Aalto University
School of Science

Kernel-based pairwise learning of drug-protein binding affinities

Anna Cichońska

1) University of Turku
2) Helsinki Institute for Information Technology HIIT, Aalto University
3) Institute for Molecular Medicine Finland FIMM, University of Helsinki

anna.cichonska@aalto.fi

8 March 2019

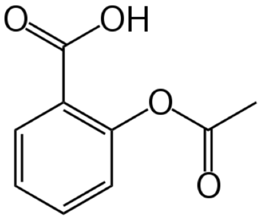
Drugs

Substances intended for use in the diagnosis, cure, mitigation, treatment or prevention of a disease.

Examples

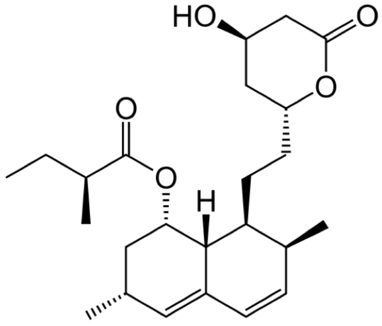
- **Aspirin** (acetylsalicylic acid)

  Treatment of pain, fever and inflammation.



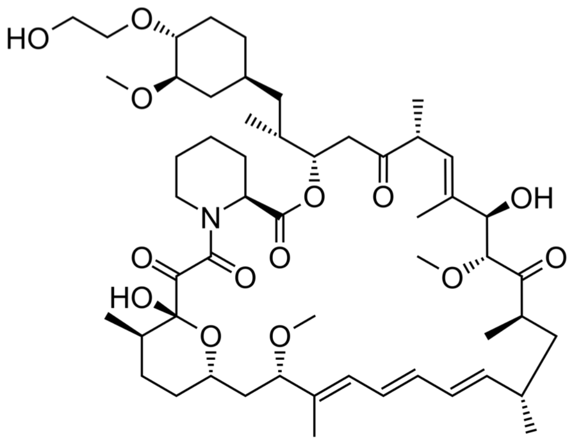
- **Lovastatin**

  Lowers cholesterol level.



- **Everolimus**

  Treatment of cancer, including cancer of the kidney, pancreas, breast, and brain. Used together with other drugs to keep the body from rejecting a transplanted kidney or liver.



Drug targets

- Drug-like chemical compounds execute their actions mainly by modulating cellular targets, including proteins, metabolites or even nucleic acids (DNA and RNA).

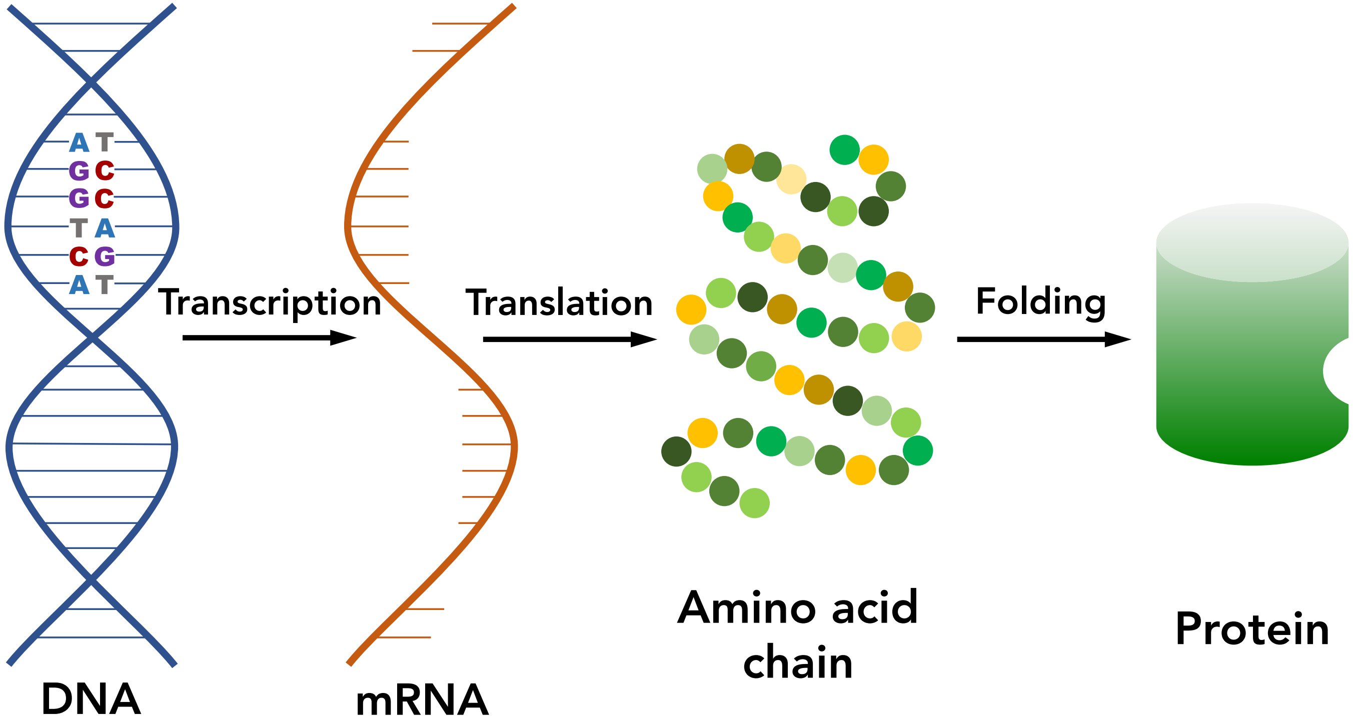
PROTEINS

- Most common drug targets.
- Large biomolecules.
- One of the most abundant molecules in living organisms.
- Perform a variety of important tasks, such as:
  - catalyzing chemical reactions (so called **enzymes**, e.g., kinases),
  - transporting other molecules,
  - identifying and neutralizing foreign particles,
  - providing structure and support for cells,

  & many more.

Drug targets: PROTEINS

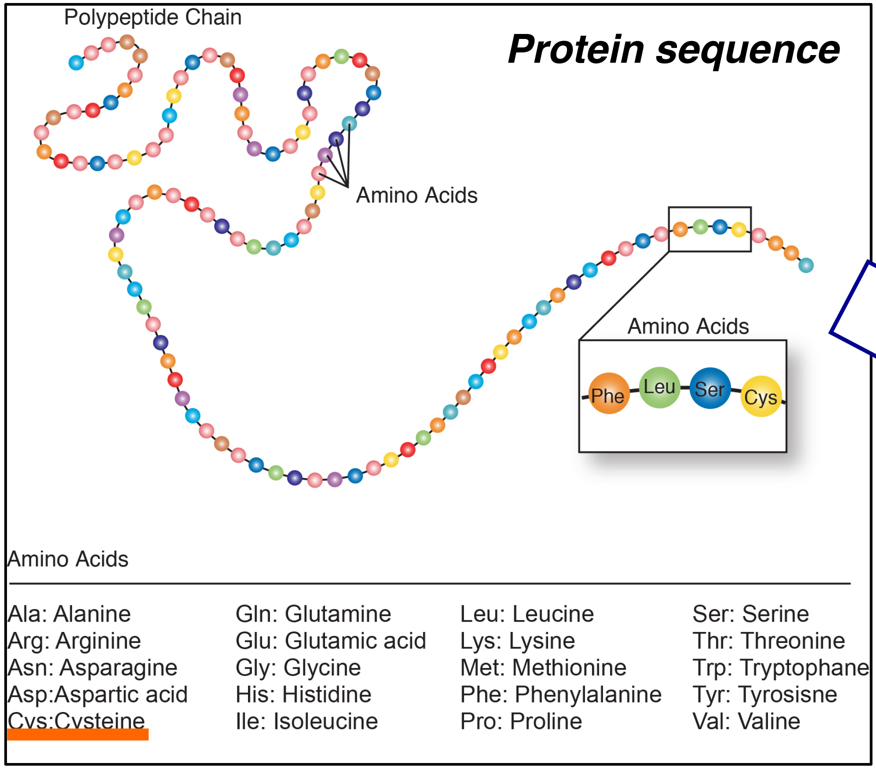
- Proteins are assembled from amino acids using information encoded in genes.



Gene expression begins with DNA and results in a protein.

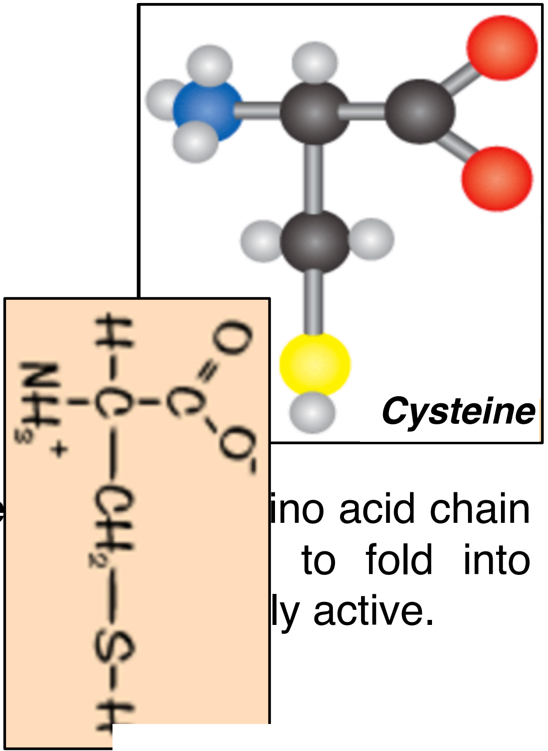
Drug targets: PROTEINS

- Proteins consist of one or more long chains of amino acid residues.
- In Eukaryotes, there are 21 proteinogenic amino acids.

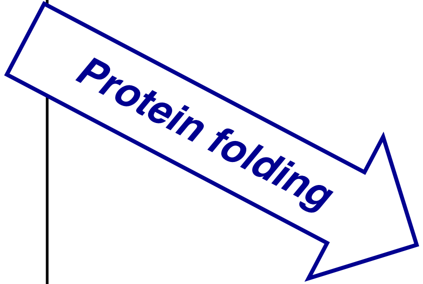


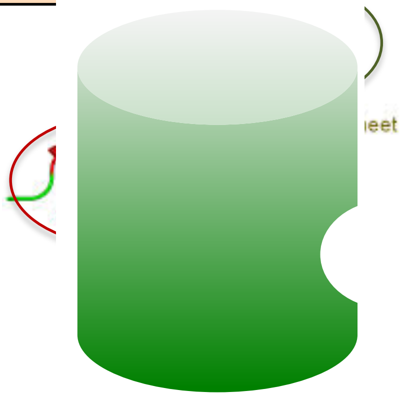
Amino Acids

Ala: Alanine	Gln: Glutamine	Leu: Leucine	Ser: Serine
Arg: Arginine	Glu: Glutamic acid	Lys: Lysine	Thr: Threonine
Asn: Asparagine	Gly: Glycine	Met: Methionine	Trp: Tryptophane
Asp:Aspartic acid	His: Histidine	Phe: Phenylalanine	Tyr: Tyrosisne
Cys:Cysteine	Ile: Isoleucine	Pro: Proline	Val: Valine



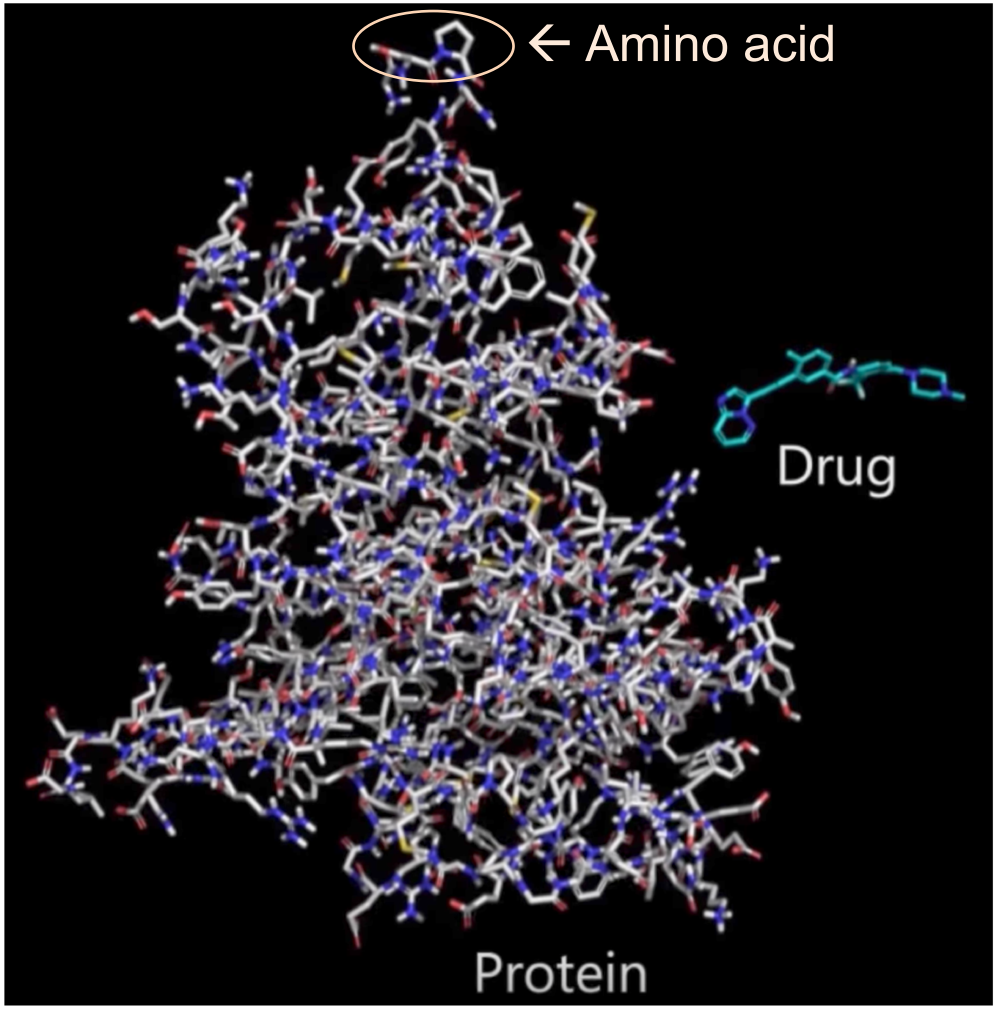
- The seque ino acid chain causes th to fold into a shape th ly active.





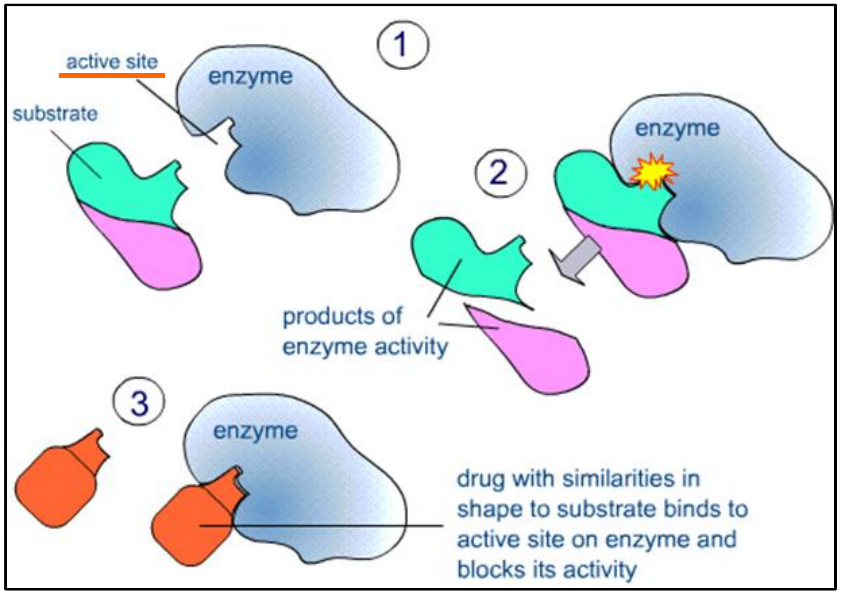
Protein structure

Drug targets: PROTEINS

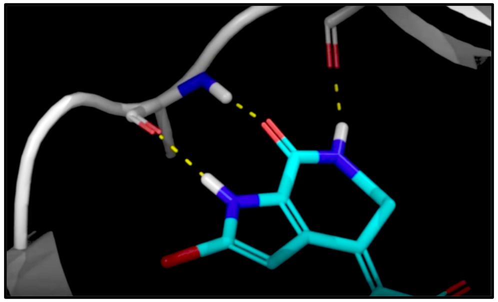


Drug-protein interaction (DPI)

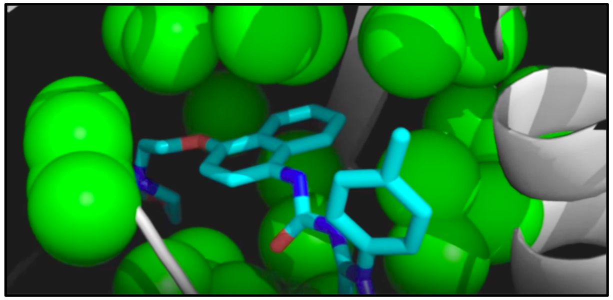
One of the most common drug's mechanism of action (MoA)


active site
enzyme
1
substrate
enzyme
2
products of enzyme activity
3
enzyme
drug with similarities in shape to substrate binds to active site on enzyme and blocks its activity

Drug-protein interaction = **molecular-level interaction**, e.g.,



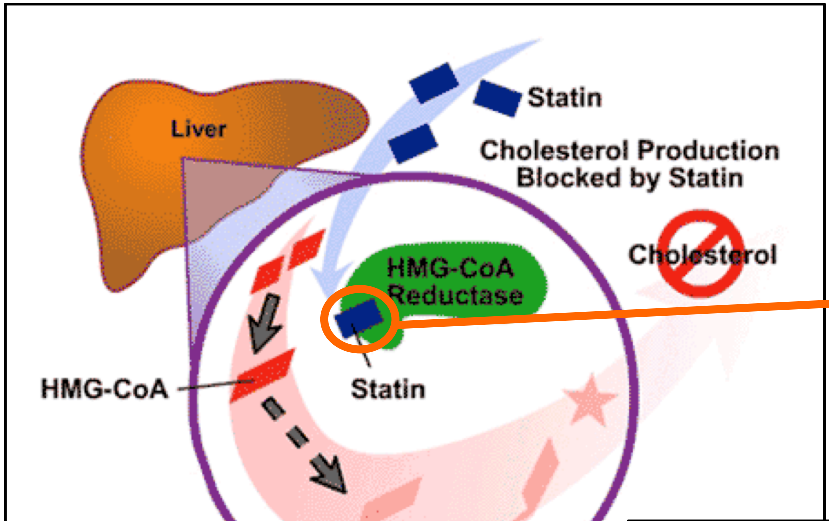
hydrogen bonds keep a **compound** tightly bound to a protein,...

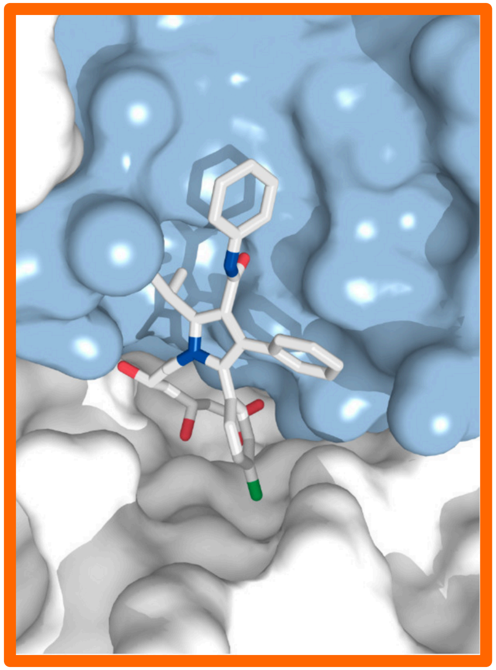


hydrophobic **protein atoms** enclose hydrophobic **compound atoms**.

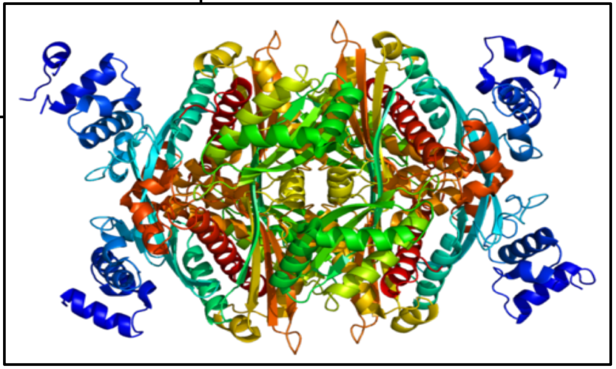
Drug-protein interaction

Statin–*HMG-CoA reductase*





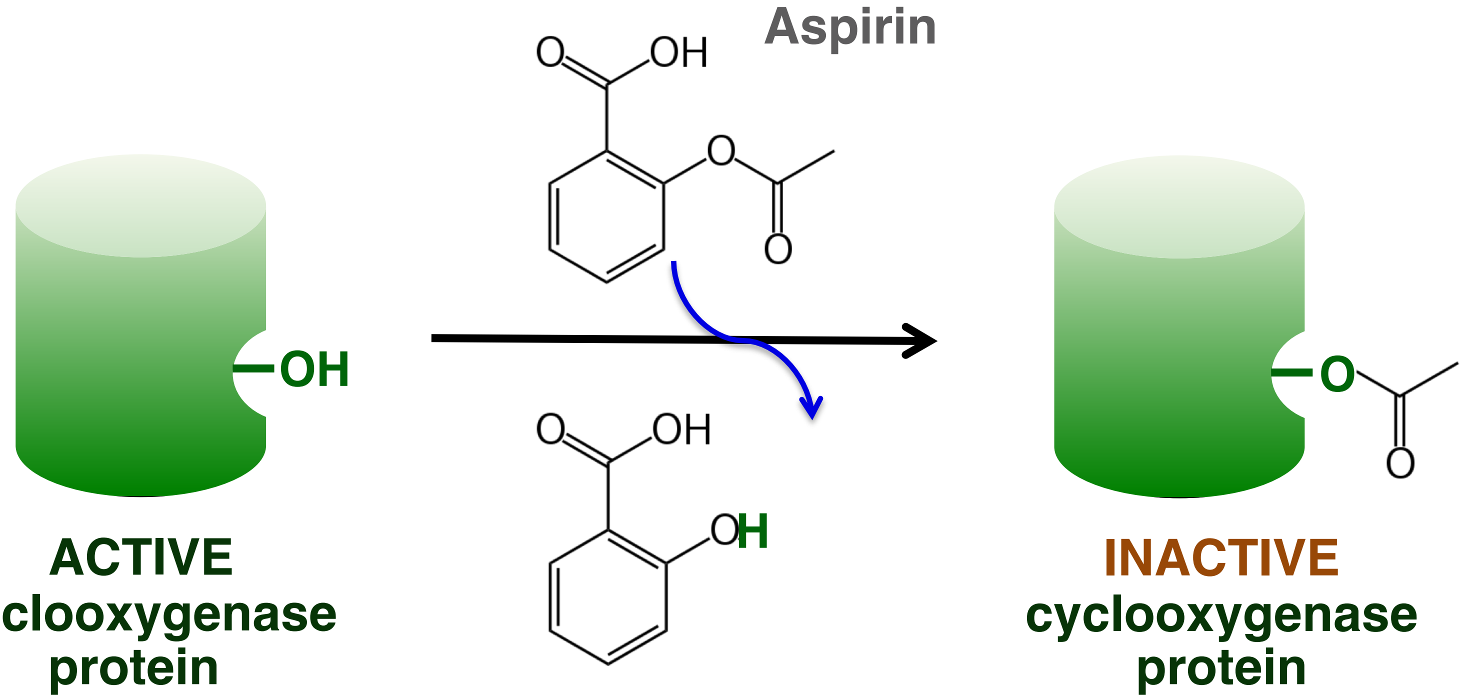
Inhibition of HMG-CoA reductase enzyme with statin



HMG-CoA reductase

Drug-protein interaction

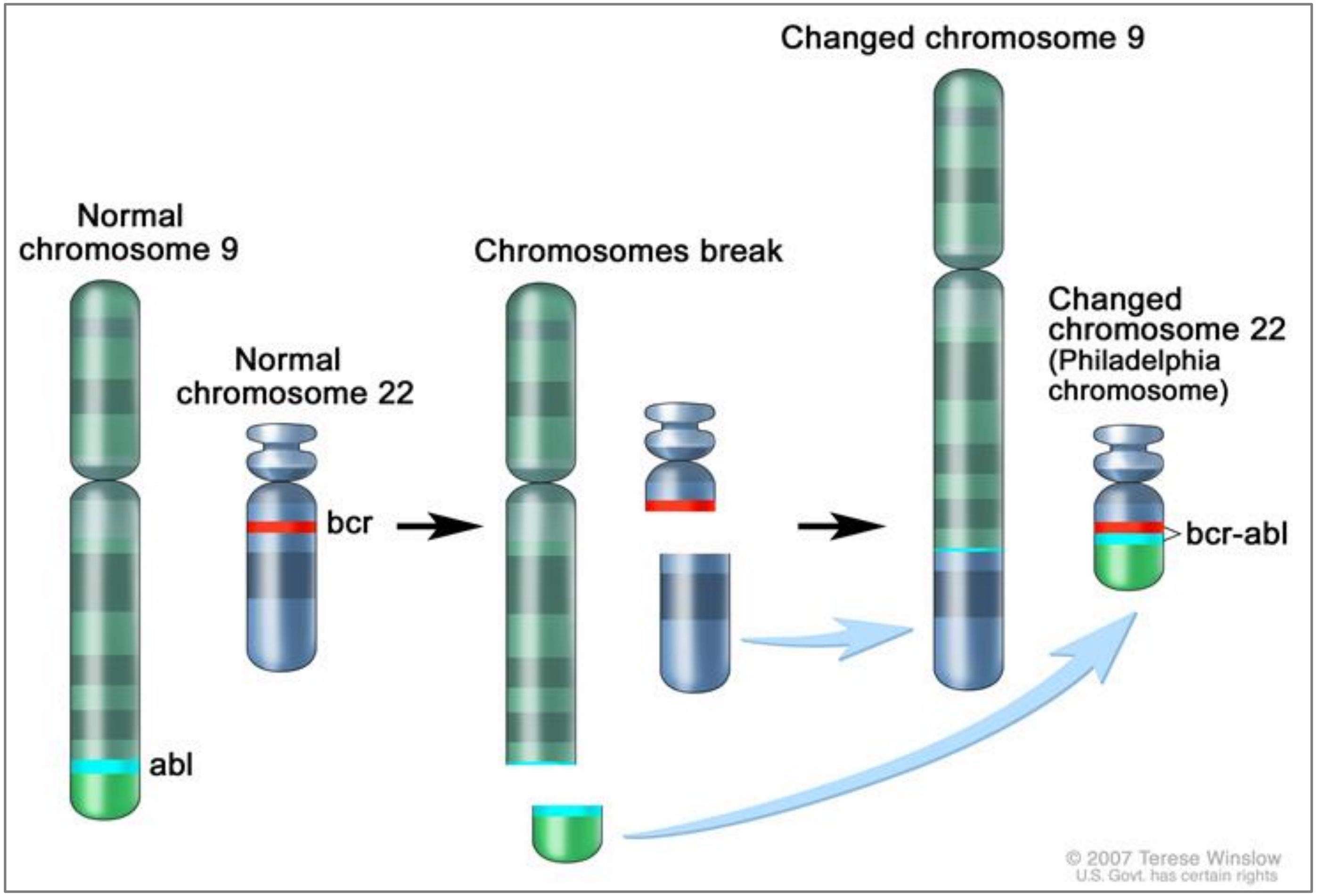
Aspirin–Cyclooxygenase



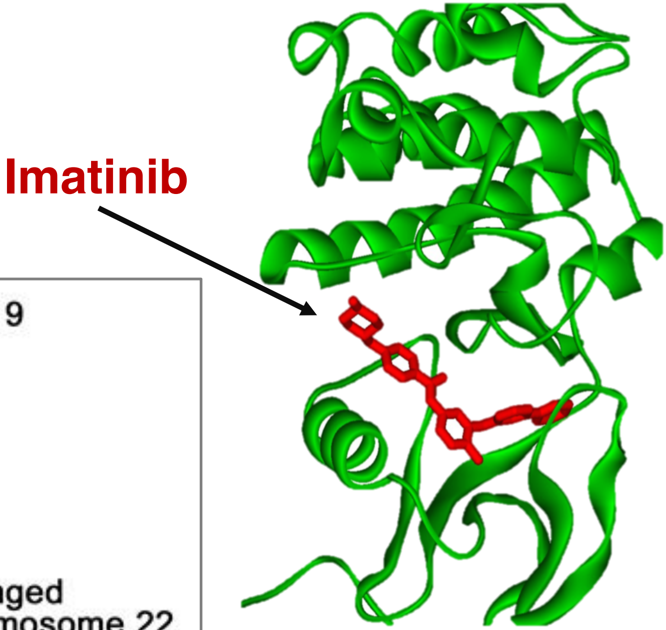
Responsible for the production of hormones causing, among others, inflammation, swelling, fever and pain.

Drug-protein interaction

Imatinib–BCR-ABL



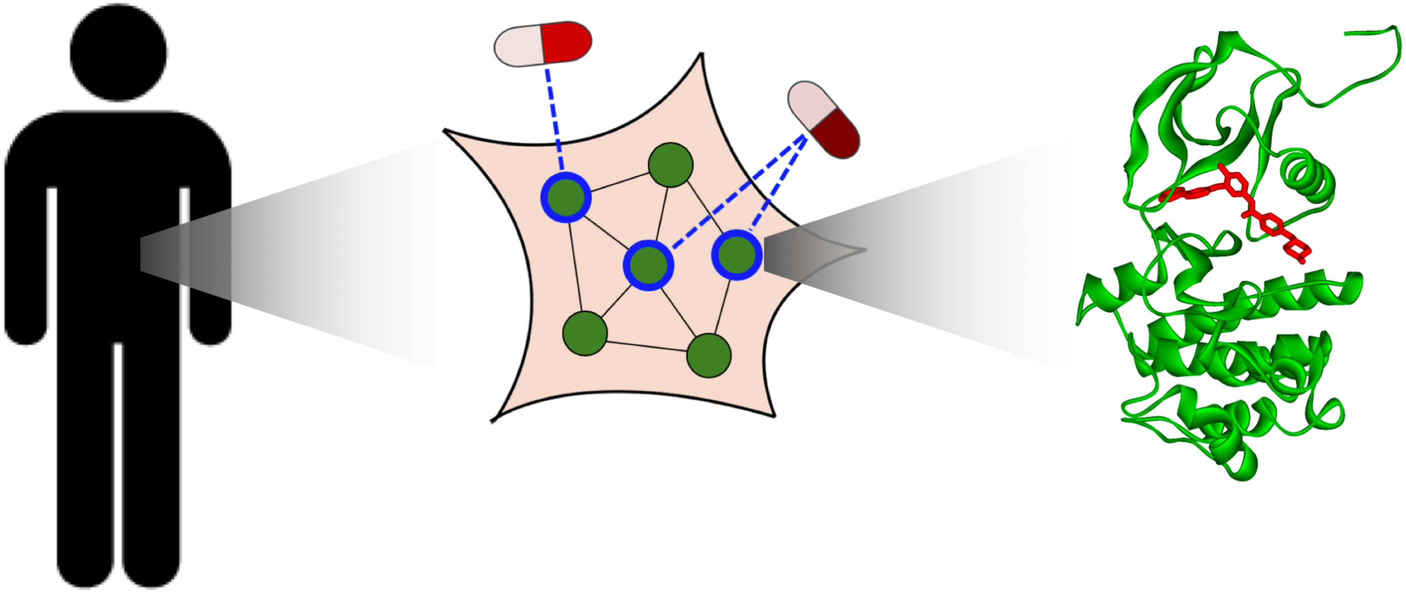
© 2007 Terese Winslow
U.S. Govt. has certain rights



Off-target interactions

- **Neutral**
- **Negative**
  - Imatinib–c-ABL → cardiotoxic side effects.
- **Positive**
  - Imatinib–KIT → treatment of gastrointestinal cancer.

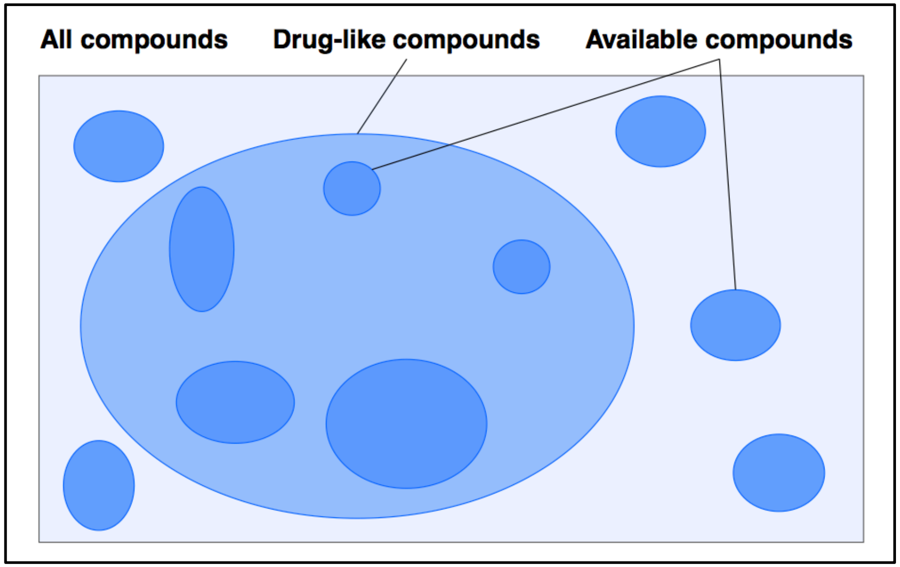
Drug-protein interaction mapping



- Expensive
- Time consuming

Enormous chemical universe

CHEMICAL SPACE



Only certain molecules have features consistent with good pharmacological properties (e.g. *Lipinski's rule of five*).

10^{20} - 10^{24} !

Lipinski's rule of five states that, in general, an orally-absorbed drug has no more than one violation of the following criteria:

- no more than 5 hydrogen bond donors;
- no more than 10 hydrogen bond acceptors;
- a molecular weight lower than 500 daltons;
- an octanol-water partition coefficient log*P* (a measure of lipophilicity) not greater than 5.

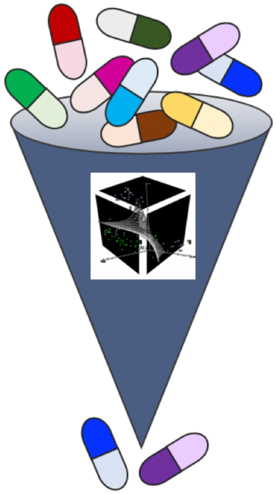
Note that the name of the rule originates from the fact that the cut-offs for all parameters are close to 5 or a multiple of 5.

Motivation for computational methods in drug discovery

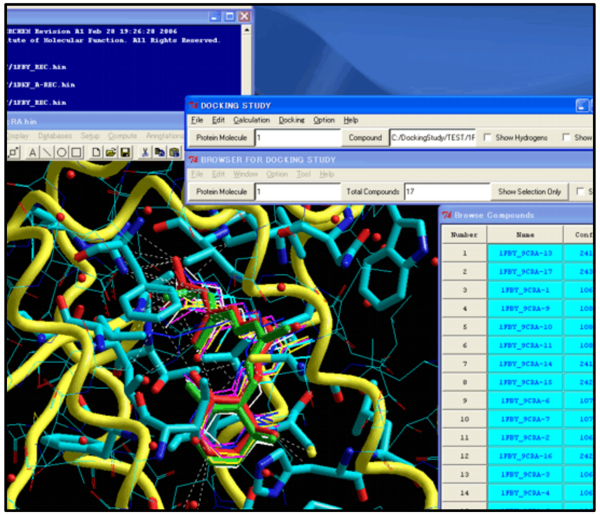
- Experimental drug-protein interaction mapping is time consuming and expensive.

- Moreover, it is simply infeasible to determine all the possible drug-protein interactions in the laboratory (10^{20} - 10^{24} drug-like compounds!)

- The hypothesis is that computational models could provide fast, large-scale and systematic pre-screening of chemical probes, toward prioritization of the most potent interactions for further *in vitro* or *ex vivo* verification in the laboratory.



In silico drug screening


DOCKING STUDY
File Edit Calculation Docking Option Help
Protein Molecule
Compound
Show Hydrogens
Show
BROWSER FOR DOCKING STUDY
File Edit Window Option Tool Help
Protein Molecule
Total Compounds
Show Selection Only
Browse Compounds
RA.hin
Display Databases Setup Compute Annotations

```
RCHEM Revision A1 Feb 28 19:26:28 2006
tute of Molecular Function. All Rights Reserved.

/1FBY_REC.hin

/1DKF_A-REC.hin

/1FBY_REC.hin
```

Protein Molecule: 1

Compound: C:/DockingStudy/TEST/1F

Protein Molecule: 1

Total Compounds: 17

Number	Name	Conf
1	1FBY_9CRA-13	241
2	1FBY_9CRA-17	243
3	1FBY_9CRA-1	106
4	1FBY_9CRA-9	108
5	1FBY_9CRA-10	108
6	1FBY_9CRA-11	108
7	1FBY_9CRA-14	241
8	1FBY_9CRA-15	242
9	1FBY_9CRA-6	107
10	1FBY_9CRA-7	107
11	1FBY_9CRA-2	106
12	1FBY_9CRA-16	242
13	1FBY_9CRA-3	106
14	1FBY_9CRA-4	106

➢ Docking Simulations

- Finding preferred orientation of one molecule to a second one when bound to each other to form a stable complex.
- **DOCK**: first docking program by Kuntz *et al.* (1982).
- Some algorithms:
  - Fragment-based methods: **FlexX**, **DOCK** (since version 4.0);
  - Monte Carlo/Simulated annealing: **QXP(Flo)**, **Autodock**, **Affinity** & **LigandFit** (Accelrys);
  - Genetic algorithms: **GOLD**, **AutoDock** (since version 3.0);
  - Systematic search: **FRED** (OpenEye), **Glide** (Schrödinger).
- Very accurate but slow.
- Require the usage of 3D molecular structures.

In silico drug screening

➢ **Machine Learning**

- Less accurate but orders of magnitude faster than docking simulations; thus, more preferable for early-stage *in silico* drugs screening.
- The objective is to derive rules from the existing bioactivity data (a phase of learning from training data) in order to build predictive models that can be then applied to infer unmeasured drug-protein binding affinities (prediction phase).
- **Drug-based methods** (quantitative structure-activity relationship QSAR models)
  Models trained using available bioactivity data + drug information.
- **Protein-based methods**
  Models trained using available bioactivity data + protein information.
- **Systems-based methods** (proteochemometric models, pairwise models)
  Models trained using available bioactivity data + both drug and protein information.
  Assumption: similar drugs are likely to interact with similar proteins.

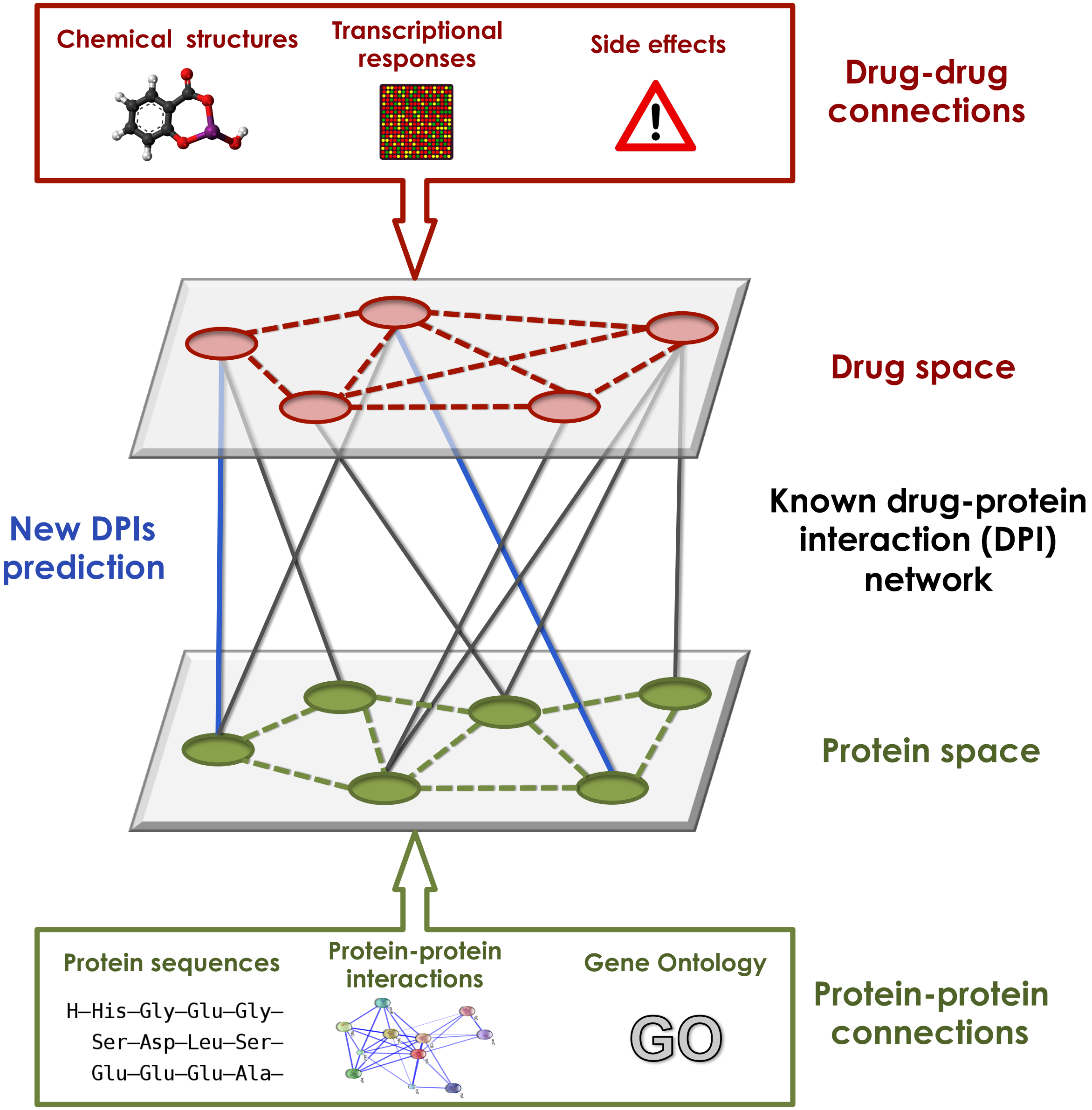
Systems-based DPI prediction methods

- Classification

  interaction/no interaction

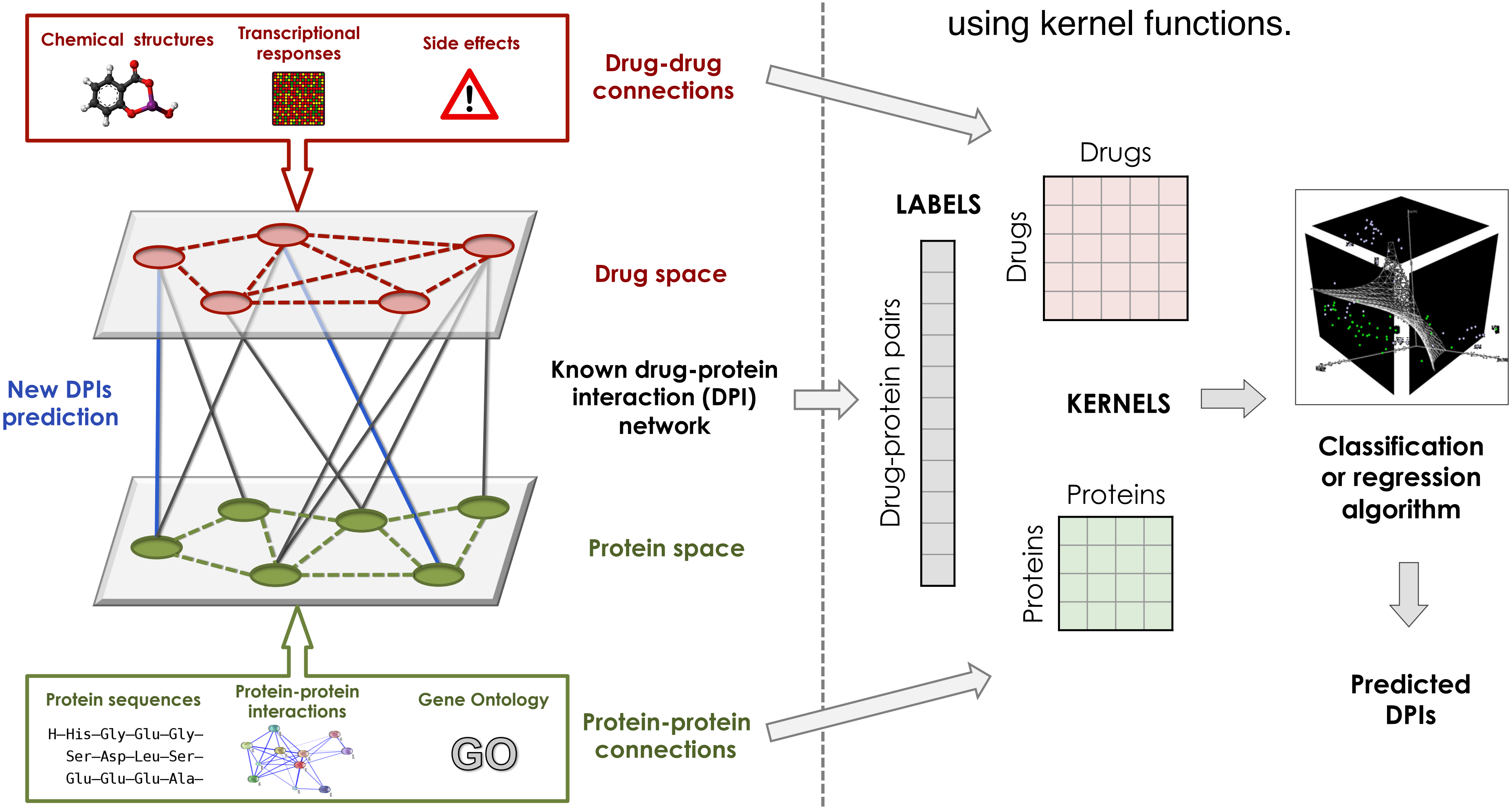
- Regression

  quantitative binding affinity



Cichonska A *et al*. (2015) "Identification of drug candidates and repurposing opportunities through compound-target interaction networks". Expert Opinion on Drug Discovery.

Similarities between molecules can be encoded using kernel functions.



Chemical structures

Transcriptional responses

Side effects

Drug-drug connections

Drug space

New DPIs prediction

Known drug-protein interaction (DPI) network

Protein space

Protein sequences

Protein-protein interactions

Gene Ontology

H–His–Gly–Glu–Gly–
Ser–Asp–Leu–Ser–
Glu–Glu–Glu–Ala–

GO

Protein-protein connections

LABELS

Drug-protein pairs

Drugs

Drugs

KERNELS

Proteins

Proteins

Classification or regression algorithm

Predicted DPIs

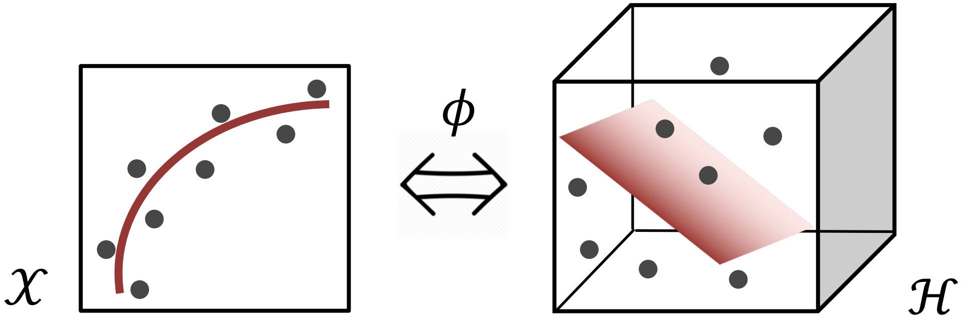
Kernels

- Kernels allow modelling **nonlinearities** in the data using well-established linear learning algorithms (in a computationally efficient manner).

- Formally, a kernel is a function that for all instances **x**, **z** ∈ $\mathcal{X}$ (e.g. drugs) satisfies

$$k(\mathbf{x}, \mathbf{z}) = \langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle,$$

where ϕ denotes the mapping from the input space $\mathcal{X}$ to an inner product high-dimensional feature space $\mathcal{H}$.

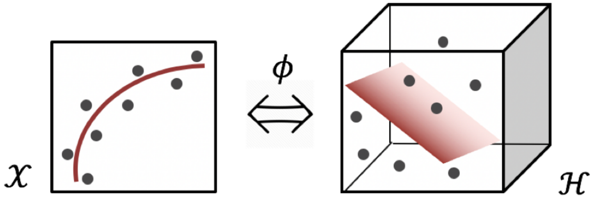


Example

$$k(\mathbf{x}, \mathbf{z}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{z}\|^2}{2\sigma^2}\right)$$

Kernels

$$k(\mathbf{x}, \mathbf{z}) = \langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle$$



- **Kernel trick**

  It is possible to avoid the explicit computation of the mapping ϕ and define the kernel directly in terms of the original input features by replacing the inner product $\langle \cdot, \cdot \rangle$ with an appropriately chosen kernel function.

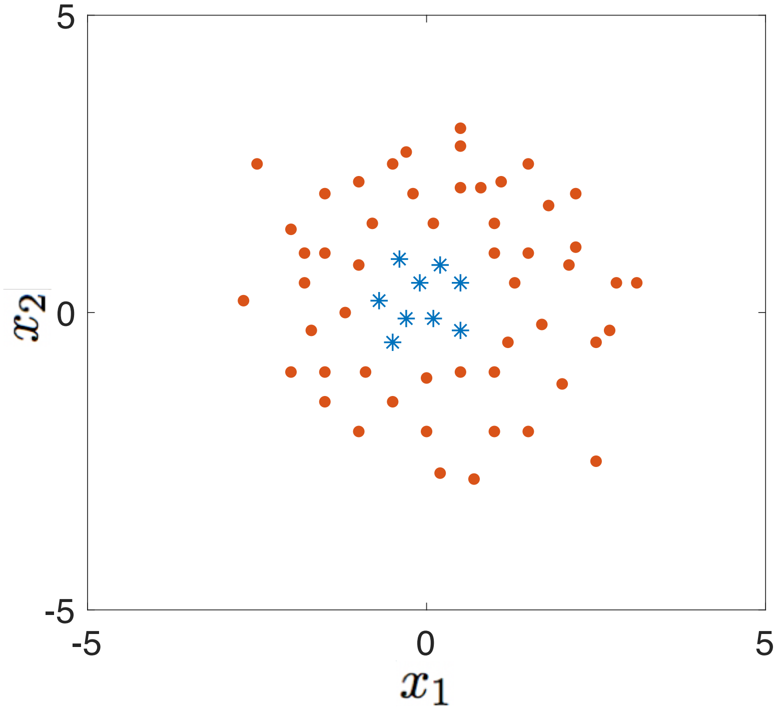
Example. Consider a two-dimensional input space together with the feature map:

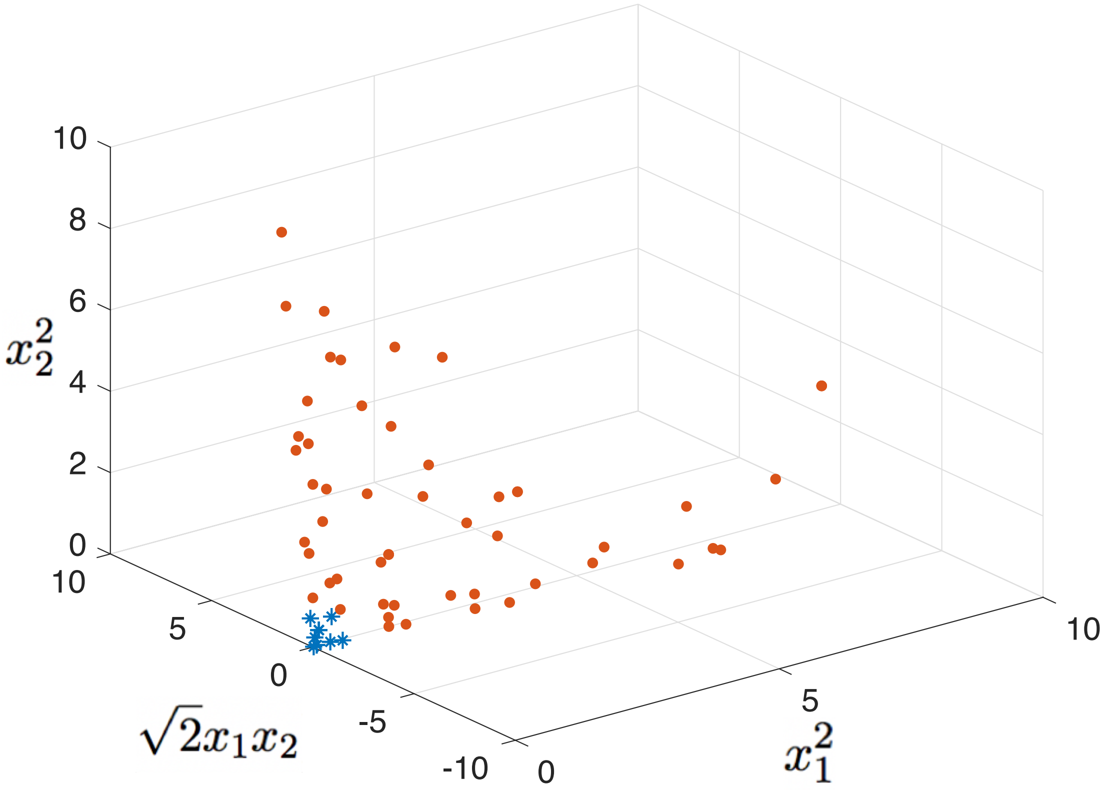
$$\mathbf{x} = (x_1, x_2) \longmapsto \phi(\mathbf{x}) = (x_1^2, x_2^2, \sqrt{2}x_1x_2)$$

$$\begin{aligned} \langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle &= \left\langle (x_1^2, x_2^2, \sqrt{2}x_1x_2), (z_1^2, z_2^2, \sqrt{2}z_1z_2) \right\rangle \\ &= x_1^2z_1^2 + x_2^2z_2^2 + 2x_1x_2z_1z_2 \\ &= (x_1z_1 + x_2z_2)^2 = \langle \mathbf{x}, \mathbf{z} \rangle^2 . \end{aligned}$$

$$\boxed{\kappa(\mathbf{x}, \mathbf{z}) = \langle \mathbf{x}, \mathbf{z} \rangle^2}$$

Kernels





$$\mathbf{x} = (x_1, x_2) \longmapsto \boldsymbol{\phi}(\mathbf{x}) = (x_1^2, x_2^2, \sqrt{2}x_1x_2)$$

Kernels

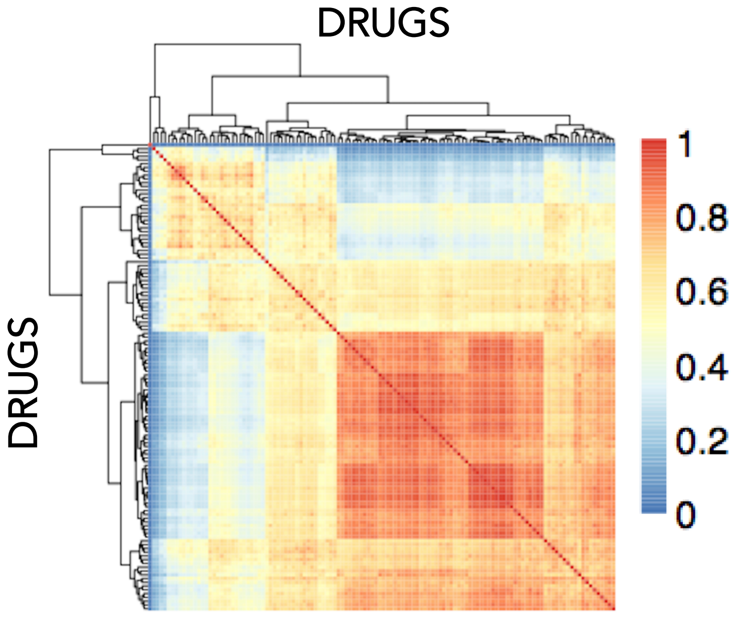
- Kernels address the challenge of

  #instances (e.g. drugs) ≪ **#features** (e.g. various chemical properties)

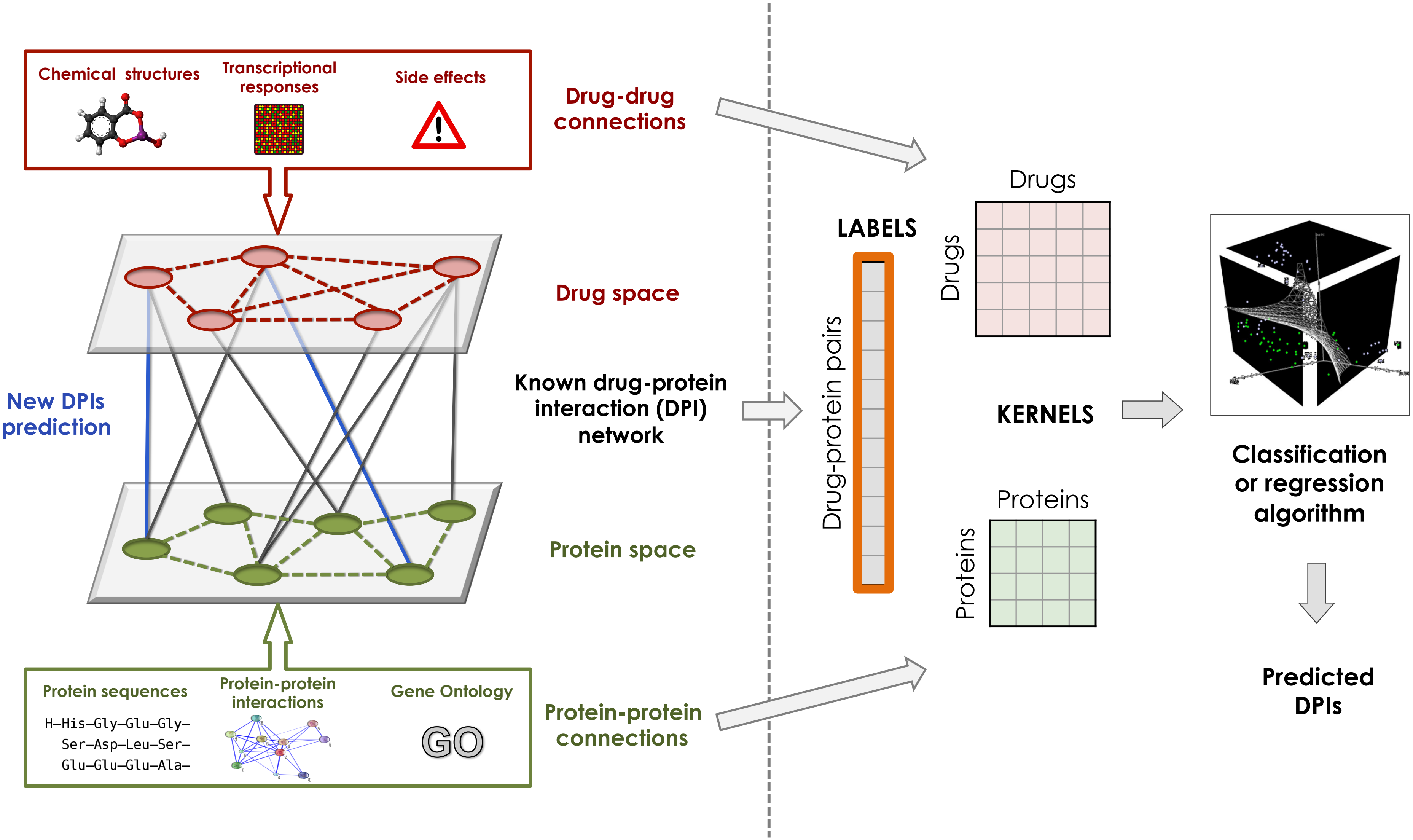
  → data appears only through the entries in the kernel matrix relating all pairs of instances.

- Kernels are well-suited for representing structured objects, such as molecules, that cannot always be accurately described by a standard feature vector.

- Kernel can be considered as a **similarity measure** between input instances.



Kernel matrix



Chemical structures

Transcriptional responses

Side effects

Drug-drug connections

Drug space

New DPIs prediction

Known drug-protein interaction (DPI) network

Protein space

Protein sequences

H–His–Gly–Glu–Gly–
Ser–Asp–Leu–Ser–
Glu–Glu–Glu–Ala–

Protein-protein interactions

Gene Ontology

GO

Protein-protein connections

LABELS

Drug-protein pairs

Drugs

Drugs

KERNELS

Proteins

Proteins

Classification or regression algorithm

Predicted DPIs

Known interactions – bioactivity databases

https://www.ebi.ac.uk/chembl/

- Searchable and downloadable.
- Data manually extracted from the literature.
- Target Report Card, Compound Report Card.
- ~2.3 mln compounds in ChEMBL 24.

https://pubchem.ncbi.nlm.nih.gov/

- Data generators deposit their data.
- Incorporates data from other databases, e.g. ChEMBL.
- Contains data on ~40 mln compounds.

http://www.drugbank.ca/

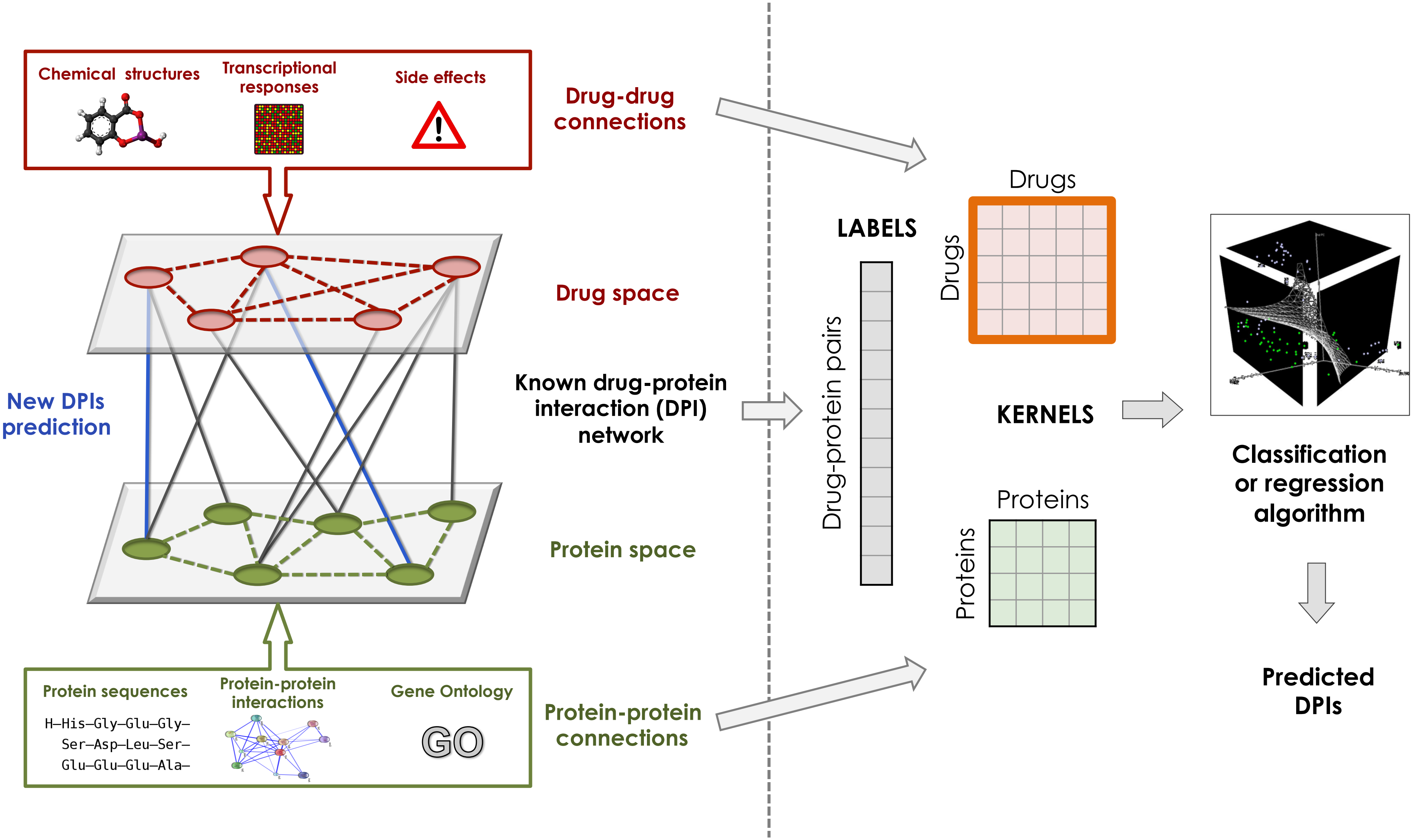
- 12 065 compounds.
- Does not contain strictly bioactivity data.
- Pharmacokinetics.

Known interactions – bioactivity databases

DRUGTARGET COMMONS

https://drugtargetcommons.fimm.fi/

- **Crowd-sourcing platform** to improve the consensus and use of drug-target interactions.
- The end users can search, view and download bioactivity data using various compound, target and publications identifiers.
- Expert users may also submit suggestions to **edit and upload new bioactivity data**, as well as participate in the **assay annotation** and **data curation** processes.



Chemical structures

Transcriptional responses

Side effects

Drug-drug connections

Drug space

New DPIs prediction

Known drug-protein interaction (DPI) network

Protein space

Protein sequences

H–His–Gly–Glu–Gly–
Ser–Asp–Leu–Ser–
Glu–Glu–Glu–Ala–

Protein-protein interactions

Gene Ontology

GO

Protein-protein connections

LABELS

Drug-protein pairs

Drugs

Drugs

KERNELS

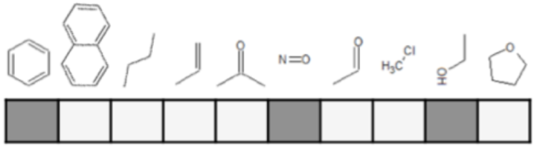
Proteins

Proteins

Classification or regression algorithm

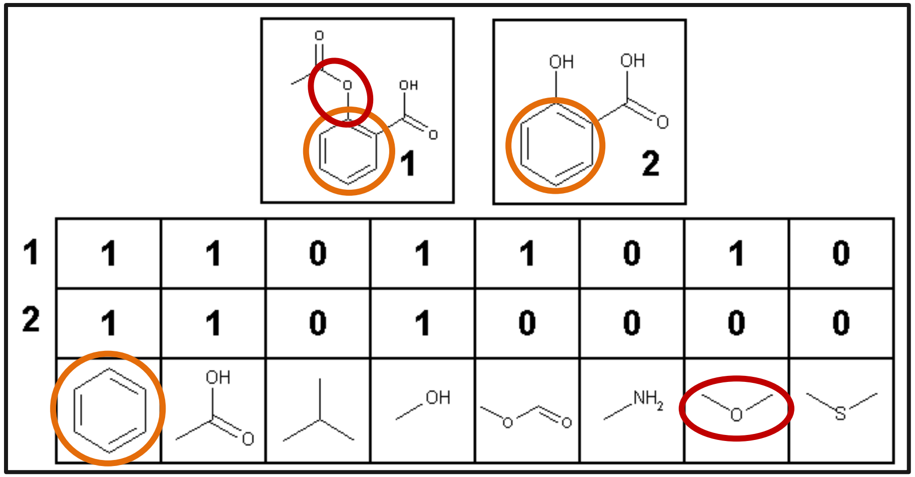
Predicted DPIs

Molecular fingerprint



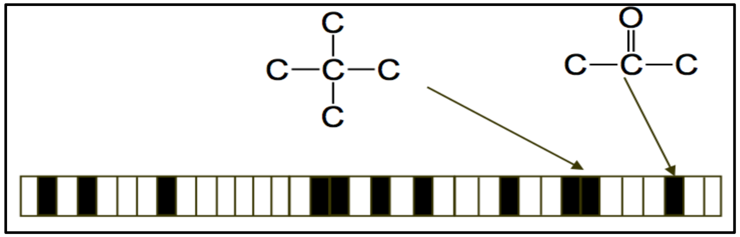
- A way of encoding the structure of a molecule.
- The most common type of fingerprint is a series of binary digits (bits) that represent the presence or absence of particular substructures in the molecule.

Example



1	1	1	0	1	1	0	1	0
2	1	1	0	1	0	0	0	0

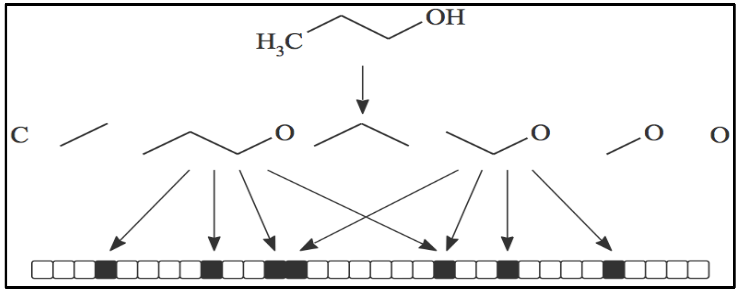
2D fingerprints



Dictionary-Based Fingerprints

Pre-defined fragments, each of which maps to a single bit.

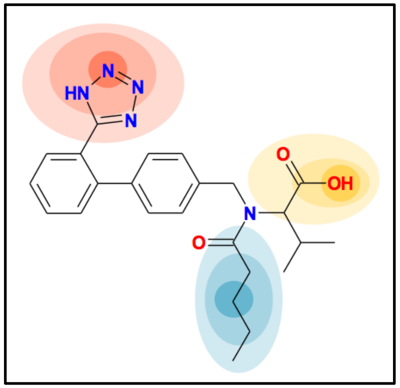
Examples: MACCS Keys, BCI.



Path-Based Hashed Fingerprints

Fragments are generated algorithmically without the need for a dictionary, e.g., all paths up to seven non-hydrogen atoms from the source atom.

Examples: Daylight, UNITY fingerprints.

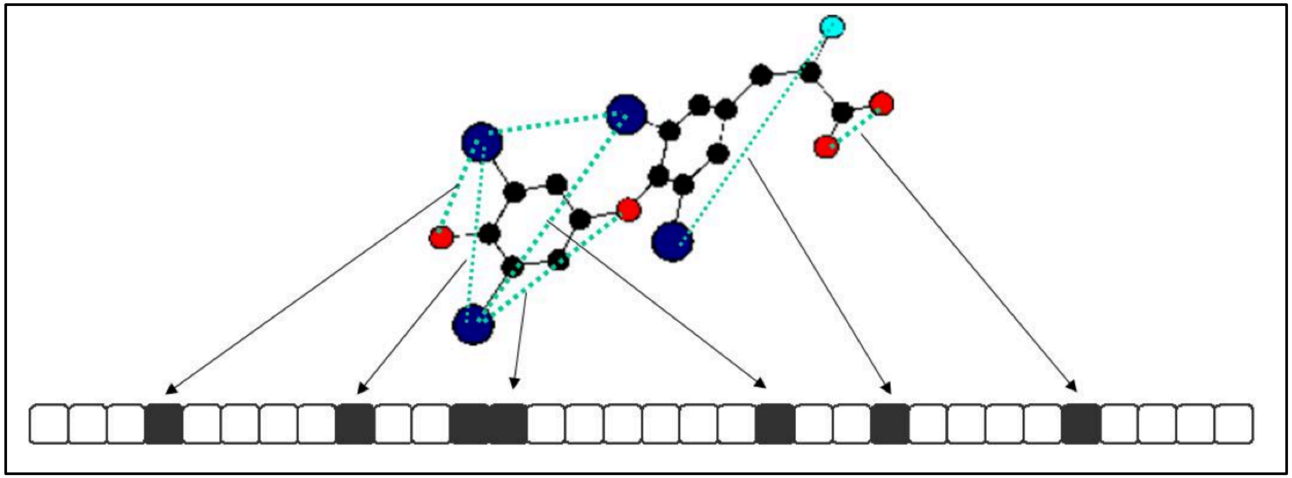


Circular Hashed Fingerprints

Each atom is represented together with its environment (neighbouring atoms as extended spheres).

Examples: ECFP2, ECFP4.

3D fingerprints



Presence or absence of geometric features, e.g., pairs/triplets of atoms at given distance, valence/torsion angles.

Fingerprint-based Tanimoto kernel

$$K(fp_1, fp_2) = \frac{N_{fp_1,fp_2}}{N_{fp_1} + N_{fp_2} - N_{fp_1,fp_2}}$$

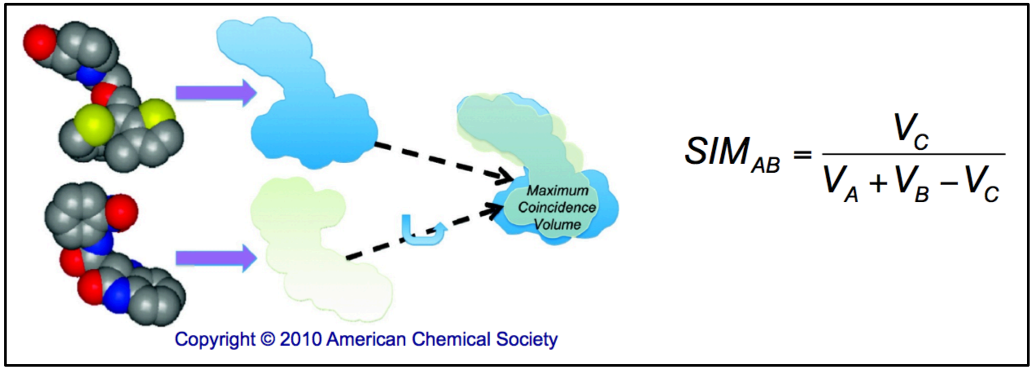
fp_i – fingerprint of the molecule i,

N_{fp_i} – number of 1-bits in the fingerprint fp_i,

N_{fp_1,fp_2} – number of 1-bits in both fingerprints.

- Computed based on the size of common substructures of the molecules represented by the fingerprints.

3D shape-based comparison

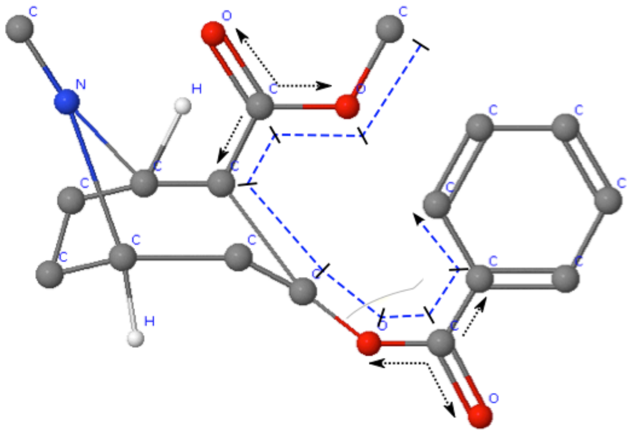


$$SIM_{AB} = \frac{V_C}{V_A + V_B - V_C}$$

Copyright © 2010 American Chemical Society

- Atoms are represented as Gaussian functions.
- Molecules are aligned in 3D.
- Similarity score is based on the common volume.

Graph kernel



- Graph kernels allow to measure the similarity between graphs.
- Chemical molecule can be represented as a labeled or unlabeled graph, where a node corresponds to an atom, and an edge indicates a bond between two atoms.

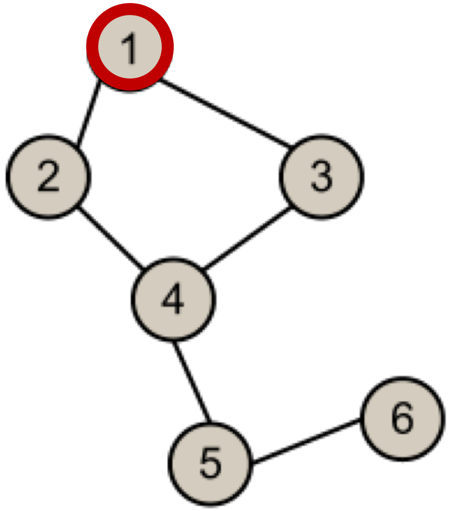
- Graph kernels can be roughly categorized into three main groups:
  1) graph kernels based on walks and paths,
  2) graph kernels based on limited-size subgraphs,
  3) graph kernels based on subtree patterns.

- **Examples**: random walk kernel, shortest-path kernel, Weisfeiler-Lehman subtree kernel.

Graph

- A graph *G* is a set of nodes (vertices) *V* and edges *E*.

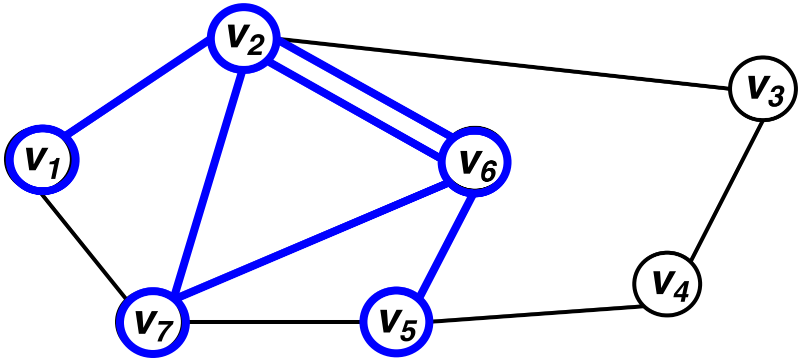
- The adjacency matrix **A** of *G* is defined as $[A]_{ij} = \begin{cases} 1 & \text{if } (v_i, v_j) \in E \\ 0 & \text{otherwise} \end{cases}$.



$$\begin{array}{c|cccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 \\
\hline
1 & 0 & 1 & 1 & 0 & 0 & 0 \\
2 & 1 & 0 & 0 & 1 & 0 & 0 \\
3 & 1 & 0 & 0 & 1 & 0 & 0 \\
4 & 0 & 1 & 1 & 0 & 1 & 0 \\
5 & 0 & 0 & 0 & 1 & 0 & 1 \\
6 & 0 & 0 & 0 & 0 & 1 & 0
\end{array}$$

Random walk

- A graph *G* is a set of nodes (vertices) *V* and edges *E*.
- The adjacency matrix **A** of *G* is defined as $[A]_{ij} = \begin{cases} 1 & \text{if } (v_i, v_j) \in E \\ 0 & \text{otherwise} \end{cases}$.
- **Walk** – a sequence of nodes, in which consecutive nodes are connected by an edge. A walk can travel over any edge and any node any number of times.



$w = (v_1, v_2, v_6, v_7, v_2, v_6, v_5)$

- Walks of length *k* can be computed by taking the adjacency matrix **A** to the power of *k*.

$A^k(v_i,v_j) = m$ → *m* walks of length *k* exist between nodes v_i and v_j.

Random walk graph kernel

- Random walk kernel computes **the number of all pairs of matching walks in a pair of graphs.**

- **TRICK**: common walks of length *k* can be calculated from the adjacency matrix of the **product graph** $G_\times$ of two input graphs G_1 and G_2.

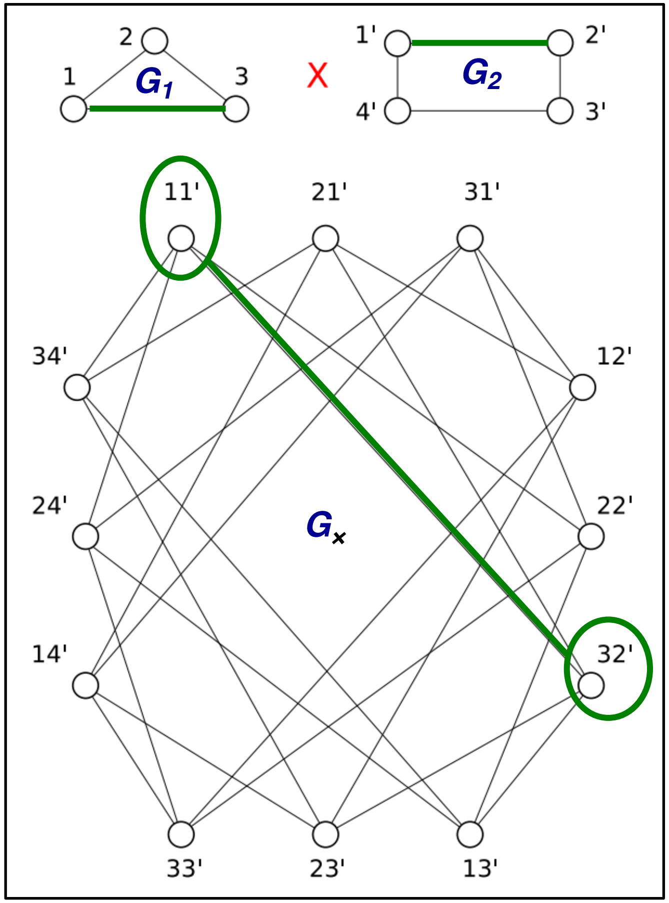
- $G_\times$ is a graph over pairs of vertices from G_1 and G_2. Two vertices in $G_\times$ are neighbours if and only if the corresponding vertices in G_1 and G_2 are both neighbours.

- **Random walk kernel**

$$K_\times(G_1,G_2) = \sum_{i,j=1}^{|V_\times|}\Big[\sum_{n=0}^{\infty}\lambda^n A_\times^n\Big]_{ij} = \sum_{i,j=1}^{|V_\times|}\big[(I-\lambda A_\times)^{-1}\big]$$

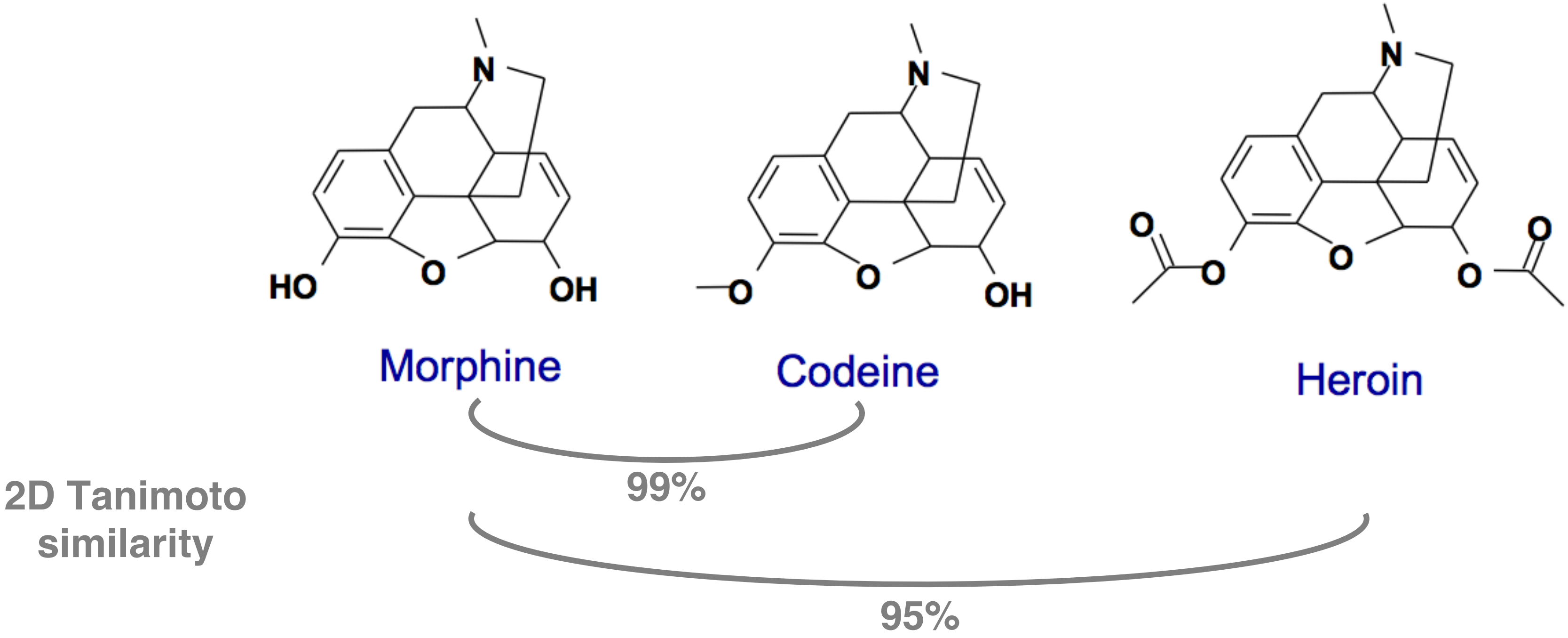
Counts all pairs of matching walks of any length

Vishwanathan S *et al.* (2010) "Graph kernels". The Journal of Machine Learning Research.



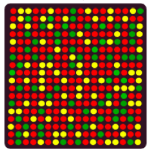
Problem with using chemical structures

- Sometimes structurally similar molecules can have different properties.



Additional information

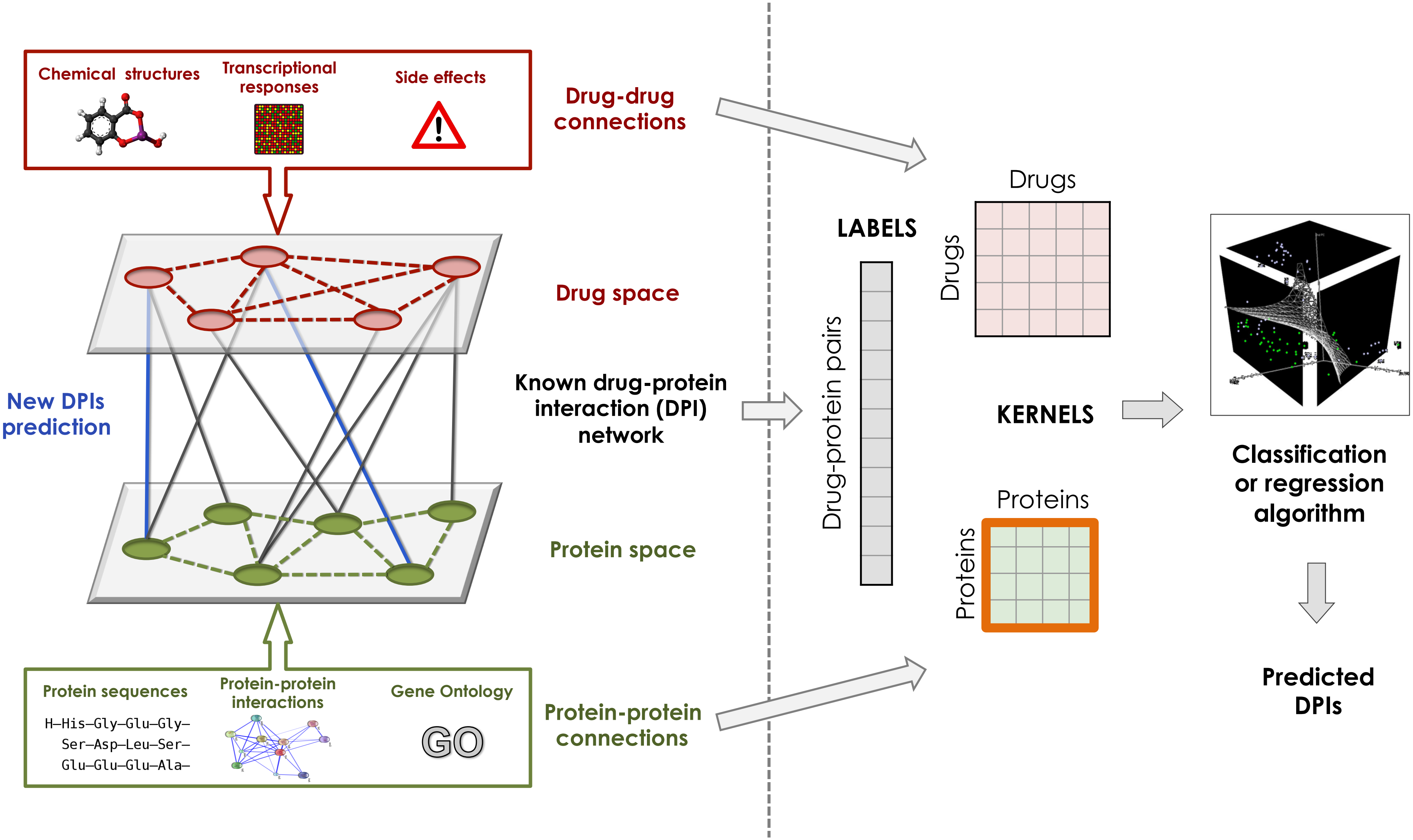
- Side effects. ⚠
- Anatomical Therapeutic Chemical (ATC) Classification System.
- Gene expression responses to drugs.



ATC

Example: ***Glibenclamide*** **(A10B B01)**

A	Alimentary tract and metabolism (main anatomical group)
A10	Drugs used in diabetes (main therapeutic group)
A10B	Oral blood-glucose-lowering drugs (pharmacological subgroup)
A10B B	Sulfonamides, urea derivatives (chemical/therapeutic subgroup)
A10B B01	Glibenclamide (subgroup for chemical substance)



Chemical structures

Transcriptional responses

Side effects

Drug-drug connections

Drug space

New DPIs prediction

Known drug-protein interaction (DPI) network

Protein space

Protein sequences

H–His–Gly–Glu–Gly–
Ser–Asp–Leu–Ser–
Glu–Glu–Glu–Ala–

Protein-protein interactions

Gene Ontology

GO

Protein-protein connections

LABELS

Drug-protein pairs

Drugs

Drugs

KERNELS

Proteins

Proteins

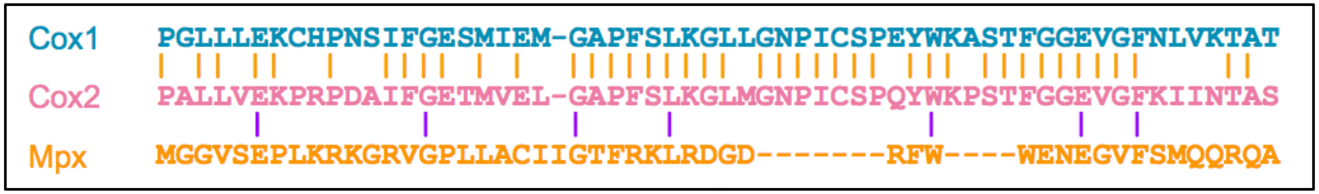
Classification or regression algorithm

Predicted DPIs

Aalto University
School of Science

Amino acid sequence alignment

- **Protein sequence alignment** is a way of arranging the amino acid sequences to identify regions of similarity that may be a consequence of functional, structural, or evolutionary relationships between the sequences.



```
Cox1  PGLLLEKCHPNSIFGESMIEM-GAPFSLKGLLGNPICSPEYWKASTFGGEVGFNLVKTAT
      | || ||  |  |||| | |  ||||||||| ||||||| ||| |||||||||    ||
Cox2  PALLVEKPRPDAIFGETMVEL-GAPFSLKGLMGNPICSPQYWKPSTFGGEVGFKIINTAS
           |        |       |    |             |       |  |
Mpx   MGGVSEPLKRKGRVGPLLACIIGTFRKLRDGD-------RFW----WENEGVFSMQQRQA
```

- **Smith-Waterman (SW) algorithm**
  - Performs local sequence alignment; uses dynamic programming to compare segments of all possible lengths.
  - To find the optimal alignment, a scoring system including a set of specified gap penalties is used (different scoring matrices, e.g. BLOSUM, PAM).
  - The algorithm assigns a score to each residue comparison between two sequences.
  - Normalized similarity between two proteins p_1 and p_2:

$$s(p_1, p_2) = \frac{SW(p_1, p_2)}{\sqrt{SW(p_1, p_1)}\sqrt{SW(p_2, p_2)}}.$$

Generic String (GS) kernel

Protein space

$$GS(\mathbf{x},\mathbf{x}',L,\sigma_p,\sigma_c)$$

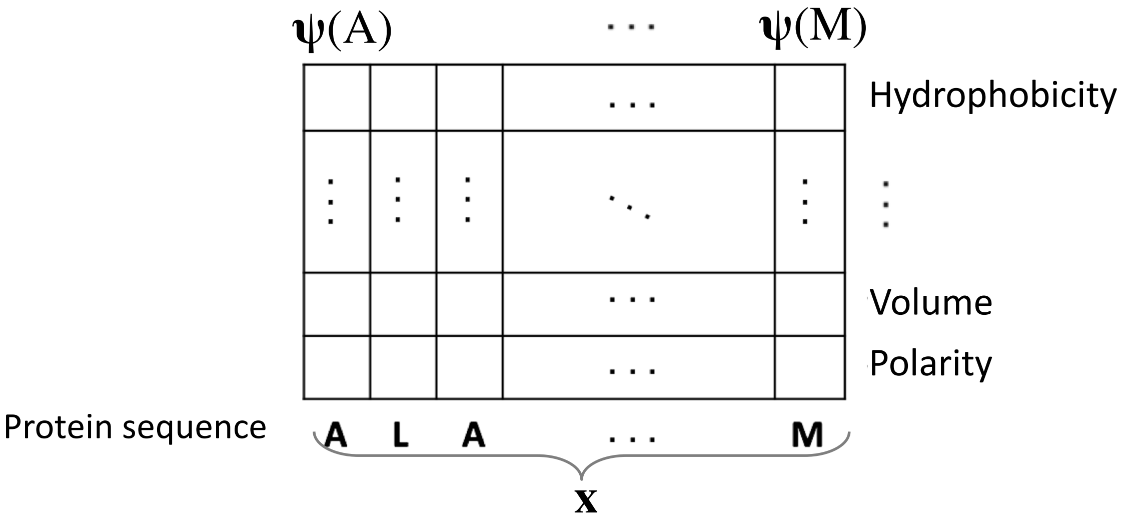
$$\stackrel{\text{def}}{=} \sum_{l=1}^{L} \sum_{i=0}^{|x|-l} \sum_{j=0}^{|x'|-l} e^{\left(\frac{-(i-j)^2}{2\sigma_p^2}\right)} e^{\left(\frac{-\|\boldsymbol{\psi}^l(x_{i+1},\ldots,x_{i+l}) - \boldsymbol{\psi}^l(x'_{j+1},\ldots,x_{j+l})\|^2}{2\sigma_c^2}\right)}$$

Shifting contribution term

Similarity of the amino acids in the substrings x and x’

Each type of amino acid a_k, $k = 1,\ldots,K$, (e.g. Asparagine) has a corresponding feature vector $\boldsymbol{\psi}(a_k)$ which defines its d properties:

$$\boldsymbol{\psi}(a_k) = (\psi_1(a_k), \psi_2(a_k),\ldots, \psi_d(a_k)).$$



Giguère S *et al.* (2013) “Learning a peptide-protein binding affinity predictor with kernel ridge regression”. BMC Bioinformatics.

Generic String (GS) kernel

Protein space

$$GS(\mathbf{x},\mathbf{x}',L,\sigma_p,\sigma_c)$$

$$\stackrel{\text{def}}{=} \sum_{l=1}^{L} \sum_{i=0}^{|\mathbf{x}|-l} \sum_{j=0}^{|\mathbf{x}'|-l} e^{\left(\frac{-(i-j)^2}{2\sigma_p^2}\right)} e^{\left(\frac{-\|\boldsymbol{\psi}^l(x_{i+1},..,x_{i+l}) - \boldsymbol{\psi}^l(x'_{j+1},..,x_{j+l})\|^2}{2\sigma_c^2}\right)}$$

Shifting contribution term

Similarity of the amino acids in the substrings x and x'

Each type of amino acid a_k, $k = 1,\ldots,K$, (e.g. Asparagine) has a corresponding feature vector $\boldsymbol{\psi}(a_k)$ which defines its d properties:

$$\boldsymbol{\psi}(a_k) = (\psi_1(a_k), \psi_2(a_k),\ldots, \psi_d(a_k)).$$

Given a string $\mathbf{x} = x_1,x_2,\ldots,x_l$, $\psi^l(\mathbf{x})$ is its encoding function which concatenates l vectors describing each amino acid the string $\mathbf{x}$ is composed of:

$$\psi^l(\mathbf{x}) = (\boldsymbol{\psi}(x_1), \boldsymbol{\psi}(x_2),\ldots, \boldsymbol{\psi}(x_l)).$$

Giguère S *et al.* (2013) "Learning a peptide-protein binding affinity predictor with kernel ridge regression". BMC Bioinformatics.

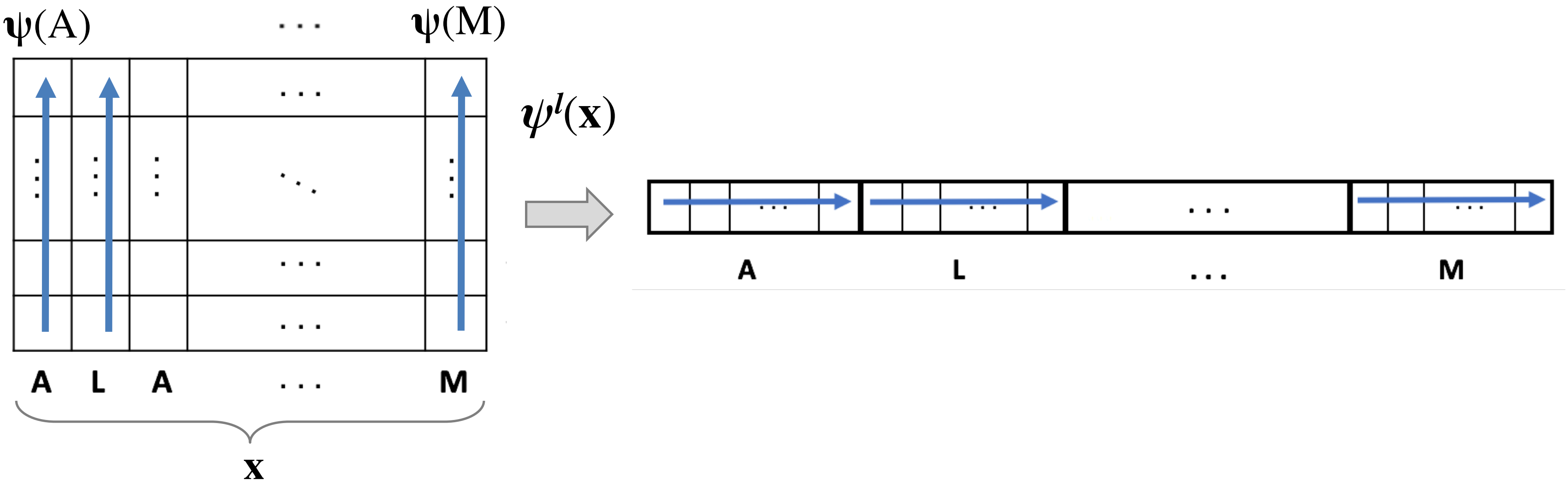
Generic String (GS) kernel

$$GS(\mathbf{x},\mathbf{x}',L,\sigma_p,\sigma_c)$$

$$\stackrel{\text{def}}{=} \sum_{l=1}^{L}\sum_{i=0}^{|x|-l}\sum_{j=0}^{|x'|-l} e^{\left(\frac{-(i-j)^2}{2\sigma_p^2}\right)} e^{\left(\frac{-\|\psi^l(x_{i+1},..,x_{i+l})-\psi^l(x'_{j+1},..,x_{j+l})\|^2}{2\sigma_c^2}\right)}$$

Shifting contribution term

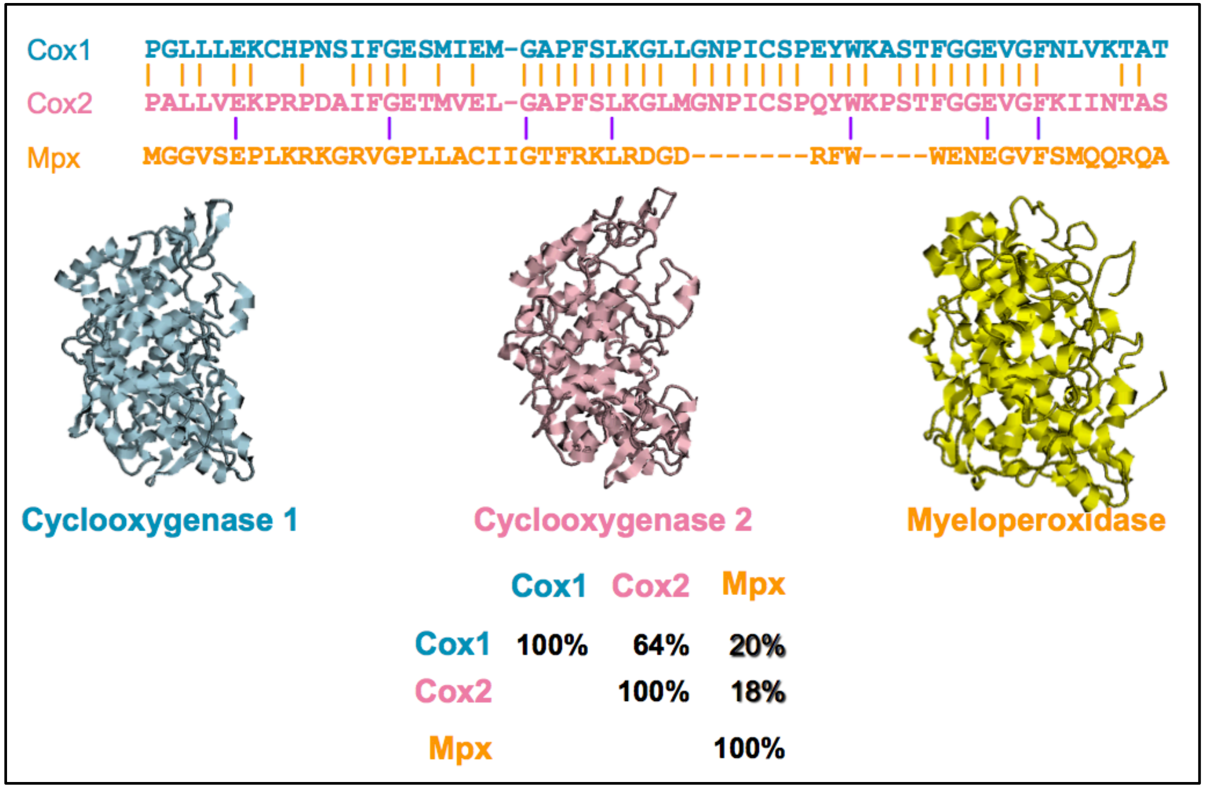
Similarity of the amino acids in the substrings x and x’



Giguère S *et al.* (2013) “Learning a peptide-protein binding affinity predictor with kernel ridge regression”. BMC Bioinformatics.

Amino acid sequence alignment

- Some proteins might have very low amino acid sequence identity but similar 3D structures.

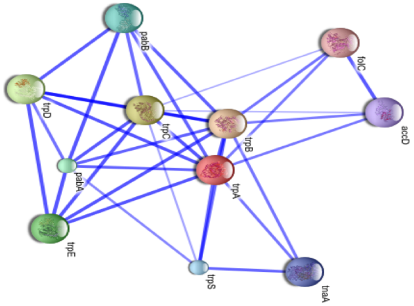


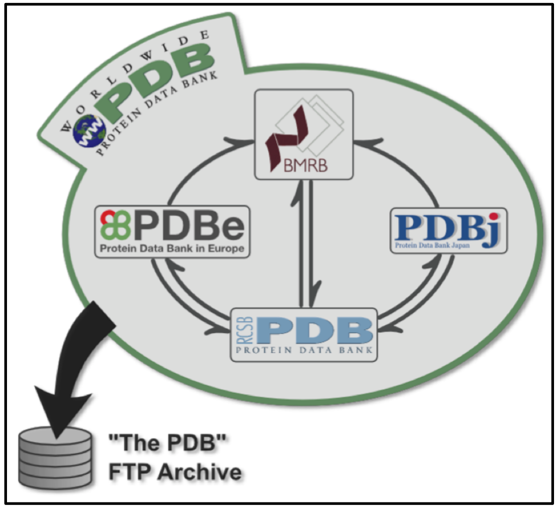
```
Cox1  PGLLLEKCHPNSIFGESMIEM-GAPFSLKGLLGNPICSPEYWKASTFGGEVGFNLVKTAT
      | || ||  |  |||| | |  ||||||||| ||||||| ||| |||||||||    ||
Cox2  PALLVEKPRPDAIFGETMVEL-GAPFSLKGLMGNPICSPQYWKPSTFGGEVGFKIINTAS
           |        |       |    |             |       |  |
Mpx   MGGVSEPLKRKGRVGPLLACIIGTFRKLRDGD-------RFW----WENEGVFSMQQRQA
```

	Cox1	Cox2	Mpx
Cox1	100%	64%	20%
Cox2		100%	18%
Mpx			100%

Additional information

- 3D protein structures (Protein Data Bank PDB, computational prediction algorithms);
- Binding sites, domains;
- Protein surface;
- Protein-protein interaction network;

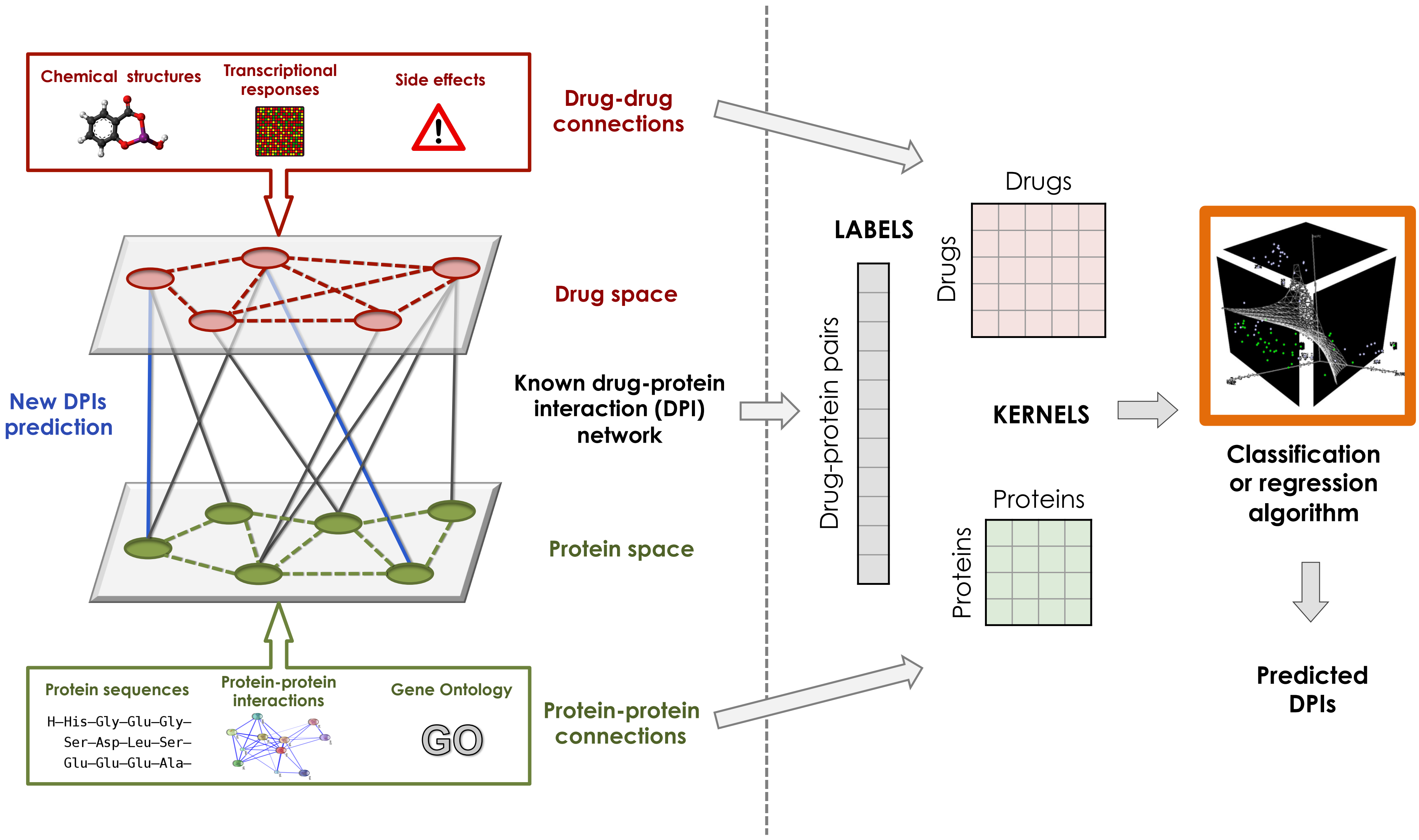



WORLDWIDE PDB PROTEIN DATA BANK
BMRB
PDBe Protein Data Bank in Europe
PDBj Protein Data Bank Japan
RCSB PDB PROTEIN DATA BANK
"The PDB" FTP Archive

http://wwpdb.org/

- Gene Ontology classifications (http://geneontology.org/).

Three domains:
1) biological processes,
2) cellular components,
3) molecular functions.



Chemical structures

Transcriptional responses

Side effects

Drug-drug connections

Drug space

New DPIs prediction

Known drug-protein interaction (DPI) network

Protein space

Protein sequences

H–His–Gly–Glu–Gly–
Ser–Asp–Leu–Ser–
Glu–Glu–Glu–Ala–

Protein-protein interactions

Gene Ontology

GO

Protein-protein connections

LABELS

Drug-protein pairs

Drugs

Drugs

KERNELS

Proteins

Proteins

Classification or regression algorithm

Predicted DPIs

Aalto University
School of Science

Supervised learning

Inferring a function from LABELED training data

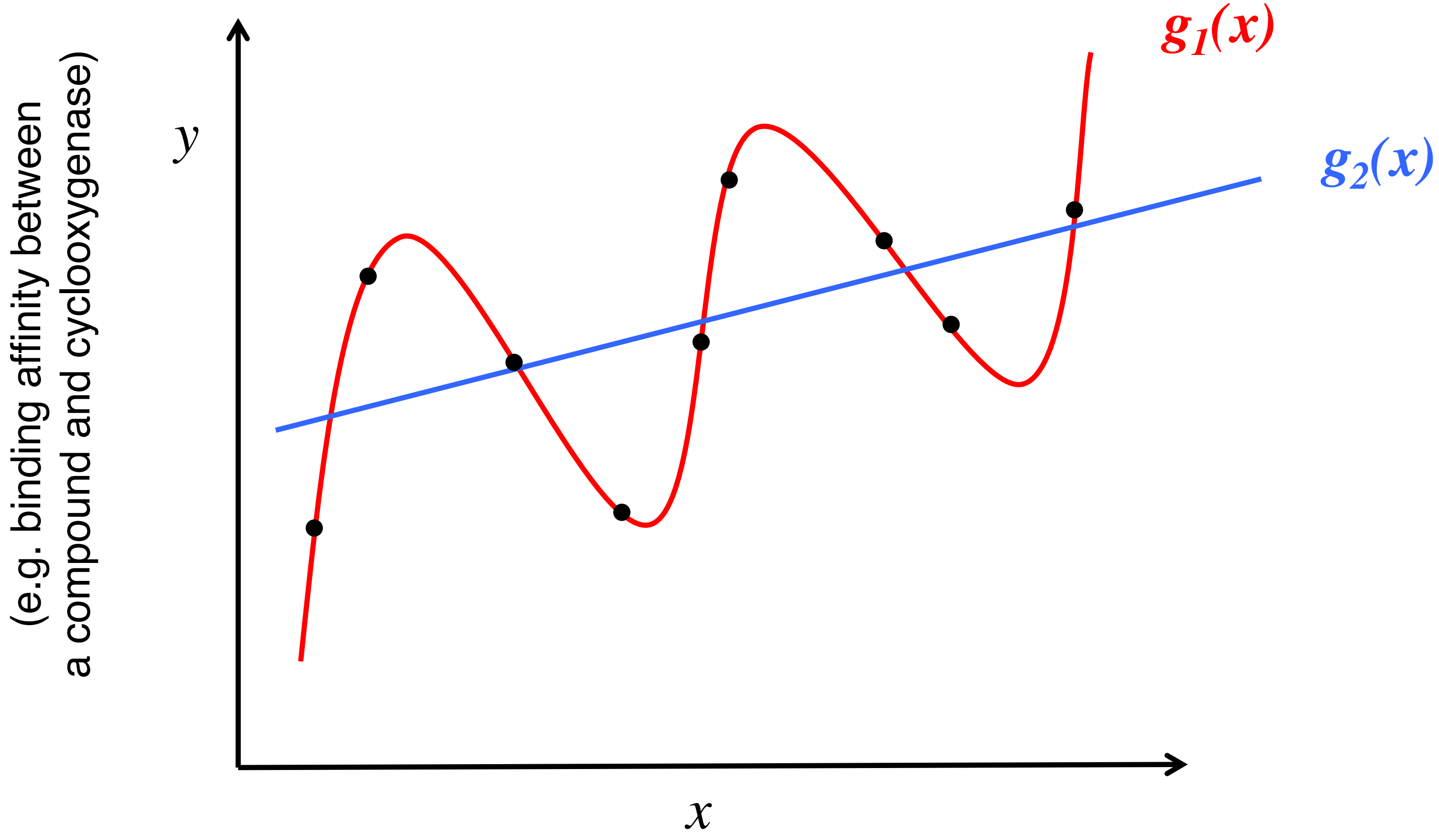
- **Classification** is the prediction of a **class label**, given attributes.
- **Regression** is the prediction of a **real number**, given attributes.

INGREDIENTS

- $\mathcal{X}$: a space of inputs.
- $\mathcal{Y}$: a space of outputs.
- $\mathcal{G}$: a set of models mapping input to output $\mathcal{G} = \{g: \mathcal{X} \mapsto \mathcal{Y}\}$.
- Training dataset S: $\{(\mathbf{x}_i, y_i)\}$, $i=1,\ldots,N$, $\mathbf{x}_i \in \mathcal{X}$, $y_i \in \mathcal{Y}$ sampled from an underlying unknown distribution $(\mathbf{x}, y) \sim \mathcal{D}$.
- $\mathcal{L}$: a loss function measuring the discrepancy between the model's predicted outputs and true outputs.

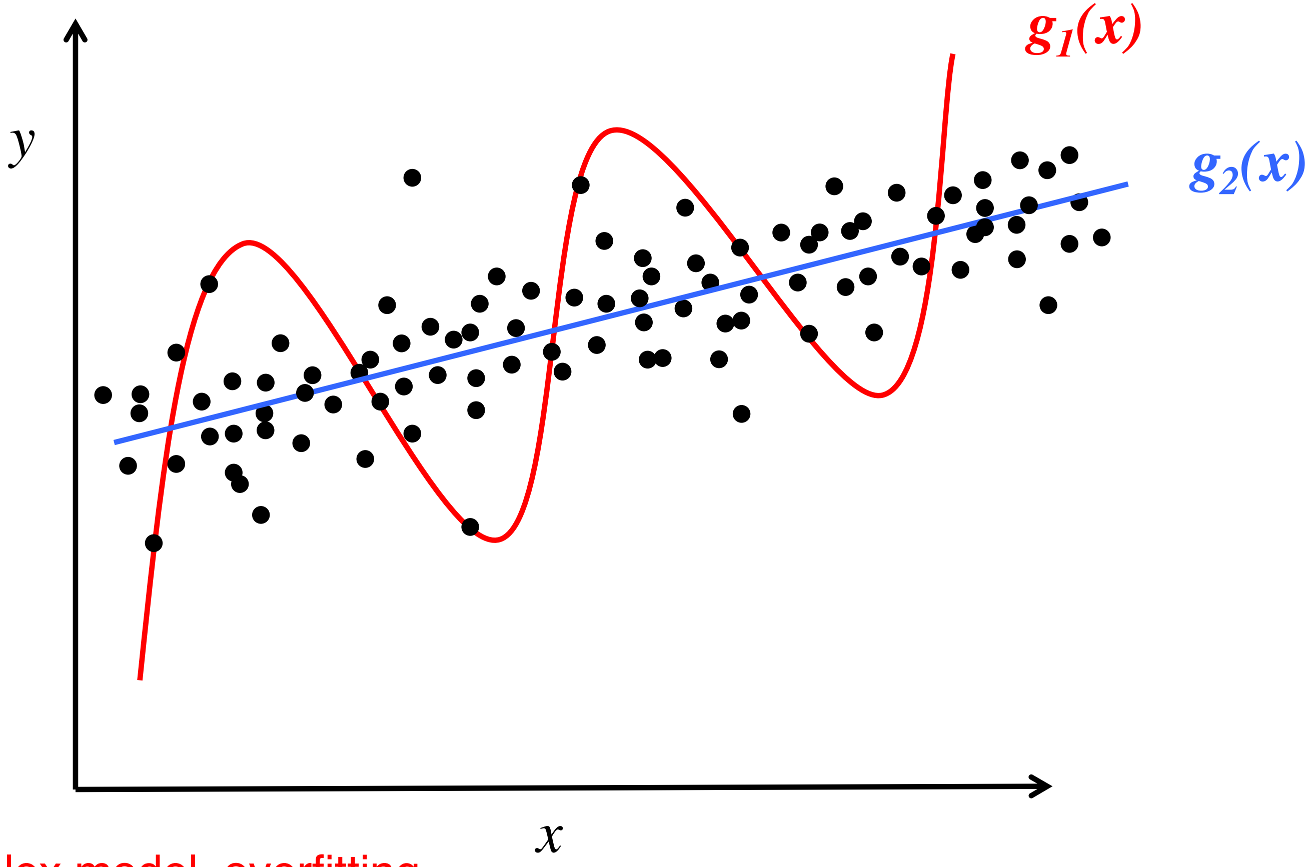
GOAL: to find a model g that minimizes the expected loss $\mathcal{L}(g(\boldsymbol{x}), y)$ on future instances.

Regression



(e.g. molecular weight of a chemical compound)

Regression



$g_1(x)$: more complex model, overfitting.

$g_2(x)$: generalizes better to new instances.

Regularization

- Regularized learning considers optimising the functions of the form:

$$\arg\min_{g \in G} \boxed{\sum_{i=1}^{N} \mathcal{L}(g(\mathbf{x}_i), y_i) + \lambda \Omega(g)}$$

Regularized empirical risk

Training error (loss), typically, squared loss: $\mathcal{L}(g(\mathbf{x}_i), y_i) = (g(\mathbf{x}_i) - y_i)^2$.

Regularizer that controls the complexity of the model g.

- Complex model g → high value of $\Omega(g)$.
- Regularization parameter $\lambda \geq 0$ controls the balance between training error and model complexity.

Linear model

- A model in the form of a linear function $g(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle = \mathbf{w}^T \mathbf{x}$, where $\mathbf{w} \in \mathbb{R}^p$ is the vector of model parameters to be found by minimizing $\sum_{i=1}^{N} \mathcal{L}(g(\mathbf{x}_i), y_i) + \lambda \Omega(g)$.

- The choice of the loss function and regularization determines the learning algorithm.

$\mathcal{L}(g(\mathbf{x}_i), y_i)$	$\Omega(g)$	Algorithm
$\max(0, 1 - g(\mathbf{x}_i) y_i)$, where $\mathbf{y} \in \{-1, +1\}$	$\|\mathbf{w}\|^2 = \langle \mathbf{w}, \mathbf{w} \rangle$	Support vector machine (SVM)
$(g(\mathbf{x}_i) - y_i)^2$, where $\mathbf{y} \in \mathbb{R}^N$	$\|\mathbf{w}\|^2 = \langle \mathbf{w}, \mathbf{w} \rangle$	Ridge regression
$(g(\mathbf{x}_i) - y_i)^2$, where $\mathbf{y} \in \mathbb{R}^N$	$\|\mathbf{w}\|_1 = \sum_{l=1}^{p} \lvert w_l \rvert$	Least absolute shrinkage and selection operator regression (LASSO)

Ridge regression

- Given the squared loss and quadratic regularizer, the optimization problem of ridge regression can be written as:

$$\arg\min_{\mathbf{w}} \sum_{i=1}^{N} (y_i - \langle \mathbf{w}, \mathbf{x}_i \rangle)^2 + \lambda ||\mathbf{w}||^2$$

$$\arg\min_{\mathbf{w}} \langle \mathbf{y} - \mathbf{X}\mathbf{w}, \mathbf{y} - \mathbf{X}\mathbf{w} \rangle + \lambda \langle \mathbf{w}, \mathbf{w} \rangle, \qquad \mathbf{y} \in \mathbb{R}^N, \mathbf{X} \in \mathbb{R}^{N \times p}$$

$$\frac{\delta}{\delta \mathbf{w}} (\langle \mathbf{y} - \mathbf{X}\mathbf{w}, \mathbf{y} - \mathbf{X}\mathbf{w} \rangle + \lambda \langle \mathbf{w}, \mathbf{w} \rangle) = 0$$

$$\mathbf{X}^T\mathbf{X}\mathbf{w} + \lambda \mathbf{w} - \mathbf{X}^T\mathbf{y} = 0$$

$$\mathbf{X}^T\mathbf{X}\mathbf{w} + \lambda \mathbf{w} = \mathbf{X}^T\mathbf{y}$$

$$\mathbf{w} = (\mathbf{X}^T\mathbf{X} + \lambda \mathbf{I}_p)^{-1} \mathbf{X}^T\mathbf{y}$$

$\mathbf{I}_p$ is a $p \times p$ identity matrix

Linear model and dual representation

- The optimal $\mathbf{w}$ can be written as a linear combination of the examples by introducing so called *dual variable* $\boldsymbol{\alpha} \in \mathbb{R}^N$:

$$\mathbf{w} = \sum_{i=1}^{N} \alpha_i \mathbf{x}_i = \mathbf{X}^T \boldsymbol{\alpha}\,.$$

- Now, we can represent model's prediction in terms of inner products of training examples → we can use **kernels**:

$$g(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle = \sum_{i=1}^{N} \alpha_i \langle \mathbf{x}_i, \mathbf{x} \rangle = \sum_{i=1}^{N} \alpha_i k(\mathbf{x}_i, \mathbf{x}) = \boldsymbol{\alpha}^T \mathbf{k},$$

where $\mathbf{k}$ is a vector with kernel values between each training example $\mathbf{x}_i$ and a test example $\mathbf{x}$ for which the prediction is made.

Ridge regression

- Given the squared loss and quadratic regularizer, the optimization problem of ridge regression can be written as:

$$\arg\min_{\mathbf{w}} \sum_{i=1}^{N} (y_i - \langle \mathbf{w}, \mathbf{x}_i \rangle)^2 + \lambda ||\mathbf{w}||^2$$

$$\arg\min_{\mathbf{w}} \langle \mathbf{y} - \mathbf{X}\mathbf{w}, \mathbf{y} - \mathbf{X}\mathbf{w} \rangle + \lambda \langle \mathbf{w}, \mathbf{w} \rangle, \qquad \mathbf{y} \in \mathbb{R}^N, \mathbf{X} \in \mathbb{R}^{N \times p}$$

$$\frac{\delta}{\delta \mathbf{w}} (\langle \mathbf{y} - \mathbf{X}\mathbf{w}, \mathbf{y} - \mathbf{X}\mathbf{w} \rangle + \lambda \langle \mathbf{w}, \mathbf{w} \rangle) = 0$$

$$\mathbf{X}^T\mathbf{X}\mathbf{w} + \lambda \mathbf{w} - \mathbf{X}^T\mathbf{y} = 0$$

$$\mathbf{X}^T\mathbf{X}\mathbf{w} + \lambda \mathbf{w} = \mathbf{X}^T\mathbf{y}$$

$$\mathbf{w} = (\mathbf{X}^T\mathbf{X} + \lambda \mathbf{I}_p)^{-1}\, \mathbf{X}^T\mathbf{y}$$

$\mathbf{I}_p$ is a $p \times p$ identity matrix

Kernel ridge regression (KRR)

$$\mathbf{w} = \sum_{i=1}^{N} \alpha_i \mathbf{x}_i = \mathbf{X}^T \boldsymbol{\alpha}.$$

- We can rewrite equation $\mathbf{X}^T\mathbf{X}\mathbf{w} + \lambda\mathbf{w} = \mathbf{X}^T\mathbf{y}$ in terms of **w** to get:

$$\mathbf{w} = \lambda^{-1}\mathbf{X}^T(\mathbf{y} - \mathbf{X}\mathbf{w}) = \mathbf{X}^T\boldsymbol{\alpha}.$$

- The solution to ridge regression in the dual space (i.e. KRR) has a closed form

$$\boldsymbol{\alpha} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}_N)^{-1}\mathbf{y}$$

$$\boldsymbol{\alpha} = (\mathbf{K} + \lambda\mathbf{I}_N)^{-1}\mathbf{y} \qquad \mathbf{K} \in \mathbb{R}^{N\times N}$$

Saunders C *et al.* (1998) "Ridge regression learning algorithm in dual variables". In Proceedings of the 15th International Conference on Machine Learning.

KRR for drug-protein binding affinity prediction

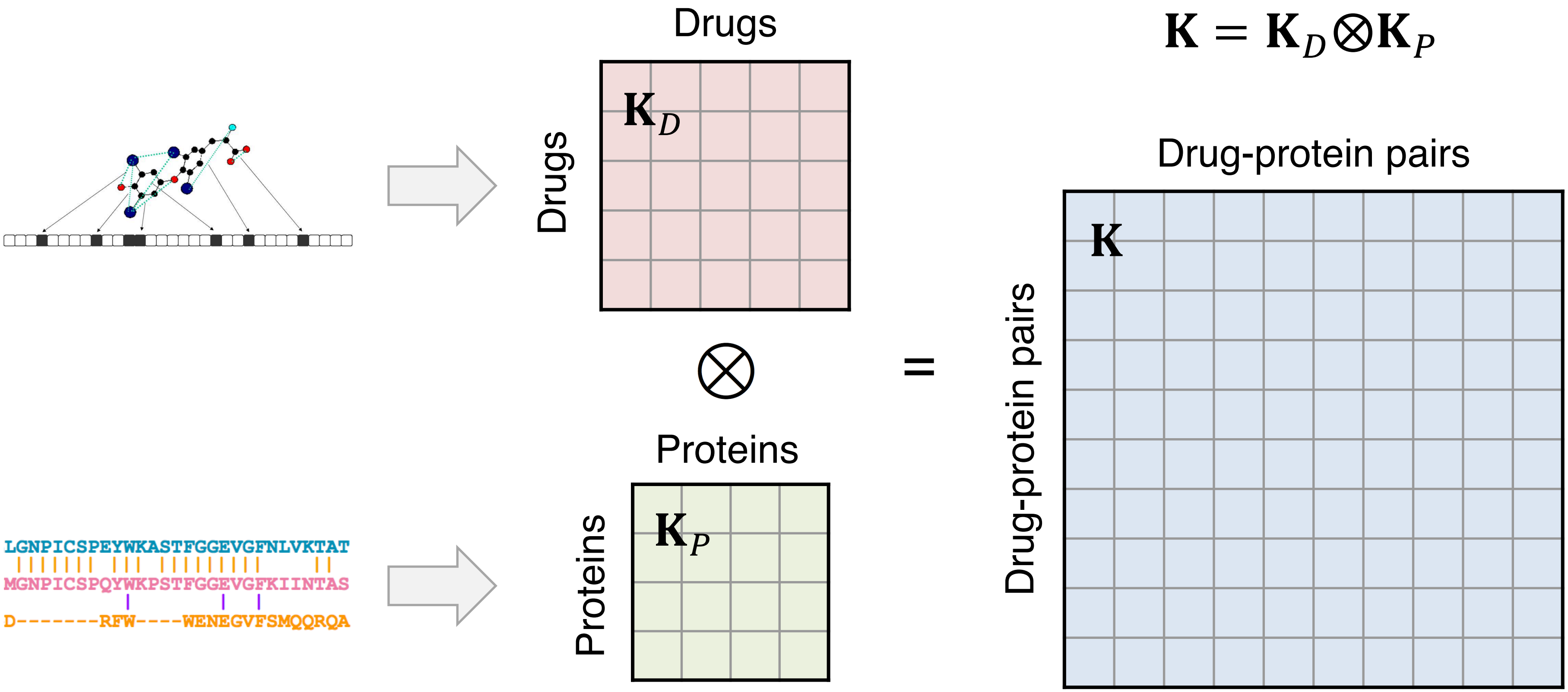
Ingredients

- A set of n_d drugs: $D = \{\mathbf{d}_1, \ldots, \mathbf{d}_{n_d}\}$
- A set of n_p proteins: $P = \{\mathbf{p}_1, \ldots, \mathbf{p}_{n_p}\}$
- A set of N training examples (**drug-protein pairs**): $X = \{(\mathbf{d}_1, \mathbf{p}_1), \ldots, (\mathbf{d}_1, \mathbf{p}_{n_p}), (\mathbf{d}_2, \mathbf{p}_1), \ldots, (\mathbf{d}_2, \mathbf{p}_{n_p}), \ldots, \ldots, (\mathbf{d}_{n_d}, \mathbf{p}_{n_p})\}$
- $N \leq n_d \times n_p$
- y_i: a real value indicating binding affinity of i^{th} drug-protein pair $\mathbf{x}_i$
- Pairwise kernel matrix $\mathbf{K} \in \mathbb{R}^{N \times N}$

Pairwise kernel

PAIRWISE KERNEL

$$\mathbf{K} = \mathbf{K}_D \otimes \mathbf{K}_P$$



Kronecker product

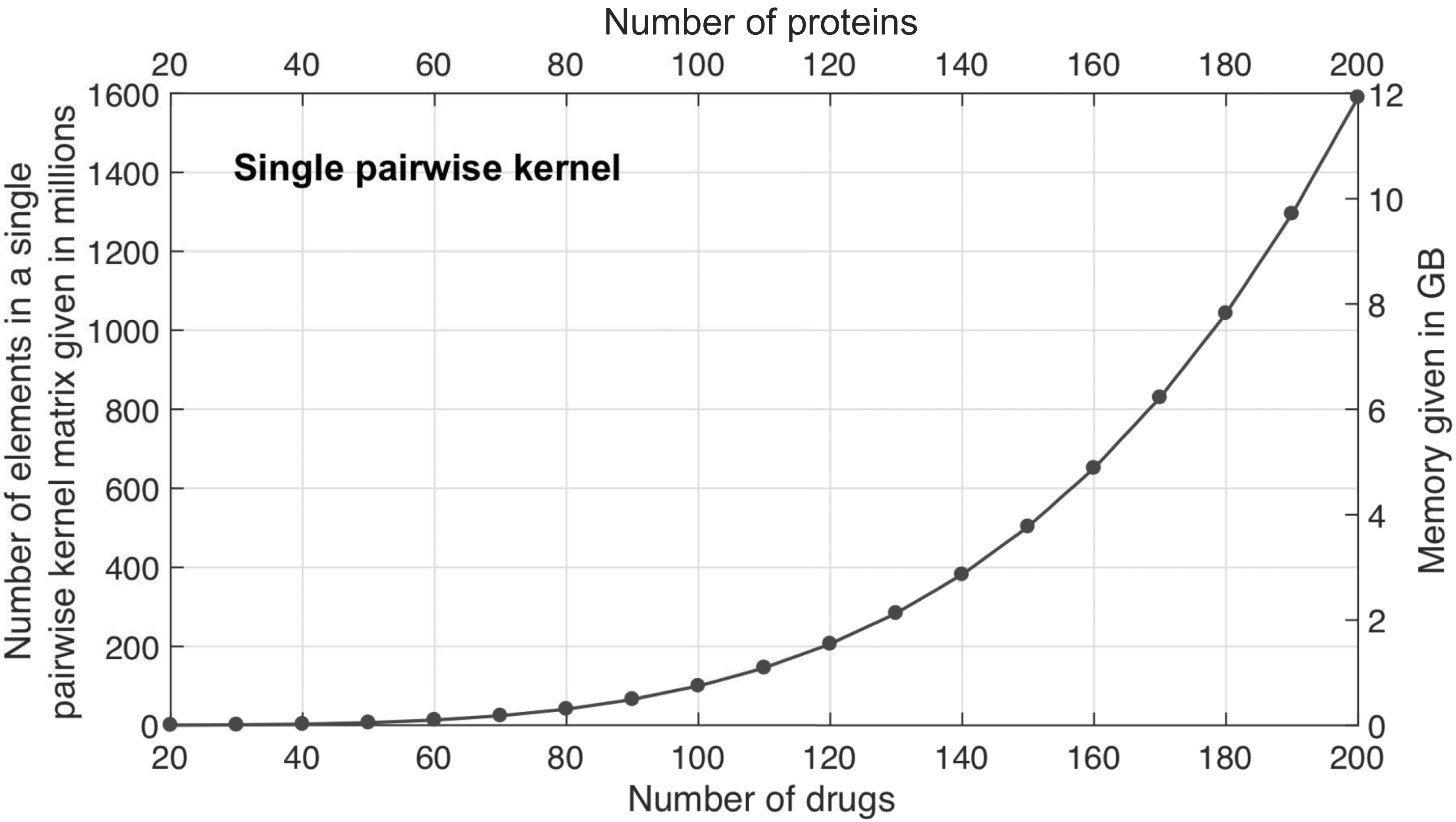
- Defined for any two matrices **B** and **C** of arbitrary size.
- Resulting matrix contains all possible products of entries of **B** and **C**.

Example

$$\begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \otimes \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix} = \left[\begin{array}{ccc|ccc} b_{11}c_{11} & b_{11}c_{12} & b_{11}c_{13} & b_{12}c_{11} & b_{12}c_{12} & b_{12}c_{13} \\ b_{11}c_{21} & b_{11}c_{22} & b_{11}c_{23} & b_{12}c_{21} & b_{12}c_{22} & b_{12}c_{23} \\ b_{11}c_{31} & b_{11}c_{32} & b_{11}c_{33} & b_{12}c_{31} & b_{12}c_{32} & b_{12}c_{33} \\ \hline b_{21}c_{11} & b_{21}c_{12} & b_{21}c_{13} & b_{22}c_{11} & b_{22}c_{12} & b_{22}c_{13} \\ b_{21}c_{21} & b_{21}c_{22} & b_{21}c_{23} & b_{22}c_{21} & b_{22}c_{22} & b_{22}c_{23} \\ b_{21}c_{31} & b_{21}c_{32} & b_{21}c_{33} & b_{22}c_{31} & b_{22}c_{32} & b_{22}c_{33} \end{array}\right]$$

Pairwise kernel

- The size of a pairwise kernel matrix $\mathbf{K}$ makes the model training computationally infeasible in typical applications.



Pairwise KRR – shortcut

- It is possible to use algebraic properties of the Kronecker product to avoid the explicit computation of the pairwise kernel, and therefore significantly speed up the model training.

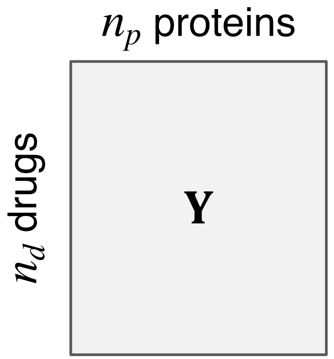
$$
\begin{aligned}
\boldsymbol{\alpha} &= (\mathbf{K} + \lambda \mathbf{I}_N)^{-1}\mathbf{y} \\
&= (\mathbf{K}_D \otimes \mathbf{K}_P + \lambda \mathbf{I}_N)^{-1}\mathrm{vec}(\mathbf{Y}) \\
&= ((\mathbf{Q}_D \boldsymbol{\Lambda}_D \mathbf{Q}_D^T) \otimes (\mathbf{Q}_P \boldsymbol{\Lambda}_P \mathbf{Q}_P^T) + \lambda \mathbf{I}_N)^{-1}\mathrm{vec}(\mathbf{Y}) \\
&= \mathrm{vec}(\mathbf{Q}_P \mathbf{R} \mathbf{Q}_D^T)
\end{aligned}
$$

Eigen-decomposition of the kernel matrices $\mathbf{K}_D$ and $\mathbf{K}_P$

$$\mathrm{vec}(\mathbf{R}) = \underbrace{(\boldsymbol{\Lambda}_D \otimes \boldsymbol{\Lambda}_P}_{} + \lambda \mathbf{I}_N)^{-1}\mathrm{vec}(\mathbf{Q}_P^T \mathbf{Y}^T \mathbf{Q}_D)$$

$$\mathrm{diag}(\boldsymbol{\Lambda}_D \otimes \boldsymbol{\Lambda}_P) = \mathrm{diag}(\boldsymbol{\Lambda}_D) \otimes \mathrm{diag}(\boldsymbol{\Lambda}_P) = \mathrm{vec}(\mathrm{diag}(\boldsymbol{\Lambda}_P)\mathrm{diag}(\boldsymbol{\Lambda}_D)^T)$$

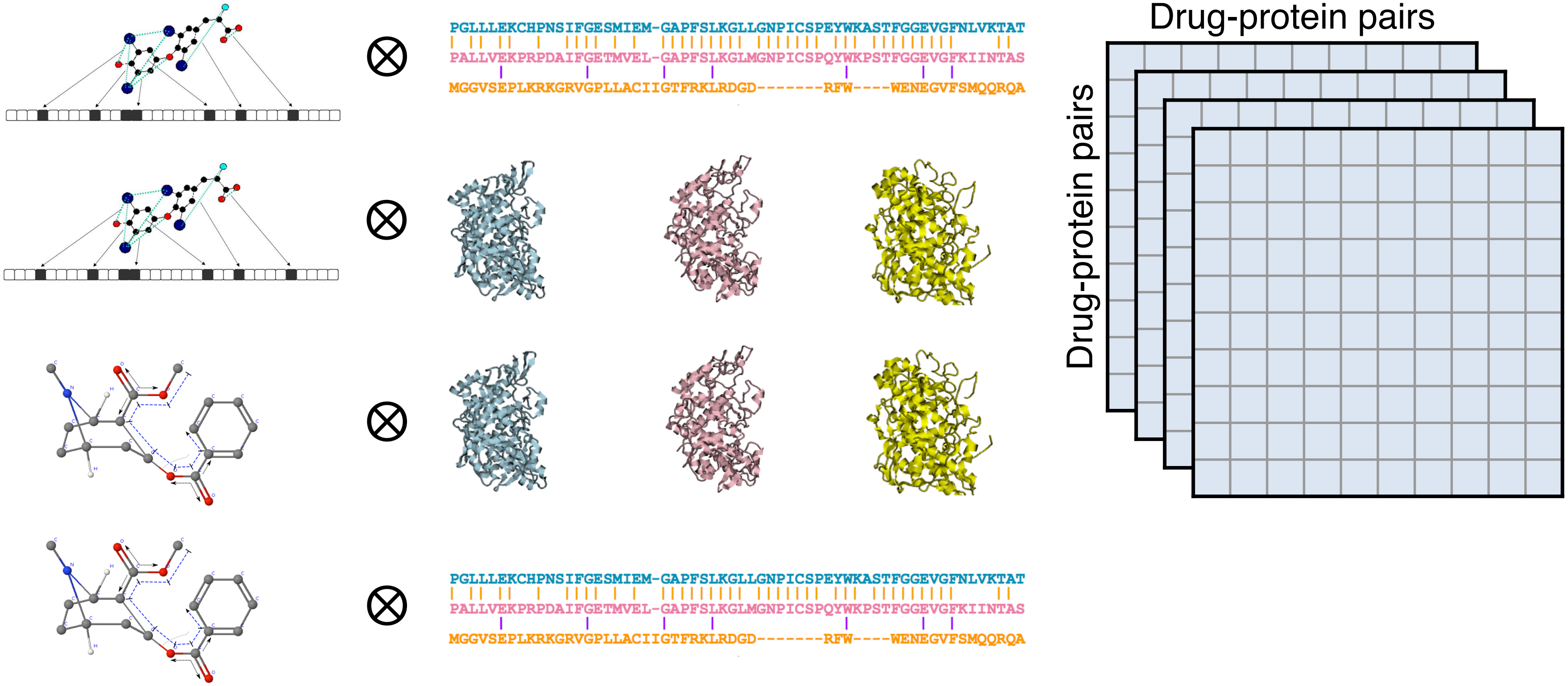
The above works if $\mathbf{Y}$ has no missing values, i.e., $N = n_d \times n_p$ (small number of missing values can be imputed as a pre-processing step).



Pahikkala T *et al.* (2013) "Efficient regularized least-squares algorithms for conditional ranking on relational data". Machine Learning.

Airola A and Pahikkala T (2017) "Fast Kronecker product kernel methods via generalized vec trick". IEEE Transactions on Neural Networks and Learning Systems.

Multiple Kernel Learning (MKL)


⊗
PGLLLEKCHPNSIFGESMIEM-GAPFSLKGLLGNPICSPEYWKASTFGGEVGFNLVKTAT
PALLVEKPRPDAIFGETMVEL-GAPFSLKGLMGNPICSPQYWKPSTFGGEVGFKIINTAS
MGGVSEPLKRKGRVGPLLACIIGTFRKLRDGD--------RFW----WENEGVFSMQQRQA
⊗
⊗
⊗
PGLLLEKCHPNSIFGESMIEM-GAPFSLKGLLGNPICSPEYWKASTFGGEVGFNLVKTAT
PALLVEKPRPDAIFGETMVEL-GAPFSLKGLMGNPICSPQYWKPSTFGGEVGFKIINTAS
MGGVSEPLKRKGRVGPLLACIIGTFRKLRDGD--------RFW----WENEGVFSMQQRQA
Drug-protein pairs
Drug-protein pairs

Multiple Kernel Learning (MKL)

- Classical kernel-based algorithms rely on a single kernel – the view resulting in the highest predictive performance is considered the best one.

- Risk of loosing some important information by dropping all the other views.

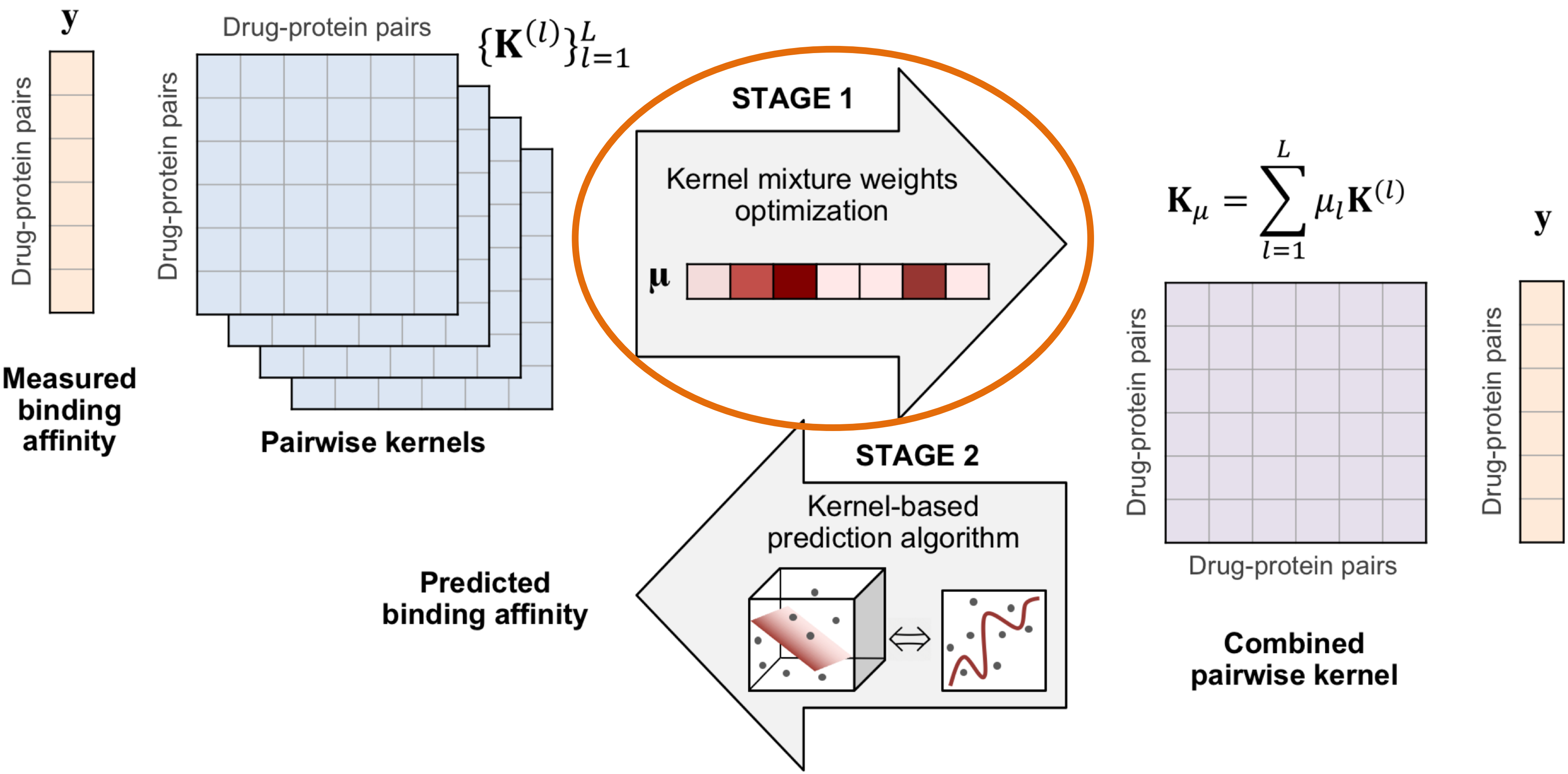
- Ideally, one would like to learn the importance of each kernel matrix in a given task, and then use a weighted combination of them:

$$\mathbf{K}_{\mu} = \sum_{l=1}^{L} \mu_l \mathbf{K}^{(l)}$$

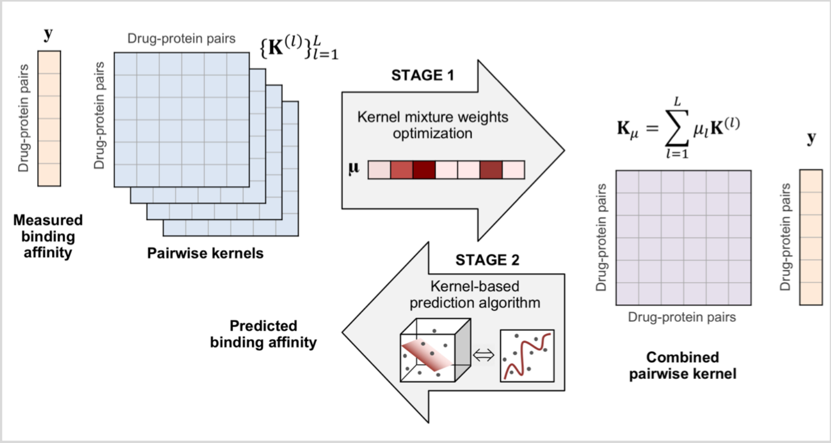
- One-stage MKL methods learn the kernel combination and prediction model parameters jointly.

- Two-stage MKL methods find the optimal kernel weights before subsequent phase of learning a classifier or regressor.

Two-stage MKL

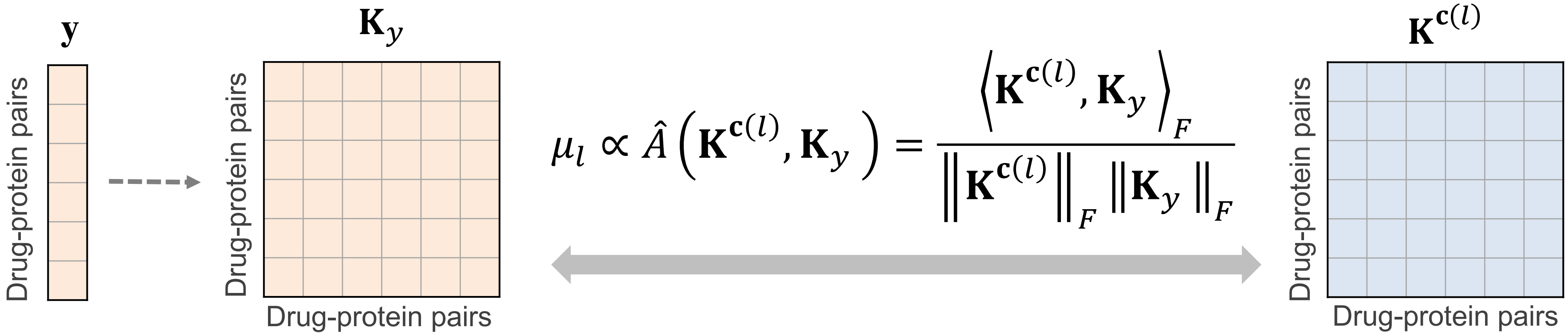


Two-stage MKL


y
Drug-protein pairs
Measured binding affinity
Drug-protein pairs
Drug-protein pairs
$\{\mathbf{K}^{(l)}\}_{l=1}^{L}$
Pairwise kernels
STAGE 1
Kernel mixture weights optimization
$\boldsymbol{\mu}$
STAGE 2
Kernel-based prediction algorithm
⇔
Predicted binding affinity
$\mathbf{K}_\mu = \sum_{l=1}^{L} \mu_l \mathbf{K}^{(l)}$
Drug-protein pairs
Drug-protein pairs
Combined pairwise kernel
y
Drug-protein pairs

- **UNIMKL**
  Equal kernel weights $\mu_l = \frac{1}{L}.$

- **ALIGN**
  Kernel weights chosen to be proportional to their centered alignment with so-called “ideal” response kernel $\mathbf{K}_y$ derived from the label values:



$$\mu_l \propto \hat{A}\left(\mathbf{K}^{\mathbf{c}(l)}, \mathbf{K}_y\right) = \frac{\left\langle \mathbf{K}^{\mathbf{c}(l)}, \mathbf{K}_y \right\rangle_F}{\left\| \mathbf{K}^{\mathbf{c}(l)} \right\|_F \left\| \mathbf{K}_y \right\|_F}$$

$$\mathbf{K}^{\mathbf{c}} = \mathbf{CKC}$$

$$\mathbf{C} = \left[\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{N}\right]$$

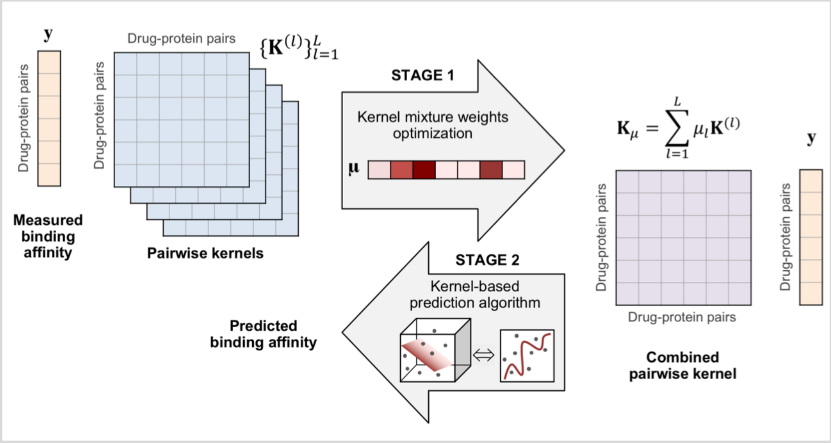
The sum of the rows (columns) of $\mathbf{K}^{\mathbf{c}}$ yields the zero vector $\mathbf{K}^{\mathbf{c}}\mathbf{1} = \mathbf{0}$ $(\mathbf{1}^T\mathbf{K}^{\mathbf{c}} = \mathbf{0}^T)$.

$$\langle \mathbf{A}, \mathbf{B} \rangle_F = \mathrm{vec}(\mathbf{A})^T \mathrm{vec}(\mathbf{B})$$

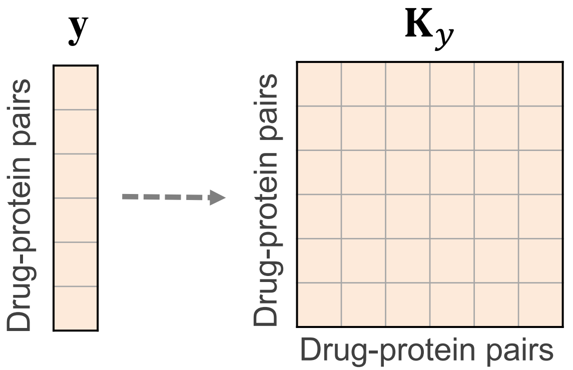
Cortes C *et al.* (2012) “Algorithms for learning kernels based on centered alignment”. Journal of Machine Learning Research.

Two-stage MKL

- **ALIGNF**
  Kernel mixture weights are determined by maximising the centered alignment between the combined kernel $\mathbf{K}_\mu$ and the response kernel $\mathbf{K}_y$:

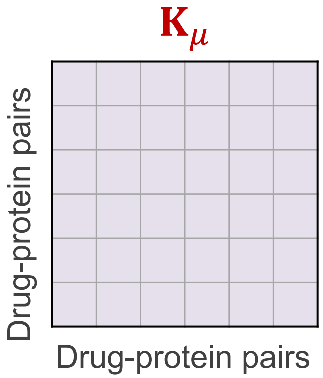


y
Drug-protein pairs
Measured binding affinity
Drug-protein pairs
Drug-protein pairs
$\{\mathbf{K}^{(l)}\}_{l=1}^{L}$
Pairwise kernels
STAGE 1
Kernel mixture weights optimization
μ
STAGE 2
Kernel-based prediction algorithm
Predicted binding affinity
$\mathbf{K}_\mu = \sum_{l=1}^{L} \mu_l \mathbf{K}^{(l)}$
y
Drug-protein pairs
Drug-protein pairs
Drug-protein pairs
Combined pairwise kernel



$$\underset{\boldsymbol{\mu}}{\arg\max}\, \hat{A}\left(\mathbf{K}^{\mathbf{c}}_{\mu}, \mathbf{K}_y\right) = \max_{\boldsymbol{\mu}} \frac{\left\langle \mathbf{K}^{\mathbf{c}}_{\mu}, \mathbf{K}_y \right\rangle_F}{\left\| \mathbf{K}^{\mathbf{c}}_{\mu} \right\|_F},$$

subject to: $\|\boldsymbol{\mu}\|_2 = 1, \boldsymbol{\mu} \geq 0.$



$$\mathbf{K}^{\mathbf{c}} = \mathbf{CKC}$$

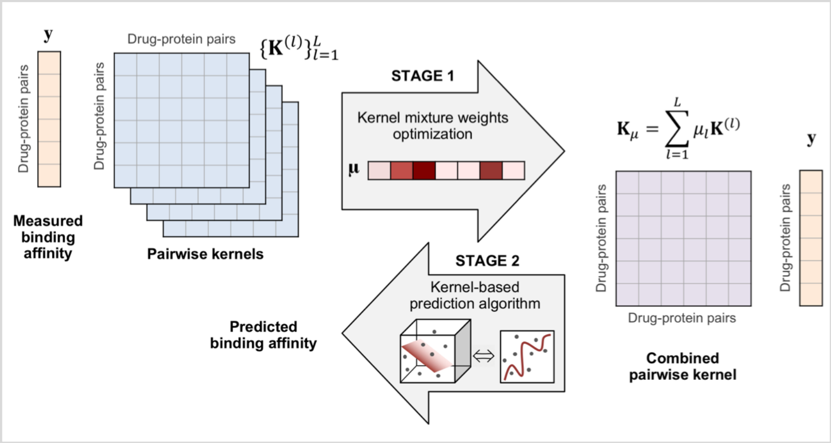
$$\mathbf{C} = \left[\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{N}\right]$$

The sum of the rows (columns) of $\mathbf{K}^{\mathbf{c}}$ yields the zero vector $\mathbf{K}^{\mathbf{c}}\mathbf{1} = \mathbf{0}$ $(\mathbf{1}^T\mathbf{K}^{\mathbf{c}} = \mathbf{0}^T)$.

$$\langle \mathbf{A}, \mathbf{B} \rangle_F = \mathrm{vec}(\mathbf{A})^T \mathrm{vec}(\mathbf{B})$$

Cortes C *et al.* (2012) “Algorithms for learning kernels based on centered alignment”. Journal of Machine Learning Research.

Two-stage MKL



y
Drug-protein pairs
Measured binding affinity
Drug-protein pairs
Drug-protein pairs
$\{\mathbf{K}^{(l)}\}_{l=1}^{L}$
Pairwise kernels
STAGE 1
Kernel mixture weights optimization
μ
$\mathbf{K}_\mu = \sum_{l=1}^{L} \mu_l \mathbf{K}^{(l)}$
y
Drug-protein pairs
Drug-protein pairs
Drug-protein pairs
Combined pairwise kernel
STAGE 2
Kernel-based prediction algorithm
⇔
Predicted binding affinity

- **ALIGNF**
  Kernel mixture weights are determined by maximising the centered alignment between the combined kernel $\mathbf{K}_\mu$ and the response kernel $\mathbf{K}_y$:

$$\arg\max_{\boldsymbol{\mu}} \hat{A}\left(\mathbf{K}_\mu^{\mathbf{c}}, \mathbf{K}_y\right) = \max_{\boldsymbol{\mu}} \frac{\left\langle \mathbf{K}_\mu^{\mathbf{c}}, \mathbf{K}_y \right\rangle_F}{\left\| \mathbf{K}_\mu^{\mathbf{c}} \right\|_F},$$

subject to: $\|\boldsymbol{\mu}\|_2 = 1, \boldsymbol{\mu} \geq 0.$

The above optimization problem can be solved via a simple **quadratic programming**:

$$\min_{\mathbf{v} \geq 0} \mathbf{v}^T \mathbf{M} \mathbf{v} - 2\mathbf{v}^T \mathbf{a},$$

$$(\mathbf{a})_i \quad = \left\langle \mathbf{K}^{\mathbf{c}(i)}, \mathbf{K}_y \right\rangle_F, \qquad i = 1, \ldots, L,$$

$$(\mathbf{M})_{ij} = \left\langle \mathbf{K}^{\mathbf{c}(i)}, \mathbf{K}^{\mathbf{c}(j)} \right\rangle_F \qquad i, j = 1, \ldots, L.$$

Optimal kernel weights are given by $\boldsymbol{\mu}^* = \frac{\mathbf{v}^*}{\|\mathbf{v}^*\|}$, where $\mathbf{v}^*$ is the solution to the above QP.

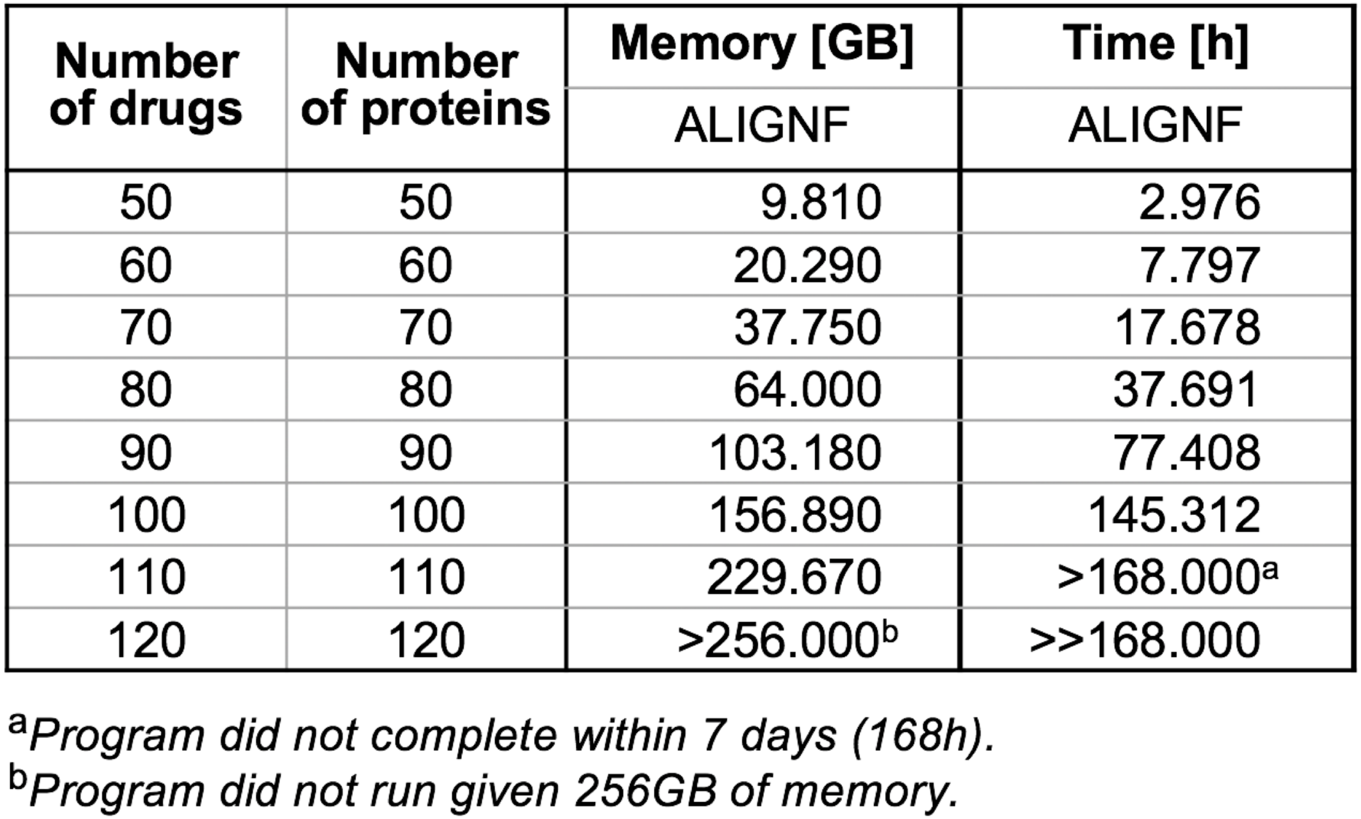
Aalto University
School of Science

Cortes C *et al.* (2012) "Algorithms for learning kernels based on centered alignment". Journal of Machine Learning Research.

MKL with pairwise kernels

10 drug kernels
12 protein kernels

Again, the immense size of pairwise kernel spaces
makes the model training infeasible in practical applications.

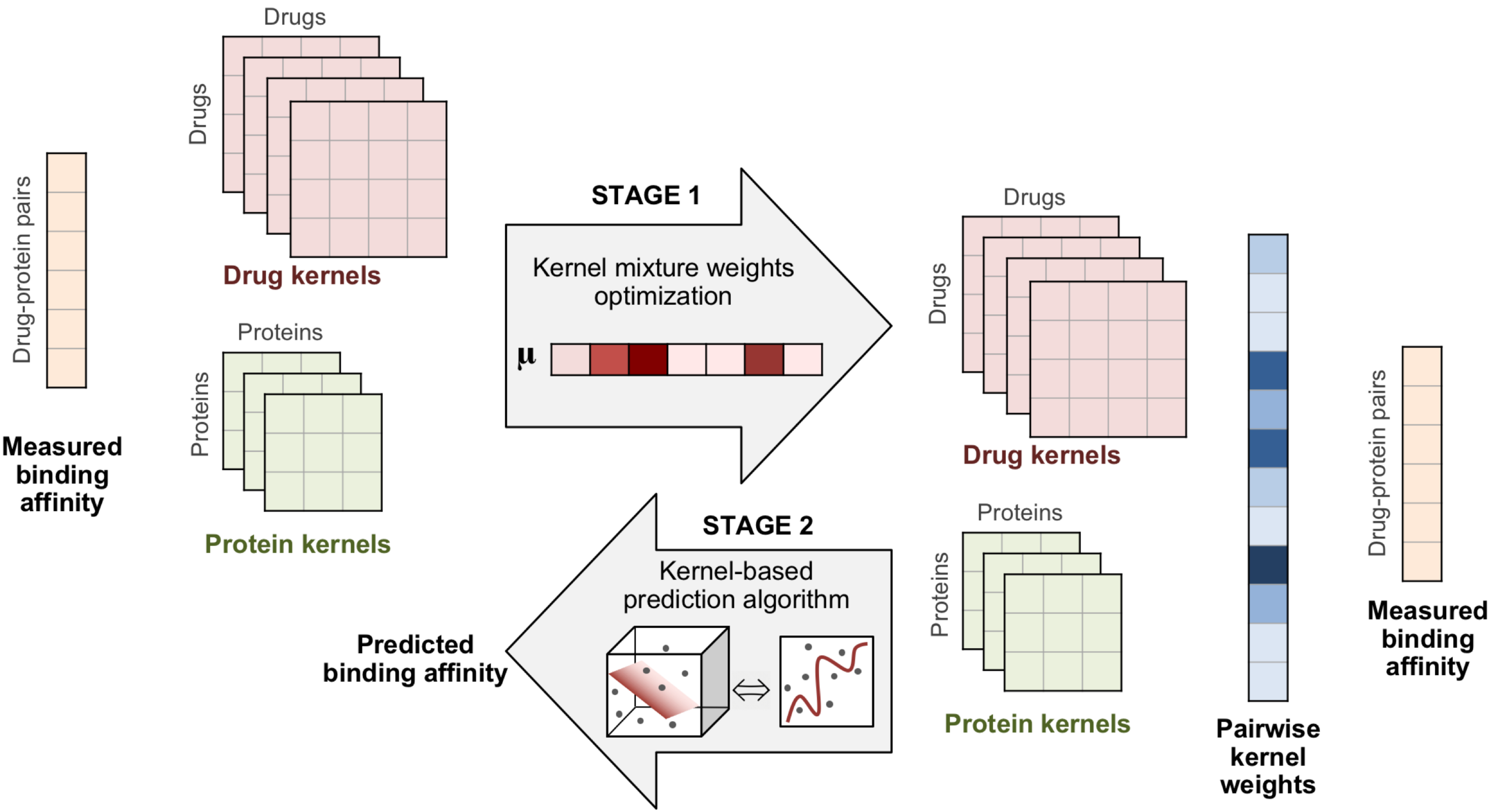


Number of drugs	Number of proteins	Memory [GB]	Time [h]
		ALIGNF	ALIGNF
50	50	9.810	2.976
60	60	20.290	7.797
70	70	37.750	17.678
80	80	64.000	37.691
90	90	103.180	77.408
100	100	156.890	145.312
110	110	229.670	>168.000[a]
120	120	>256.000[b]	>>168.000

[a] *Program did not complete within 7 days (168h).*

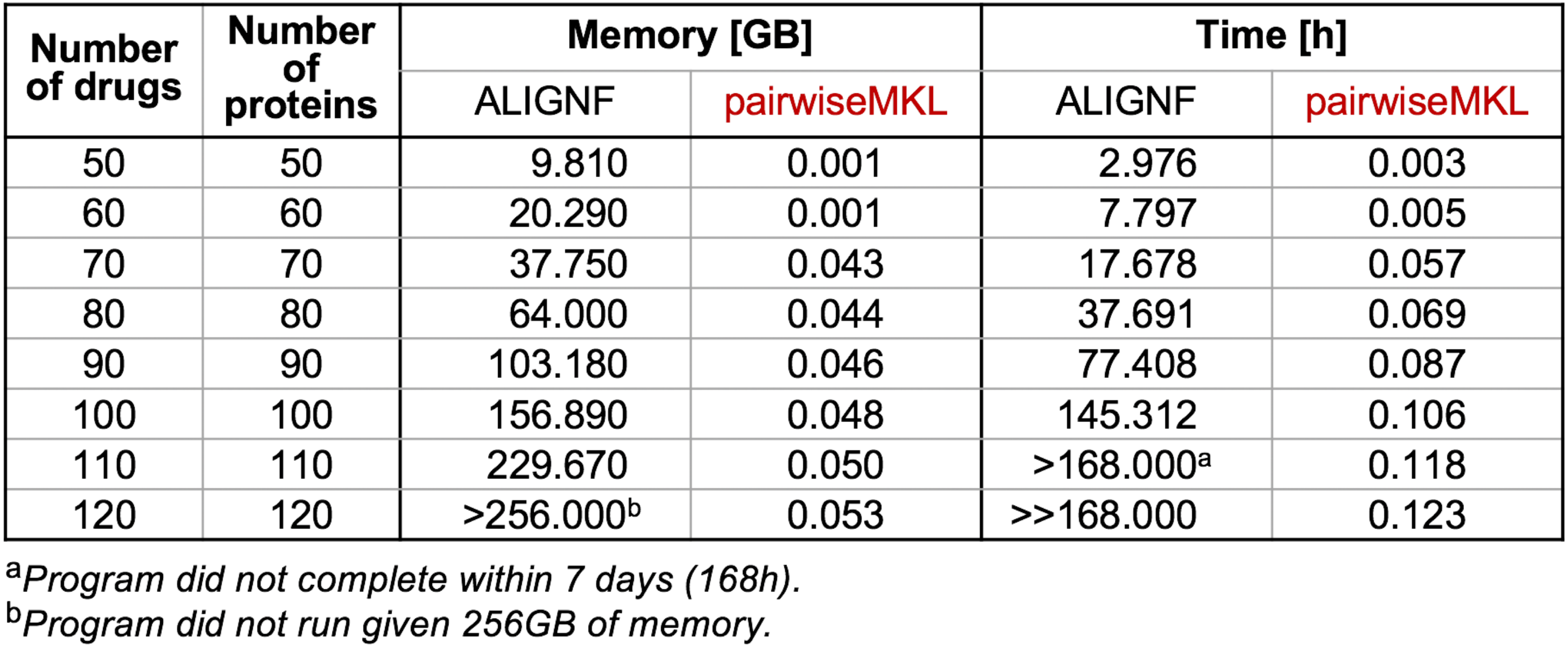
[b] *Program did not run given 256GB of memory.*

pairwiseMKL (i.e. ALIGNF for pairwise kernels)



Cichonska A *et al.* (2018) "Learning with multiple pairwise kernels for drug bioactivity prediction". Bioinformatics.

pairwiseMKL



Number of drugs	Number of proteins	Memory [GB]		Time [h]	
		ALIGNF	pairwiseMKL	ALIGNF	pairwiseMKL
50	50	9.810	0.001	2.976	0.003
60	60	20.290	0.001	7.797	0.005
70	70	37.750	0.043	17.678	0.057
80	80	64.000	0.044	37.691	0.069
90	90	103.180	0.046	77.408	0.087
100	100	156.890	0.048	145.312	0.106
110	110	229.670	0.050	>168.000[a]	0.118
120	120	>256.000[b]	0.053	>>168.000	0.123

[a] *Program did not complete within 7 days (168h).*

[b] *Program did not run given 256GB of memory.*

Cichonska A *et al*. (2018) “Learning with multiple pairwise kernels for drug bioactivity prediction”. Bioinformatics.

pairwiseMKL

- **Bottleneck** in using ALIGNF with pairwise kernels is the centering of the kernel, required by the algorithm

$$\mathbf{K}^{\mathbf{c}} = \mathbf{C}(\mathbf{K}_D \otimes \mathbf{K}_P)\mathbf{C}$$

- **Key contribution**: factorized form for the centering operator

$$\mathbf{C} = \left[\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{N}\right] = \sum_{q=1}^{2} \mathbf{Q}_D^{(q)} \otimes \mathbf{Q}_P^{(q)}$$

- Now, the quantities (inner products) required by ALIGNF can be computed without explicitly building the huge pairwise kernels

$$(\mathbf{M})_{ij} = \left\langle \mathbf{K}^{\mathbf{c}(i)}, \mathbf{K}^{\mathbf{c}(j)} \right\rangle_F = \sum_{q=1}^{2} \sum_{r=1}^{2} \mathrm{tr}(\mathbf{Q}_D^{(q)} \mathbf{K}_D^{(i)} \mathbf{Q}_D^{(r)} \mathbf{K}_D^{(j)})\, \mathrm{tr}(\mathbf{Q}_P^{(q)} \mathbf{K}_P^{(i)} \mathbf{Q}_P^{(r)} \mathbf{K}_P^{(j)})$$

$(\mathbf{a})_i \;\; = \left\langle \mathbf{K}^{\mathbf{c}(i)}, \mathbf{K}_y \right\rangle_F = \langle \mathbf{y}, \mathbf{h} \rangle,$ where

$$\mathbf{h} = \sum_{q=1}^{2} \sum_{r=1}^{2} \mathrm{vec}\left((\mathbf{Q}_P^{(q)} \mathbf{K}_P^{(i)} \mathbf{Q}_P^{(r)}) \mathbf{Y} (\mathbf{Q}_D^{(q)} \mathbf{K}_D^{(i)} \mathbf{Q}_D^{(r)}) \right) \qquad \mathbf{y} = \mathrm{vec}(\mathbf{Y})$$

pairwiseMKL

- In multiple kernel learning for classification tasks, it is usual to choose the **response kernel** of the form:

$$(\mathbf{K}_y)_{ij} = y_i y_j = \begin{cases} +1, & \text{if } \; y_i = y_j \\ -1, & \text{if } \; y_i \neq y_j \end{cases}$$

- This works in binary classification, since positive and negative classes are perfectly separated.

- However, it fails completely with real values, as large numbers get large kernel values, and small numbers get small kernel values.

$$y_i = y_j = 1 \quad \Rightarrow \quad y_i y_j = 1$$

$$y_i = 1,\, y_j = 1000 \quad \Rightarrow \quad y_i y_j = 1000$$

- The Gaussian kernel would work better as it is translation invariant.

$$(\mathbf{K}_y)_{ij} = \exp\left(-\frac{\left\|y_i - y_j\right\|^2}{2\sigma^2}\right)$$

- However, the factorized centering procedure requires explicit representation of the response matrix $\mathbf{Y}$ ($\mathbf{y} = \mathrm{vec}(\mathbf{Y})$).

pairwiseMKL

- We start fitting a mixture of Gaussians onto the frequency histogram of the response variable, obtaining a density f(b) for each bin b.

- For each value y, a window of S bins around it is defined: $(b_y, b_y + 1, \dots, b_y + S - 1)$

- **Feature vector for** y is read off the bin densities, and normalized.

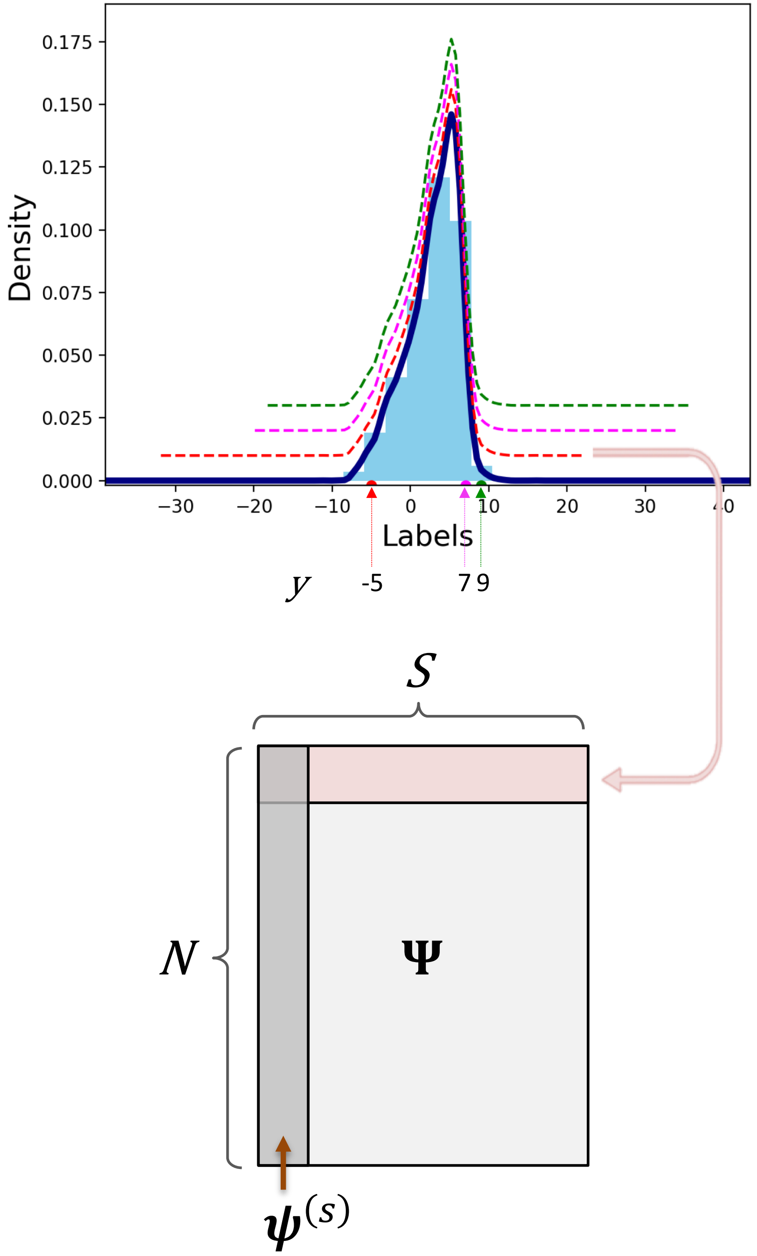
$$\boldsymbol{\psi}(y) = (f(b_y), f(b_y + 1), \dots, f(b_y + S - 1))$$

- **Kernel**: sum of products of S bin densities.

$$\mathbf{K}_y = \sum_{s=1}^{S} \boldsymbol{\psi}^{(s)}\, \boldsymbol{\psi}^{(s)T}$$

$$\boldsymbol{\psi}^{(s)} = \left(\psi^{(s)}(y_1), \dots, \psi^{(s)}(y_N)\right)$$

- Intuitively, the kernel measures the alignment of the original density f with f shifted by b_{yi}-b_{yj} bins.



pairwiseMKL

- We start fitting a mixture of Gaussians onto the frequency histogram of the response variable, obtaining a density f(b) for each bin b.

- For each value y, a window of S bins around it is defined: $(b_y, b_y + 1, \dots, b_y + S - 1)$

- **Feature vector for** y is read off the bin densities, and normalized.

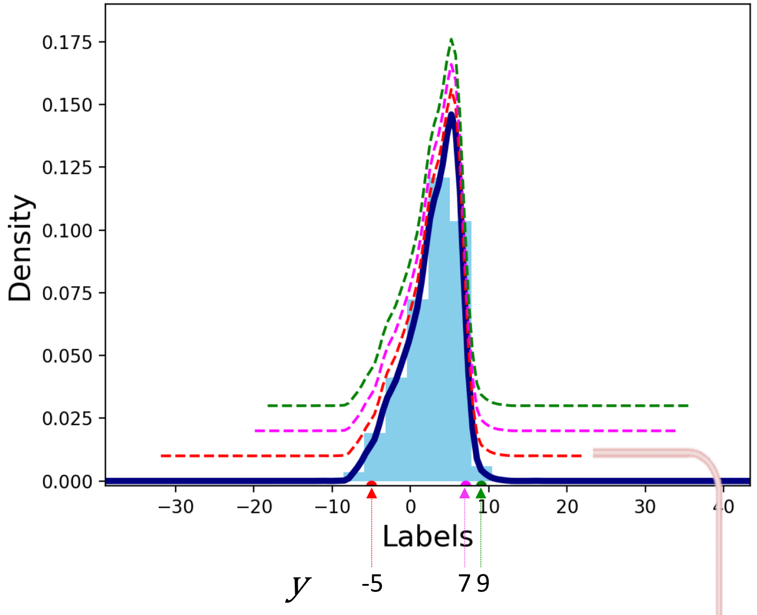
$$\boldsymbol{\psi}(y) = (f(b_y), f(b_y + 1), \dots, f(b_y + S - 1))$$

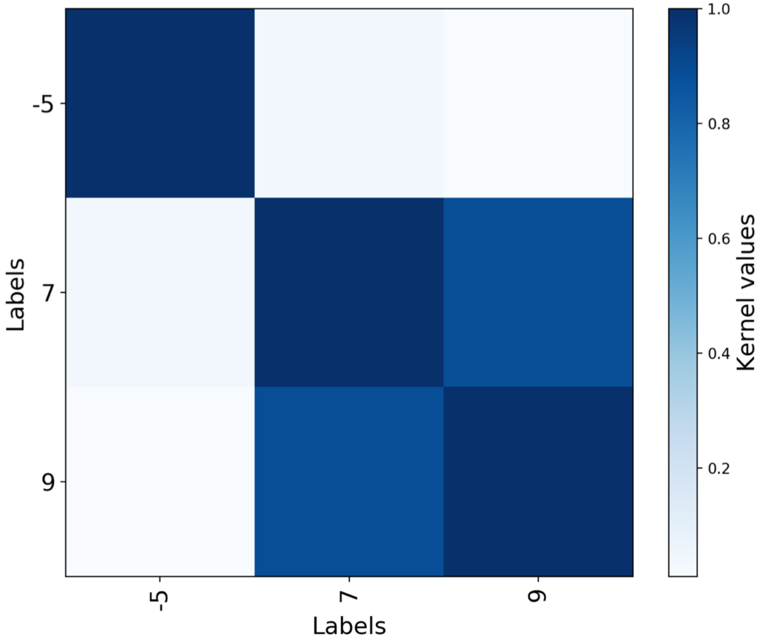
- **Kernel**: sum of products of S bin densities.

$$\mathbf{K}_y = \sum_{s=1}^{S} \boldsymbol{\psi}^{(s)} \boldsymbol{\psi}^{(s)T}$$

$$\boldsymbol{\psi}^{(s)} = \left(\psi^{(s)}(y_1), \dots, \psi^{(s)}(y_N)\right)$$

- Intuitively, the kernel measures the alignment of the original density f with f shifted by b_{yi}-b_{yj} bins.





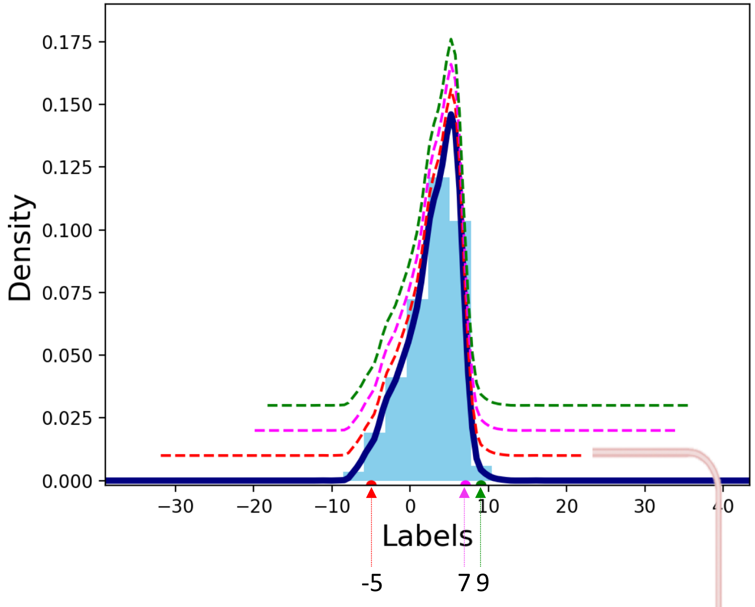
pairwiseMKL

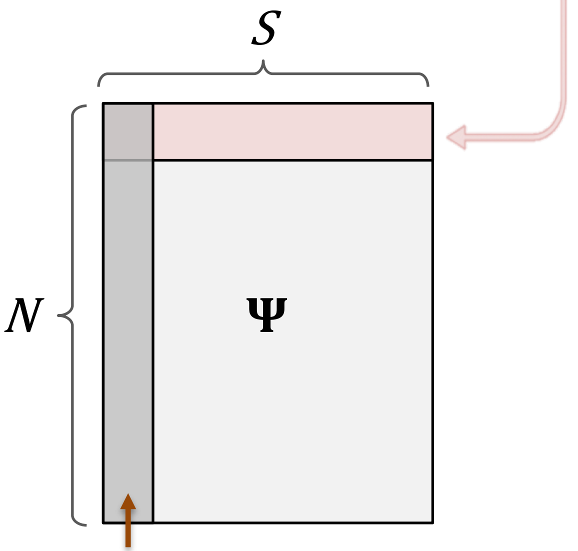
$$\mathbf{K}_y = \sum_{s=1}^{S} \boldsymbol{\psi}^{(s)} \boldsymbol{\psi}^{(s)T}$$

- We can now compute the centered kernel alignment between each input kernel and the Gaussian response kernel:

$$(\mathbf{a})_i \;= \left\langle \mathbf{K}^{\mathbf{c}(i)}, \mathbf{K}_y \right\rangle_F = \sum_{s=1}^{S} \left\langle \boldsymbol{\psi}^{(s)}, \mathbf{w} \right\rangle,$$

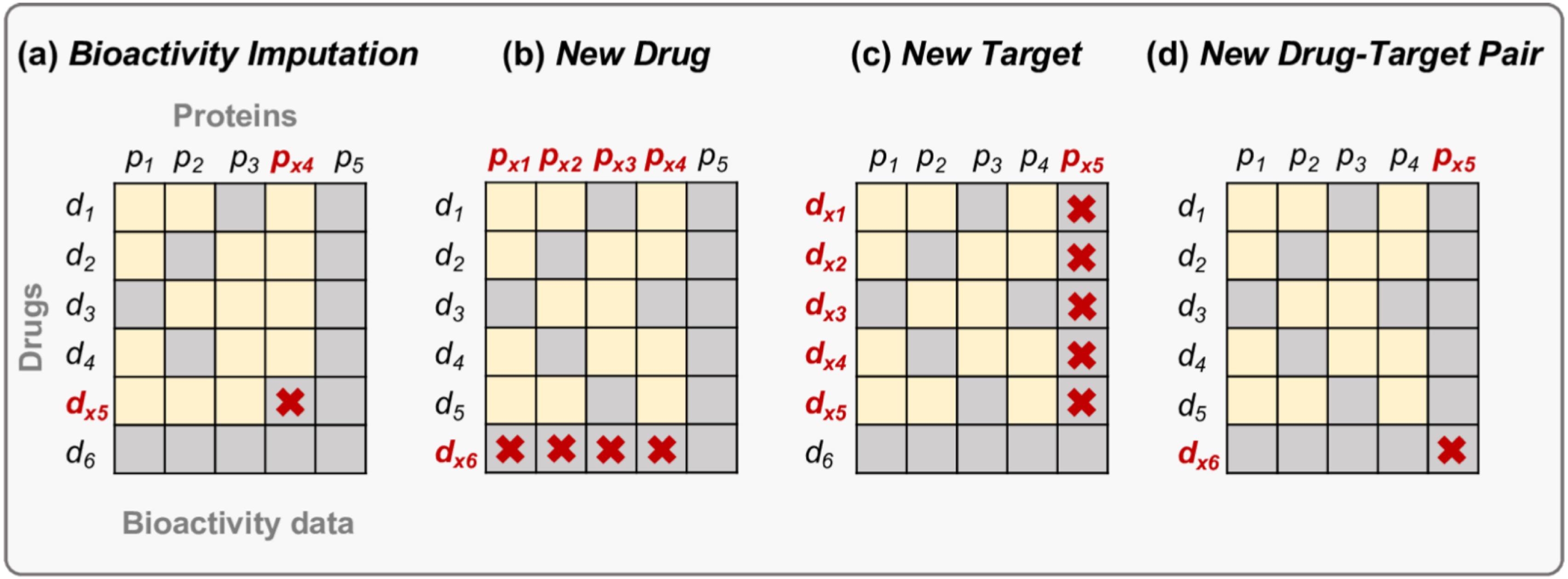
$$\mathbf{w} = \sum_{q=1}^{2} \sum_{r=1}^{2} \mathrm{vec}\left((\mathbf{Q}_P^{(q)} \mathbf{K}_P^{(i)} \mathbf{Q}_P^{(r)}) \mathbf{Z} (\mathbf{Q}_D^{(q)} \mathbf{K}_D^{(i)} \mathbf{Q}_D^{(r)}) \right)$$





$$\boldsymbol{\psi}^{(s)} = \mathrm{vec}(\mathbf{Z})$$

Pairwise prediction scenarios



(a) ***Bioactivity Imputation***

	p_1	p_2	p_3	$\boldsymbol{p_{x4}}$	p_5
d_1	🟨	🟨	⬜	🟨	⬜
d_2	🟨	⬜	🟨	🟨	⬜
d_3	⬜	🟨	🟨	🟨	⬜
d_4	🟨	⬜	🟨	🟨	⬜
$\boldsymbol{d_{x5}}$	🟨	🟨	🟨	✖	⬜
d_6	⬜	⬜	⬜	⬜	⬜

(b) ***New Drug***

	$\boldsymbol{p_{x1}}$	$\boldsymbol{p_{x2}}$	$\boldsymbol{p_{x3}}$	$\boldsymbol{p_{x4}}$	p_5
d_1	🟨	🟨	⬜	🟨	⬜
d_2	🟨	⬜	🟨	🟨	⬜
d_3	⬜	🟨	🟨	⬜	⬜
d_4	🟨	⬜	🟨	🟨	⬜
d_5	🟨	🟨	⬜	🟨	⬜
$\boldsymbol{d_{x6}}$	✖	✖	✖	✖	⬜

(c) ***New Target***

	p_1	p_2	p_3	p_4	$\boldsymbol{p_{x5}}$
$\boldsymbol{d_{x1}}$	🟨	🟨	⬜	🟨	✖
$\boldsymbol{d_{x2}}$	🟨	⬜	🟨	🟨	✖
$\boldsymbol{d_{x3}}$	⬜	🟨	🟨	⬜	✖
$\boldsymbol{d_{x4}}$	🟨	⬜	🟨	🟨	✖
$\boldsymbol{d_{x5}}$	🟨	🟨	⬜	🟨	✖
d_6	⬜	⬜	⬜	⬜	⬜

(d) ***New Drug-Target Pair***

	p_1	p_2	p_3	p_4	$\boldsymbol{p_{x5}}$
d_1	🟨	🟨	⬜	🟨	⬜
d_2	🟨	⬜	🟨	🟨	⬜
d_3	⬜	🟨	🟨	⬜	⬜
d_4	🟨	⬜	🟨	🟨	⬜
d_5	🟨	🟨	⬜	🟨	⬜
$\boldsymbol{d_{x6}}$	⬜	⬜	⬜	⬜	✖

🟨 Experimentally-measured binding affinity

⬜ Unmeasured binding affinity

✖ Unmeasured binding affinity to be predicted

d	Drug
p	Protein
(d_x, p_x)	Query drug-protein pair, the binding affinity of which is to be predicted

Cichonska A *et al*. (2017) "Computational-experimental approach to drug-target interaction mapping: A case study on kinase inhibitors". PLOS Computational Biology.

IDG-DREAM Drug Kinase Binding Prediction Challenge

DREAM CHALLENGES powered by Sage Bionetworks
FIMM
IDG
Icahn School of Medicine at Mount Sinai
SageBionetworks
SGC
UNC Eshelman School of Pharmacy
UNIVERSITY OF COPENHAGEN

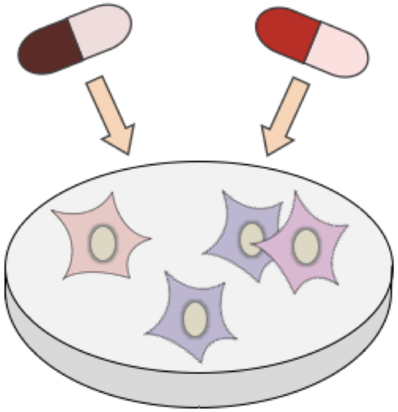
www.synapse.org/DrugKinaseChallenge

OVERALL AIMS

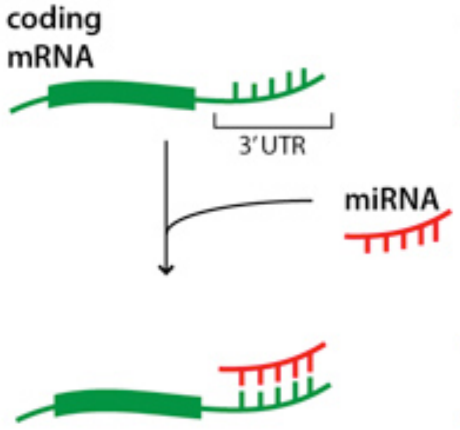
- To evaluate machine learning models as systematic tools for guiding drug-protein mapping efforts to prioritize most potent and selective agents for further experimental evaluation.
- The participating teams are challenged with two overall questions:
  - **What are the best machine learning approaches for predicting drug-protein binding affinities?**
  - **What are the most predictive features for drug compounds and protein targets?**
- Specific focus on kinase inhibitors, due to their clinical importance, toward extending the druggability of human kinome space.

Other pairwise learning problems in bioinformatics

- ➢ Drug response in cancer cell line prediction.



- ➢ mRNA-miRNA interaction prediction.



- ➢ Protein-protein interaction prediction.