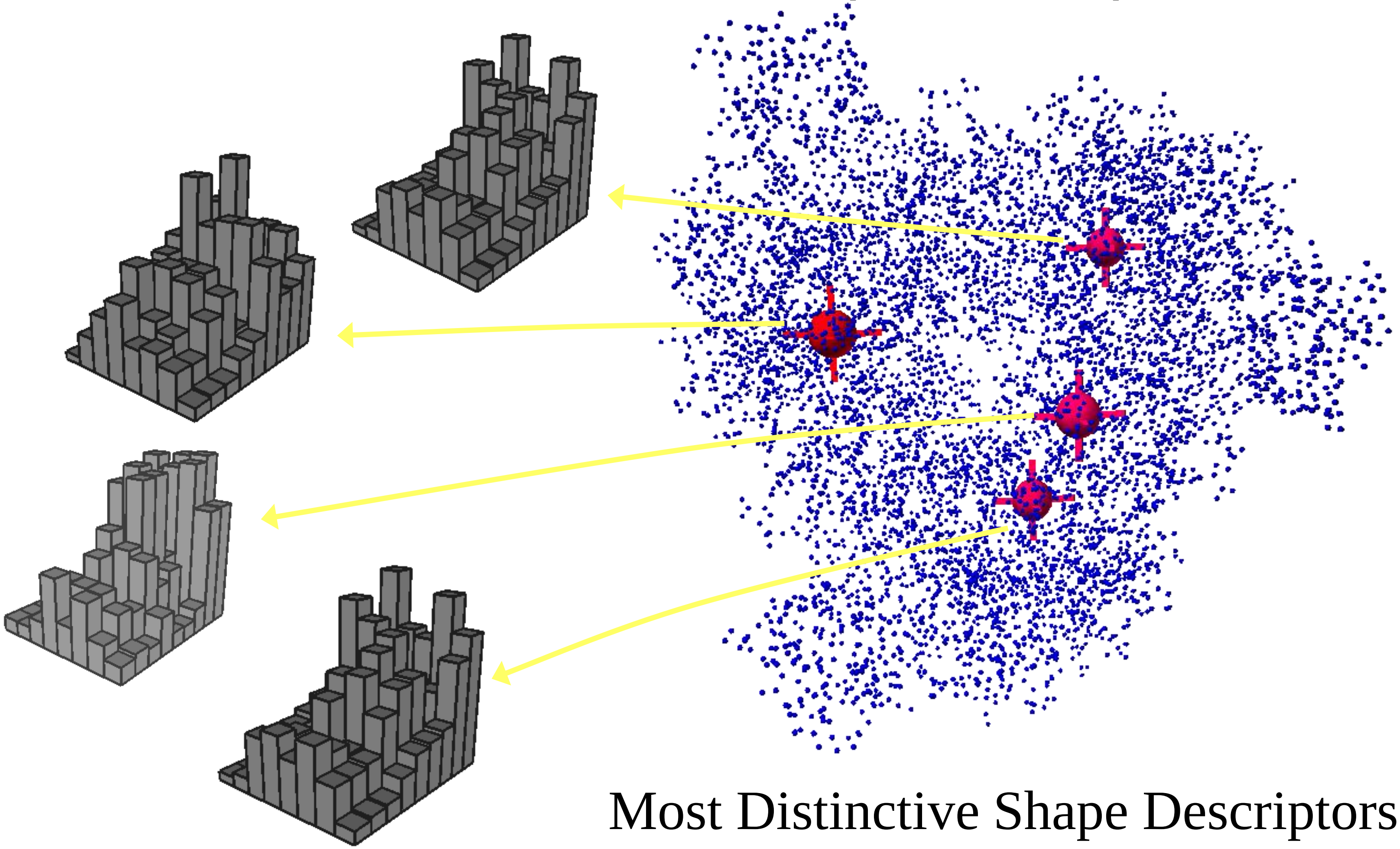
4a) Select only the most distinctive shape descriptors



Most Distinctive Shape Descriptors

Methods

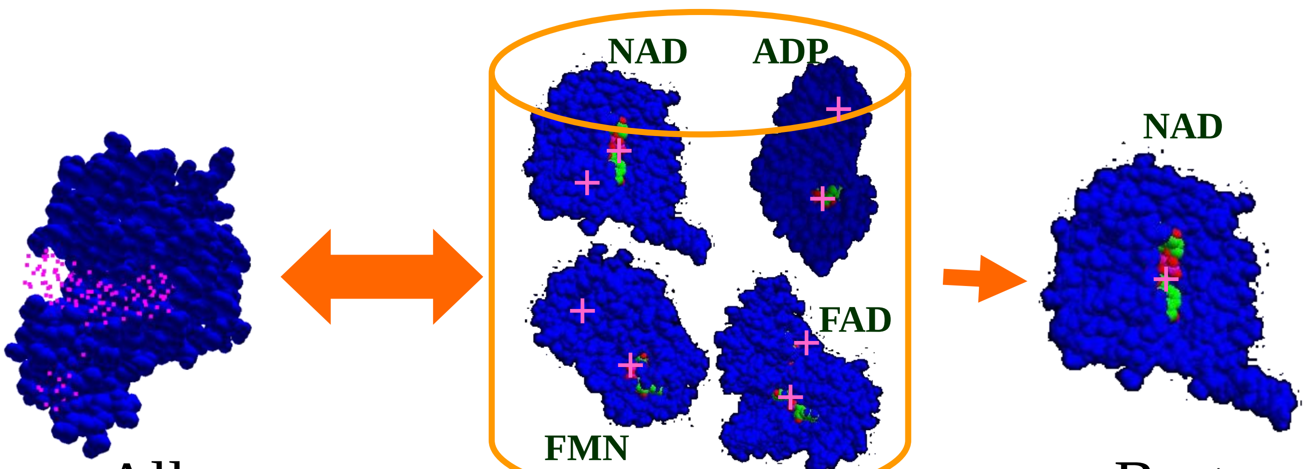
4b) Represent target proteins (in known classes) with small set of distinctive shape descriptors



Most Distinctive Shape Descriptors

Methods

5) Match all shape descriptors of query protein to most distinctive descriptors of labeled proteins



All Shape Descriptors of Query Protein

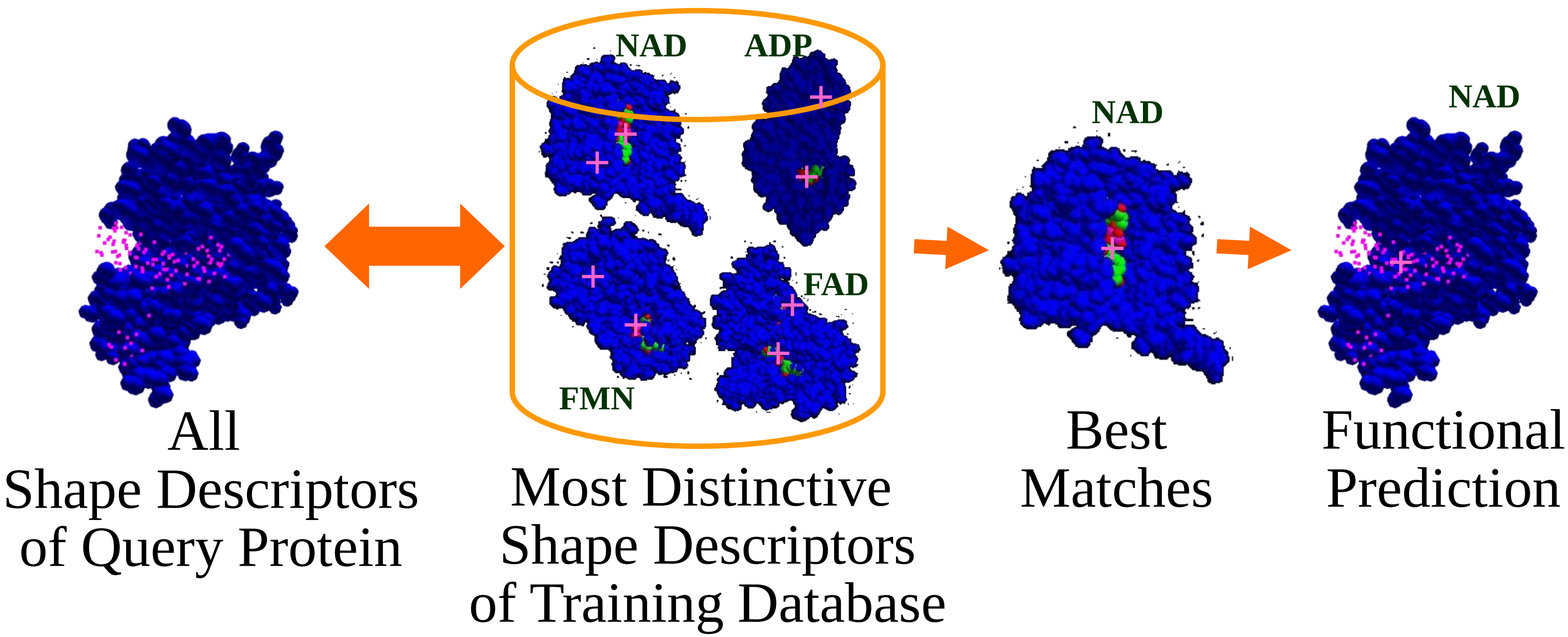
Most Distinctive Shape Descriptors of Training Database

Best Matches

Methods

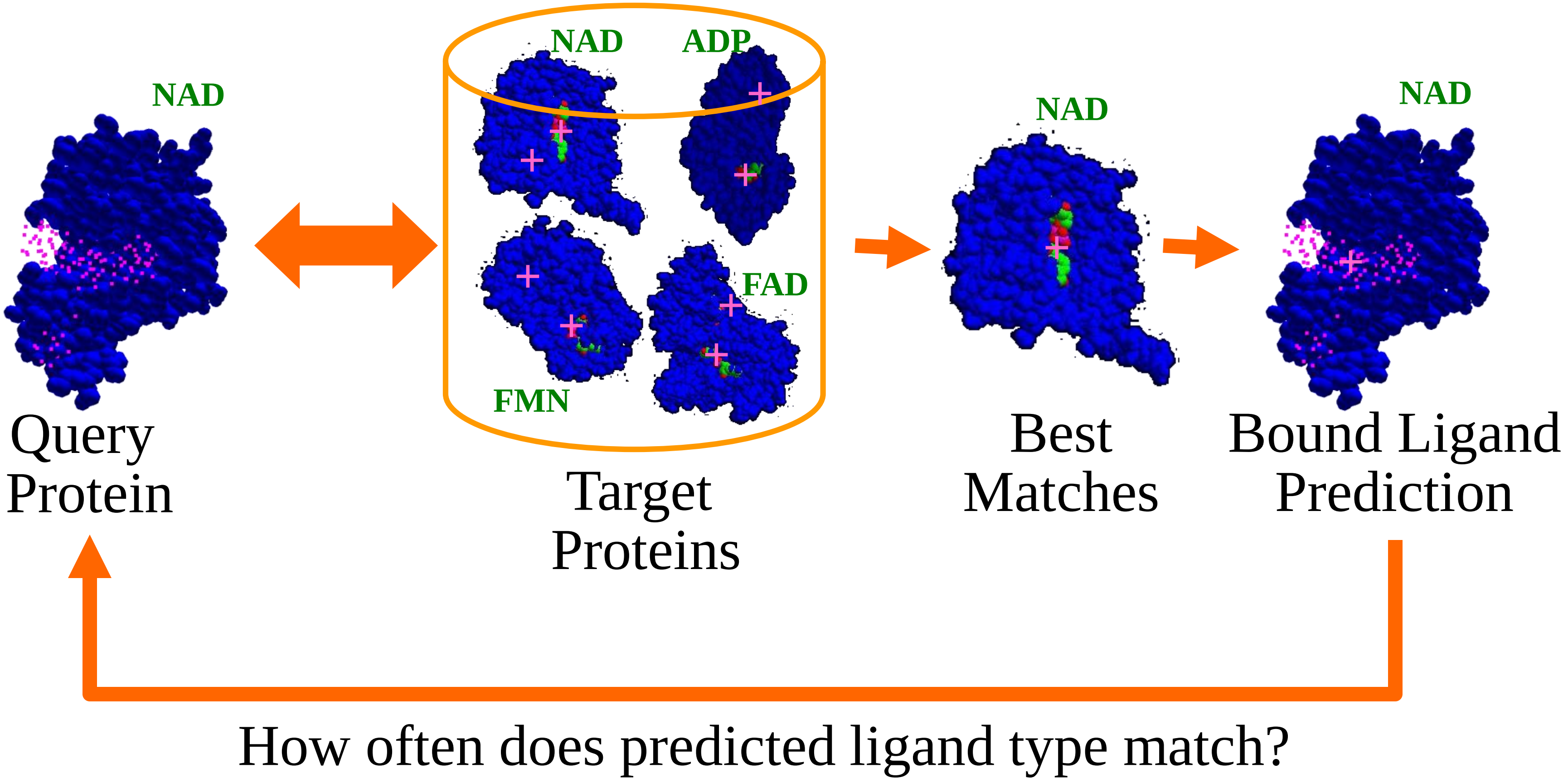
6) Make functional prediction based on best matches

- Nearest neighbor classifier



Experiment Design

Leave-one-out classification experiments



Experimental Data Sets

Data Set 1: [Kahraman07]

- 100 nonhomologous proteins with bound ligands in PDB
- 9 ligand types
  - AMP, ATP, FAD, FMN, GLC, HEM, NAD, PO4, Steroid

Data Set 2:

- 157 nonhomologous proteins with both bound and unbound structures in PDB
- 17 ligand types, 94 confusers (other ligand types)
  - 5GP, 8HG, A3P, ADP, AMP, ANP, ATP, C8E, GDP, HEM, MAL, NAD, OLA, RBF, SAM, SIA

Experimental Results – Data Set 1

Evaluation of classification performance:

Algorithm	Classification Rate	Preprocessing Time	Query Time
Proposed Method	68%	150 sec	0.001 sec
12Å descriptor centered on bound ligand	46%	1 sec	0.001 sec
FASTA	19%	-	2 sec
Random	12%	-	-

Experimental Results – Data Set 1

Effect of selecting distinctive sites:

Algorithm	Classification Rate	Preprocessing Time	Query Time
Proposed Method	68%	150 sec	0.001 sec
Without Distinctive Site Selection	13%	150 sec	0.1 sec

Experimental Results – Data Set 2

Data Set 2 (157 bound and unbound structures):

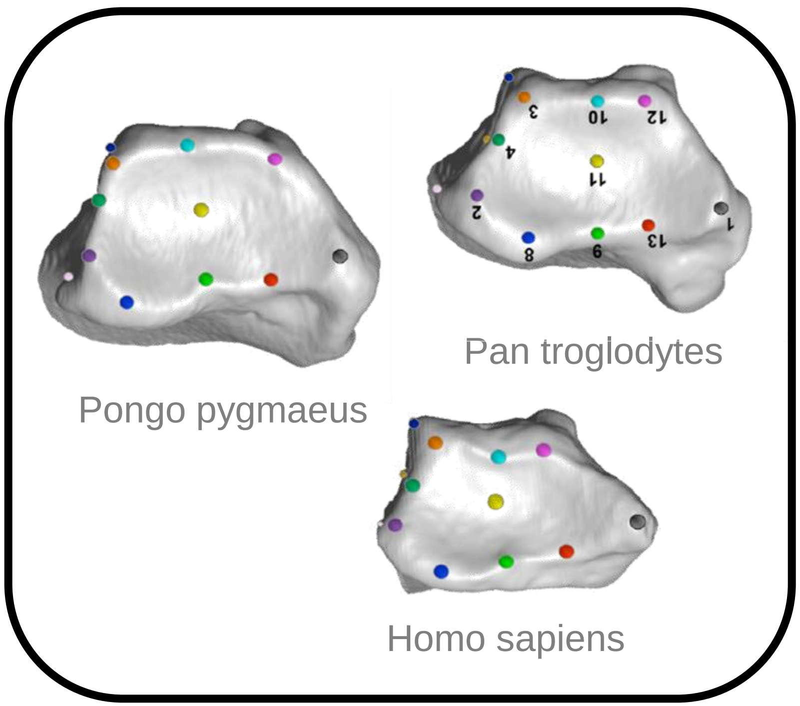
Algorithm	Classification Rate	
Distinctive Site Matching	63%	61%
FASTA	3%	3%
Random	3%	3%

	Bound Structures	Unbound Structures

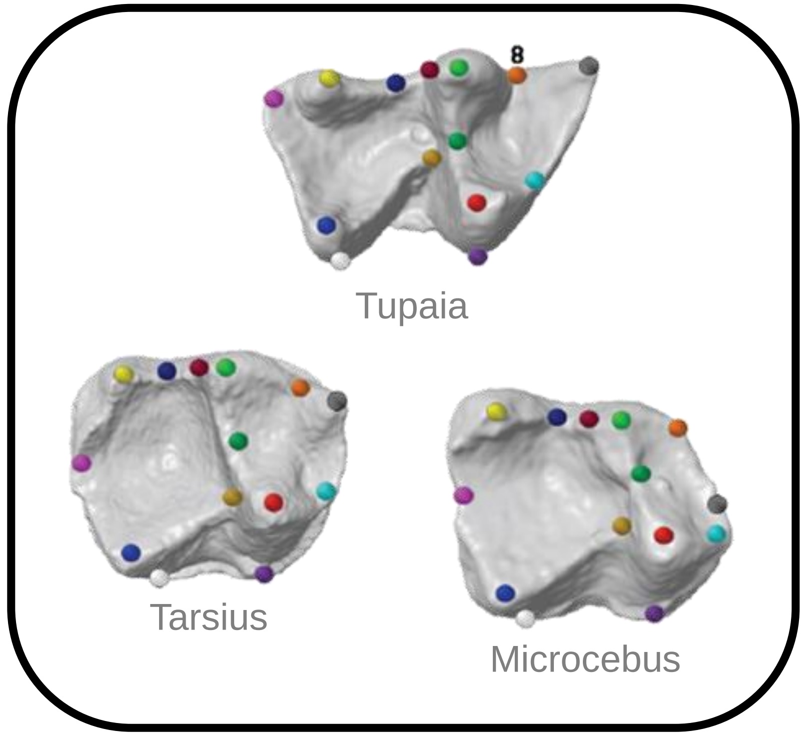
Paleontology:
Matching Fossil Surfaces

Goal

Automatically quantify the geometric similarity of anatomical surfaces



Distal Radius



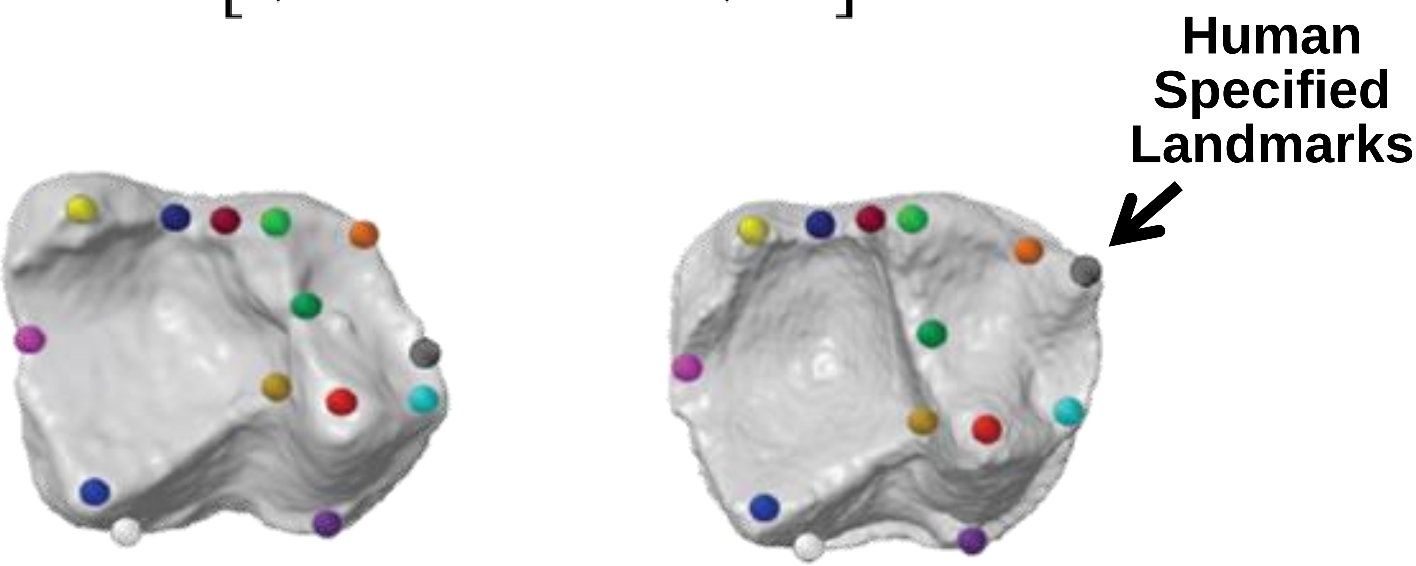
Mandibular Molar

[Boyer, Lipman, St. Clair, Puente, Patel, Funkhouser, Jernvall, and Daubechies, 2011]

Previous Work

Traditional Procrustes distance:

$$d(X,Y) = min_R\left[\left(\sum_{i=1}^{N}\|R(X_i) - Y_i\|^2\right)^{1/2}\right]$$

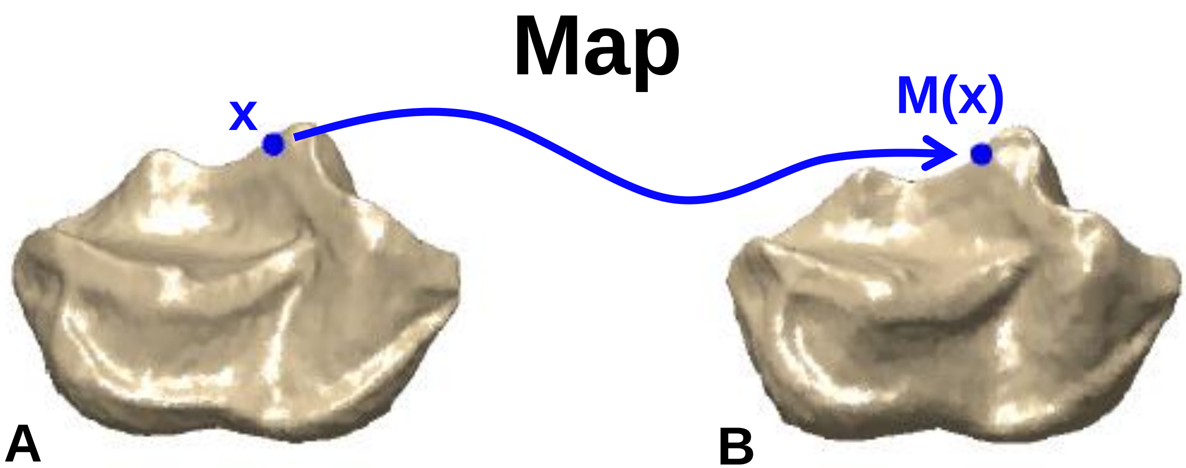


$X = \{ X_i \}$ $Y = \{ Y_i \}$

Target Approach

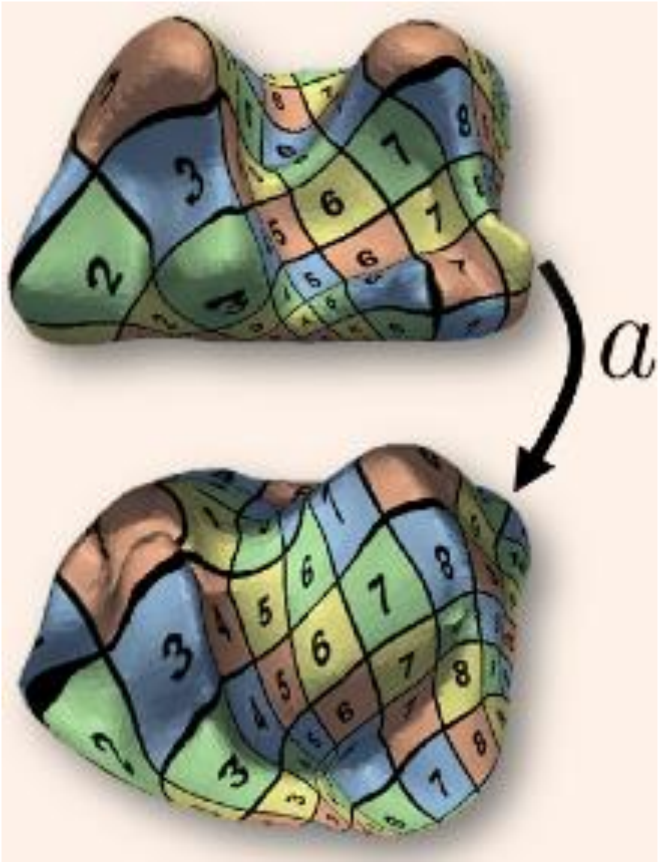
New continuous Procrustes distance:

$$d(A,B) = min_{R,M}\left[\left(\int_A \|R(x) - M(x)\|^2\, dx\right)^{1/2}\right]$$



Surface Matching

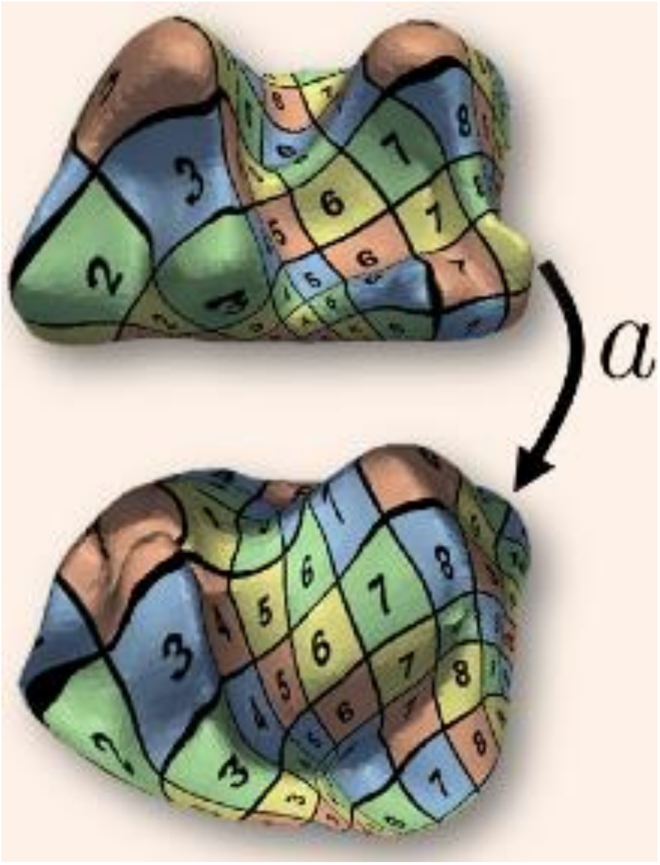
Goal: find map between surfaces



Surface Matching

Goal: find map between surfaces

- Non-rigid
- Bijective
- Smooth
- Shape preserving
- Automatic
- Efficient computation
- Provide metric
- Semantic alignment



Applications

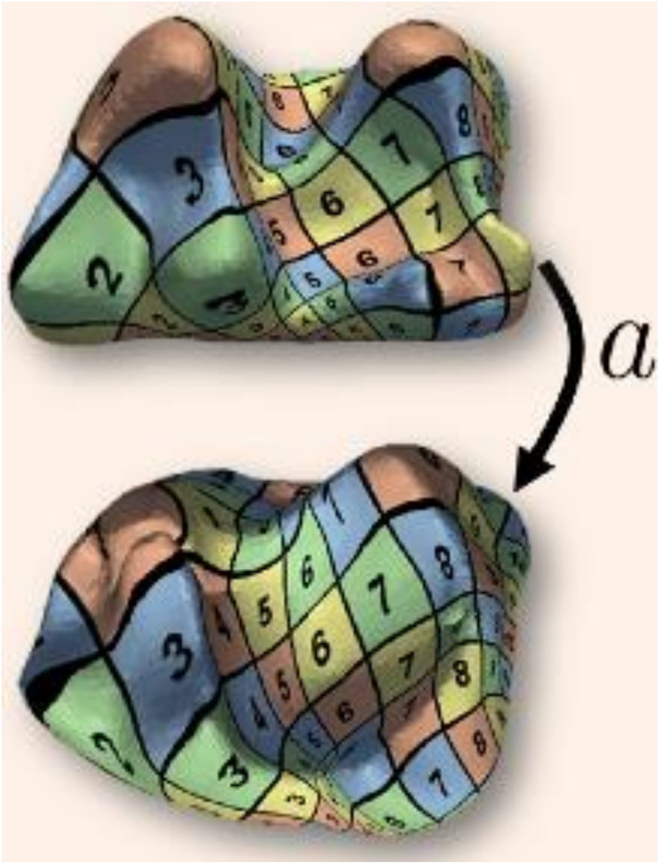
Registration

Comparison

Property transfer

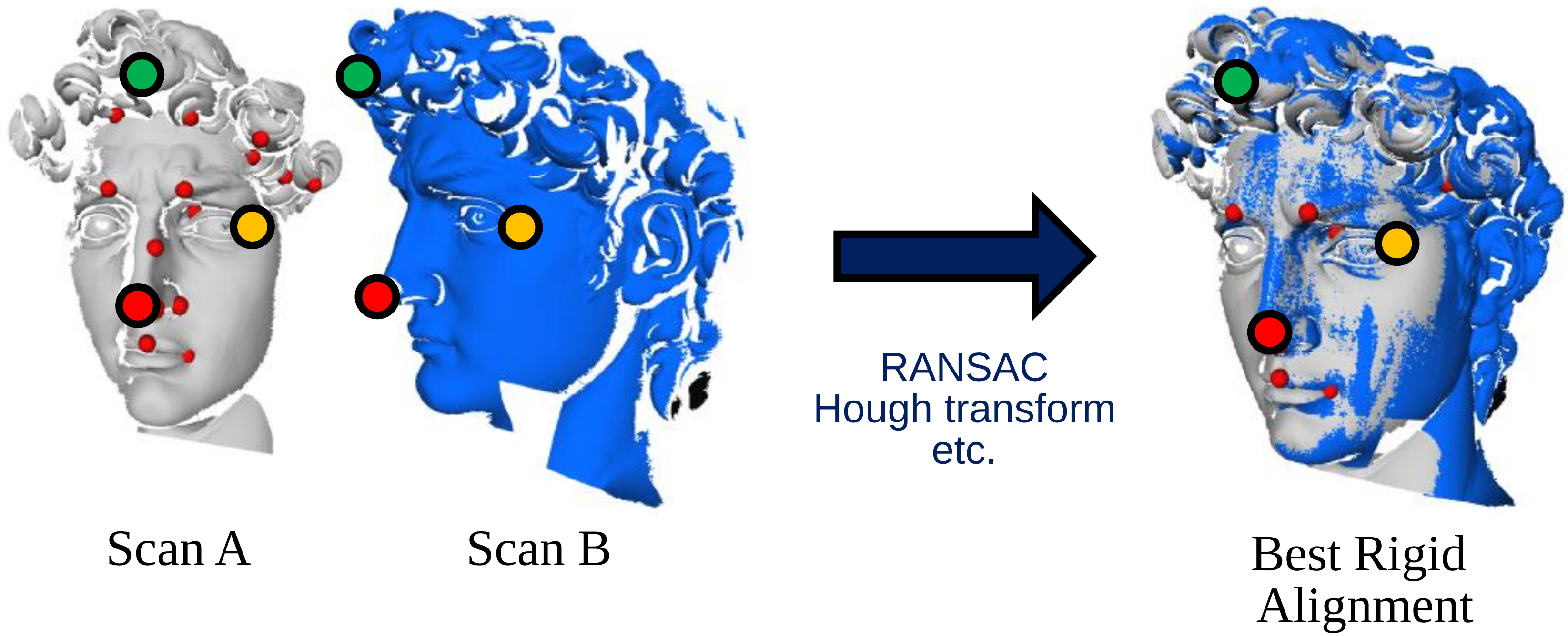
Morphing

etc.



Possible Approach

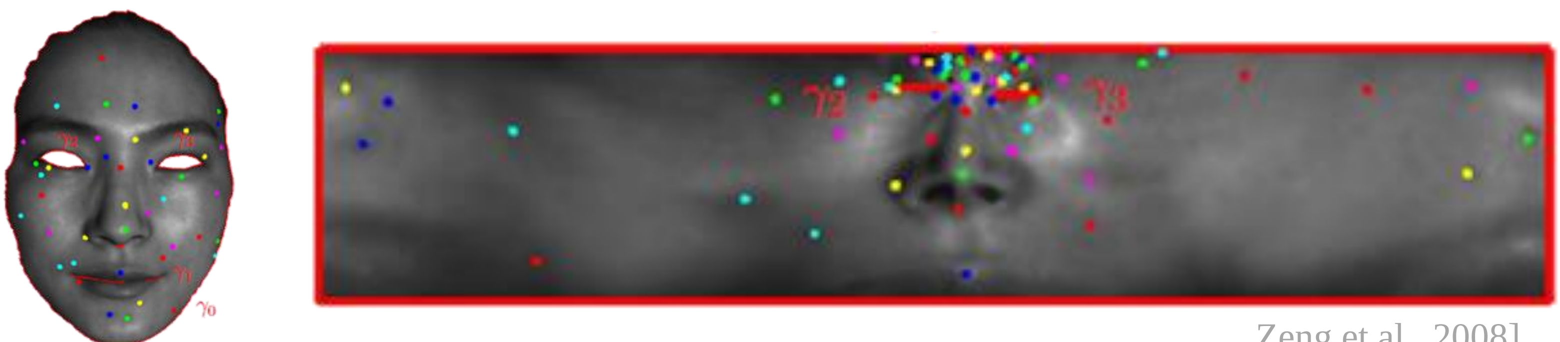
Find feature correspondences and solve for map that best aligns them



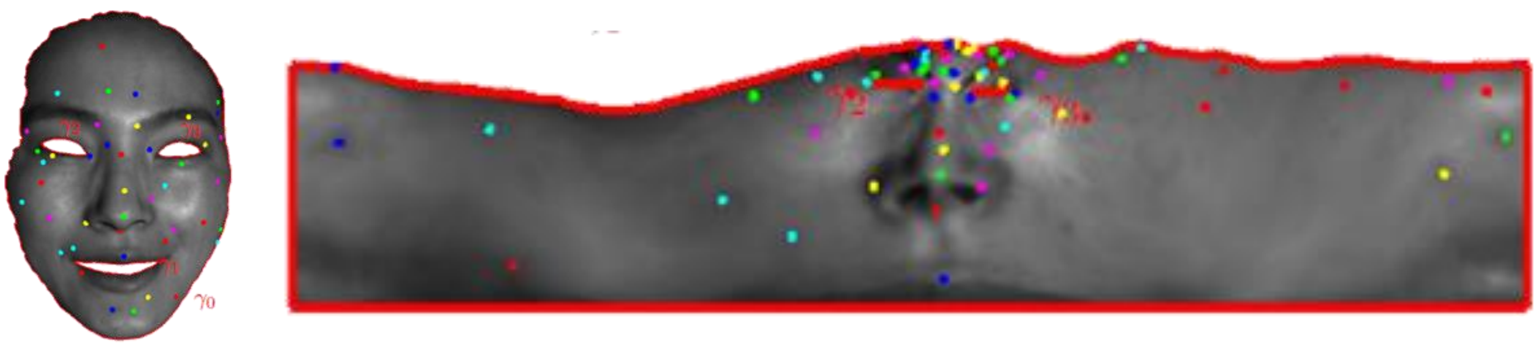
Suitable only for “low-dimensional” maps

Challenge

Many feature points are needed for most maps between surfaces



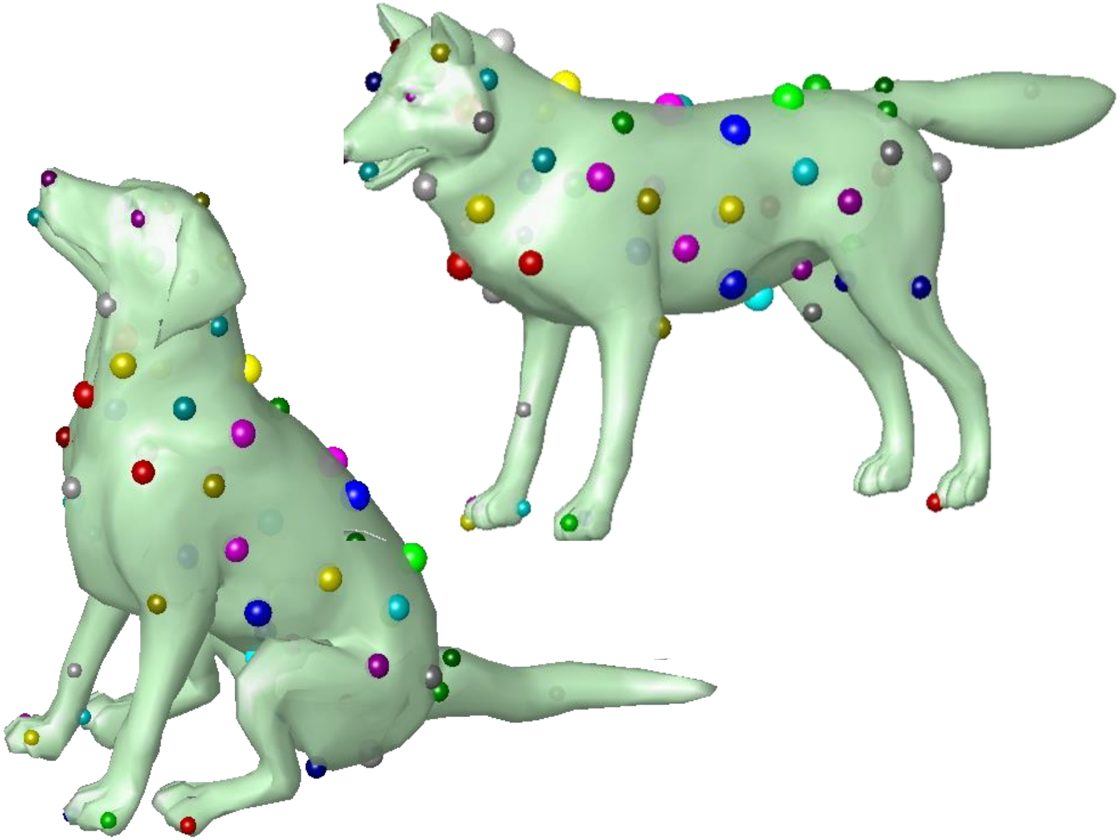
Zeng et al., 2008]



Least Squares Conformal Map
(preserve angles as best as possible)

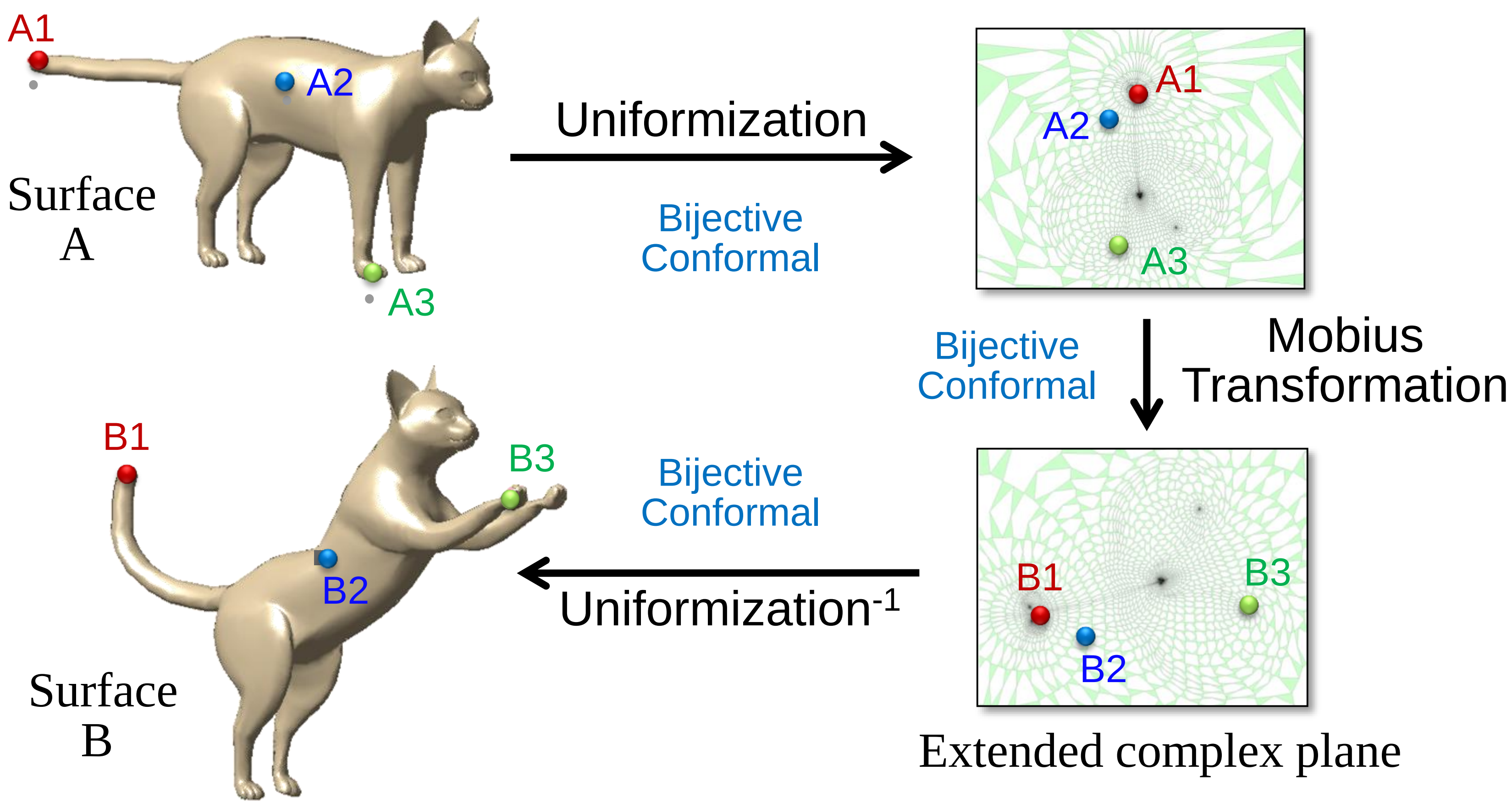
Problem

Automatically finding many correspondences is difficult for surfaces



Key Observation

Any three point correspondences define a bijective, conformal map between genus zero surfaces



Key Observation

We can search for the “lowest distortion” bijective, conformal map between genus zero surfaces using algorithms that sample triplets of correspondences (e.g., RANSAC, Hough transform, etc.)

Polynomial-time algorithm
for non-rigid surface mapping

Surface Mapping Algorithm

Example: RANSAC algorithm

- For i = 1 to ~N^3
  - Sample three points (A1,A2,A3) on surface A
  - Sample three points (B1,B2,B3) on surface B
  - Compute conformal map M: (A1,A2,A3)→(B1,B2,B3)
  - Remember M if distortion is smallest

Surface Mapping Algorithm

Example: RANSAC algorithm

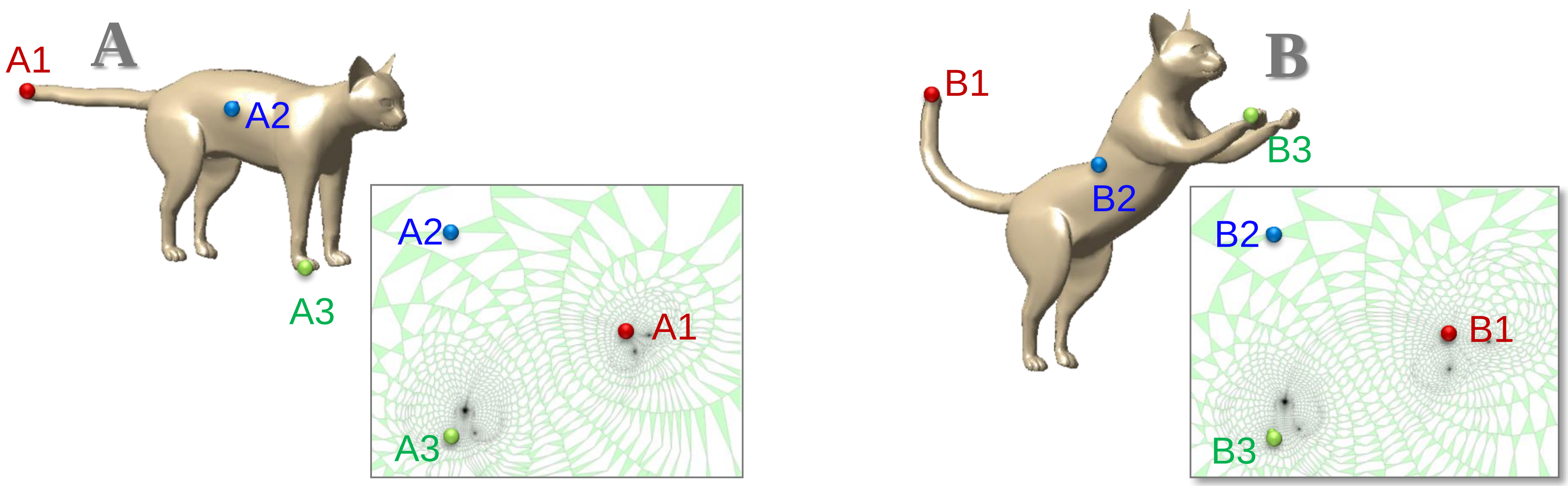
For i = 1 to ~N^3

Sample three points (A1,A2,A3) on surface A

Sample three points (B1,B2,B3) on surface B

Compute conformal map M: (A1,A2,A3)→(B1,B2,B3)

Remember M if distortion is smallest



Measure distortion by relative change of area
(deviation from isometry)

Surface Mapping Algorithm

Example: RANSAC algorithm

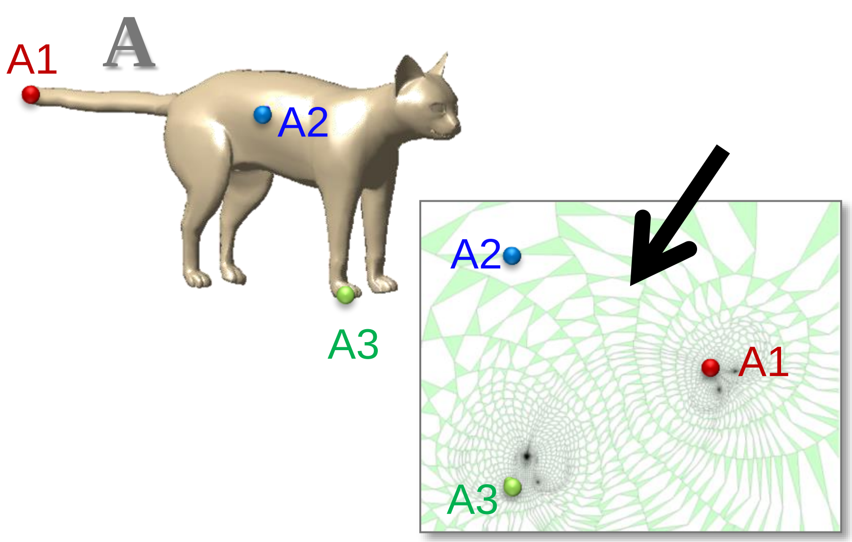
For i = 1 to ~N^3

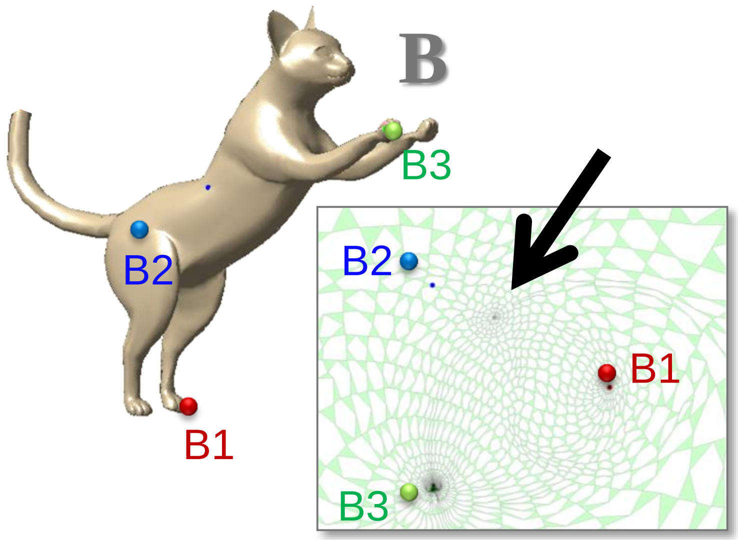
Sample three points (A1,A2,A3) on surface A

Sample three points (B1,B2,B3) on surface B

Compute conformal map M: (A1,A2,A3)→(B1,B2,B3)

Remember M if distortion is smallest





Measure distortion by relative change of area
(deviation from isometry)

Surface Mapping Algorithm

RANSAC algorithm properties:

- Non-rigid
- Bijective
- Smooth
- Shape preserving
- Automatic
- Efficient computation
- Provides metric
- Semantic alignment?

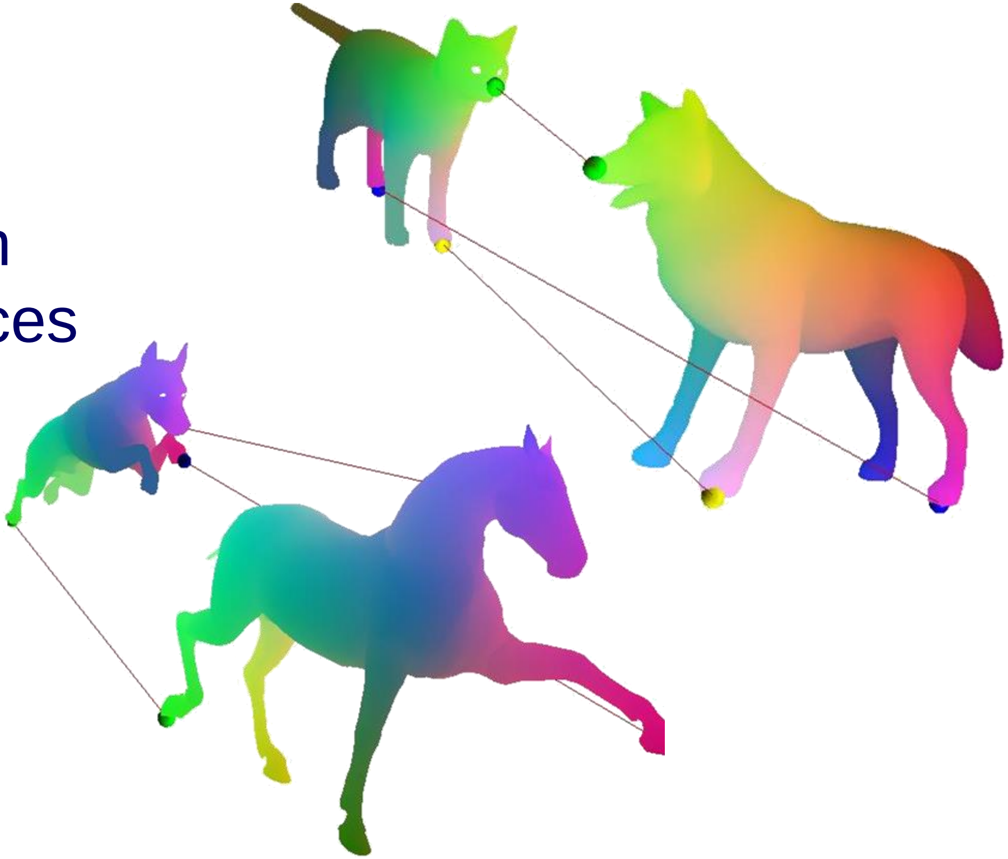
Experimental Results

Data:

- 51 pairs of meshes representing animals from TOSCA and SHREC Watertight data sets

Methodology:

- Predict surface maps
- Compare to ground truth semantic correspondences

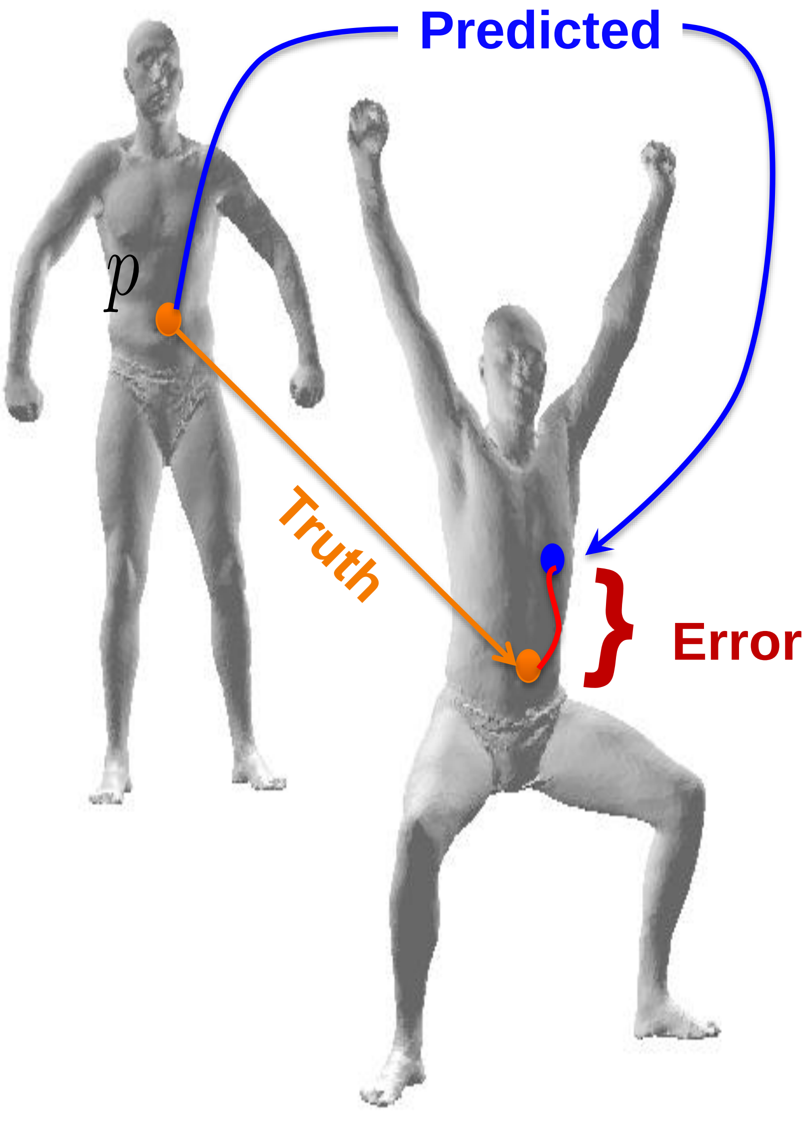


Experimental Results

Evaluation:

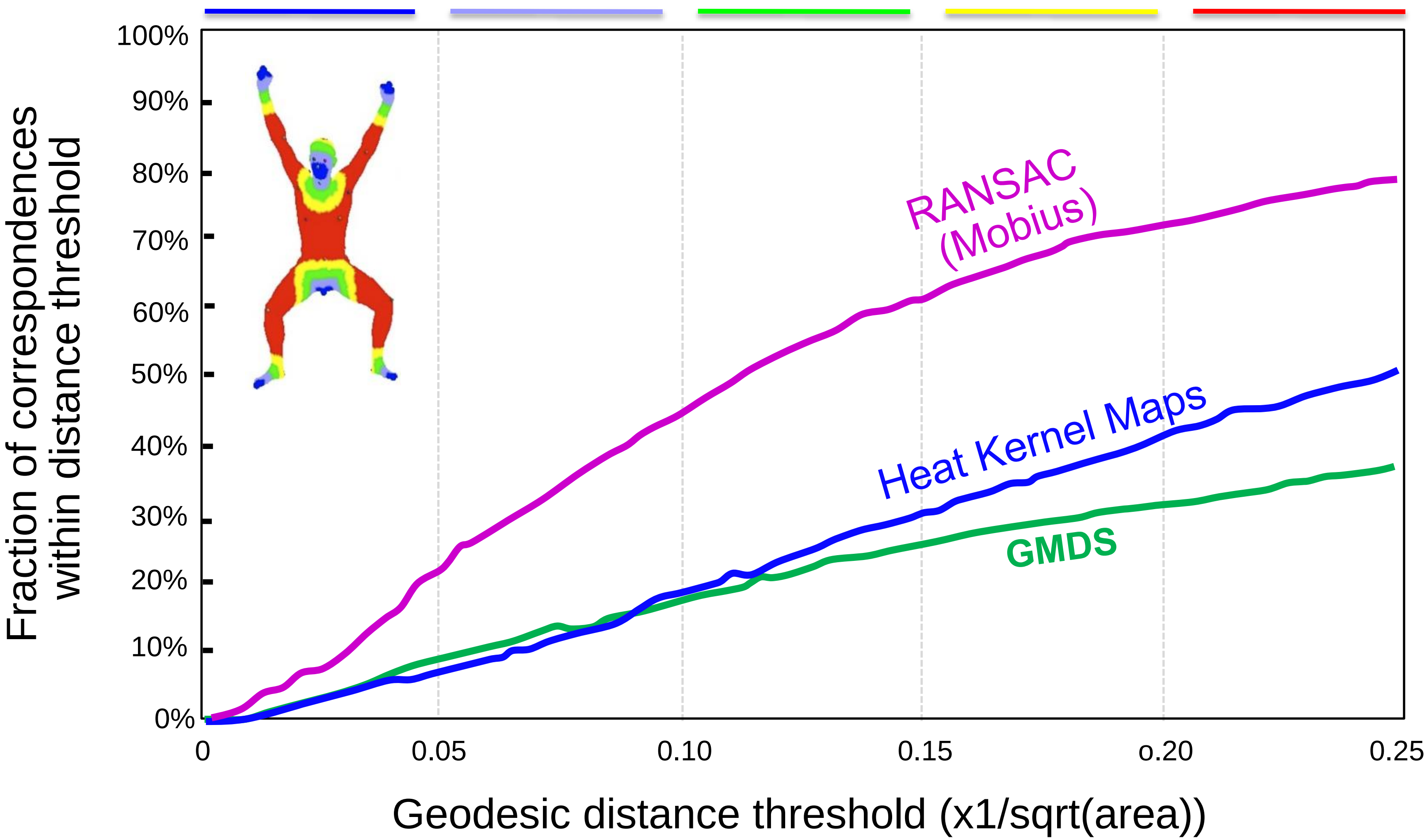
1. For every point with a ground truth correspondence, measure geodesic distance between predicted correspondence and ground truth correspondence

2. Plot fraction of points within geodesic error threshold



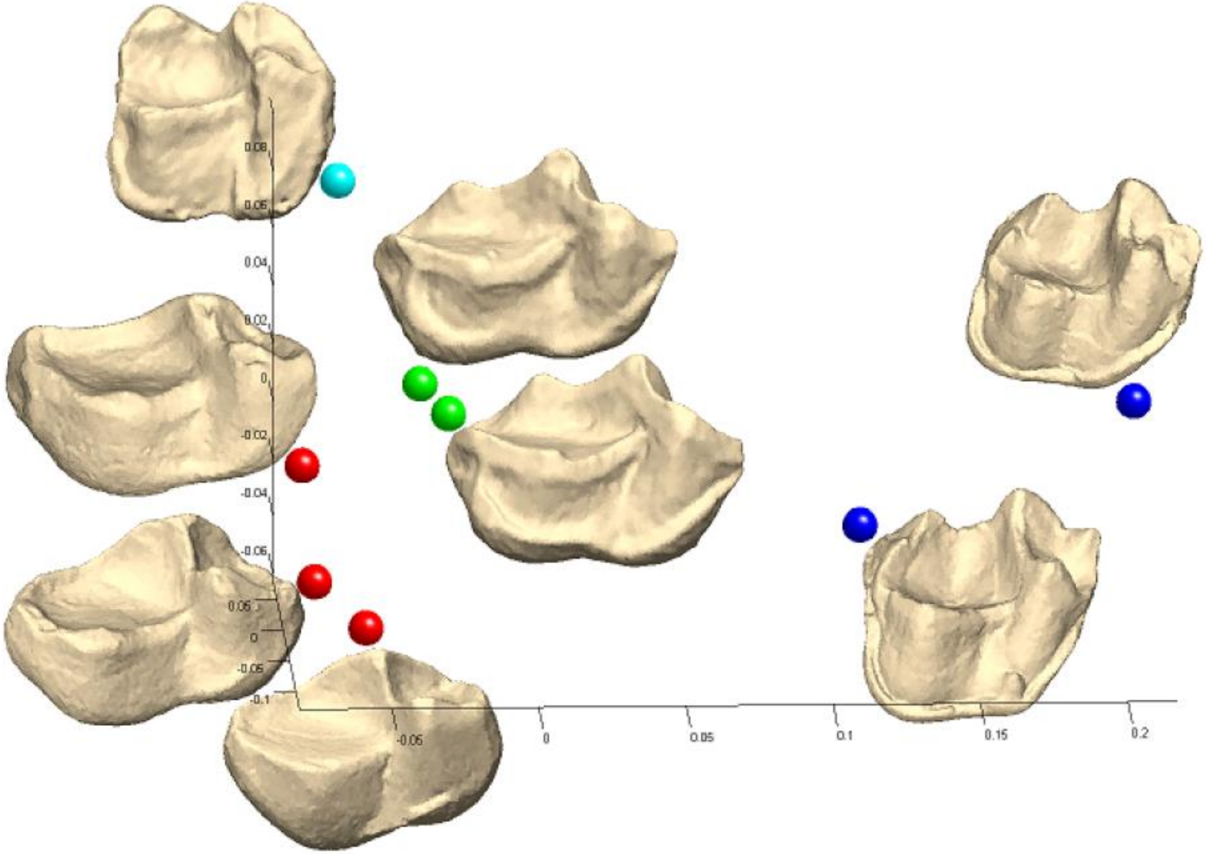
Experimental Results

Results:



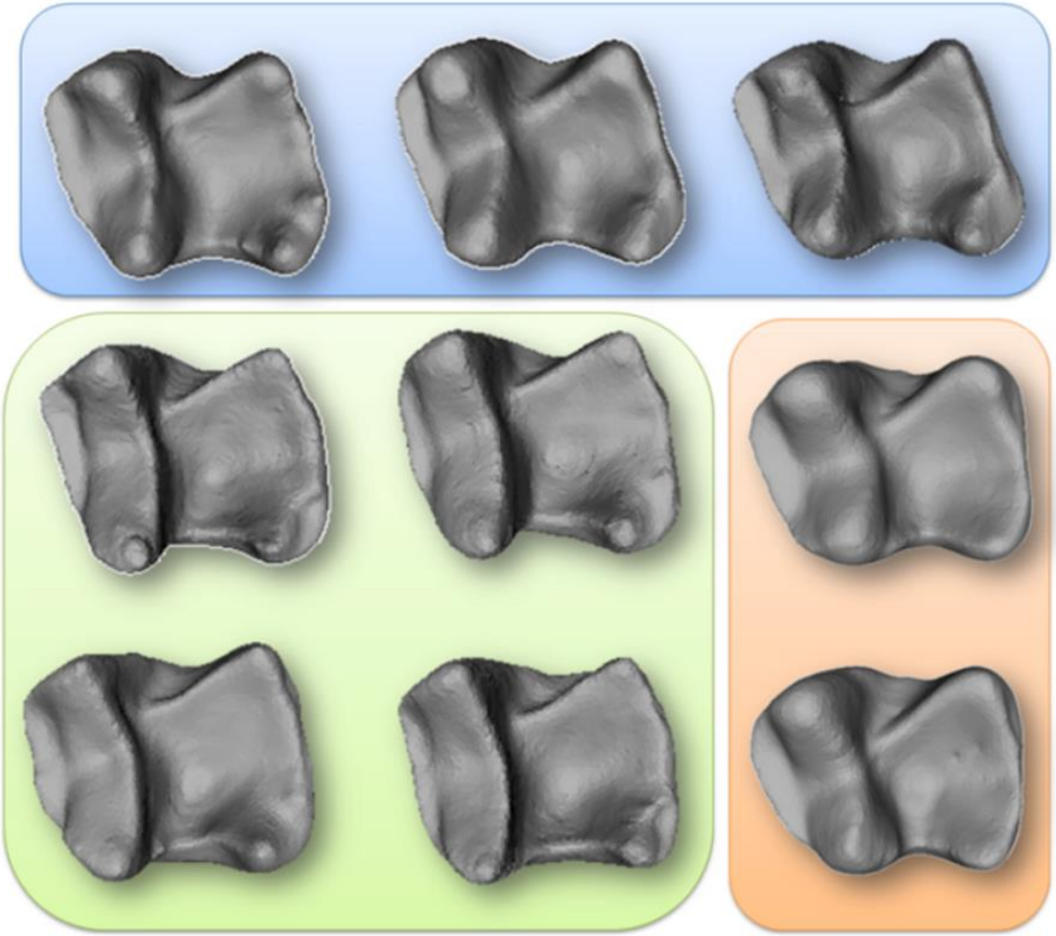
Application

Embedding based on new distance



Application

Clustering based on new distance



Species Groups of Galaga Genus

Application

Classification based on nearest-neighbors

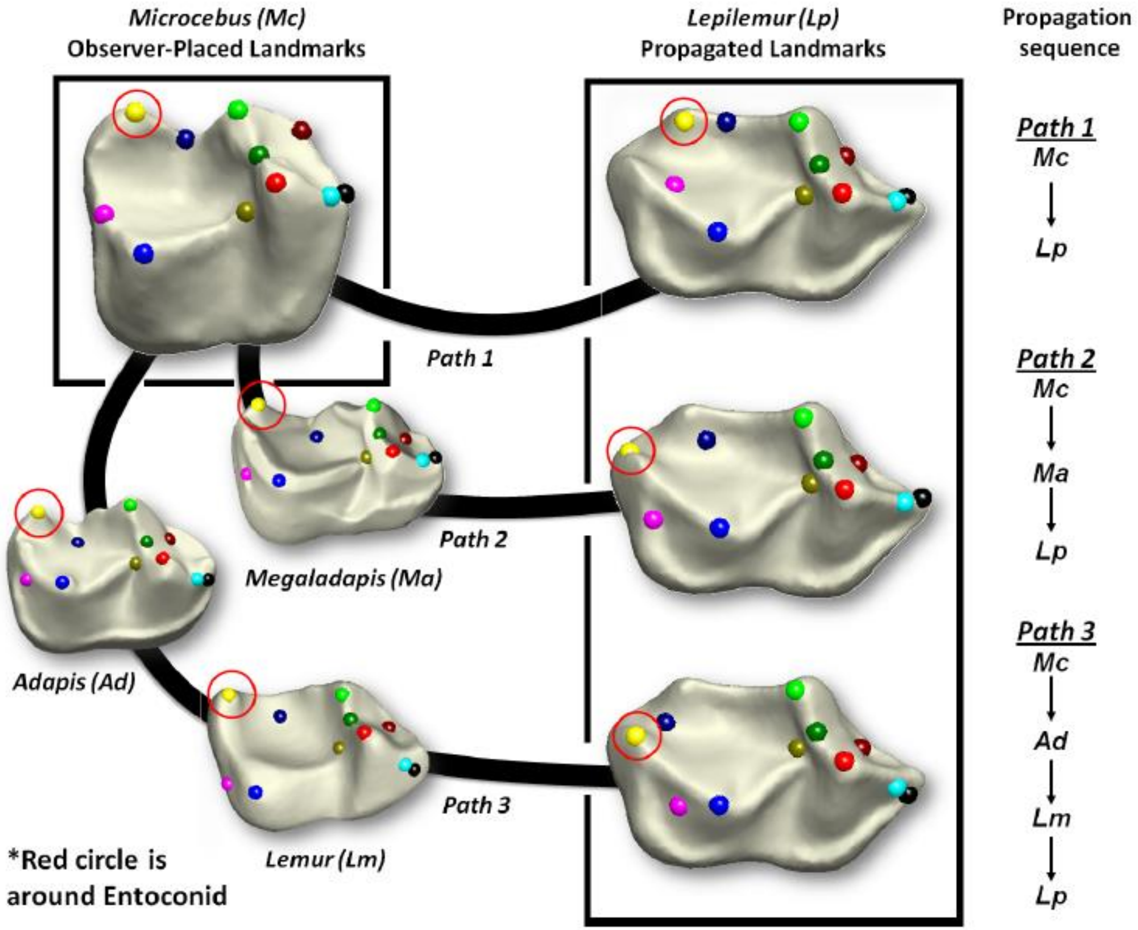
Mandibular Molar	# Groups	# Objects	New Distance	Human Landmarks
Genus	24	99	90.9%	91.9%
Family	17	106	92.5%	94.3%
Order	5	116	94.8%	95.7%

First Metatarsal	# Groups	# Objects	New Distance	Human1 Landmarks	Human2 Landmarks
Genus	13	59	79.9%	76.3%	88.1%
Family	9	61	91.8%	83.6%	93.4%
Superfamily	2	61	100%	100%	100%

Distal Radius	# Groups	# Objects	New Distance	Human Landmarks
Genus	4	45	84.4%	77.7%

Application

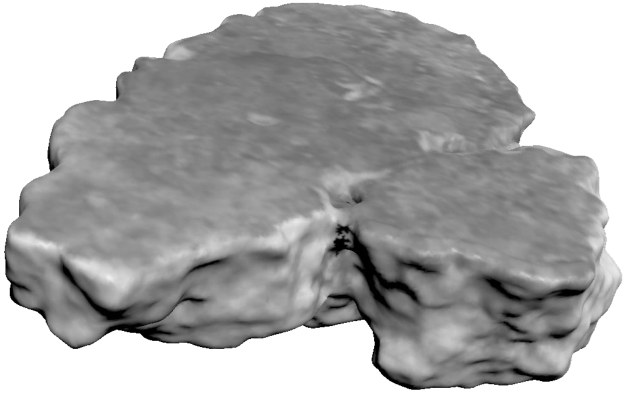
Propagating correspondences

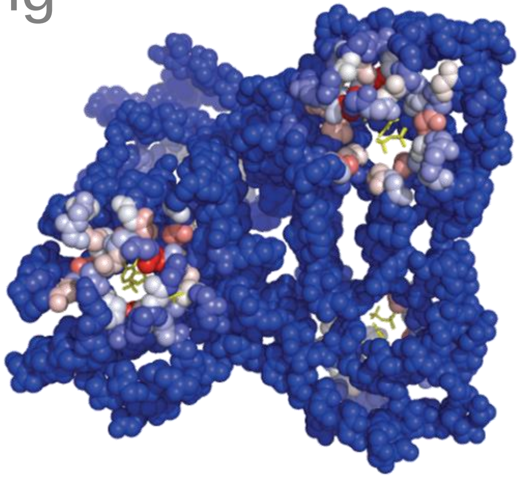


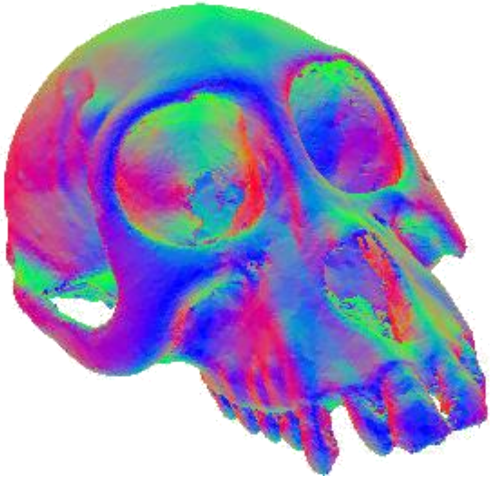
Summary

Shape matching applications:

- Archaeology
- Molecular biology
- Paleontology
- Neuroscience
- Urban planning
- Numismatics
- Geometric modeling
- Medicine
- Art
- etc.







Summary

3D shape matching uses many of the same techniques as 2D image analysis

- Feature detectors
- Feature descriptors
- Feature matching
- etc.

Images courtesy of
Georgia Tech and
www.dreamhorse.com

A Quick Diversion …

Which is harder to recognize by a computer?

3D Model

2D Image

Summary

3D shape matching uses many of the same techniques as 2D image analysis

- Feature detectors
- Feature descriptors
- Feature matching
- etc.

except, …

- Fewer high-frequency features
- More complex topology
- Irregular sampling
- One more dimension
- etc.

Acknowledgments

Archaeology

- Brown, Dobkin, Doumas, Garcia-Castelano, Rusinkiewicz, Shin, Steiglitz, Strife, Toler-Franklin, Vlachopoulos, Weiss, Weyrich

Structural bioinformatics

- Capra, Glaser, Kahraman, Kazhdan, Lazkowski, Morris, Najmanovich, Shilane, Singh, Thornton

Paleontology

- Boyer, Daubechies, Jernvall, Lipman, Patel, Puente, St. Clair