Learning Scoring Function Parameters from Binary Data

Markus Seifert
4SC AG

ACS National Meeting 2009
Salt Lake City
4SC – Company Facts

- Founded in 1997, since 2005 listed at the Frankfurt Stock Exchange

- Development pipeline:

<table>
<thead>
<tr>
<th>Indication</th>
<th>Discovery</th>
<th>Pre-Clinic</th>
<th>Phase I</th>
<th>Phase II</th>
<th>Phase III</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rheumatoid Arthritis</td>
<td></td>
<td></td>
<td>4SC-101 (DHODH)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IBD</td>
<td></td>
<td></td>
<td>4SC-101 (DHODH)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IBD/RA</td>
<td>4SC-102 (Immunmodul.)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Viral infections</td>
<td>4SC-301 (NFκB)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Oncology</td>
<td></td>
<td></td>
<td>4SC-201 (HDAC)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Oncology</td>
<td>4SC-206 (Proteasome)*</td>
<td></td>
<td></td>
<td></td>
<td>*licensed to ViroLogik for antiviral applications</td>
</tr>
<tr>
<td>Oncology</td>
<td>4SC-203 (Multi Kinase Inhibitor)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Solid Tumours</td>
<td>4SC-202 (HDAC)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Solid Tumours</td>
<td>4SC-205 (Eg5-kinesin)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Objective

- A method to parameterize a scoring function for virtual screening efficiently, reliably, and automatically
- Example: protein-ligand docking
Why binary data?

✔ Data availability
  - Minimum amount of data you have in early stages
  - binary data i.e. some active and inactive molecules

✔ Noise reduction
  - Binning reduces experimental noise
  - If done properly
What influences docking?

- Docking Parameters
- Software-specific Heuristics
What influences docking?

Docking Parameters

Software-specific Heuristics
Finding one hundred in a million …

\[
\log \frac{p(a \mid x)}{1 - p(a \mid x)} = \log \frac{p(a)}{1 - p(a)} + \log \frac{\sigma_i}{\sigma_a} + \frac{1}{2} \left[ \left( \frac{x - \bar{x}_i}{\sigma_i} \right)^2 - \left( \frac{x - \bar{x}_a}{\sigma_a} \right)^2 \right]
\]
Finding one hundred in a million …

AUR = 0.76
P(active|score = -4) = 0.003

top 0.2%  
~ 2 hits

actives  inactives

1σ
Finding one hundred in a million …

\[ \text{AUR} = 0.92 \]
\[ P(\text{active}|\text{score} = -4) = 0.04 \]

Top 0.2% 
~ 16 hits
Finding one hundred in a million …

\[ \text{AUR} = 0.98 \]
\[ P(\text{active}|\text{score} = -4) = 0.15 \]

Top 0.2% ~ 50 hits

[Diagram showing a normal distribution with AUR and score values labeled]
Iterative learning

- Docking algorithm
  - Dock and score active and inactive cpds.

- Objective function
  - Measure difference between score distributions of active and inactive cpds.

- Optimization algorithm
  - Take a step in parameter space that maximizes objective function
Analysis of Variance (ANOVA)
- $\eta^2$ similar to correlation coefficient
- $F$ measures significance

Why ANOVA?
- well established
- statistically sound
- very versatile
### Two-way ANOVA

<table>
<thead>
<tr>
<th>Activity</th>
<th>Target</th>
<th>Factor A</th>
<th>Factor B</th>
<th>Interaction</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>inactive</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>0 0</td>
<td>2.1702</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>-1</td>
<td>0 1 0</td>
<td>0 1 0</td>
<td>-2.4917</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>-1</td>
<td>0 0 1</td>
<td>0 0 0</td>
<td>3.3639</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>-1</td>
<td>-1 -1 -1</td>
<td>1 1 1</td>
<td>1.7591</td>
</tr>
<tr>
<td>active</td>
<td>1</td>
<td>1 1 0</td>
<td>0 1 0</td>
<td>1 1 0</td>
<td>-1.5612</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1 0 1 0</td>
<td>0 1 1 0</td>
<td>0 1 1 1 0</td>
<td>-3.4595</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>1 0 0</td>
<td>0 1 0</td>
<td>0 0 0 1</td>
<td>1.5426</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>1 -1 -1</td>
<td>-1 -1 -1</td>
<td>1 -1 -1 -1</td>
<td>7.7478</td>
</tr>
</tbody>
</table>

\[
F_A = \frac{\left(R_{y,x_A x_B x_{AxB}}^2 - R_{y,x_{B}x_{AxB}}^2\right) \cdot (N - p \cdot q)}{\left(1 - R_{y,x_A x_B x_{AxB}}^2\right) \cdot (p - 1)}
\]
Ingredient 2: Dividing rectangles (DIRECT)

⇒ Deterministic global optimization algorithm

D-TOP approach

- **Direct Target-class optimization**
  - Multiple target proteins simultaneously = higher efficiency
  - Multiple protein conformations
  - Multiple tautomers of ligands

- **Global optimization**
  - Two-way ANOVA
  - DIRECT

---

Eval. on full DUD data sets: 4 targets

Sampling ~400 cpds.

Optimized parameters

Eval. on full DUD data sets: 4 targets

External validation 2 targets

iterative learning
Validation

Small molecules from the “directory of useful decoys”

Training:
- **CDK2** (PDB 2R3M, 2R3O)  
  72 : 2074
- **EGFR** (PDB 1M17)  
  475 : 15996
- **Src** (PDB 2SRC)  
  159 : 6319
- **P38α** (PDB 1A9U, 1WBS)  
  454 : 9141

Test (DUD):
- **VEGFR2** (PDB 2P2H)  
  88 : 2906
- **FGFR1** (PDB 1AGW)  
  120 : 4550

Summary of Results: D-TOP
Raw data: kinase docking initially
Raw data: kinase docking internal validation
Raw data: kinase docking external validation
Confidence intervals

FGFR, DUD

FGFR, Rand.

VEGFR, DUD

VEGFR, Rand.

Area: p<0.0001
TPF 5%: p<0.0001

Area: p=0.0012
TPF 5%: n.s.

Area: p=0.0004
TPF 5%: p=0.05
Why should I do such complicated things?
It’s much easier to make a QSAR model …
D-TOP: internal validation

Scoring function: 7 parameters
D-TOP: external validation

Scoring function: 7 parameters
Naïve Bayes: internal validation

Binary QSAR: 7 independent components
Naïve Bayes: external validation

Binary QSAR: 7 independent components
Why does virtual screening improve?

![Graphs showing the improvement in virtual screening](image-url)
Bottom line

- Target-specific optimization is flexible and efficient

- This approach is generally applicable
  - for optimizing many other things for virtual screening
  - other software, other parameters, etc.
Acknowledgement

- Daniel Vitt
- Bernd Kramer
- Thomas Herz
- Jürgen Kraus