Kernel methods for *in silico* chemogenomics

Laurent Jacob
Jean-Philippe Vert

Center for Computational Biology, Mines ParisTech
Bioinformatics Department, Institut Curie
U900 Inserm

CAp 2008

Outline

1. Introduction
   - Drug discovery
   - Chemogenomics
2. Method
   - Formalization
   - Representation of pairs
   - Kernel for ligands and targets
3. Results
4. Conclusion

Outline

1. Introduction
   - Drug discovery
   - Chemogenomics
2. Method
   - Formalization
   - Representation of pairs
   - Kernel for ligands and targets
3. Results
4. Conclusion

Outline

1. Introduction
    - Drug discovery
    - Chemogenomics
2. Method
    - Formalization
    - Representation of pairs
    - Kernel for ligands and targets
3. Results
4. Conclusion

Outline

1. Introduction
    - Drug discovery
    - Chemogenomics
2. Method
    - Formalization
    - Representation of pairs
    - Kernel for ligands and targets
3. Results
4. Conclusion

Outline

1. Introduction
   - Drug discovery
   - Chemogenomics
2. Method
3. Results
4. Conclusion

Drug discovery

Classical approach

- Imitate traditional remedies.
- Accidental discoveries.

New trend

1. Understand underlying biological process.
2. Identify targets (typically proteins).
3. Identify modulators of these targets.

Typical targets

- G-protein-coupled-receptors (GPCR).
- Enzymes.
- Ion channels.

Drug discovery

Classical approach

- Imitate traditional remedies.
- Accidental discoveries.

New trend

1. Understand underlying biological process.
2. Identify targets (typically proteins).
3. Identify modulators of these targets.

Typical targets

- G-protein-coupled-receptors (GPCR).
- Enzymes.
- Ion channels.

Drug discovery

Classical approach

- Imitate traditional remedies.
- Accidental discoveries.

New trend

1. Understand underlying biological process.
2. Identify targets (typically proteins).
3. Identify modulators of these targets.

Typical targets

- G-protein-coupled-receptors (GPCR).
- Enzymes.
- Ion channels.

Drug discovery

Classical approach

- Imitate traditional remedies.
- Accidental discoveries.

New trend

1. Understand underlying biological process.
2. Identify targets (typically proteins).
3. Identify modulators of these targets.

Typical targets

- G-protein-coupled-receptors (GPCR).
- Enzymes.
- Ion channels.

Drug discovery

Classical approach

- Imitate traditional remedies.
- Accidental discoveries.

New trend

1. Understand underlying biological process.
2. Identify targets (typically proteins).
3. Identify modulators of these targets.

Typical targets

- G-protein-coupled-receptors (GPCR).
- Enzymes.
- Ion channels.

Drug discovery

Classical approach

- Imitate traditional remedies.
- Accidental discoveries.

New trend

1. Understand underlying biological process.
2. Identify targets (typically proteins).
3. Identify modulators of these targets.

Typical targets

- G-protein-coupled-receptors (GPCR).
- Enzymes.
- Ion channels.

Virtual screening

Motivation

- Need to test a huge number of candidate molecule against a target.
- *In silico* interaction prediction is therefore a key element.

Classical paradigms

- Ligand-based : compare candidate ligand to known ligands of the target (*e.g.* using machine learning).
- Structure-based (docking) : use the 3D structure of the target to determine how well each candidate binds.

Virtual screening

Motivation

- Need to test a huge number of candidate molecule against a target.
- *In silico* interaction prediction is therefore a key element.

Classical paradigms

- Ligand-based : compare candidate ligand to known ligands of the target (*e.g.* using machine learning).
- Structure-based (docking) : use the 3D structure of the target to determine how well each candidate binds.

Virtual screening

Motivation

- Need to test a huge number of candidate molecule against a target.
- *In silico* interaction prediction is therefore a key element.

Classical paradigms

- Ligand-based : compare candidate ligand to known ligands of the target (*e.g.* using machine learning).
- Structure-based (docking) : use the 3D structure of the target to determine how well each candidate binds.

Virtual screening

Motivation

- Need to test a huge number of candidate molecule against a target.
- *In silico* interaction prediction is therefore a key element.

Classical paradigms

- Ligand-based : compare candidate ligand to known ligands of the target (*e.g.* using machine learning).
- Structure-based (docking) : use the 3D structure of the target to determine how well each candidate binds.

Outline

1. Introduction
   - Drug discovery
   - Chemogenomics

2. Method

3. Results

4. Conclusion

Chemogenomics approach

Limits of the classical paradigms

- Ligand-based : need to know (enough) ligands of a given target to produce (accurate) predictors.
- Structure-based :
  - Time-consuming.
  - Need to know the 3D structure of the target.

Chemogenomic framework

- Idea: mine the chemical space (small molecules) against the *whole* biological space (targets).
- Similar molecules bind similar targets.
- Advantage: ligand-based approaches on targets with no (or few) known ligands can take advantage of similar targets with known ligands.

Chemogenomics approach

Limits of the classical paradigms

- Ligand-based : need to know (enough) ligands of a given target to produce (accurate) predictors.
- Structure-based :
  - Time-consuming.
  - Need to know the 3D structure of the target.

Chemogenomic framework

- Idea: mine the chemical space (small molecules) against the *whole* biological space (targets).
- Similar molecules bind similar targets.
- Advantage: ligand-based approaches on targets with no (or few) known ligands can take advantage of similar targets with known ligands.

Chemogenomics approach

Limits of the classical paradigms

- Ligand-based : need to know (enough) ligands of a given target to produce (accurate) predictors.
- Structure-based :
  - Time-consuming.
  - Need to know the 3D structure of the target.

Chemogenomic framework

- Idea: mine the chemical space (small molecules) against the *whole* biological space (targets).
- Similar molecules bind similar targets.
- Advantage: ligand-based approaches on targets with no (or few) known ligands can take advantage of similar targets with known ligands.

Chemogenomics approach

Limits of the classical paradigms

- Ligand-based : need to know (enough) ligands of a given target to produce (accurate) predictors.
- Structure-based :
  - Time-consuming.
  - Need to know the 3D structure of the target.

Chemogenomic framework

- Idea: mine the chemical space (small molecules) against the *whole* biological space (targets).
- Similar molecules bind similar targets.
- Advantage: ligand-based approaches on targets with no (or few) known ligands can take advantage of similar targets with known ligands.

Chemogenomics approach

Limits of the classical paradigms

- Ligand-based : need to know (enough) ligands of a given target to produce (accurate) predictors.
- Structure-based :
  - Time-consuming.
  - Need to know the 3D structure of the target.

Chemogenomic framework

- Idea: mine the chemical space (small molecules) against the *whole* biological space (targets).
- Similar molecules bind similar targets.
- Advantage: ligand-based approaches on targets with no (or few) known ligands can take advantage of similar targets with known ligands.

Kernel methods for chemogenomics

- Cast the interaction problem in a joint ligand-target space (consider ligand-target pairs).
- Apply existing machine learning algorithms in this space.
- Similar to Bock and Gough (2005) and Erhan *et al.* (2006).
- Already applied to MHC-I-epitope binding prediction (Heckerman *et al.* 2005, Jacob *et al.* 2008).

Kernel methods for chemogenomics

- Cast the interaction problem in a joint ligand-target space (consider ligand-target pairs).
- Apply existing machine learning algorithms in this space.
- Similar to Bock and Gough (2005) and Erhan *et al.* (2006).
- Already applied to MHC-I-epitope binding prediction (Heckerman *et al.* 2005, Jacob *et al.* 2008).

Kernel methods for chemogenomics

- Cast the interaction problem in a joint ligand-target space (consider ligand-target pairs).
- Apply existing machine learning algorithms in this space.
- Similar to Bock and Gough (2005) and Erhan *et al.* (2006).
- Already applied to MHC-I-epitope binding prediction (Heckerman *et al.* 2005, Jacob *et al.* 2008).

Kernel methods for chemogenomics

- Cast the interaction problem in a joint ligand-target space (consider ligand-target pairs).
- Apply existing machine learning algorithms in this space.
- Similar to Bock and Gough (2005) and Erhan *et al.* (2006).
- Already applied to MHC-I-epitope binding prediction (Heckerman *et al.* 2005, Jacob *et al.* 2008).

Outline

1. Introduction

2. Method
   - Formalization
   - Representation of pairs
   - Kernel for ligands and targets

3. Results

4. Conclusion

Formalization

Single-target screening

- Target t with known ligands $c_1, \ldots, c_n$.
- For each t, learn a function $f_t(c)$ from the c_i that predicts if unseen candidate c is a ligand of t.
- Linear case: given a description $\Phi(c)$ of the molecule, $f_t(c) = w_t^\top \Phi(c)$.

Chemogenomics setting

- Consider training pairs $(t, c)_i$ (known to interact or not to interact), represented by vectors $\Phi\left((t, c)_i\right)$.
- Learn a single function $f(t, c) = w^\top \Phi(t, c)$ in the joint space to predict if the candidate pair (t, c) interacts.
- How to choose the pair representation Φ?

Formalization

Single-target screening

- Target t with known ligands $c_1, \ldots, c_n$.
- For each t, learn a function $f_t(c)$ from the c_i that predicts if unseen candidate c is a ligand of t.
- Linear case: given a description $\Phi(c)$ of the molecule, $f_t(c) = w_t^\top \Phi(c)$.

Chemogenomics setting

- Consider training pairs $(t, c)_i$ (known to interact or not to interact), represented by vectors $\Phi\left((t, c)_i\right)$.
- Learn a single function $f(t, c) = w^\top \Phi(t, c)$ in the joint space to predict if the candidate pair (t, c) interacts.
- How to choose the pair representation Φ?

Formalization

Single-target screening

- Target t with known ligands $c_1, \ldots, c_n$.
- For each t, learn a function $f_t(c)$ from the c_i that predicts if unseen candidate c is a ligand of t.
- Linear case: given a description $\Phi(c)$ of the molecule, $f_t(c) = w_t^\top \Phi(c)$.

Chemogenomics setting

- Consider training pairs $(t, c)_i$ (known to interact or not to interact), represented by vectors $\Phi\left((t, c)_i\right)$.
- Learn a single function $f(t, c) = w^\top \Phi(t, c)$ in the joint space to predict if the candidate pair (t, c) interacts.
- How to choose the pair representation Φ?

Formalization

Single-target screening

- Target t with known ligands $c_1, \ldots, c_n$.
- For each t, learn a function $f_t(c)$ from the c_i that predicts if unseen candidate c is a ligand of t.
- Linear case: given a description $\Phi(c)$ of the molecule, $f_t(c) = w_t^\top \Phi(c)$.

Chemogenomics setting

- Consider training pairs $(t, c)_i$ (known to interact or not to interact), represented by vectors $\Phi\left((t, c)_i\right)$.
- Learn a single function $f(t, c) = w^\top \Phi(t, c)$ in the joint space to predict if the candidate pair (t, c) interacts.
- How to choose the pair representation Φ?

Formalization

Single-target screening

- Target t with known ligands $c_1, \ldots, c_n$.
- For each t, learn a function $f_t(c)$ from the c_i that predicts if unseen candidate c is a ligand of t.
- Linear case: given a description $\Phi(c)$ of the molecule, $f_t(c) = w_t^\top \Phi(c)$.

Chemogenomics setting

- Consider training pairs $(t, c)_i$ (known to interact or not to interact), represented by vectors $\Phi\left((t, c)_i\right)$.
- Learn a single function $f(t, c) = w^\top \Phi(t, c)$ in the joint space to predict if the candidate pair (t, c) interacts.
- How to choose the pair representation Φ?

Formalization

Single-target screening

- Target t with known ligands $c_1, \ldots, c_n$.
- For each t, learn a function $f_t(c)$ from the c_i that predicts if unseen candidate c is a ligand of t.
- Linear case: given a description $\Phi(c)$ of the molecule, $f_t(c) = w_t^\top \Phi(c)$.

Chemogenomics setting

- Consider training pairs $(t, c)_i$ (known to interact or not to interact), represented by vectors $\Phi\left((t, c)_i\right)$.
- Learn a single function $f(t, c) = w^\top \Phi(t, c)$ in the joint space to predict if the candidate pair (t, c) interacts.
- How to choose the pair representation Φ?

Outline

1. Introduction

2. Method
   - Formalization
   - Representation of pairs
   - Kernel for ligands and targets

3. Results

4. Conclusion

Vector representation of pairs

Ligand representation

- A lot of existing work to represent a molecule t by a vector $\Phi_{ligand}(c) \in \mathbb{R}^{d_c}$.
- Physico-chemical, structural properties of the molecules.

Target representation

- Similarly, much work devoted to the construction of descriptors for a given protein t by a vector $\Phi_{target}(t) \in \mathbb{R}^{d_t}$.
- Properties of the sequence, structure of the protein.

Vector representation of pairs

Ligand representation

- A lot of existing work to represent a molecule t by a vector $\Phi_{ligand}(c) \in \mathbb{R}^{d_c}$.
- Physico-chemical, structural properties of the molecules.

Target representation

- Similarly, much work devoted to the construction of descriptors for a given protein t by a vector $\Phi_{target}(t) \in \mathbb{R}^{d_t}$.
- Properties of the sequence, structure of the protein.

Vector representation of pairs

Pair representation

- Use products of features of c and features of t.
- Idea (binary case): indicate that both c and t carry given features.
- May be strongly correlated with the fact that they interact.
- Set of all possible products of features of c and t is given by the tensor product:

$$\Phi(c,t) = \Phi_{ligand}(c) \otimes \Phi_{target}(t) . \tag{1}$$

- Potential issue: $d_c \times d_t$ vector, may be prohibitively large.

Vector representation of pairs

Pair representation

- Use products of features of c and features of t.
- Idea (binary case): indicate that both c and t carry given features.
- May be strongly correlated with the fact that they interact.
- Set of all possible products of features of c and t is given by the tensor product:

$$\Phi(c,t) = \Phi_{ligand}(c) \otimes \Phi_{target}(t)\,. \tag{1}$$

- Potential issue: $d_c \times d_t$ vector, may be prohibitively large.

Vector representation of pairs

Pair representation

- Use products of features of c and features of t.
- Idea (binary case): indicate that both c and t carry given features.
- May be strongly correlated with the fact that they interact.
- Set of all possible products of features of c and t is given by the tensor product:

$$\Phi(c,t) = \Phi_{ligand}(c) \otimes \Phi_{target}(t) \,. \tag{1}$$

- Potential issue: $d_c \times d_t$ vector, may be prohibitively large.

Vector representation of pairs

Pair representation

- Use products of features of c and features of t.
- Idea (binary case): indicate that both c and t carry given features.
- May be strongly correlated with the fact that they interact.
- Set of all possible products of features of c and t is given by the tensor product:

$$\Phi(c,t) = \Phi_{ligand}(c) \otimes \Phi_{target}(t)\,. \tag{1}$$

- Potential issue: $d_c \times d_t$ vector, may be prohibitively large.

Vector representation of pairs

Pair representation

- Use products of features of c and features of t.
- Idea (binary case): indicate that both c and t carry given features.
- May be strongly correlated with the fact that they interact.
- Set of all possible products of features of c and t is given by the tensor product:

$$\Phi(c,t) = \Phi_{ligand}(c) \otimes \Phi_{target}(t)\,. \tag{1}$$

- Potential issue: $d_c \times d_t$ vector, may be prohibitively large.

Kernels for ligand-target pairs

Kernel trick

- Need to compute efficiently the inner product between pairs.
- Classical property of tensor products :

$$\begin{aligned}\Phi(c,t)^\top \Phi(c',t') &= \left(\Phi_{lig}(c) \otimes \Phi_{tar}(t)\right)^\top \left(\Phi_{lig}(c') \otimes \Phi_{tar}(t')\right) \\ &= \Phi_{lig}(c)^\top \Phi_{lig}(c') \times \Phi_{tar}(t)^\top \Phi_{tar}(t') \,.\end{aligned}$$

More generally

Denoting

$$K_{lig}(c,c') = \Phi_{lig}(c)^\top \Phi_{lig}(c'), \quad K_{tar}(t,t') = \Phi_{tar}(t)^\top \Phi_{tar}(t') \,,$$

we obtain the inner product between tensor products by:

$$K\left((c,t),(c',t')\right) = K_{tar}(t,t') \times K_{lig}(c,c'). \tag{2}$$

Kernels for ligand-target pairs

Kernel trick

- Need to compute efficiently the inner product between pairs.
- Classical property of tensor products :

$$\begin{aligned}\Phi(c,t)^\top\Phi(c',t') &= (\Phi_{lig}(c)\otimes\Phi_{tar}(t))^\top\left(\Phi_{lig}(c')\otimes\Phi_{tar}(t')\right)\\ &= \Phi_{lig}(c)^\top\Phi_{lig}(c')\times\Phi_{tar}(t)^\top\Phi_{tar}(t')\,.\end{aligned}$$

More generally

Denoting

$$K_{lig}(c,c') = \Phi_{lig}(c)^\top\Phi_{lig}(c'),\quad K_{tar}(t,t') = \Phi_{tar}(t)^\top\Phi_{tar}(t')\,,$$

we obtain the inner product between tensor products by:

$$K\left((c,t),(c',t')\right) = K_{tar}(t,t')\times K_{lig}(c,c'). \tag{2}$$

Kernels for ligand-target pairs

Kernel trick

- Need to compute efficiently the inner product between pairs.
- Classical property of tensor products :

$$\begin{aligned}\Phi(c,t)^\top \Phi(c',t') &= \left(\Phi_{lig}(c)\otimes\Phi_{tar}(t)\right)^\top \left(\Phi_{lig}(c')\otimes\Phi_{tar}(t')\right)\\ &= \Phi_{lig}(c)^\top\Phi_{lig}(c')\times\Phi_{tar}(t)^\top\Phi_{tar}(t')\,.\end{aligned}$$

More generally

Denoting

$$K_{lig}(c,c') = \Phi_{lig}(c)^\top\Phi_{lig}(c'),\quad K_{tar}(t,t') = \Phi_{tar}(t)^\top\Phi_{tar}(t')\,,$$

we obtain the inner product between tensor products by:

$$K\left((c,t),(c',t')\right) = K_{tar}(t,t')\times K_{lig}(c,c'). \qquad (2)$$

Outline

1. Introduction

2. Method
    - Formalization
    - Representation of pairs
    - Kernel for ligands and targets

3. Results

4. Conclusion

Kernel for ligands

2D kernel for ligands

- We chose the *Tanimoto kernel* (state-of-the-art performances in general).
- Characterizes the molecules by the occurrences of linear subgraphs of length 8 or less.

Kernels for targets: non-informative approaches

Dirac kernel

$$K_{dirac}(t, t') = \delta(t, t').$$

Equivalent to performing independant learning for each target.

Multitask kernel

$$K_{multitask}(t, t') = K_{dirac}(t, t') + 1.$$

- Naive information sharing.
- Penalize individual norm and variance among individual functions.

Kernels for targets: non-informative approaches

Dirac kernel

$$K_{dirac}(t, t') = \delta(t, t').$$

Equivalent to performing independant learning for each target.

Multitask kernel

$$K_{multitask}(t, t') = K_{dirac}(t, t') + 1.$$

- Naive information sharing.
- Penalize individual norm and variance among individual functions.

Kernels for targets including biological information

Sequence-based kernels

Classical mismatch and local alignment kernel on whole sequences.

Hierarchy kernel

Use KEGG hierarchy between the targets: number of common ancestors in the corresponding hierarchy plus one.

Kernels for targets including biological information

Sequence-based kernels

Classical mismatch and local alignment kernel on whole sequences.

Hierarchy kernel

Use KEGG hierarchy between the targets: number of common ancestors in the corresponding hierarchy plus one.

KEGG Benchmark

Data generation

- KEGG data base (Kanehisa *et al.*, 2002).
- Ligand data for GPCR, enzymes and ion channels.
- For each positive pair, generate a negative ligand-target pair (same target, random ligand among existing ligands).

Final benchmark

- 2436 pairs for enzymes, 798 for GPCR and 2330 for ion channels.
- First experiment: 10-fold cross validation (assess the incidence of using ligands from other targets on the accuracy of the learned classifier for a given target).
- Second experiment: for each t learn a classifier using only interactions that do not involve t and test on the points that involve t (simulate the behavior when making predictions for orphan targets).

KEGG Benchmark

Data generation

- KEGG data base (Kanehisa *et al.*, 2002).
- Ligand data for GPCR, enzymes and ion channels.
- For each positive pair, generate a negative ligand-target pair (same target, random ligand among existing ligands).

Final benchmark

- 2436 pairs for enzymes, 798 for GPCR and 2330 for ion channels.
- First experiment: 10-fold cross validation (assess the incidence of using ligands from other targets on the accuracy of the learned classifier for a given target).
- Second experiment: for each t learn a classifier using only interactions that do not involve t and test on the points that involve t (simulate the behavior when making predictions for orphan targets).

KEGG Benchmark

Data generation

- KEGG data base (Kanehisa *et al.*, 2002).
- Ligand data for GPCR, enzymes and ion channels.
- For each positive pair, generate a negative ligand-target pair (same target, random ligand among existing ligands).

Final benchmark

- 2436 pairs for enzymes, 798 for GPCR and 2330 for ion channels.
- First experiment: 10-fold cross validation (assess the incidence of using ligands from other targets on the accuracy of the learned classifier for a given target).
- Second experiment: for each t learn a classifier using only interactions that do not involve t and test on the points that involve t (simulate the behavior when making predictions for orphan targets).

KEGG Benchmark

Data generation

- KEGG data base (Kanehisa *et al.*, 2002).
- Ligand data for GPCR, enzymes and ion channels.
- For each positive pair, generate a negative ligand-target pair (same target, random ligand among existing ligands).

Final benchmark

- 2436 pairs for enzymes, 798 for GPCR and 2330 for ion channels.
- First experiment: 10-fold cross validation (assess the incidence of using ligands from other targets on the accuracy of the learned classifier for a given target).
- Second experiment: for each t learn a classifier using only interactions that do not involve t and test on the points that involve t (simulate the behavior when making predictions for orphan targets).

KEGG Benchmark

Data generation

- KEGG data base (Kanehisa *et al.*, 2002).
- Ligand data for GPCR, enzymes and ion channels.
- For each positive pair, generate a negative ligand-target pair (same target, random ligand among existing ligands).

Final benchmark

- 2436 pairs for enzymes, 798 for GPCR and 2330 for ion channels.
- First experiment: 10-fold cross validation (assess the incidence of using ligands from other targets on the accuracy of the learned classifier for a given target).
- Second experiment: for each t learn a classifier using only interactions that do not involve t and test on the points that involve t (simulate the behavior when making predictions for orphan targets).

KEGG Benchmark

Data generation

- KEGG data base (Kanehisa *et al.*, 2002).
- Ligand data for GPCR, enzymes and ion channels.
- For each positive pair, generate a negative ligand-target pair (same target, random ligand among existing ligands).

Final benchmark

- 2436 pairs for enzymes, 798 for GPCR and 2330 for ion channels.
- First experiment: 10-fold cross validation (assess the incidence of using ligands from other targets on the accuracy of the learned classifier for a given target).
- Second experiment: for each t learn a classifier using only interactions that do not involve t and test on the points that involve t (simulate the behavior when making predictions for orphan targets).

First experiment

$K_{tar} \backslash$ Target	Enzymes	GPCR	Channels
Dirac	0.536 ± 0.005	0.682 ± 0.022	0.701 ± 0.017
multitask	0.874 ± 0.008	0.595 ± 0.030	0.797 ± 0.017
hierarchy	0.907 ± 0.008	0.817 ± 0.025	0.857 ± 0.015
local alignment	0.544 ± 0.007	0.696 ± 0.033	0.824 ± 0.015

Prediction accuracy for the first protocol on each dataset with various target kernels.

- Strong improvement when using data from the other targets.
- Stronger improvement when using prior information (hierarchy kernel).

First experiment

$K_{tar} \backslash$ Target	Enzymes	GPCR	Channels
Dirac	0.536 ± 0.005	0.682 ± 0.022	0.701 ± 0.017
multitask	0.874 ± 0.008	0.595 ± 0.030	0.797 ± 0.017
hierarchy	0.907 ± 0.008	0.817 ± 0.025	0.857 ± 0.015
local alignment	0.544 ± 0.007	0.696 ± 0.033	0.824 ± 0.015

Prediction accuracy for the first protocol on each dataset with various target kernels.

- Strong improvement when using data from the other targets.
- Stronger improvement when using prior information (hierarchy kernel).

Second experiment

$K_{tar} \backslash$ Target	Enzymes	GPCR	Channels
Dirac	0.500 ± 0.000	0.500 ± 0.000	0.500 ± 0.000
multitask	0.856 ± 0.009	0.477 ± 0.025	0.636 ± 0.021
hierarchy	0.862 ± 0.009	0.776 ± 0.026	0.805 ± 0.018
local alignment	0.521 ± 0.004	0.647 ± 0.030	0.722 ± 0.019

Prediction accuracy for the second protocol on each dataset with various target kernels.

Still possible to obtain reasonable results when no ligand is known for the target.

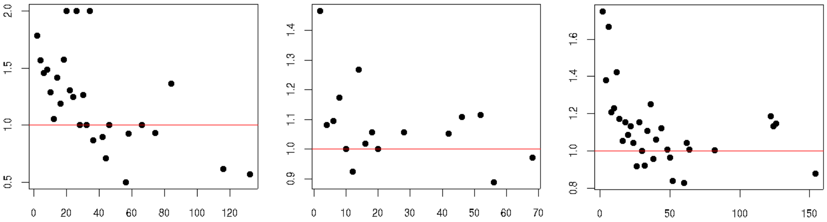
Second experiment

$K_{tar} \backslash$ Target	Enzymes	GPCR	Channels
Dirac	0.500 ± 0.000	0.500 ± 0.000	0.500 ± 0.000
multitask	0.856 ± 0.009	0.477 ± 0.025	0.636 ± 0.021
hierarchy	0.862 ± 0.009	0.776 ± 0.026	0.805 ± 0.018
local alignment	0.521 ± 0.004	0.647 ± 0.030	0.722 ± 0.019

Prediction accuracy for the second protocol on each dataset with various target kernels.

Still possible to obtain reasonable results when no ligand is known for the target.

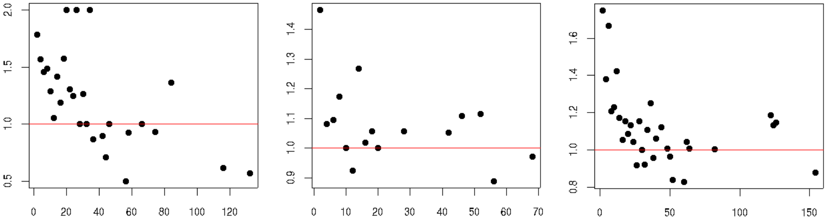
General behavior



Relative improvement of the *hierarchy* kernel against the *Dirac* kernel as a function of the number of known ligands for enzymes, GPCR and ion channel datasets.

- Strong improvement when few training points available.
- After a certain point, using similar targets can deteriorates accuracy.
- Suggests that the method could be improved by learning for each target independently how much information should be shared.

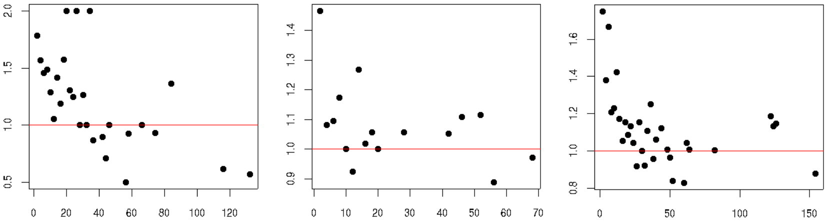
General behavior



Relative improvement of the *hierarchy* kernel against the *Dirac* kernel as a function of the number of known ligands for enzymes, GPCR and ion channel datasets.

- Strong improvement when few training points available.
- After a certain point, using similar targets can deteriorates accuracy.
- Suggests that the method could be improved by learning for each target independently how much information should be shared.

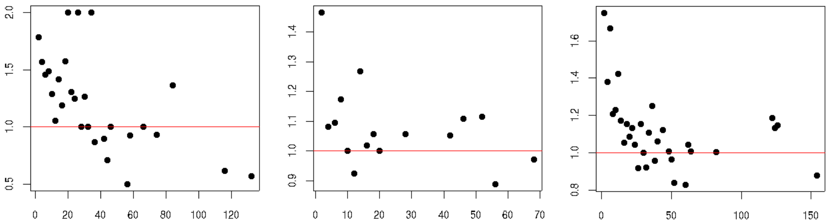
General behavior



Relative improvement of the *hierarchy* kernel against the *Dirac* kernel as a function of the number of known ligands for enzymes, GPCR and ion channel datasets.

- Strong improvement when few training points available.
- After a certain point, using similar targets can deteriorates accuracy.
- Suggests that the method could be improved by learning for each target independently how much information should be shared.

General behavior



Relative improvement of the *hierarchy* kernel against the *Dirac* kernel as a function of the number of known ligands for enzymes, GPCR and ion channel datasets.

- Strong improvement when few training points available.
- After a certain point, using similar targets can deteriorates accuracy.
- Suggests that the method could be improved by learning for each target independently how much information should be shared.

Conclusion

- General method to predict interaction between any chemical compound and any biological target.
- Using target kernels allowing to share information across the targets improves the prediction.
- Accuracy improvement depends on the number of known ligands.
- Possible improvements:
  - Other kernels (using molecule 3D information or sequence/structure of targets).
  - Adapt the amount of information sharing to each target.
  - Other regularizations.

Conclusion

- General method to predict interaction between any chemical compound and any biological target.
- Using target kernels allowing to share information across the targets improves the prediction.
- Accuracy improvement depends on the number of known ligands.
- Possible improvements:
  - Other kernels (using molecule 3D information or sequence/structure of targets).
  - Adapt the amount of information sharing to each target.
  - Other regularizations.

Conclusion

- General method to predict interaction between any chemical compound and any biological target.
- Using target kernels allowing to share information across the targets improves the prediction.
- Accuracy improvement depends on the number of known ligands.
- Possible improvements:
  - Other kernels (using molecule 3D information or sequence/structure of targets).
  - Adapt the amount of information sharing to each target.
  - Other regularizations.

Conclusion

- General method to predict interaction between any chemical compound and any biological target.
- Using target kernels allowing to share information across the targets improves the prediction.
- Accuracy improvement depends on the number of known ligands.
- Possible improvements:
  - Other kernels (using molecule 3D information or sequence/structure of targets).
  - Adapt the amount of information sharing to each target.
  - Other regularizations.