

# **Vorlesung Bioinformatik**

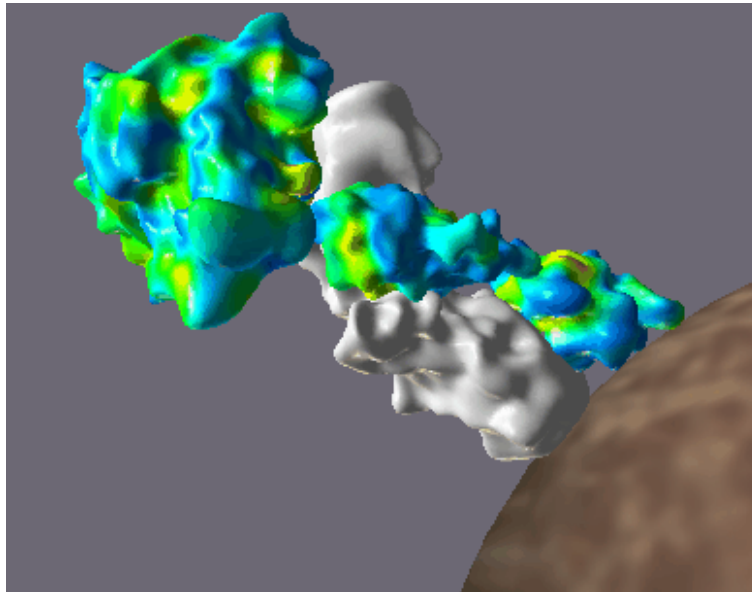
## **Proteininteraktionen und Geometric Hashing**

Dr. Axel Mosig

25. Mai 2004

# Interaktionen zwischen Proteinen

- Wirkung von Proteinen entfaltet sich häufig durch Interaktion mit anderen Proteinen:



# Interaktionen zwischen Proteinen

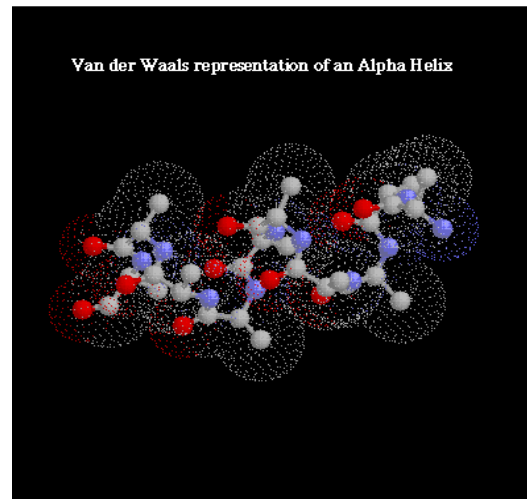
- $\rightsquigarrow$  Interaktion zwischen Moleküloberflächen
- Zwei Teilprobleme:
  - I. Wie lässt sich die Oberfläche eines Proteins sinnvoll repräsentieren?
  - II. Wie gut interagieren zwei Proteine?
    - $\rightsquigarrow$  geometrische/chemische Komplementarität  $\rightsquigarrow$  Geometric Hashing

# I. Moleküloberflächen

- Diverse Arten der Oberflächen Repräsentierung, z.B.:
  - Van der Waals-Oberfläche
  - Connolly-Oberfläche
  - $\alpha$ -shapes (Edelsbrunner et al.)
  - ...

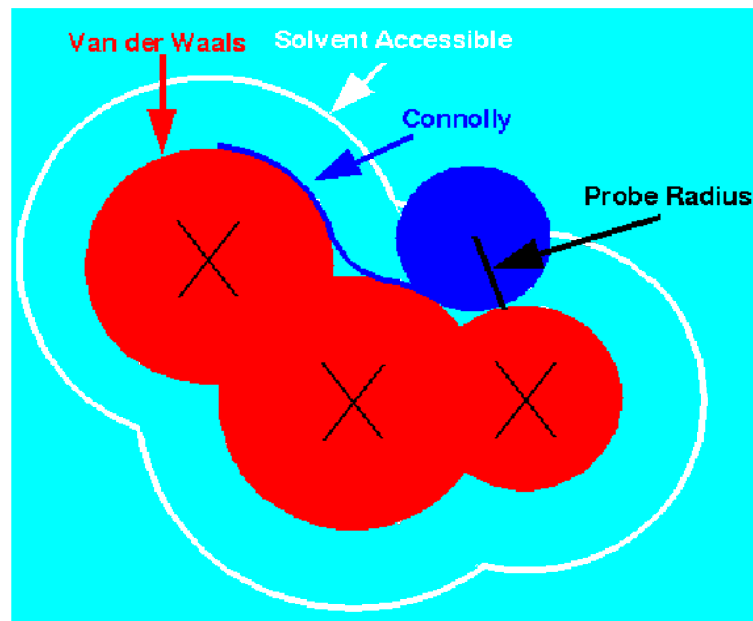
# Van der Waals-Oberfläche

- Lege um jedes Atom eine Kugel vom Radius des mittleren Abstands der Elektronen vom Atomkern
- Die so entstehende Oberfläche ist die *van-der-Waals-Oberfläche*



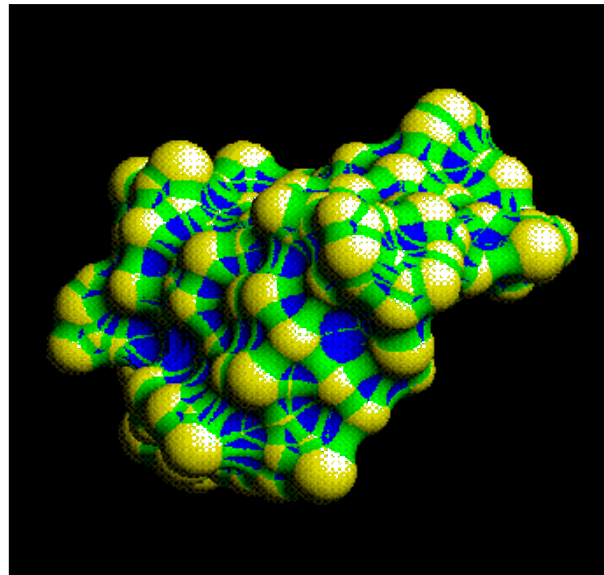
# Connolly Oberfläche

- Ball mit Durchmesser eines Wassermoleküls wird über vdW-Oberfläche “abgerollt”



## Connolly Oberfläche

- Beim “abrollen” entstehen konvexe (gelb), konkave (blau) und sattelförmige (grün) Bereiche.



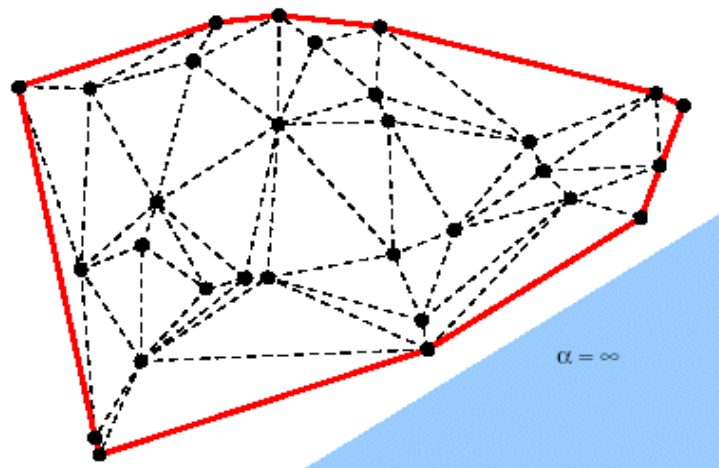
- Aus jedem Oberflächenstück werden üblicherweise Punkte gesampelt
- gesampelte Punkte zusammen mit Normalenvektoren repräsentieren Oberfläche.

## Konvexe Hülle und $\alpha$ -Shapes

- Weitere Form der Oberflächen-Repräsentation eines Moleküls
- $\alpha$ -shape einer Punktmenge ( $\equiv$  Atom-Koordinaten eines Moleküls) ist Verallgemeinerung der konvexen Hülle.
- Berechnung auf Grundlage der Delaunay-Triangulation

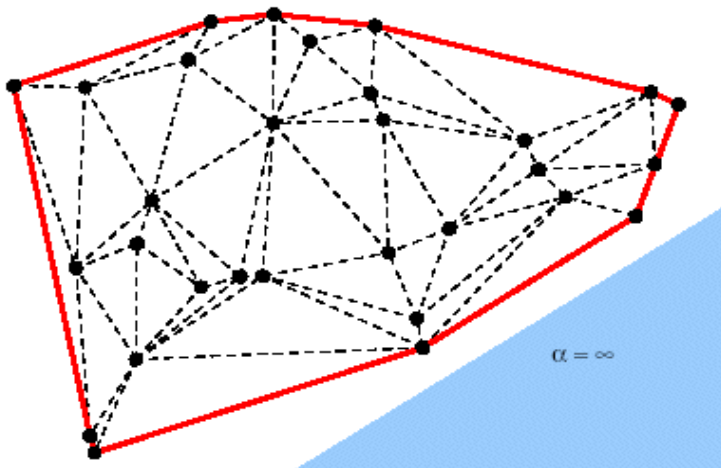
# Konvexe Hülle

- Konvexe Hülle einer Punktmenge  $M$ : kleinste konvexe Menge, die  $M$  enthält



# Konvexe Hülle

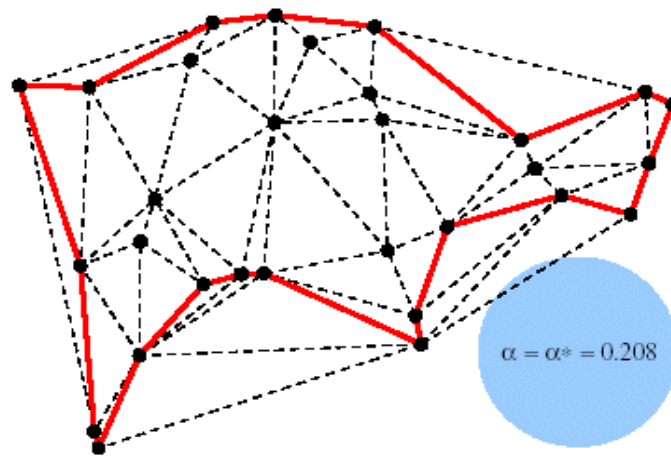
- Andere Sichtweise: Menge aller Punkte, die von einer Halbebene berührt werden können;
- dabei soll kein Punkt aus  $M$  innerhalb der Halbebene liegen.



## $\alpha$ -Shapes

- Verallgemeinerung der konvexen Hülle: Fasse Halbebene als Kreis vom Radius  $\alpha = \infty$  auf.
- $\alpha$ -shape von  $M$ :
- Menge aller Punkte, die von einem Kreis vom Radius  $\alpha$  berührt werden können;
- dabei soll kein Punkt aus  $M$  innerhalb des Kreises liegen.

## $\alpha$ -Shapes



- Verallgemeinerung auf dreidimensionale Punktmengen kanonisch.
- Effiziente Algorithmen u. Implementierung in 3D: Edelsbrunner/Mücke (1994).

## Extraktion signifikanter Merkmale

- *Problem:* Connolly-Oberfläche (und auch  $\alpha$ -shape) liefert (zu) viele Punkte
- Für Protein-Protein-Docking werden daher signifikante Punkte extrahiert
- Norel et al. (1995): Signifikante Punkte an Extremstellen von Aushöhlungen und Ausbeulungen

# Geometric Hashing: Problemstellung

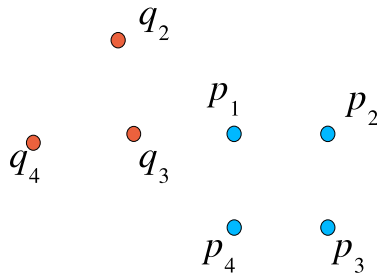
- Gegeben Features zweier Proteinoberflächen  $P$  und  $Q$
- Die Lage von  $Q$  ist variabel – i.e., kann *verschoben* und *rotiert* werden  
 $\rightsquigarrow$  Transformation durch *starre Bewegung*
- Starre Bewegung  $g \rightsquigarrow gQ \equiv Q$  transformiert durch  $g$
- *Für welche starre Bewegung  $g$  haben  $P$  und  $gQ$  die größtmögliche komplementäre Teilstruktur?*

# Geometric Hashing

- *Fragen:*
  - Wie lässt sich Komplementarität bewerten?
  - Wie finden wir eine Transformation, die Komplementarität maximiert?

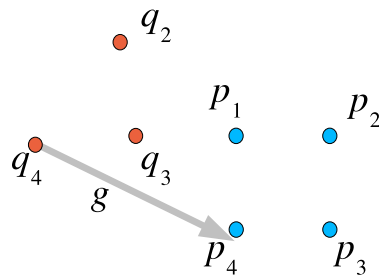
# Geometric Hashing – Vereinfachtes Szenario

- Zunächst beschränken wir uns nur auf *Translationen*.
- Gegeben eine Menge von Mustern  $P_1, \dots, P_K$ .
- *Zusätzliche Vereinfachung*: Suche Translation  $g$ , so dass  $gQ$  als Teilkonstellation in einem  $P_k$ ,  $k \in [1 : K]$  vorkommt:



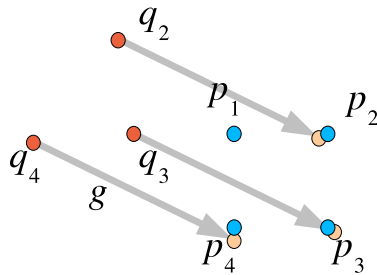
# Geometric Hashing – Vereinfachtes Szenario

- Zunächst beschränken wir uns nur auf *Translationen*.
- Gegeben eine Menge von Mustern  $P_1, \dots, P_K$ .
- *Zusätzliche Vereinfachung*: Suche Translation  $g$ , so dass  $gQ$  als Teilkonstellation in einem  $P_k$ ,  $k \in [1 : K]$  vorkommt:



# Geometric Hashing – Vereinfachtes Szenario

- Zunächst beschränken wir uns nur auf *Translationen*.
- Gegeben eine Menge von Mustern  $P_1, \dots, P_K$ .
- *Zusätzliche Vereinfachung*: Suche Translation  $g$ , so dass  $gQ$  als Teilkonstellation in einem  $P_k$ ,  $k \in [1 : K]$  vorkommt:



## Aufbau der Hash-Tables

- Unterteile Ebene durch diskretes uniformes Gitter
- Jeder Punkt in der Ebene lässt sich einer Gitterregion zuordnen
- Für jede Gitterregion lege eine Hash-Tabelle an

## Aufbau der Hash-Tables

```
Init-Hash-Tables( $P_1, \dots, P_k$ )  
  for  $k \in [1 : K]$   
    //  $P_k = \{p_1, \dots, p_{n_k}\}$   
    for  $i \in [1 : n_k]$   
      ermittle jene Translation  $t_i$ , welche  $p_i$  in den  
      Ursprung verschiebt;  
      for  $j \in [1 : n_k]$   
        Trage das Tupel  $(k, i)$  in die zum Punkt  $p_j - t_i$   
        korrespondierende Hash-Tabelle ein;  
end.
```

## Durchführen der Mustersuche

- Benutze Hash-Tabellen, um nach verschobenem Muster  $Q = \{q_1, \dots, q_m\}$  zu suchen.
- Wähle ein beliebiges  $q_i$  aus.
- $h$  sei jene Translation, die  $q_i$  auf den Ursprung verschiebt.
- Bilde die Vereinigung  $T$  von allen Hash-Tabellen, die von irgendeinem  $hq_j$  “gertroffen” werden.
- Ermittle jenes Tupel  $(k, j)$ , das in  $T$  am häufigsten vorkommt.

## Verallgemeinerung auf starre Bewegungen

- *Problem:* Wenn  $P_k$  durch  $t_i$  verschoben wird, kann  $t_i P_k$  noch beliebig rotiert werden.
- *Lösung:* Um eine eindeutige starre Bewegung zu erhalten, wähle beim Aufbau der Hash-Tabellen immer *drei* Punkte  $p_\alpha, p_\beta, p_\gamma$ .
- $\exists$  eindeutig (!?) definierte starre Bewegung  $g$  mit Eigenschaften:
  - (R1)  $gp_\alpha$  liegt im Ursprung
  - (R2) Der durch  $gp_\alpha$  und  $gp_\beta$  definierte Vektor ist parallel zur  $x$ -Achse
  - (R3) Das durch  $gp_\alpha, gp_\beta$  und  $gp_\gamma$  definierte Dreieck liegt in der durch die  $x$ - und die  $y$ -Achse definierten Ebene

## Verallgemeinerung auf starre Bewegungen

- Beim durchführen der Mustersuche wähle drei Punkte  $q_m, q_\nu, q_\kappa$ .
- Bestimme eindeutig definierte starre Bewegung  $h$ , so dass  $h$  (R1)–(R3) erfüllt.
- Vereinige alle Hash-Tabellen, die von einem  $hq_j$  “getroffen” werden.
- Ermittle jenes Tupel  $(k, (\alpha, \beta, \gamma))$ , das in  $T$  am häufigsten vorkommt.
- Chemische Merkmale der Oberfläche lassen sich ohne weiteres integrieren.