

Evolutionary Methods for Protein Structure Comparison

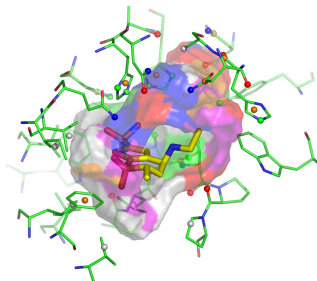
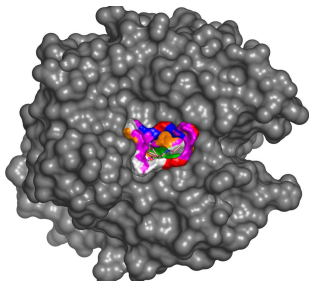
Thomas Fober

Knowledge Engineering & Bioinformatics Group
Mathematics and Computer Science Department



RoGerS, Sibiu, 2009

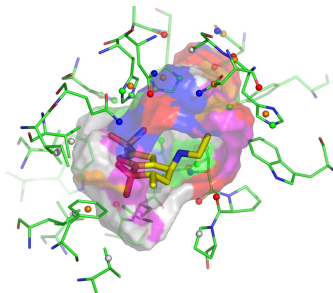
Protein binding sites



- small cavities on the surface of a protein
- protein consists of amino acids; surface of binding site, too
- amino acids define physico-chemical properties
- abstraction: summarize patch into a spatial point

Why are we interested in binding sites?

- small molecules bind these sites and cause a reaction of protein
- pharmaceutical chemistry is interested in binding sites and similarity measures between them
 - ▶ inhibit binding site to cause or suppress a reaction of the protein



Applications

1 cross reactivities

- ▶ binding site of target is known
- ▶ search for proteins with similar binding site
- ▶ these proteins may also be influenced by ligand

2 functional analysis

- ▶ set of proteins with certain function
- ▶ investigate binding site of whole set
- ▶ which parts are responsible for function

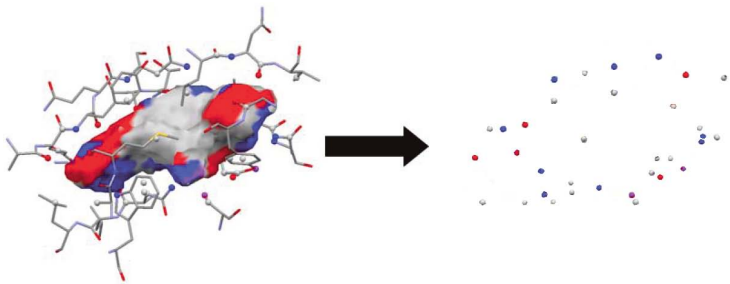
3 prediction of the function

- ▶ protein with unknown function but known structure
- ▶ search for proteins with known function and similar binding site

Goals

- 1 machine readable representation of a protein binding site
- 2 simultaneous investigation of $m > 2$ protein binding sites

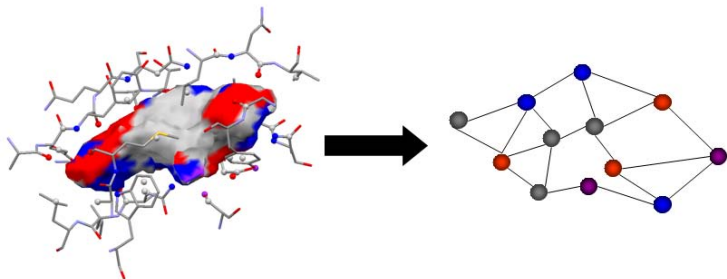
Point cloud representation of protein binding sites



Definition (labeled geometrical point cloud)

a labeled geometrical point cloud is a set P of n points p_i , $i = 1, \dots, n$, with associated functions $l_p : P \rightarrow \mathcal{L}$ and $l_c : P \rightarrow \mathbb{R}^3$

Graph-representation of protein binding sites



Definition (graph)

$G = (V, E, I_V, I_E)$ is a node-labeled and edge weighted graph, where V is a finite set of nodes and $E \subseteq V \times V$ a set of edges, and where $I_V : V \rightarrow \mathcal{L}_V$ and $I_E : E \rightarrow \mathbb{R}$ are functions that assign labels

Similarity measures on graphs

Definition (subgraph)

Let $G = (V, E)$ be a graph. $G' = (V', E')$ is called subgraph of G ($G' \subseteq G$) if $V' \subseteq V$ and $E' \subseteq E$. G' is called induced subgraph ($G' \subseteq_i G$) if additionally holds $E' = E \cap V' \times V'$

Definition (graph isomorphism)

A graph G is isomorphic to another graph G' ($G \simeq G'$) if there exists a mapping $f : V_G \rightarrow V_{G'}$ so that if $(u, v) \in E_G \Leftrightarrow (f(u), f(v)) \in E_{G'}$.

Overview

- graph isomorphism
- subgraph isomorphism
 - ▶ maximum common subgraph
 - ▶ minimum common supergraph
- graph edit distance
- graph kernels

Criteria

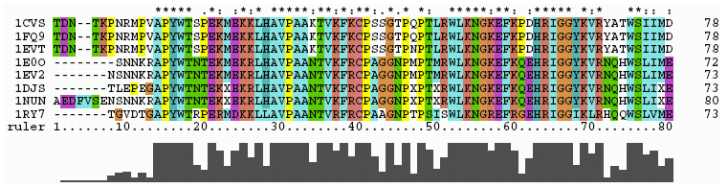
- flexibility?
- efficiency?
- error-tolerance?
- simultaneous computation on $m > 2$ graphs?

how to define well-suited similarity measures?

Outline

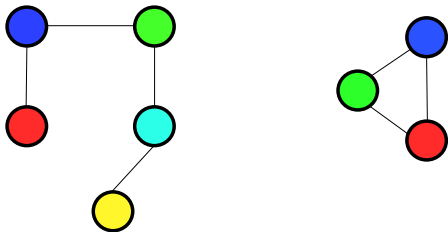
- 1 Introduction
- 2 Multiple graph alignment**
- 3 Point cloud superposition
- 4 Experiments
- 5 Conclusions

Multiple Sequence Alignment

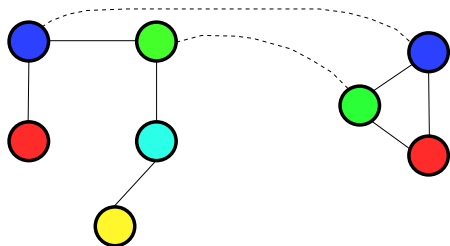


- important concept in bioinformatics
- efficient solution led to a breakthrough in biology

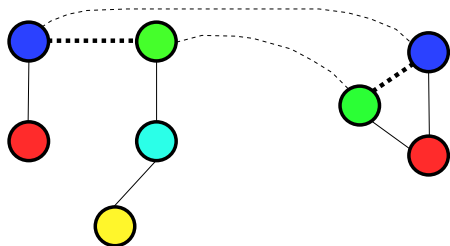
Multiple Graph Alignment



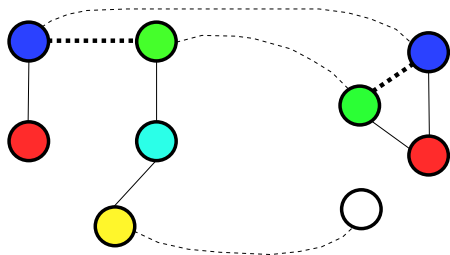
Multiple Graph Alignment



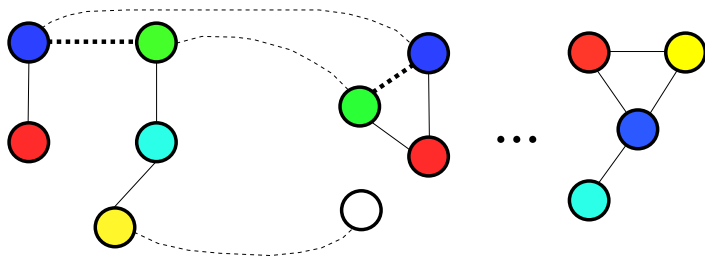
Multiple Graph Alignment



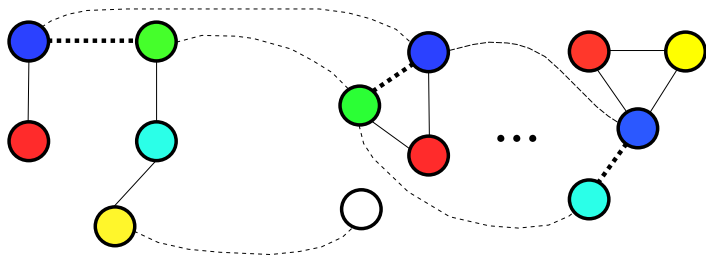
Multiple Graph Alignment



Multiple Graph Alignment



Multiple Graph Alignment



Definition (multiple graph alignment)

$\mathcal{A} \subseteq (V_1 \cup \{\perp\}) \times \dots \times (V_m \cup \{\perp\})$ is a multiple alignment of the graphs $G_1 = (V_1, E_1), \dots, G_m = (V_m, E_m)$ iff

- 1 for all $i = 1 \dots m$ and for each $v \in V_i$ there exists exactly $a = (a_1 \dots a_m) \in \mathcal{A}$, such that $v = a_i$.
- 2 for each $a = (a_1 \dots a_m) \in \mathcal{A}$ there exists at least one $1 \leq i \leq m$, such that $a_i \neq \perp$

Definition (sum-of-pairs measure)

$$f(\mathcal{A}) = \sum_{a_i \in \mathcal{A}} \text{score}_V(a_i) + \sum_{\substack{a_i \in \mathcal{A}, a_j \in \mathcal{A} \\ i < j}} \text{score}_E(a_i, a_j)$$

with

$$\text{score}_V(a_i) = \text{score}_V \begin{pmatrix} a_1^i \\ \vdots \\ a_m^i \end{pmatrix} = \sum_{1 \leq j < k \leq m} \begin{cases} c_{\text{match}} & l_V(a_j^i) = l_V(a_k^i) \\ c_{\text{mismatch}} & l_V(a_j^i) \neq l_V(a_k^i) \\ c_{\text{dummy}} & a_j^i = \perp, a_k^i \neq \perp \\ c_{\text{dummy}} & a_j^i \neq \perp, a_k^i = \perp \end{cases}$$

calculation of edge-score analog

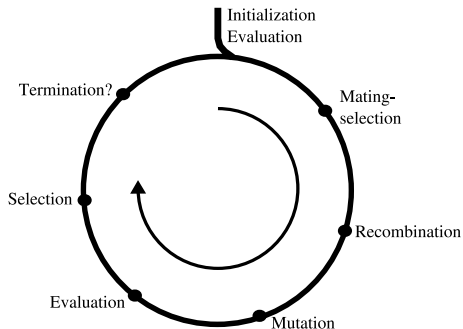
- edit-operations are penalized by $c_i < 0$
- $c_i \geq 0$ rewards a mapping that does not require an edit operation

optimal alignment

$$\mathcal{A}^* \in \mathit{arg\ max}_{\mathcal{A}} f(\mathcal{A})$$

- how to calculate?
- Weskamp et al. (2007): Multiple Graph Alignment for the Structural Analysis of Protein Active Sites, IEEE Transactions on Computational Biology and Bioinformatics

GAVEO

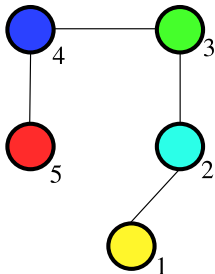


Representation

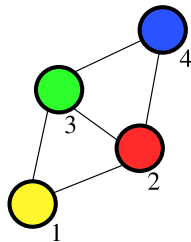
$m \times L$ matrix, $L = \max_{i=1, \dots, m} \{|V_i|\} + 1$

A: 1 3 5 2 4 .

B: 3 4 . 2 1 .



A



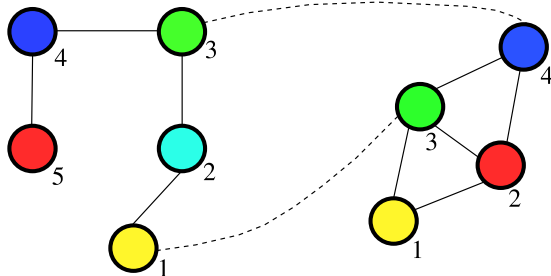
B

Representation

$m \times L$ matrix, $L = \max_{i=1, \dots, m} \{|V_i|\} + 1$

A: 1 3 5 2 4 .

B: 3 4 . 2 1 .



A

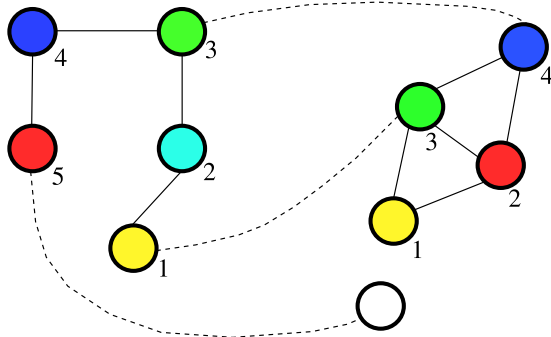
B

Representation

$m \times L$ matrix, $L = \max_{i=1, \dots, m} \{|V_i|\} + 1$

A: 1 3 5 2 4 .

B: 3 4 . 2 1 .



A

B

Initialization and fitness evaluation

- initialization

- ▶ given m graphs; determine $L = \max_{i=1, \dots, m} \{|V_i|\} + 1$
- ▶ in each row j :
 - ★ random permutation over $\{1, \dots, L\}$
 - ★ substitute elements $x > |V_j|$ by dummy nodes

- evaluation

- ▶ sum-of-pairs measure
- ▶ ignore dummy columns

Recombination, $rec : I^2 \rightarrow I$

- recombination of 2 parents
- mechanism required that establishes an ordering
 - choose 1 pivot-graph G_c
 - columns are arranged according to G_c

2	.	1	3	4	3	.	2	4	1	2	.	1	3	4
5	3	1	2	4	1	3	2	4	5	5	3	1	2	4
3	.	2	1	4	2	.	1	4	3	3	.	2	1	4
4	.	2	3	1	1	.	4	3	2	2	.	1	4	3
2	1	4	.	3	2	4	1	.	3	3	4	2	1	.

Mutation, $mut : I \rightarrow I$

- each point in the search space should be reachable
- choose one row and two columns at random
- swap both cells

1	.	2	3	4
1	2	4	.	3
1	.	2	3	4
1	2	4	.	3

↓

1	.	2	3	4
1	2	3	.	4
1	.	2	3	4
1	2	3	.	4

Alignment length

maximal length

1	2	3	4	
.	.	.	.	1	4	2	3	
.	3	2	4	1	5	.	.	.	
.	2	1	3	4

minimal length

1	5	2	3	4
3	2	4	.	1
1	.	2	3	.
4	2	3	.	1

- optimal length?

→ dynamic length adaptation

Dynamic length adaptation

optimal case

```

1 . 2 3 .
3 2 1 . .
1 . 3 2 .
2 1 . 3 .

```

too short

```

1 . 2 . 3
3 2 1 . .
1 . 3 2 .
2 1 . 3 .

```

too large

```

1 . 2 . 3 . .
3 2 1 . . . .
1 . 3 2 . . .
2 1 . 3 . . .

```

EA: Other operators

- mating selection, $I^\mu \rightarrow I^2$: uniformly random
- selection, $I^{\mu+\lambda} \rightarrow I^\mu$: deterministic plus-selection
- termination: after T generations without improvement

Summary

- multiple graph alignment for identification of common patterns
- weight of edit distance can be used as similarity
- calculation expensive
- geometrical information is not used during calculation, instead later restored

Outline

- 1 Introduction
- 2 Multiple graph alignment
- 3 Point cloud superposition**
- 4 Experiments
- 5 Conclusions

Point cloud is *natural* representation

- biological data is often given by a set of labeled points in 3D
 - ▶ Molfile
 - ▶ SDfile
 - ▶ protein binding sites are geometric objects; pseudocenters spatial points
- transforming these point clouds into graphs leads to a loss of information
- idea: use point clouds directly to calculate similarity
- will be less flexible than MGA since geometry is rigid

Similarity on labeled point clouds

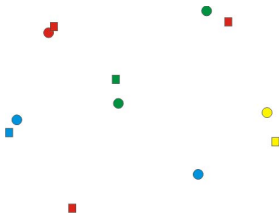
- a point cloud is a set of points
- classical equivalence of sets

$$A = B \Leftrightarrow A \subseteq B \wedge A \supseteq B$$

- adapt equivalence for sets to point clouds
- differences:
 - ▶ coordinates
 - ▶ labels
 - ▶ error tolerant measure in range $[0, 1]$

$A \subseteq B$

... by measuring the distances $d(i, B)$ from each point i in A with certain label to closest point in the other cloud B with same label

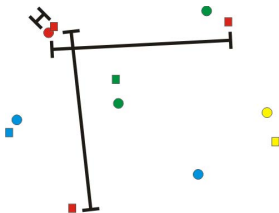


- inclusion in unit-interval \rightarrow use mapping $d \mapsto e^{-d}$
- inclusion by

$$inc(A, B) = \frac{1}{|A|} \cdot \sum_{x_i \in A} e^{-d(x_i, B)}$$

$A \subseteq B$

... by measuring the distances $d(i, B)$ from each point i in A with certain label to closest point in the other cloud B with same label



- inclusion in unit-interval \rightarrow use mapping $d \mapsto e^{-d}$
- inclusion by

$$inc(A, B) = \frac{1}{|A|} \cdot \sum_{x_i \in A} e^{-d(x_i, B)}$$

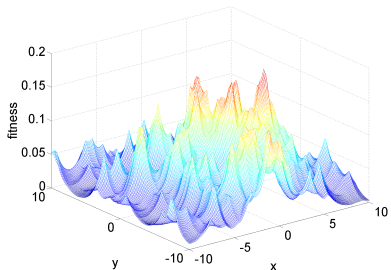
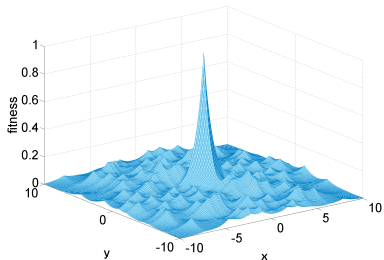
- point clouds have relative coordinates, so *inc* depends on position and orientation in 3D
- optimize orientation and position of 2nd point cloud w.r.t. maximizing the similarity

$$INC(A, B) = \max_{t \in \mathbb{R}^3 \times [0, 2\pi]^3} inc(TF(B, t), A)$$

- always 6 parameters $(t_1, t_2, t_3, \alpha, \beta, \gamma)$ to optimize \rightarrow independent of problem size

- use evolutionary strategy for optimization
- use restarts to find global maximum
- optimized inclusion will give the real inclusion between two point clouds

Fitness landscape



Similarity of point clouds

$$\text{sim}(A, B) = \frac{1}{2}(\text{INC}(A, B) + \text{INC}(B, A))$$

- symmetric measure
- similarity values in $[0, 1]$

Construction of geometrical Alignments

- is point cloud superposition only a similarity measure? No!
- given distances between points we can construct an alignment using optimal assignment algorithms
- pairwise alignment can be extended to multiple one by using star alignment

	a'	b'	c'	\perp	\perp	\perp
a	$d(a, a')$	$d(a, b')$	$d(a, c')$	t	t	t
b	$d(b, a')$	$d(b, b')$	$d(b, c')$	t	t	t
c	$d(c, a')$	$d(c, b')$	$d(c, c')$	t	t	t
d	$d(d, a')$	$d(d, b')$	$d(d, c')$	t	t	t
\perp	t	t	t	0	0	0
\perp	t	t	t	0	0	0

Summary

- point cloud is often the natural representation of real data
- working directly on point clouds will not lead to a loss of information
- using geometrical information during calculation accelerates the process
- point cloud superposition for measuring similarity and finding common substructures

Outline

- 1 Introduction
- 2 Multiple graph alignment
- 3 Point cloud superposition
- 4 Experiments**
- 5 Conclusions

Methods

- 1 Westkamp et al. (2007) approach
- 2 GAVEO
- 3 LPCS
- 4 kernels
 - ▶ kernels are used in machine learning to solve non-linear problems with linear classifiers
 - ▶ map from object- to feature space using function ϕ
 - ▶ kernel trick: $k(G, G') = \langle \phi(G), \phi(G') \rangle = \phi(G)^T \phi(G')$
 - ▶ a function k is a kernel, if
 - ★ $k : \mathcal{G} \times \mathcal{G} \rightarrow \mathbb{R}$
 - ★ k is symmetric
 - ★ k is positive definite
 - ▶ R-convolution:

$$k(G, G') = \sum_{\substack{g \in R^{-1}(G) \\ g' \in R^{-1}(G')}} \kappa(g, g')$$

Classification

Dataset: NADH/ATP

- data set consists of 355 protein binding sites
- 241 pockets bind to ATP ligands
- 141 pockets bind to NADH ligands
- 2 class classification problem

- use approaches as similarity measure (pairwise case)
- better similarity \Rightarrow higher classification rates
- two-class ATP/NADH data set
 - predict to which ligand a pocket binds
 - 1-NN classifier
 - leave-one-out cross validation

	RW	SP	Weskamp et al.	GAVEO	LPCS
acc	59.70%	60.60%	76.62%	78.87%	92.11%
rt	65.5 \pm 89.1	9.8 \pm 97.8	74.2 \pm 85.6	—	20.0 \pm 24.7

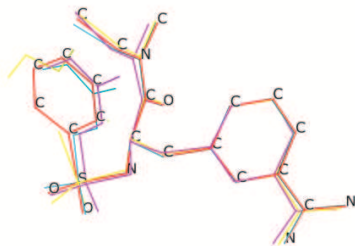
Structure retrieval

Dataset: benzamidine

- 87 chemical compounds
- graphs descriptors contain 47 – 100 nodes
- all compounds share a common core fragment
- data set has a clear and unambiguous pattern

- 87 compounds of benzamidine derivates
- all derivates share the benzamidine core fragment
- search for conserved structures

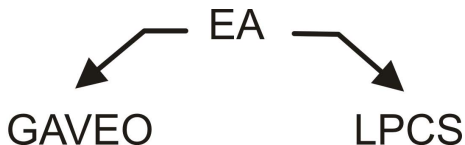
m	Weskamp et al.	GAVEO	LPCS
2	58%	100%	96%
4	38%	96%	92%
8	14%	94%	80%



Outline

- 1 Introduction
- 2 Multiple graph alignment
- 3 Point cloud superposition
- 4 Experiments
- 5 Conclusions**

Conclusions



- very flexible
- similarity measure
- finds common substructures
- expensive
- uses geometrical information
- independent of problem size
- high classification rates
- almost parameter-less