Extensions of marginalized graph kernels

Pierre Mahé , Jean-Philippe Vert
Ecole des Mines de Paris
Computational Biology group

Nobuhisa Ueda , Tatsuya Akutsu , Jean-Luc Perret
Kyoto University, Bioinformatics Center

21st ICML , Banff (Canada) , July 4th, 2004.

Introduction

- Motivation : design of state-of-the-art kernel functions for chemical compounds represented as graphs
  - ⋆ many applications in computational biology/chemistry
- Foundations : *Marginalized kernels between labeled graphs* (Kashima et al., 2003)
  - ⋆ general, customizable formulation
- Contribution : 2 modular extensions aiming at :
  - ⋆ increasing the kernel similarity measure
  - ⋆ reducing the kernel complexity

Outline of the talk

- Part 1 : Reminder of the original kernel
- Part 2 : First extension : Label enrichment
- Part 3 : Second extension : Preventing totters
- Part 4 : Results and conclusion

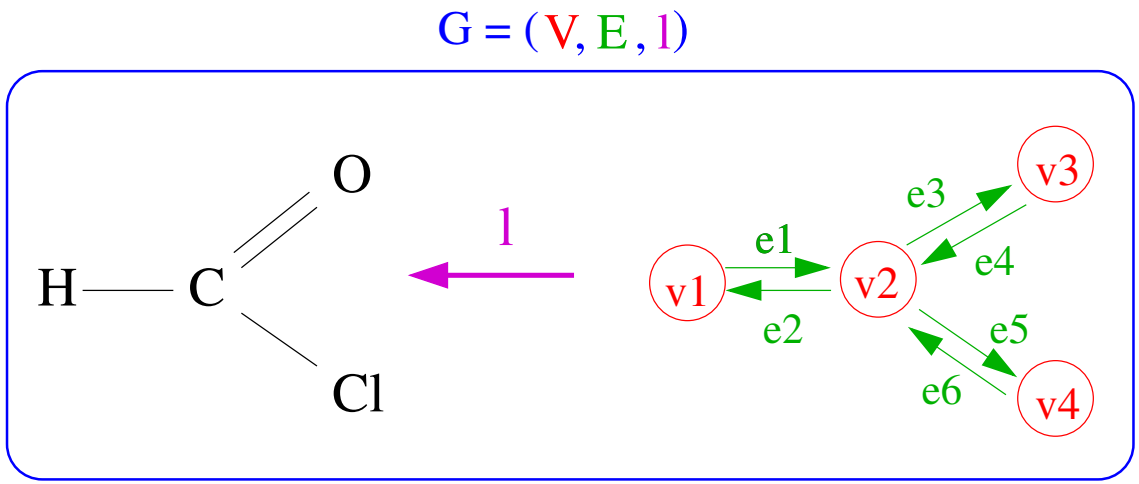
Part 1

Marginalized kernels between labeled graphs

(Kashima et al., 2003)

Definitions

- Labeled graph : $G = (V, E, l)$
  - ⋆ $E \subset (V \times V)$
  - ⋆ $l : V \cup E \rightarrow A$



Definitions

- Labeled graph : $G = (V, E, l)$
  - $\star$ $E \subset (V \times V)$
  - $\star$ $l : V \cup E \rightarrow A$

- Path of a graph : $h = (v_1, ..., v_n)$ with $(v_i, v_{i+1}) \in E$

  h $\rightarrow$ $\begin{cases} \text{length} : |h| = n \\ \text{label} : l(h) = l(v_1) \circ l((v_1, v_2)) \circ ... \circ l((v_{n-1}, v_n)) \circ l(v_n) \end{cases}$

Definitions

- Labeled graph : $G = (V, E, l)$
  - $\star$ $E \subset (V \times V)$
  - $\star$ $l : V \cup E \rightarrow A$

- Path of a graph : $h = (v_1, ..., v_n)$ with $(v_i, v_{i+1}) \in E$

  h $\rightarrow$ $\begin{cases} \text{length} : |h| = n \\ \text{label} : l(h) = l(v_1) \circ l((v_1, v_2)) \circ ... \circ l((v_{n-1}, v_n)) \circ l(v_n) \end{cases}$

- $H(G) = \{$ paths of graph $G \}$

Formulation

$$K(G_1, G_2) = \sum_{\substack{(h_1,h_2) \\ \in H(G_1)\times H(G_2)}} p_{G_1}(h_1) p_{G_2}(h_2) K_L(l(h_1), l(h_2))$$

- p_G = probability distribution on $H(G)$ (i.e : $\sum_{h \in H(G)} p_G(h) = 1$)
- K_L = label kernel between paths labels

$\Rightarrow$ Marginalized Kernel (Tsuda et al., 2002)

$\Rightarrow$ Infinite-dimensional feature-space

Parametrization

- Label kernel = Dirac function
  - ⋆ $K_L(l_1, l_2) = \begin{cases} 1 & \text{if } l_1 = l_2 \\ 0 & \text{otherwise} \end{cases}$
- Probability distribution : 1st order random walk model
  - ⋆ $p_G(v_1 \dots v_n) = p_s(v_1) \prod_{i=2}^{n} p_t(v_i | v_{i-1})$

$\Rightarrow$ NB : Kashima et al. parametrization

Computation

- $K_L = \delta \Longleftrightarrow K = \sum_{(h1,h2)\in\mathcal{C}(G_1,G_2)} p_{G_1}(h_1)p_{G_2}(h_2)$

  with $\mathcal{C}(G_1, G_2) =$ common paths $= \{(h_1, h_2) \in H(G_1) \times H(G_2) \ : \ l(h_1) = l(h_2)\}$

- Graph Product : $\mathcal{G} = G_1 \times G_2 \ : \ H(\mathcal{G}) = \mathcal{C}(G_1, G_2)$

$$\Rightarrow K = \sum_{h\in H(\mathcal{G})} f(h) \text{ with } f(h) = q_s(v_1) \prod_{l=2}^{|h|} q_t(v_{l-1}, v_l)$$

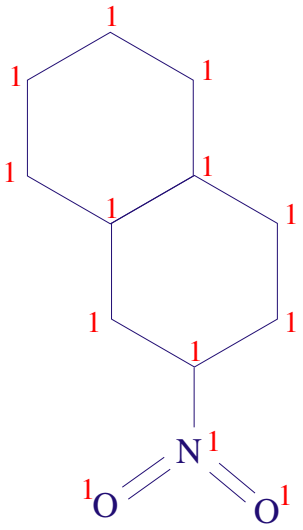
- Complexity $= \mathcal{O}(|\mathcal{G}|^3)$ (inversion of $[Q]_{i,j} = q_t(v_i, v_j)$)
  - ⋆ NB : $|\mathcal{G}| \leq |G_1| \times |G_2|$
  - ⋆ inverse approximation + sparsity $\Rightarrow$ complexity $\ll \mathcal{O}(|\mathcal{G}|^3)$

Part 2

First Extension : Label Enrichment

Label enrichment : algorithm

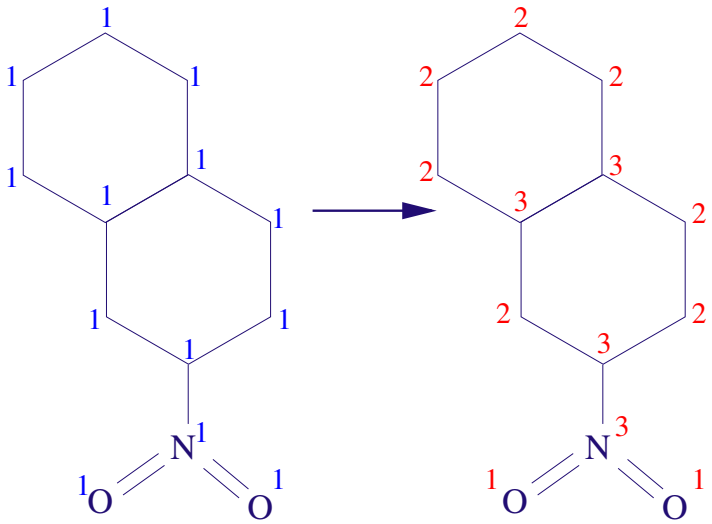
- Enrichment with vertex connectivity properties
  $\longrightarrow$ extended connectivity descriptor :



Original Graph

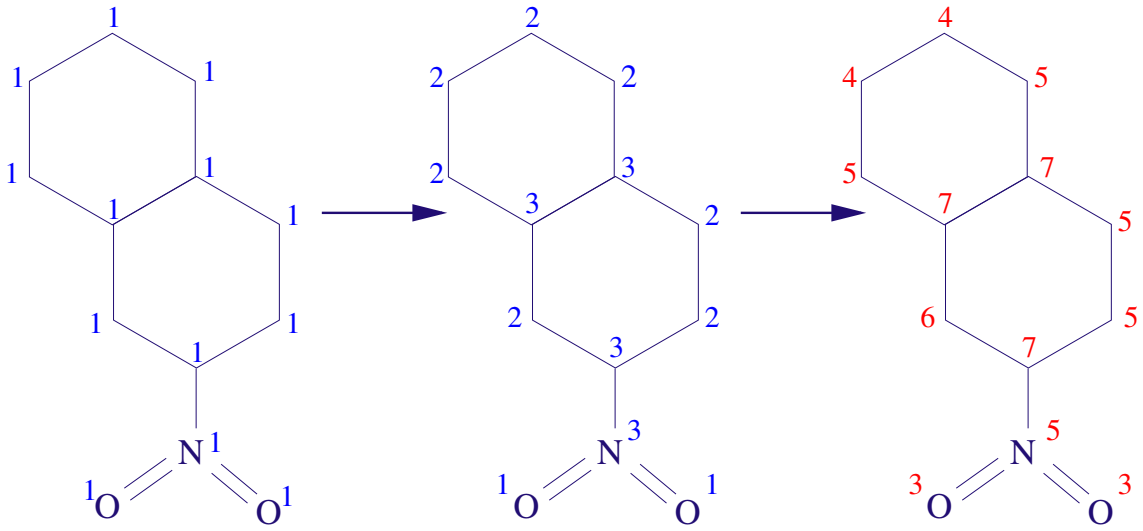
Label enrichment : algorithm

- Enrichment with vertex connectivity properties
  $\longrightarrow$ extended connectivity descriptor :



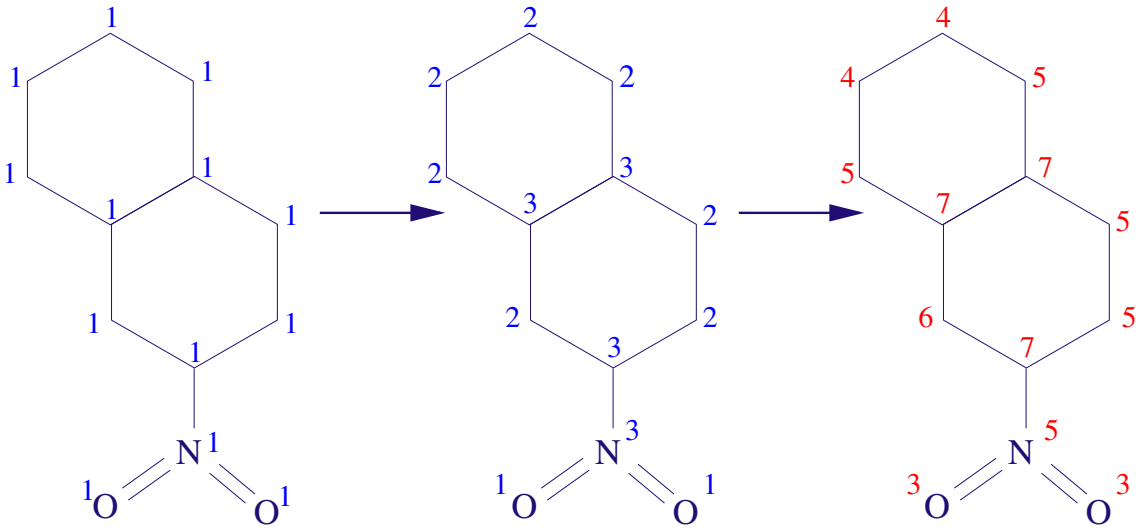
Label enrichment : algorithm

- Enrichment with vertex connectivity properties
  $\longrightarrow$ extended connectivity descriptor :



Label enrichment : algorithm

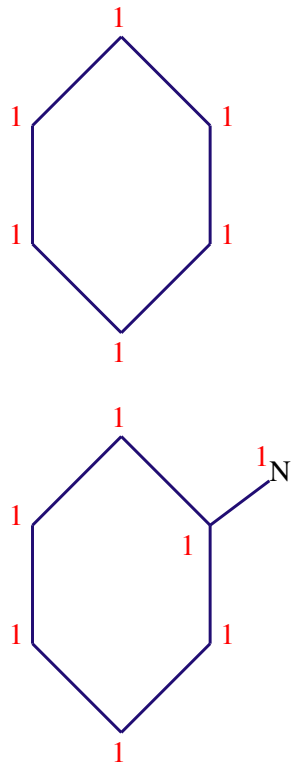
- Enrichment with vertex connectivity properties
  $\longrightarrow$ extended connectivity descriptor :



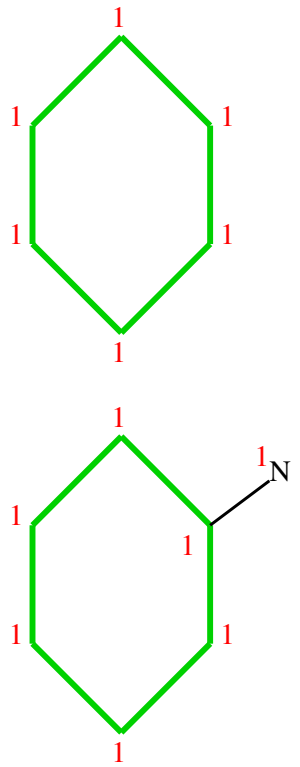
Algorithm :

- $M_0(v) = 1, \ \forall\, v$
- $M_t(v) = \sum_{\mathsf{neig}(v)} M_{t-1}(u)$
- $\Rightarrow$ New label :
  $l_t(v) = l(v) \circ M_t(v)$

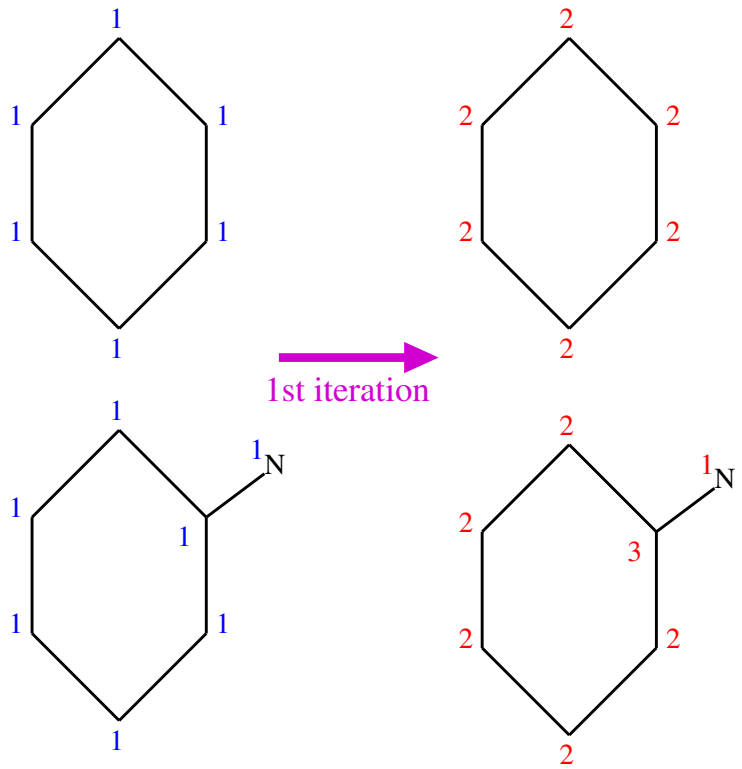
Label enrichment : illustration



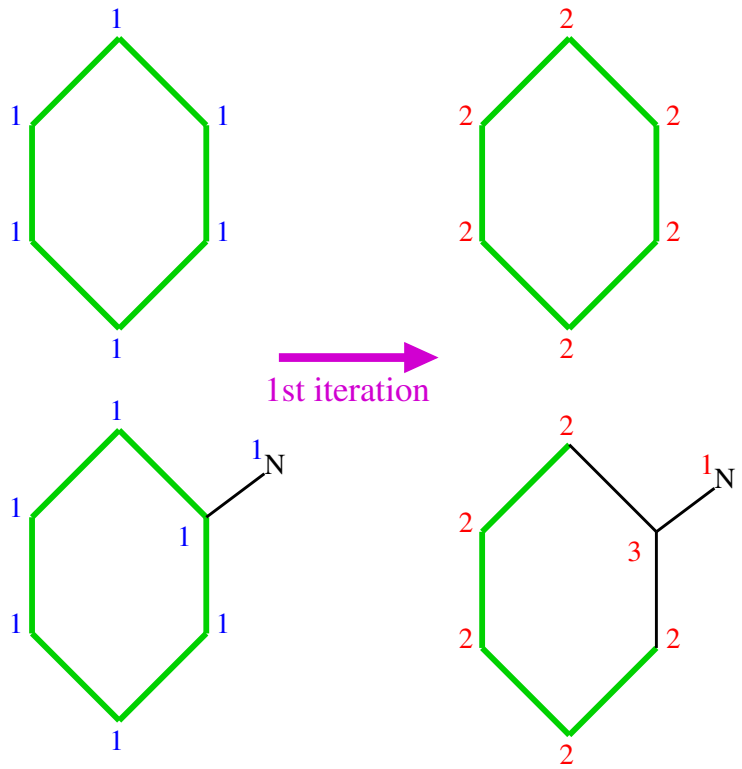
Label enrichment : illustration



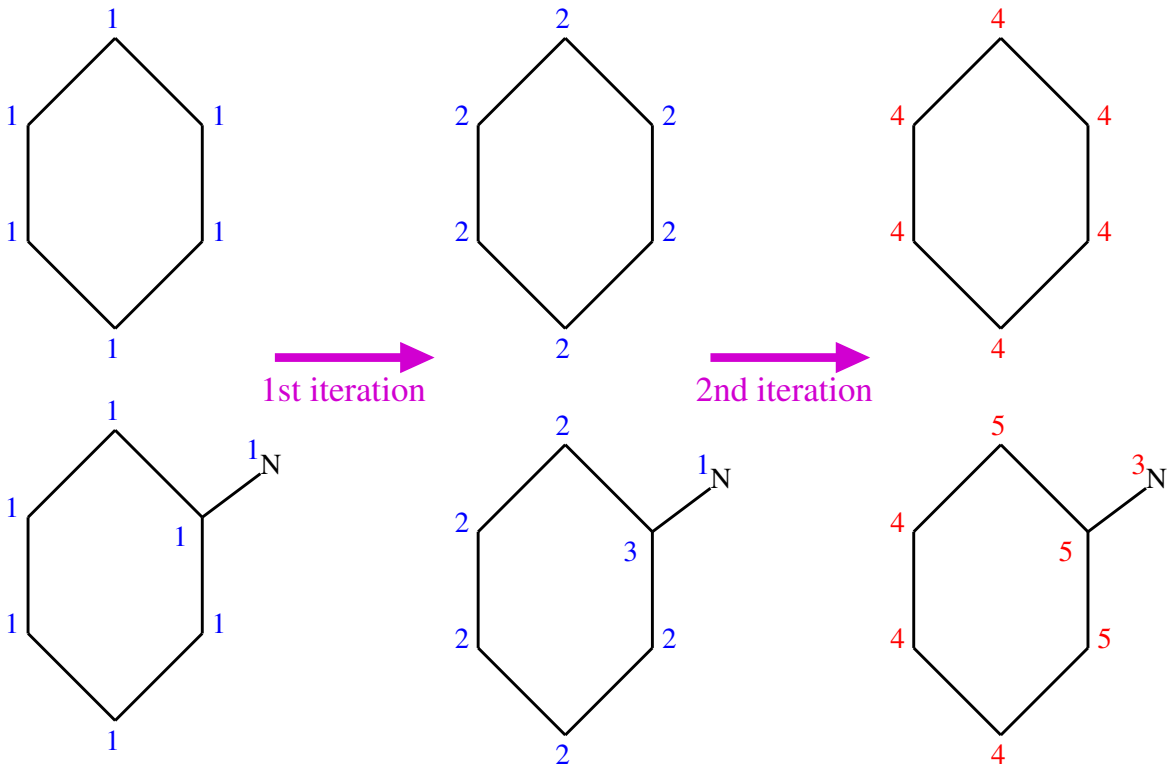
Label enrichment : illustration



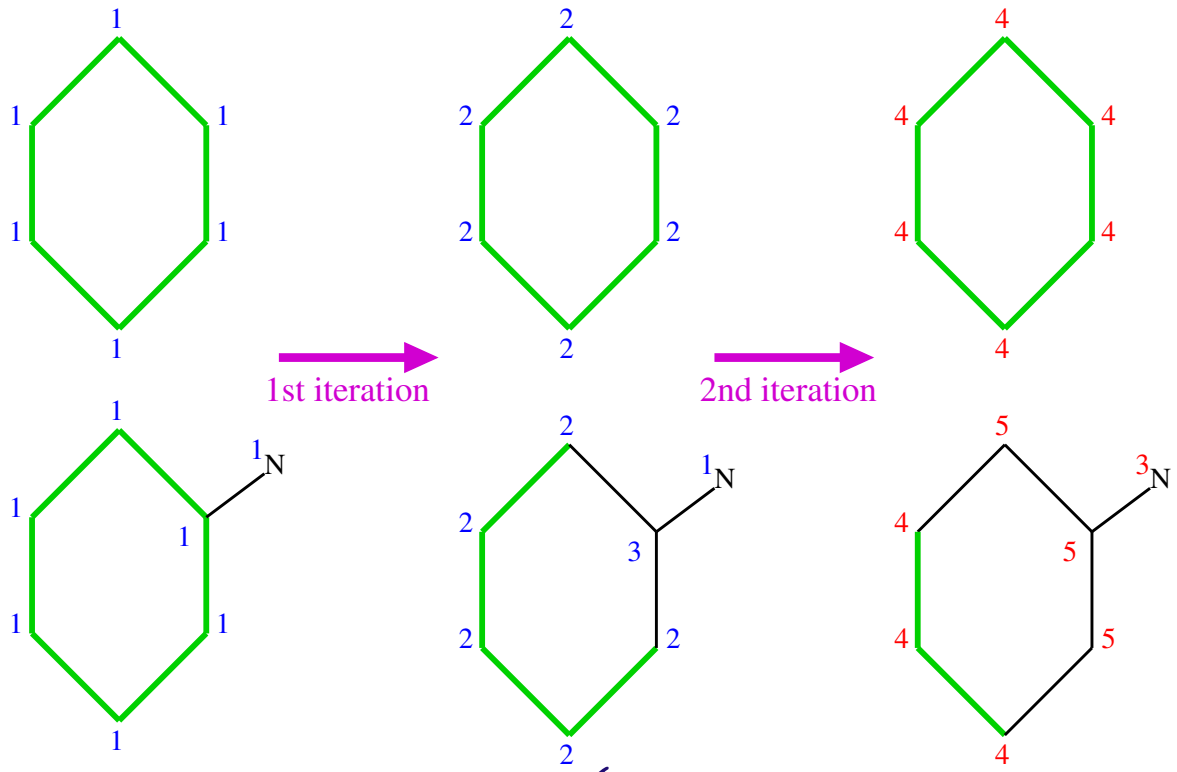
Label enrichment : illustration



Label enrichment : illustration



Label enrichment : illustration



$\Rightarrow$ Family of kernels K_n : $\begin{cases} \text{decreased kernel complexity } (\mathcal{O}(\lvert G_1 \times G_2 \rvert^3)) \\ \text{(potential) increased kernel expressivity} \end{cases}$

Part 3

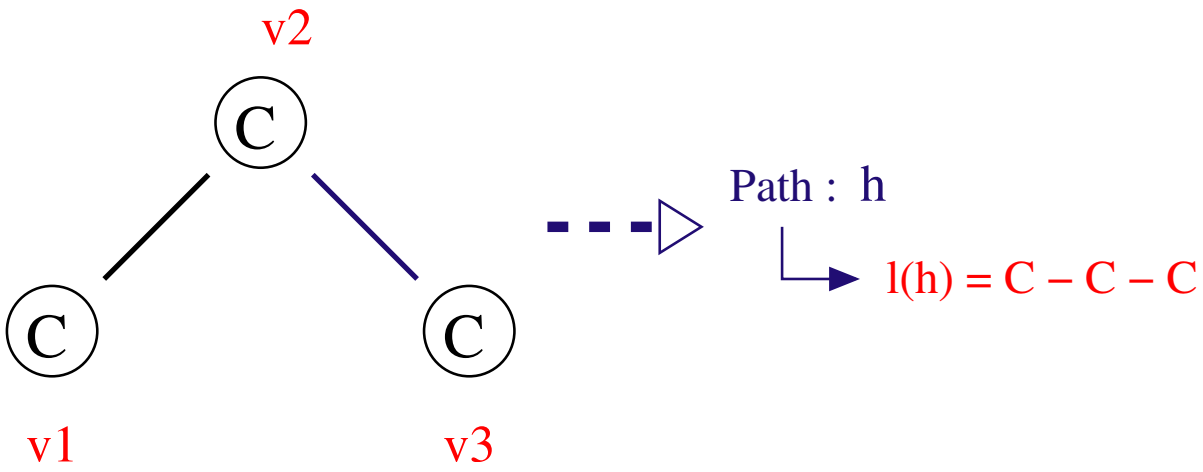
Second Extension : Preventing Totters

Preventing totters

- Tottering path : $h = (v_1, \ldots, v_n)$, $\exists\, i : v_{i+2} = v_i$.

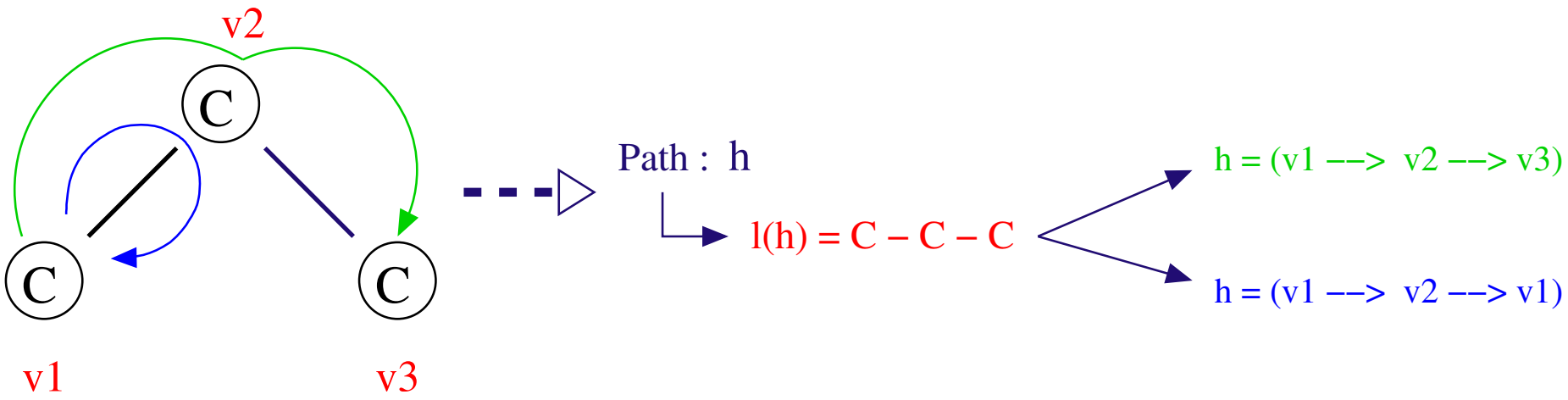
Preventing totters

- Tottering path : $h = (v_1, \ldots, v_n)$, $\exists\, i : v_{i+2} = v_i$.



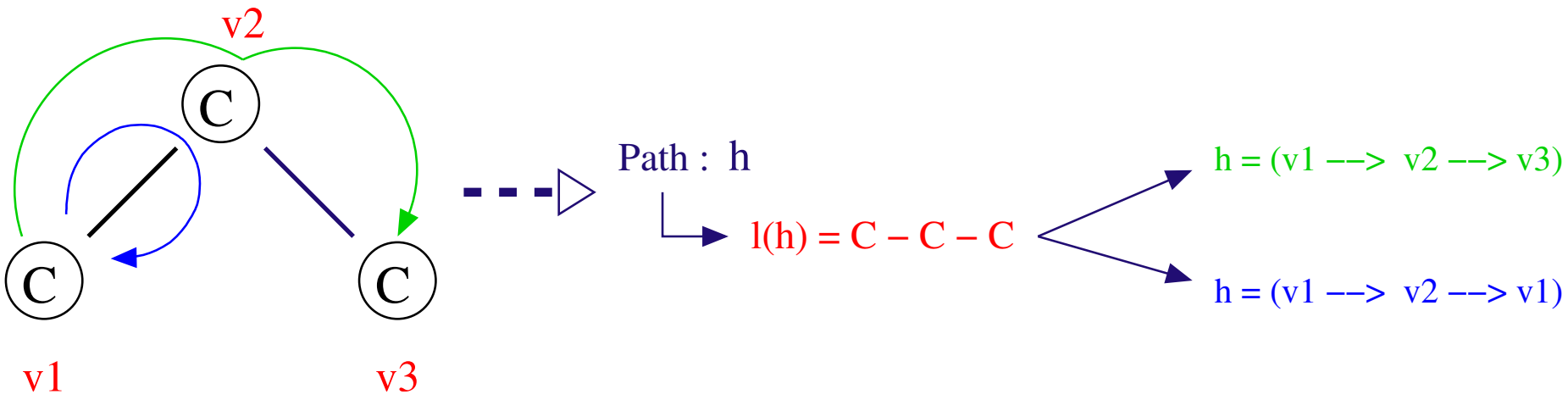
Preventing totters

- Tottering path : $h = (v_1, \ldots, v_n)$, $\exists\, i : v_{i+2} = v_i$.



Preventing totters

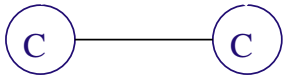
- Tottering path : $h = (v_1, \ldots, v_n)$, $\exists\, i : v_{i+2} = v_i$.

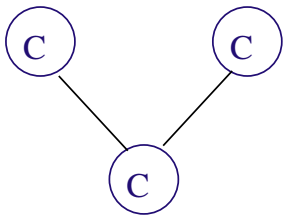


$\Rightarrow$ preventing totters $\Leftrightarrow$ filtering blue path.

Preventing totters

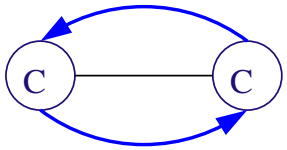
- Motivation:



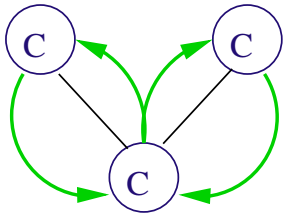


Preventing totters

- Motivation:

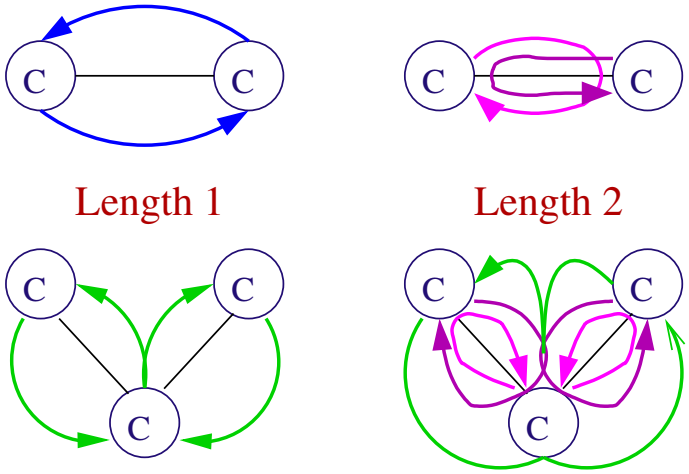


Length 1



Preventing totters

- Motivation:

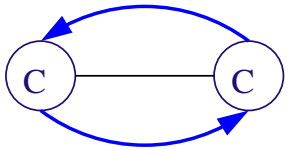


⋆ Every path of G_2 can be matched to a tottering path of G_1

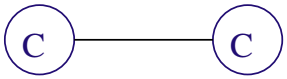
⋆ $\Rightarrow$ Compounds are considered as identical

Preventing totters

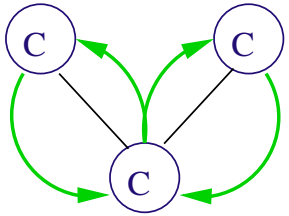
- Motivation:

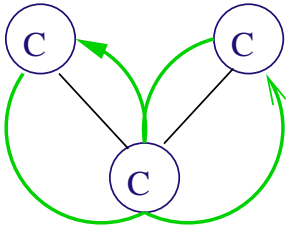


Length 1



Length 2, no tottering

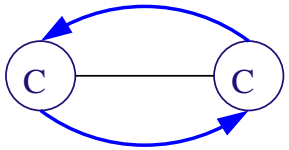




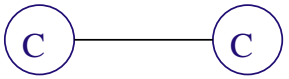
- ⋆ Only "real" chemical path are matched
- ⋆ $\Rightarrow$ Compounds are now seen as different

Preventing totters

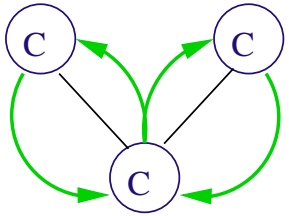
- Motivation:

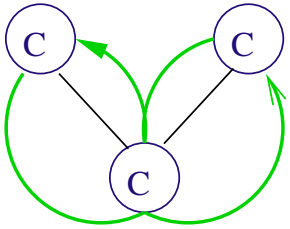


Length 1



Length 2, no tottering





⋆ Only "real" chemical path are matched

⋆ $\Rightarrow$ Compounds are now seen as different

- Solution : increase the order of the random walk model :

$\Rightarrow p_G(h) = p_s(v_1)p_t(v_2|v_1)\prod_{i=3}^{n} p_t(v_i|v_{i-2}, v_{i-1})$

Graph Transformation

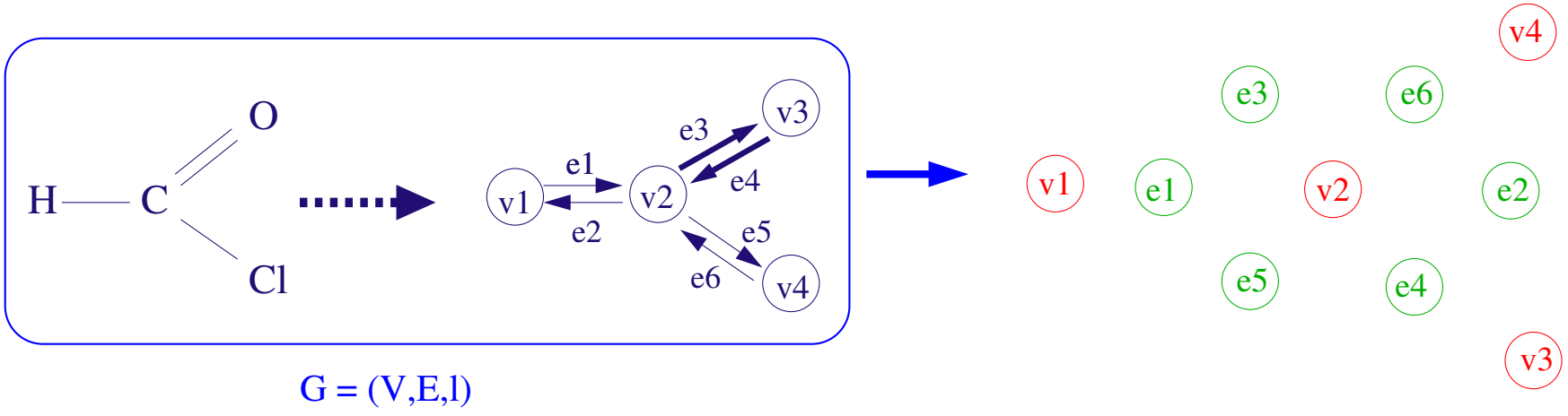
Transformation : $G = (V, E, l) \Rightarrow G' = (V', E', l')$ where :

- $V' = V \cup E$
- $E' = \{(v, (v, t)) \,|v \in V, (v, t) \in E\}$
  $\cup \{((u, v), (v, t)) \,|\, (u, v), (v, t) \in E, u \neq t\}$

Graph Transformation

Transformation : $G = (V, E, l) \Rightarrow G' = (V', E', l')$ where :

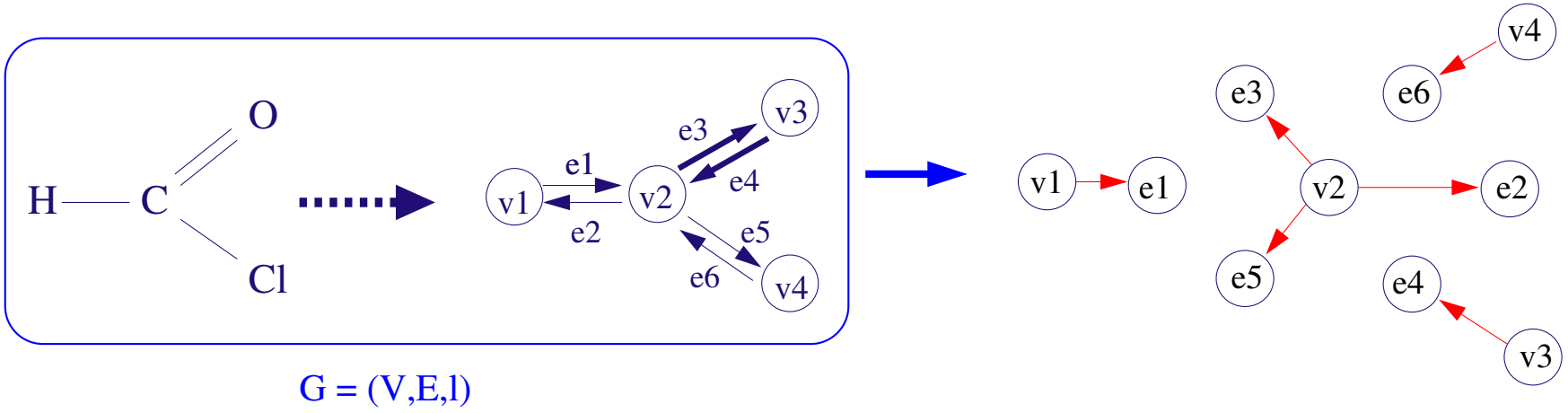
- $V' = V \cup E$
- $E' = \{(v, (v, t)) \,|v \in V, (v, t) \in E\}$
  $\cup \{((u, v), (v, t)) \,|\, (u, v), (v, t) \in E, u \neq t\}$



Graph Transformation

Transformation : $G = (V, E, l) \Rightarrow G' = (V', E', l')$ where :

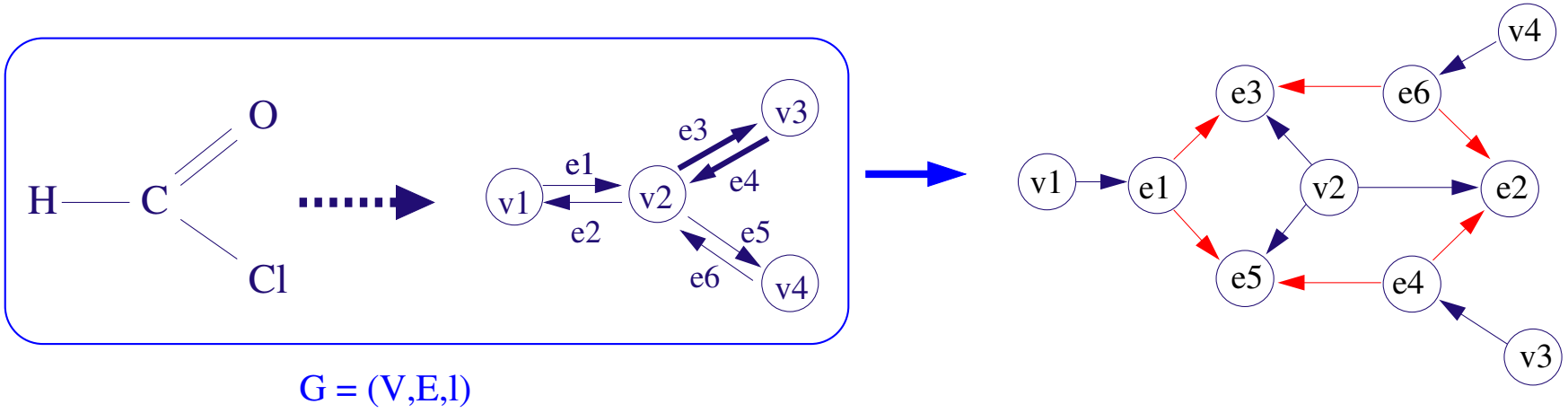
- $V' = V \cup E$
- $E' = \{(v, (v, t)) \,|v \in V, (v, t) \in E\}$
  $\cup \{((u, v), (v, t)) \,| (u, v), (v, t) \in E, u \neq t\}$



Graph Transformation

Transformation : $G = (V, E, l) \Rightarrow G' = (V', E', l')$ where :

- $V' = V \cup E$
- $E' = \{(v, (v, t)) \,|v \in V, (v, t) \in E\}$
  $\cup \ \{((u, v), (v, t)) \,|\, (u, v), (v, t) \in E, u \neq t\}$



Kernel computation (1/2)

- Consider : $\begin{cases} H_0(G) = \{\text{Non tottering paths of G}\} \\ H_1(G') = \{\text{Paths of } G' \text{ starting from a node } v \in V\ \} \end{cases}$
- The mapping $f : H_0(G) \longrightarrow H_1(G')$ defined by

$$h = (v_1, ..., v_n) \mapsto h' = (v'_1, ..., v'_n) \text{ such that } \begin{cases} v'_1 = v_1 \\ v'_i = (v_{i-1}, v_i) \end{cases}$$

establishes a bijection between $H_0(G)$ and $H_1(G')$

Kernel computation (1/2)

- Consider : $\begin{cases} H_0(G) = \{\text{Non tottering paths of G}\} \\ H_1(G') = \{\text{Paths of } G' \text{ starting from a node } v \in V\ \} \end{cases}$
- The mapping $f : H_0(G) \rightarrow H_1(G')$ defined by

$$h = (v_1, ..., v_n) \mapsto h' = (v'_1, ..., v'_n) \text{ such that } \begin{cases} v'_1 = v_1 \\ v'_i = (v_{i-1}, v_i) \end{cases}$$

establishes a bijection between $H_0(G)$ and $H_1(G')$

- Let p' be the image of p_G by f:

$$\forall h' \in H_1(G'), \quad p'(h') := p_G\left(f^{-1}(h')\right)$$

Kernel computation (2/2)

- Theorem: p' factorizes as

$$p'(h') = p'_s(v'_1) \prod_{i=2}^{n} p'_t(v'_i|v'_{i-1})$$

  - ⋆ $p'_s(v') = p_s(v')$
  - ⋆ $p'_t(u'|v') = \begin{cases} p_t(u|v') \text{ if } v' \in V \text{ and } u' = (v', u) \in E \\ p_t(u|v, w) \text{ if } v' = (v, w) \text{ and } u' = (w, u) \in E \end{cases}$

Kernel computation (2/2)

- Theorem: p' factorizes as

$$p'(h') = p'_s(v'_1) \prod_{i=2}^{n} p'_t(v'_i|v'_{i-1})$$

  - ⋆ $p'_s(v') = p_s(v')$
  - ⋆ $p'_t(u'|v') = \begin{cases} p_t(u|v') \text{ if } v' \in V \text{ and } u' = (v', u) \in E \\ p_t(u|v, w) \text{ if } v' = (v, w) \text{ and } u' = (w, u) \in E \end{cases}$

- Corollary :

$$\left.\begin{array}{l} \text{- graph transformation} \\ \text{- original graph kernel} \end{array}\right\} \Rightarrow \text{ tottering paths removed}$$

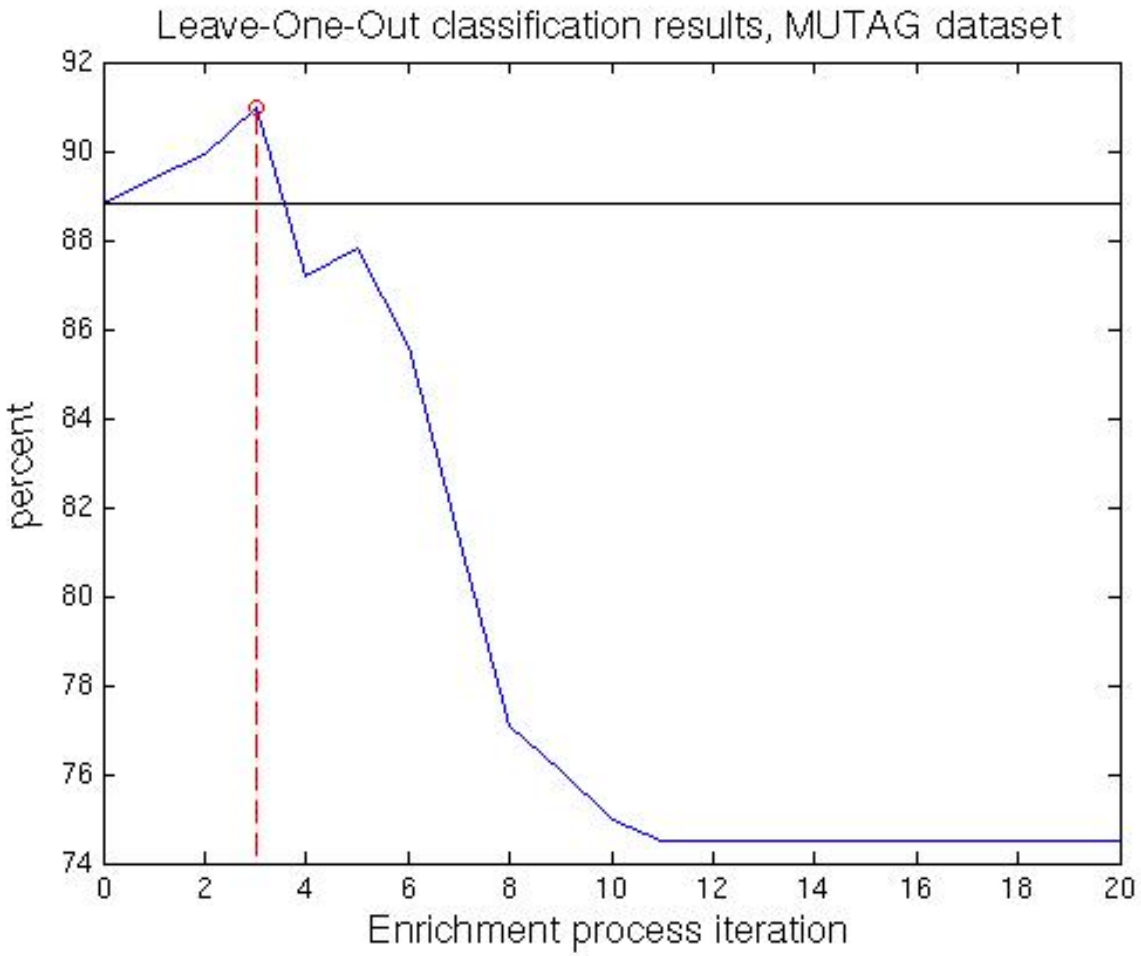
Part 4

Results and conclusion

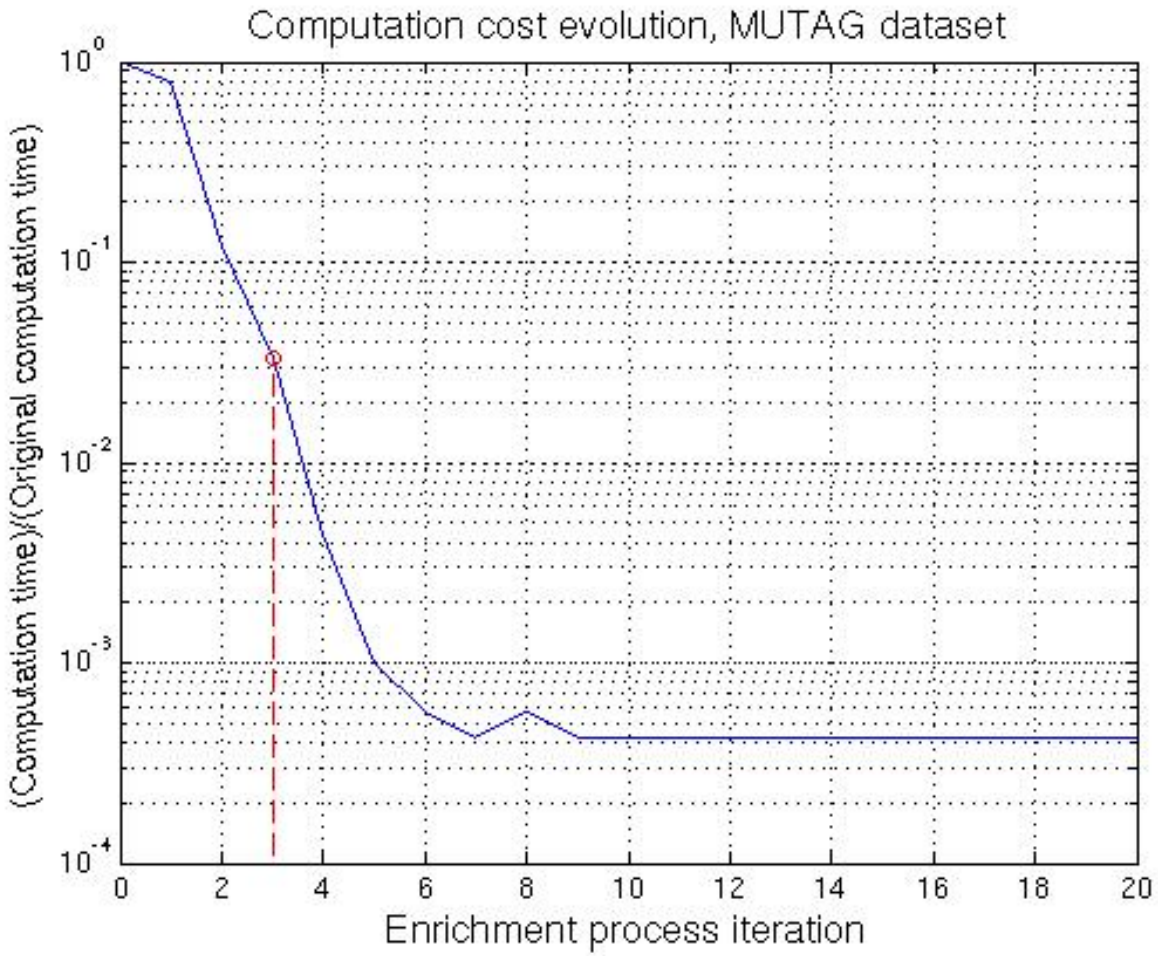
Experimental settings

- Chemical dataset : MUTAG (mutagenicity)
- Binary classification task
- Evaluation by Leave-One-Out error

MUTAG : first extension



MUTAG : first extension



MUTAG

- State of the art results :

Linear Reg.	Neural Nets	Decision Trees	I.L.P
89.3%	89.4%	88.3%	87.8%

- Our best results :
  - ⋆ No extension : 89.9%
    → Kashima et al. kernel
  - ⋆ 1st extension : 91%
    → $\frac{1}{30}$ computation cost
  - ⋆ 2nd extension : 90.4%

Conclusion

- Chemical intuitions $\Rightarrow$ 2 general and modular extensions
  - ⋆ Result : family of (+/- efficient) kernels
- Future work :
  - ⋆ Kernels combination/optimisation
  - ⋆ Chemistry oriented :
    - ∗ Definition of a "chemically-relevant" kernel K_L
    - ∗ Combine graph-kernel and traditional chemoinformatics approaches