

Discriminant Mixture of 3D Molecular Surface Models

Pascal Lamblin

Joint work with Yoshua Bengio, Dan Popovici, Benoit Crompton
and Pierre-Jean L'Heureux

- 1 Intro
 - QSAR and Virtual Screening
- 2 The 3D Surface Model
 - Surface and Surface Template
 - Theoretical Motivation
 - The Alignment Process
 - Alignment Results
- 3 Discriminant Architecture
 - The Scores
 - The architecture
- 4 Results and Future Work
 - Results
 - Future Work
 - Conclusion

QSAR

- Quantitative Structure-Activity Relationship
- Try to predict the activity of a molecule from its structure (its formula)
- Activity: against some predefined target

QSAR

- Quantitative Structure-Activity Relationship
- Try to predict the activity of a molecule from its structure (its formula)
- Activity: against some predefined target

QSAR

- Quantitative Structure-Activity Relationship
- Try to predict the activity of a molecule from its structure (its formula)
- Activity: against some predefined target

Virtual Screening

- Part of the process of drug discovery (pharmaceutical industry)
- **Screening:** find compounds active against an interesting target
- **Virtual:** without testing the actual chemical reaction
- We don't have much information on the target (we cannot use other computational chemistry tools)
- Use data banks full of molecules, only a small fraction are active
- We have samples of known (actually tested) actives and inactives

Virtual Screening

- Part of the process of drug discovery (pharmaceutical industry)
- **Screening:** find compounds active against an interesting target
- **Virtual:** without testing the actual chemical reaction
- We don't have much information on the target (we cannot use other computational chemistry tools)
- Use data banks full of molecules, only a small fraction are active
- We have samples of known (actually tested) actives and inactives

Virtual Screening

- Part of the process of drug discovery (pharmaceutical industry)
- **Screening:** find compounds active against an interesting target
- **Virtual:** without testing the actual chemical reaction
- We don't have much information on the target (we cannot use other computational chemistry tools)
- Use data banks full of molecules, only a small fraction are active
- We have samples of known (actually tested) actives and inactives

Virtual Screening

- Part of the process of drug discovery (pharmaceutical industry)
- **Screening:** find compounds active against an interesting target
- **Virtual:** without testing the actual chemical reaction
- We don't have much information on the target (we cannot use other computational chemistry tools)
- Use data banks full of molecules, only a small fraction are active
- We have samples of known (actually tested) actives and inactives

Virtual Screening

- Part of the process of drug discovery (pharmaceutical industry)
- **Screening:** find compounds active against an interesting target
- **Virtual:** without testing the actual chemical reaction
- We don't have much information on the target (we cannot use other computational chemistry tools)
- Use data banks full of molecules, only a small fraction are active
- We have samples of known (actually tested) actives and inactives

Virtual Screening

- Part of the process of drug discovery (pharmaceutical industry)
- **Screening:** find compounds active against an interesting target
- **Virtual:** without testing the actual chemical reaction
- We don't have much information on the target (we cannot use other computational chemistry tools)
- Use data banks full of molecules, only a small fraction are active
- We have samples of known (actually tested) actives and inactives

Model Overview

- We focus on the surface of the molecule, since it is the part that directly interacts with the target
- We consider the shape of the surface, and the value of some chemical features (electrostatic charges, hydrophobicity, distance to closer O atom...) on this surface
- We suppose that there is a number of “perfect” template surfaces that fit as well as possible into the target site
- Some parts of the template surface (or some properties) can play a more or less important role in determining the activity

Model Overview

- We focus on the surface of the molecule, since it is the part that directly interacts with the target
- We consider the shape of the surface, and the value of some chemical features (electrostatic charges, hydrophobicity, distance to closer O atom...) on this surface
- We suppose that there is a number of “perfect” template surfaces that fit as well as possible into the target site
- Some parts of the template surface (or some properties) can play a more or less important role in determining the activity

Model Overview

- We focus on the surface of the molecule, since it is the part that directly interacts with the target
- We consider the shape of the surface, and the value of some chemical features (electrostatic charges, hydrophobicity, distance to closer O atom...) on this surface
- We suppose that there is a number of “perfect” template surfaces that fit as well as possible into the target site
- Some parts of the template surface (or some properties) can play a more or less important role in determining the activity

Model Overview

- We focus on the surface of the molecule, since it is the part that directly interacts with the target
- We consider the shape of the surface, and the value of some chemical features (electrostatic charges, hydrophobicity, distance to closer O atom...) on this surface
- We suppose that there is a number of “perfect” template surfaces that fit as well as possible into the target site
- Some parts of the template surface (or some properties) can play a more or less important role in determining the activity

Molecular Surface

A molecular surface m is represented as a list of points, where point i has:

- 3D spatial coordinates (x_i^m, y_i^m, z_i^m)
- for each chemical property k , its value $p_{i,k}^m$

Molecular Surface

A molecular surface m is represented as a list of points, where point i has:

- 3D spatial coordinates (x_i^m, y_i^m, z_i^m)
- for each chemical property k , its value $p_{i,k}^m$

Molecular Surface

A molecular surface m is represented as a list of points, where point i has:

- 3D spatial coordinates (x_i^m, y_i^m, z_i^m)
- for each chemical property k , its value $p_{i,k}^m$

Molecular Surface Template

A template t is also represented as a list of points, containing

- 3D spatial coordinates (x_i^t, y_i^t, z_i^t)
- the standard deviation σ_i^t of a 3D spherical Gaussian centered on the spatial coordinates
- for each chemical property k , the mean $\mu_{i,k}^t$ and standard deviation $\sigma_{i,k}^t$ of a Gaussian

Molecular Surface Template

A template t is also represented as a list of points, containing

- 3D spatial coordinates (x_i^t, y_i^t, z_i^t)
- the standard deviation σ_i^t of a 3D spherical Gaussian centered on the spatial coordinates
- for each chemical property k , the mean $\mu_{i,k}^t$ and standard deviation $\sigma_{i,k}^t$ of a Gaussian

Molecular Surface Template

A template t is also represented as a list of points, containing

- 3D spatial coordinates (x_i^t, y_i^t, z_i^t)
- the standard deviation σ_i^t of a 3D spherical Gaussian centered on the spatial coordinates
- for each chemical property k , the mean $\mu_{i,k}^t$ and standard deviation $\sigma_{i,k}^t$ of a Gaussian

Molecular Surface Template

A template t is also represented as a list of points, containing

- 3D spatial coordinates (x_i^t, y_i^t, z_i^t)
- the standard deviation σ_i^t of a 3D spherical Gaussian centered on the spatial coordinates
- for each chemical property k , the mean $\mu_{i,k}^t$ and standard deviation $\sigma_{i,k}^t$ of a Gaussian

Molecular Surface Template

A template t is also represented as a list of points, containing

- 3D spatial coordinates (x_i^t, y_i^t, z_i^t)
- the standard deviation σ_i^t of a 3D spherical Gaussian centered on the spatial coordinates
- for each chemical property k , the mean $\mu_{i,k}^t$ and standard deviation $\sigma_{i,k}^t$ of a Gaussian

And a label $a_t \in \{0, 1\}$ (active or inactive)

Generative Model

- We can consider that the templates define a generative model, and that a molecule surface \mathbf{x} is drawn from template t by this process:
 - 1 For each point i of the template, sample (x_i, y_i, z_i) and $p_{i,k}$
 - 2 Sample a rigid transformation T from a prior distribution $P(T)$
 - 3 Apply T to each point sampled in 1
- The likelihood $P(\mathbf{x}|t)$ can be written:

$$P(\mathbf{x}|t) = \int P(\mathbf{x}|T, t)P(T)dT$$

where $P(\mathbf{x}|T, t) = \prod_i \mathcal{N}(T^{-1}(x_i, y_i, z_i); (x_i^t, y_i^t, z_i^t), \sigma_i^t I)$

- The integral is intractable, so we perform an approximate maximization over T , using ICP
- We train the model parameters discriminatively, because we don't have the exact likelihood

Generative Model

- We can consider that the templates define a generative model, and that a molecule surface \mathbf{x} is drawn from template t by this process:
 - 1 For each point i of the template, sample (x_i, y_i, z_i) and $p_{i,k}$
 - 2 Sample a rigid transformation T from a prior distribution $P(T)$
 - 3 Apply T to each point sampled in 1
- The likelihood $P(\mathbf{x}|t)$ can be written:

$$P(\mathbf{x}|t) = \int P(\mathbf{x}|T, t)P(T)dT$$

where $P(\mathbf{x}|T, t) = \prod_i \mathcal{N}(T^{-1}(x_i, y_i, z_i); (x_i^t, y_i^t, z_i^t), \sigma_i^t I)$

- The integral is intractable, so we perform an approximate maximization over T , using ICP
- We train the model parameters discriminatively, because we don't have the exact likelihood

Generative Model

- We can consider that the templates define a generative model, and that a molecule surface \mathbf{x} is drawn from template t by this process:
 - 1 For each point i of the template, sample (x_i, y_i, z_i) and $p_{i,k}$
 - 2 Sample a rigid transformation T from a prior distribution $P(T)$
 - 3 Apply T to each point sampled in 1
- The likelihood $P(\mathbf{x}|t)$ can be written:

$$P(\mathbf{x}|t) = \int P(\mathbf{x}|T, t)P(T)dT$$

where $P(\mathbf{x}|T, t) = \prod_i \mathcal{N}(T^{-1}(x_i, y_i, z_i); (x_i^t, y_i^t, z_i^t), \sigma_i^t I)$

- The integral is intractable, so we perform an approximate maximization over T , using ICP
- We train the model parameters discriminatively, because we don't have the exact likelihood

Generative Model

- We can consider that the templates define a generative model, and that a molecule surface \mathbf{x} is drawn from template t by this process:
 - 1 For each point i of the template, sample (x_i, y_i, z_i) and $p_{i,k}$
 - 2 Sample a rigid transformation T from a prior distribution $P(T)$
 - 3 Apply T to each point sampled in 1
- The likelihood $P(\mathbf{x}|t)$ can be written:

$$P(\mathbf{x}|t) = \int P(\mathbf{x}|T, t)P(T)dT$$

where $P(\mathbf{x}|T, t) = \prod_i \mathcal{N}(T^{-1}(x_i, y_i, z_i); (x_i^t, y_i^t, z_i^t), \sigma_i^t I)$

- The integral is intractable, so we perform an approximate maximization over T , using ICP
- We train the model parameters discriminatively, because we don't have the exact likelihood

Generative Model

- We can consider that the templates define a generative model, and that a molecule surface \mathbf{x} is drawn from template t by this process:
 - 1 For each point i of the template, sample (x_i, y_i, z_i) and $p_{i,k}$
 - 2 Sample a rigid transformation T from a prior distribution $P(T)$
 - 3 Apply T to each point sampled in 1
- The likelihood $P(\mathbf{x}|t)$ can be written:

$$P(\mathbf{x}|t) = \int P(\mathbf{x}|T, t)P(T)dT$$

where $P(\mathbf{x}|T, t) = \prod_i \mathcal{N}(T^{-1}(x_i, y_i, z_i); (x_i^t, y_i^t, z_i^t), \sigma_i^t I)$

- The integral is intractable, so we perform an approximate maximization over T , using ICP
- We train the model parameters discriminatively, because we don't have the exact likelihood

Generative Model

- We can consider that the templates define a generative model, and that a molecule surface \mathbf{x} is drawn from template t by this process:
 - 1 For each point i of the template, sample (x_i, y_i, z_i) and $p_{i,k}$
 - 2 Sample a rigid transformation T from a prior distribution $P(T)$
 - 3 Apply T to each point sampled in 1
- The likelihood $P(\mathbf{x}|t)$ can be written:

$$P(\mathbf{x}|t) = \int P(\mathbf{x}|T, t)P(T)dT$$

where $P(\mathbf{x}|T, t) = \prod_i \mathcal{N}(T^{-1}(x_i, y_i, z_i); (x_i^t, y_i^t, z_i^t), \sigma_i^t I)$

- The integral is intractable, so we perform an approximate maximization over T , using ICP
- We train the model parameters discriminatively, because we don't have the exact likelihood

Generative Model

- We can consider that the templates define a generative model, and that a molecule surface \mathbf{x} is drawn from template t by this process:
 - 1 For each point i of the template, sample (x_i, y_i, z_i) and $p_{i,k}$
 - 2 Sample a rigid transformation T from a prior distribution $P(T)$
 - 3 Apply T to each point sampled in 1
- The likelihood $P(\mathbf{x}|t)$ can be written:

$$P(\mathbf{x}|t) = \int P(\mathbf{x}|T, t)P(T)dT$$

where $P(\mathbf{x}|T, t) = \prod_i \mathcal{N}(T^{-1}(x_i, y_i, z_i); (x_i^t, y_i^t, z_i^t), \sigma_i^t I)$

- The integral is intractable, so we perform an approximate maximization over T , using ICP
- We train the model parameters discriminatively, because we don't have the exact likelihood

Our Goal

- Learn the templates, so that we know
 - its perfect shape and properties
 - where it is important to have them
- Be able to recognize actives and inactives
- Become rich, healthy, and famous, and live happily ever after

Our Goal

- Learn the templates, so that we know
 - its perfect shape and properties
 - where it is important to have them
- Be able to recognize actives and inactives
- Become rich, healthy, and famous, and live happily ever after

Our Goal

- Learn the templates, so that we know
 - its perfect shape and properties
 - where it is important to have them
- Be able to recognize actives and inactives
- Become rich, healthy, and famous, and live happily ever after

Our Goal

- Learn the templates, so that we know
 - its perfect shape and properties
 - where it is important to have them
- Be able to recognize actives and inactives
- Become rich, healthy, and famous, and live happily ever after

Our Goal

- Learn the templates, so that we know
 - its perfect shape and properties
 - where it is important to have them
- Be able to recognize actives and inactives
- Become rich, healthy, and famous, and live happily ever after

Overview

We need a similarity measure between a template and a surface

- This measure should be invariant by translation and rotation
- Find the most likely spacial alignment (using both geometry and chemical features) by an approximate method: ICP
- Align the template on the molecule surface
- From the aligned surfaces, compute a **score**:

$$\begin{aligned} & -\frac{1}{2} \sum_i w_i \frac{(x_i^t - x_{j_i}^m)^2 + (y_i^t - y_{j_i}^m)^2 + (z_i^t - z_{j_i}^m)^2}{\sigma_i^{t2}} \\ & -\frac{1}{2} \sum_i \sum_k \frac{(\mu_{i,k}^t - p_{j_i,k}^m)^2}{\sigma_{i,k}^{t2}} \\ & -\sum_i \log \sigma_i^t - \sum_i \sum_k \log \sigma_{i,k}^t \end{aligned}$$

Overview

We need a similarity measure between a template and a surface

- This measure should be invariant by translation and rotation
- Find the most likely spacial alignment (using both geometry and chemical features) by an approximate method: ICP
- Align the template on the molecule surface
- From the aligned surfaces, compute a **score**:

$$\begin{aligned} & -\frac{1}{2} \sum_i w_i \frac{(x_i^t - x_{j_i}^m)^2 + (y_i^t - y_{j_i}^m)^2 + (z_i^t - z_{j_i}^m)^2}{\sigma_i^{t2}} \\ & -\frac{1}{2} \sum_i \sum_k \frac{(\mu_{i,k}^t - p_{j_i,k}^m)^2}{\sigma_{i,k}^{t2}} \\ & - \sum_i \log \sigma_i^t - \sum_i \sum_k \log \sigma_{i,k}^t \end{aligned}$$

Overview

We need a similarity measure between a template and a surface

- This measure should be invariant by translation and rotation
- Find the most likely spacial alignment (using both geometry and chemical features) by an approximate method: ICP
- Align the template on the molecule surface
- From the aligned surfaces, compute a **score**:

$$\begin{aligned} & -\frac{1}{2} \sum_i w_i \frac{(x_i^t - x_{j_i}^m)^2 + (y_i^t - y_{j_i}^m)^2 + (z_i^t - z_{j_i}^m)^2}{\sigma_i^{t2}} \\ & -\frac{1}{2} \sum_i \sum_k \frac{(\mu_{i,k}^t - p_{j_i,k}^m)^2}{\sigma_{i,k}^{t2}} \\ & -\sum_i \log \sigma_i^t - \sum_i \sum_k \log \sigma_{i,k}^t \end{aligned}$$

Overview

We need a similarity measure between a template and a surface

- This measure should be invariant by translation and rotation
- Find the most likely spacial alignment (using both geometry and chemical features) by an approximate method: ICP
- Align the template on the molecule surface
- From the aligned surfaces, compute a **score**:

$$\begin{aligned} & -\frac{1}{2} \sum_i w_i \frac{(x_i^t - x_{j_i}^m)^2 + (y_i^t - y_{j_i}^m)^2 + (z_i^t - z_{j_i}^m)^2}{\sigma_i^{t2}} \\ & -\frac{1}{2} \sum_i \sum_k \frac{(\mu_{i,k}^t - p_{j_i,k}^m)^2}{\sigma_{i,k}^{t2}} \\ & - \sum_i \log \sigma_i^t - \sum_i \sum_k \log \sigma_{i,k}^t \end{aligned}$$

Overview

We need a similarity measure between a template and a surface

- This measure should be invariant by translation and rotation
- Find the most likely spacial alignment (using both geometry and chemical features) by an approximate method: ICP
- Align the template on the molecule surface
- From the aligned surfaces, compute a **score**:

$$\begin{aligned} & -\frac{1}{2} \sum_i w_i \frac{(x_i^t - x_{j_i}^m)^2 + (y_i^t - y_{j_i}^m)^2 + (z_i^t - z_{j_i}^m)^2}{\sigma_i^t{}^2} \\ & -\frac{1}{2} \sum_i \sum_k \frac{(\mu_{i,k}^t - p_{j_i,k}^m)^2}{\sigma_{i,k}^t{}^2} \\ & - \sum_i \log \sigma_i^t - \sum_i \sum_k \log \sigma_{i,k}^t \end{aligned}$$

The alignment method: ICP

Iterative method, usual for registration of 2D or 3D shapes

- For each point i on the first surface, find its nearest neighbor j_i on the other surface.

$$\forall i, j_i = \operatorname{argmin}_j (x_i^t - x_j^m)^2 + (y_i^t - y_j^m)^2 + (z_i^t - z_j^m)^2$$

- Compute the rigid transformation minimizing the sum of square distances between the pairs of nearest neighbors

$$(R, T) = \min \sum_i (\hat{x}_i^t - x_{j_i}^m)^2 + (\hat{y}_i^t - y_{j_i}^m)^2 + (\hat{z}_i^t - z_{j_i}^m)^2$$

with $(\hat{x}_i^t, \hat{y}_i^t, \hat{z}_i^t)' = R(x_i^t, y_i^t, z_i^t) + T$

- Apply this transformation and iterate, until convergence.

The alignment method: ICP

Iterative method, usual for registration of 2D or 3D shapes

- For each point i on the first surface, find its nearest neighbor j_i on the other surface.

$$\forall i, j_i = \underset{j}{\operatorname{argmin}} (x_i^t - x_j^m)^2 + (y_i^t - y_j^m)^2 + (z_i^t - z_j^m)^2$$

- Compute the rigid transformation minimizing the sum of square distances between the pairs of nearest neighbors

$$(R, T) = \min \sum_i (\hat{x}_i^t - x_{j_i}^m)^2 + (\hat{y}_i^t - y_{j_i}^m)^2 + (\hat{z}_i^t - z_{j_i}^m)^2$$

with $(\hat{x}_i^t, \hat{y}_i^t, \hat{z}_i^t)' = R(x_i^t, y_i^t, z_i^t) + T$

- Apply this transformation and iterate, until convergence.

The alignment method: ICP

Iterative method, usual for registration of 2D or 3D shapes

- For each point i on the first surface, find its nearest neighbor j_i on the other surface.

$$\forall i, j_i = \operatorname{argmin}_j (x_i^t - x_j^m)^2 + (y_i^t - y_j^m)^2 + (z_i^t - z_j^m)^2$$

- Compute the rigid transformation minimizing the sum of square distances between the pairs of nearest neighbors

$$(R, T) = \min \sum_i (\hat{x}_i^t - x_{j_i}^m)^2 + (\hat{y}_i^t - y_{j_i}^m)^2 + (\hat{z}_i^t - z_{j_i}^m)^2$$

with $(\hat{x}_i^t, \hat{y}_i^t, \hat{z}_i^t)' = R(x_i^t, y_i^t, z_i^t) + T$

- Apply this transformation and iterate, until convergence.

The alignment method: ICP

Iterative method, usual for registration of 2D or 3D shapes

- For each point i on the first surface, find its nearest neighbor j_i on the other surface.

$$\forall i, j_i = \operatorname{argmin}_j (x_i^t - x_j^m)^2 + (y_i^t - y_j^m)^2 + (z_i^t - z_j^m)^2$$

- Compute the rigid transformation minimizing the sum of square distances between the pairs of nearest neighbors

$$(R, T) = \min \sum_i (\hat{x}_i^t - x_{j_i}^m)^2 + (\hat{y}_i^t - y_{j_i}^m)^2 + (\hat{z}_i^t - z_{j_i}^m)^2$$

with $(\hat{x}_i^t, \hat{y}_i^t, \hat{z}_i^t)' = R(x_i^t, y_i^t, z_i^t) + T$

- Apply this transformation and iterate, until convergence.

The alignment method: ICP

Iterative method, usual for registration of 2D or 3D shapes

- For each point i on the first surface, find its nearest neighbor j_i on the other surface.

$$\forall i, j_i = \operatorname{argmin}_j (x_i^t - x_j^m)^2 + (y_i^t - y_j^m)^2 + (z_i^t - z_j^m)^2$$

- Compute the rigid transformation minimizing the sum of square distances between the pairs of nearest neighbors

$$(R, T) = \min \sum_i (\hat{x}_i^t - x_{j_i}^m)^2 + (\hat{y}_i^t - y_{j_i}^m)^2 + (\hat{z}_i^t - z_{j_i}^m)^2$$

$$\text{with } (\hat{x}_i^t, \hat{y}_i^t, \hat{z}_i^t)' = R(x_i^t, y_i^t, z_i^t) + T$$

- Apply this transformation and iterate, until convergence.

Since ICP is sensitive to local minima, we try different initial conditions

The modified method

Use also chemical features and template's deviations during the nearest-neighbors computations

- Geometry only:

$$\forall i, j_i = \operatorname{argmin}_j (x_i^t - x_j^m)^2 + (y_i^t - y_j^m)^2 + (z_i^t - z_j^m)^2$$

- With chemical features and deviations:

$$\forall i, j_i = \operatorname{argmin}_j \frac{(x_i^t - x_j^m)^2 + (y_i^t - y_j^m)^2 + (z_i^t - z_j^m)^2}{\sigma_i^{t2}} + \sum_k \frac{(\mu_{i,k}^t - p_{j,k}^m)^2}{\sigma_{i,k}^{t2}}$$

The modified method

Use also chemical features and template's deviations during the nearest-neighbors computations

- Geometry only:

$$\forall i, j_i = \operatorname{argmin}_j (x_i^t - x_j^m)^2 + (y_i^t - y_j^m)^2 + (z_i^t - z_j^m)^2$$

- With chemical features and deviations:

$$\forall i, j_i = \operatorname{argmin}_j \frac{(x_i^t - x_j^m)^2 + (y_i^t - y_j^m)^2 + (z_i^t - z_j^m)^2}{\sigma_i^{t2}} + \sum_k \frac{(\mu_{i,k}^t - p_{j,k}^m)^2}{\sigma_{i,k}^{t2}}$$

The modified method

Use also chemical features and template's deviations during the nearest-neighbors computations

- Geometry only:

$$\forall i, j_i = \operatorname{argmin}_j (x_i^t - x_j^m)^2 + (y_i^t - y_j^m)^2 + (z_i^t - z_j^m)^2$$

- With chemical features and deviations:

$$\forall i, j_i = \operatorname{argmin}_j \frac{(x_i^t - x_j^m)^2 + (y_i^t - y_j^m)^2 + (z_i^t - z_j^m)^2}{\sigma_i^{t2}} + \sum_k \frac{(\mu_{i,k}^t - p_{j,k}^m)^2}{\sigma_{i,k}^{t2}}$$

The modified method

Use of chemical distances for weighting

- Without weighting:

$$(R, T) = \min \sum_i ((\hat{x}_i^t - x_{j_i}^m)^2 + (\hat{y}_i^t - y_{j_i}^m)^2 + (\hat{z}_i^t - z_{j_i}^m)^2)$$

- With chemical features and weighting:

$$(R, T) = \min \sum_i w_i (\hat{x}_i^t - x_{j_i}^m)^2 + (\hat{y}_i^t - y_{j_i}^m)^2 + (\hat{z}_i^t - z_{j_i}^m)^2$$

$$\text{with } w_i = \text{sigmoid} \left(\beta \left(\alpha - \sqrt{\sum_k \frac{(\mu_{i,k}^t - p_{j_i,k}^m)^2}{\sigma_{i,k}^t}} \right) \right)$$

The modified method

Use of chemical distances for weighting

- Without weighting:

$$(R, T) = \min \sum_i ((\hat{x}_i^t - x_{j_i}^m)^2 + (\hat{y}_i^t - y_{j_i}^m)^2 + (\hat{z}_i^t - z_{j_i}^m)^2)$$

- With chemical features and weighting:

$$(R, T) = \min \sum_i w_i (\hat{x}_i^t - x_{j_i}^m)^2 + (\hat{y}_i^t - y_{j_i}^m)^2 + (\hat{z}_i^t - z_{j_i}^m)^2$$

$$\text{with } w_i = \text{sigmoid} \left(\beta \left(\alpha - \sqrt{\sum_k \frac{(\mu_{i,k}^t - p_{j_i,k}^m)^2}{\sigma_{i,k}^t}} \right) \right)$$

where $(\hat{x}_i^t, \hat{y}_i^t, \hat{z}_i^t)' = R(x_i^t, y_i^t, z_i^t) + T$

The modified method

Use of chemical distances for weighting

- Without weighting:

$$(R, T) = \min \sum_i ((\hat{x}_i^t - x_{j_i}^m)^2 + (\hat{y}_i^t - y_{j_i}^m)^2 + (\hat{z}_i^t - z_{j_i}^m)^2)$$

- With chemical features and weighting:

$$(R, T) = \min \sum_i w_i (\hat{x}_i^t - x_{j_i}^m)^2 + (\hat{y}_i^t - y_{j_i}^m)^2 + (\hat{z}_i^t - z_{j_i}^m)^2$$

$$\text{with } w_i = \text{sigmoid} \left(\beta \left(\alpha - \sqrt{\sum_k \frac{(\mu_{i,k}^t - p_{j_i,k}^m)^2}{\sigma_{i,k}^t}} \right) \right)$$

where $(\hat{x}_i^t, \hat{y}_i^t, \hat{z}_i^t)' = R(x_i^t, y_i^t, z_i^t) + T$

Visualizing alignments

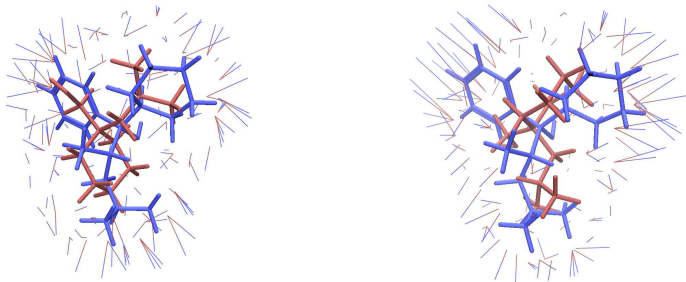


Figure: Without chemical information, with chemical information

Utility of using chemical features

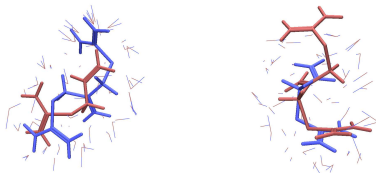


Figure: Without chemical information

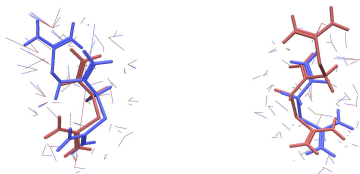


Figure: With chemical information

Formula of the Score

- The alignment score between template t and molecular surface m is:

$$\begin{aligned} S_{mt} = & -\frac{1}{2} \sum_i w_i \frac{(\hat{x}_i^t - x_{j_i}^m)^2 + (\hat{y}_i^t - y_{j_i}^m)^2 + (\hat{z}_i^t - z_{j_i}^m)^2}{\sigma_i^{t^2}} \\ & -\frac{1}{2} \sum_i \sum_k \frac{(\mu_{i,k}^t - p_{j_i,k}^m)^2}{\sigma_{i,k}^{t^2}} \\ & -\sum_i \log \sigma_i^t - \sum_i \sum_k \log \sigma_{i,k}^t \end{aligned}$$

where $(\hat{x}_i^t, \hat{y}_i^t, \hat{z}_i^t)' = R(x_i^t, y_i^t, z_i^t) + T$, and (R, T) is obtained through ICP.

- Approximate likelihood that m was generated from t

Formula of the Score

- The alignment score between template t and molecular surface m is:

$$\begin{aligned} S_{mt} = & -\frac{1}{2} \sum_i w_i \frac{(\hat{x}_i^t - x_{j_i}^m)^2 + (\hat{y}_i^t - y_{j_i}^m)^2 + (\hat{z}_i^t - z_{j_i}^m)^2}{\sigma_i^{t^2}} \\ & -\frac{1}{2} \sum_i \sum_k \frac{(\mu_{i,k}^t - p_{j_i,k}^m)^2}{\sigma_{i,k}^{t^2}} \\ & -\sum_i \log \sigma_i^t - \sum_i \sum_k \log \sigma_{i,k}^t \end{aligned}$$

where $(\hat{x}_i^t, \hat{y}_i^t, \hat{z}_i^t)' = R(x_i^t, y_i^t, z_i^t) + T$, and (R, T) is obtained through ICP.

- Approximate likelihood that m was generated from t

A neural net

- The scores with all templates are the input of an ordinary Neural Network
- The network discriminates between actives and inactives (cross-entropy)

A neural net

- The scores with all templates are the input of an ordinary Neural Network
- The network discriminates between actives and inactives (cross-entropy)

Training

We train the architecture by backpropagating the error gradient

- to the output weights
- to the input weights
- to the template parameters (σ_i^t , $\mu_{i,k}^t$, $\sigma_{i,k}^t$, α and β)

Training

We train the architecture by backpropagating the error gradient

- to the output weights
- to the input weights
- to the template parameters (σ_i^t , $\mu_{i,k}^t$, $\sigma_{i,k}^t$, α and β)

Training

We train the architecture by backpropagating the error gradient

- to the output weights
- to the input weights
- to the template parameters (σ_i^t , $\mu_{i,k}^t$, $\sigma_{i,k}^t$, α and β)

Training

We train the architecture by backpropagating the error gradient

- to the output weights
- to the input weights
- to the template parameters (σ_i^t , $\mu_{i,k}^t$, $\sigma_{i,k}^t$, α and β)

Some implementation tricks

- Since we are more interested in actives, we replicate the active surfaces in the training set, in order to have at least as many active as inactives
- We initialize the templates from randomly-picked actives and inactives from the training set
- The scores need to be normalized in order not to saturate the input neurons, we initialize then learn the normalizing factors

Some implementation tricks

- Since we are more interested in actives, we replicate the active surfaces in the training set, in order to have at least as many active as inactives
- We initialize the templates from randomly-picked actives and inactives from the training set
- The scores need to be normalized in order not to saturate the input neurons, we initialize then learn the normalizing factors

Some implementation tricks

- Since we are more interested in actives, we replicate the active surfaces in the training set, in order to have at least as many active as inactives
- We initialize the templates from randomly-picked actives and inactives from the training set
- The scores need to be normalized in order not to saturate the input neurons, we initialize then learn the normalizing factors

What training achieves

After the training phase, we should have learned:

- the templates, including standard deviations
- a discriminant system, telling us if a surface is likely to be active
- that it is not enough to get rich and famous

What training achieves

After the training phase, we should have learned:

- the templates, including standard deviations
- a discriminant system, telling us if a surface is likely to be active
- that it is not enough to get rich and famous

What training achieves

After the training phase, we should have learned:

- the templates, including standard deviations
- a discriminant system, telling us if a surface is likely to be active
- that it is not enough to get rich and famous

What training achieves

After the training phase, we should have learned:

- the templates, including standard deviations
- a discriminant system, telling us if a surface is likely to be active
- that it is not enough to get rich and famous

Results on McMaster contest data set

- Dataset of molecules tested against E. Coli dihydrofolate reductase
- 33 actives out of 50 000
- We selected 93 inactives (as diverse as possible)
- Comparison with PLS (Partial Least Squares), we reported

$$\text{Lift} = \frac{a_s/n_s}{a/n}$$

Results on McMaster contest data set

- Dataset of molecules tested against E. Coli dihydrofolate reductase
- 33 actives out of 50 000
- We selected 93 inactives (as diverse as possible)
- Comparison with PLS (Partial Least Squares), we reported

$$\text{Lift} = \frac{a_s/n_s}{a/n}$$

Results on McMaster contest data set

- Dataset of molecules tested against E. Coli dihydrofolate reductase
- 33 actives out of 50 000
- We selected 93 inactives (as diverse as possible)
- Comparison with PLS (Partial Least Squares), we reported

$$\text{Lift} = \frac{a_s/n_s}{a/n}$$

Results on McMaster contest data set

- Dataset of molecules tested against E. Coli dihydrofolate reductase
- 33 actives out of 50 000
- We selected 93 inactives (as diverse as possible)
- Comparison with PLS (Partial Least Squares), we reported

$$\text{Lift} = \frac{a_s/n_s}{a/n}$$

Results on McMaster contest data set

- Dataset of molecules tested against E. Coli dihydrofolate reductase
- 33 actives out of 50 000
- We selected 93 inactives (as diverse as possible)
- Comparison with PLS (Partial Least Squares), we reported

$$\text{Lift} = \frac{a_s/n_s}{a/n}$$

SPLIT	SURFACE TEMPLATE LEARNING	PLS
1	173.96	149.11
2	149.11	149.11
3	149.11	173.96

Future Work

- Also update the templates' spatial coordinates during the learning phase (not only the deviations)
- Add molecular-level chemical properties as inputs of the neural net
- Find a way to speed up the computation of the alignments (approximate or faster nearest neighbors finding, better set of initial transformations)
- More experiments...

Future Work

- Also update the templates' spatial coordinates during the learning phase (not only the deviations)
- Add molecular-level chemical properties as inputs of the neural net
- Find a way to speed up the computation of the alignments (approximate or faster nearest neighbors finding, better set of initial transformations)
- More experiments...

Future Work

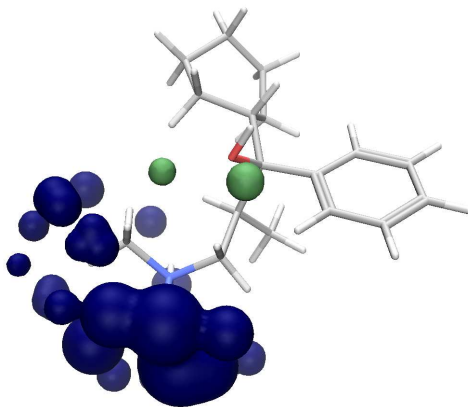
- Also update the templates' spatial coordinates during the learning phase (not only the deviations)
- Add molecular-level chemical properties as inputs of the neural net
- Find a way to speed up the computation of the alignments (approximate or faster nearest neighbors finding, better set of initial transformations)
- More experiments...

Future Work

- Also update the templates' spatial coordinates during the learning phase (not only the deviations)
- Add molecular-level chemical properties as inputs of the neural net
- Find a way to speed up the computation of the alignments (approximate or faster nearest neighbors finding, better set of initial transformations)
- More experiments...

Future Work

Design tools to easily visualize and exploit learned templates



Conclusion

We have a method that:

- gives results as good as state of the art
- produces surface templates, interpretable by chemists
- does not need to compare each pair of molecule in the database

Conclusion

We have a method that:

- gives results as good as state of the art
- produces surface templates, interpretable by chemists
- does not need to compare each pair of molecule in the database

Conclusion

We have a method that:

- gives results as good as state of the art
- produces surface templates, interpretable by chemists
- does not need to compare each pair of molecule in the database

Conclusion

We have a method that:

- gives results as good as state of the art
- produces surface templates, interpretable by chemists
- does not need to compare each pair of molecule in the database

Questions?

The End