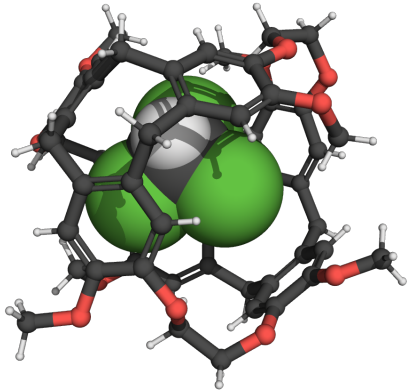
Computational Generation of Substrate-Specific Molecular Cages

Noé Demange, Yann Strozecki and Sandrine Vial

DAVID lab
UVSQ - Université Paris-Saclay, Versailles, France

Context



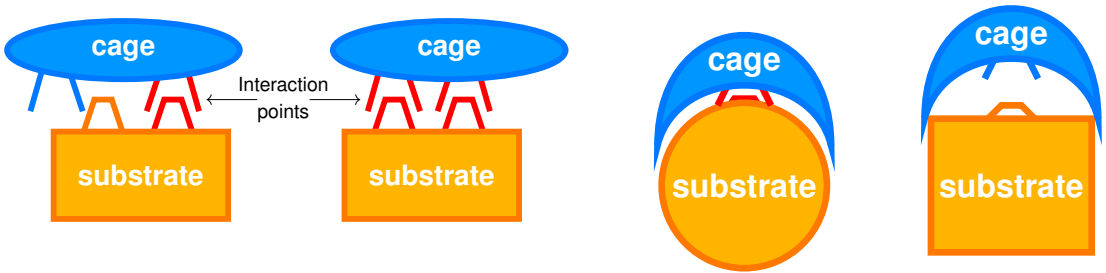
Applications: Gas capture, Drug delivery, ...

Current generation method: Generation of a molecular cage followed by the search for substrates that it captures.

Difficulty: Find the right cage when starting with a substrate.

Problem

Problem: Given a substrate, can we generate a molecular cage model adapted to it ?

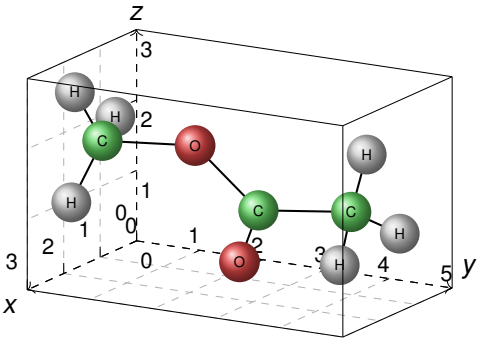


Modeling

Molecular graph: A triple (G, coord, type), where $G = (V, E)$, $\text{coord} : V \to \mathbb{R}^3$, $\text{type} : V \to A$

Constraints:

- Covalent bond length:
  $\forall (u, v) \in E,\ \|u - v\| = \text{cov}(\text{type}(u), \text{type}(v))$
- No atomic collision:
  $\forall u, v \in V \text{ with } (u, v) \notin E,\ \|u - v\| > \text{col}(\text{type}(u), \text{type}(v))$
- Degree constraint:
  $\forall v \in V,\ \deg(v) \leq \deg_{\max}(\text{type}(v))$
- Bond angle:
  $\forall e_1 = (u, v), e_2 = (v, w) \in E, Angle(e_1, e_2) = VSEPR(type(v))$



Pattern: Small molecular graph serving as a building block.

Cage Construction Overview : Our Approach

Two stages:

1. Interaction points generation with binding patterns [Bricage, 2018]
2. Molecular path construction between binding patterns

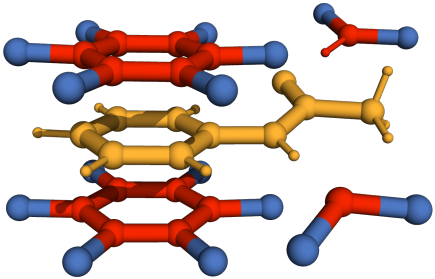


Figure: Stage 1

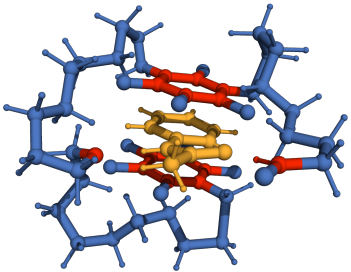
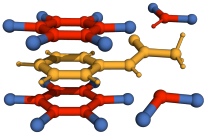


Figure: Stage 2

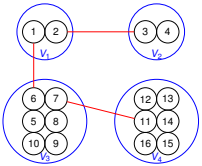
Molecular Path Overview

Objective: Obtain a connected molecular graph from the binding patterns.

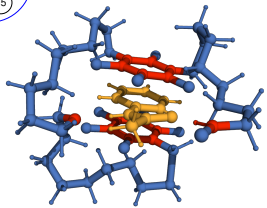
Input : Substrate (G_S) + Binding patterns (G)



Interconnection trees enumeration



For each selected pair of vertices: Generation of a molecular path



Molecular Path Construction Problem

Input:

- Substrate G_S
- Partial cage G
- Two attachment vertices $s, t \in G$

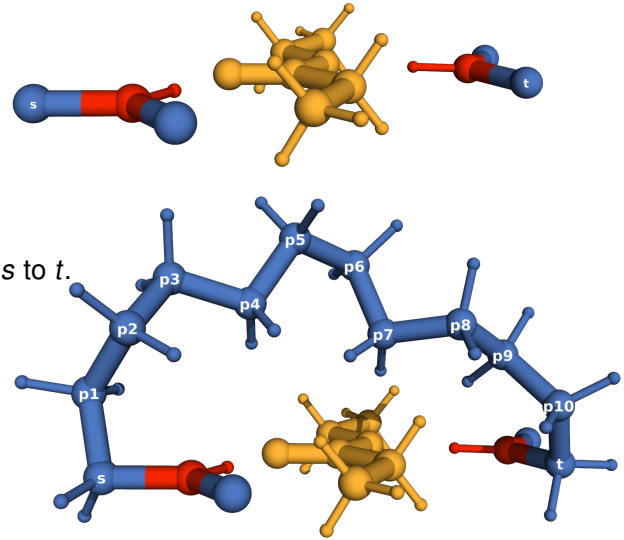
Output:

Molecular path $P = (p_1, \ldots, p_k)$ connecting s to t.
p_i is a *connection pattern*.

Constraints:

- $G \cup P$ is a molecular graph
- No collision with the substrate

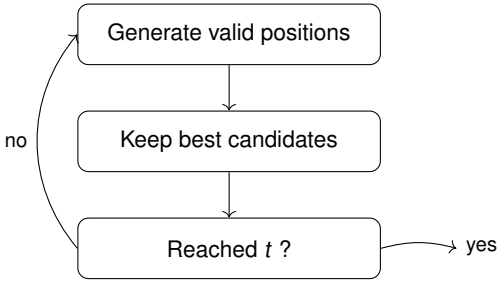
Optimization: $\min(|P|, \mathrm{NRMSD}(P))$

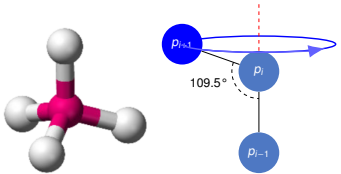


Molecular Path Construction

Single connection pattern: Tetrahedral carbon atom.

Idea: Incrementally construct a path from s to t.





Infinite possible positions

⇓

Discretization of valid positions

Search strategy: Depth-first Branch-and-Bound

Valid Positions and Discretization

Valid positions: $\forall u \in V_{G_S} \cup V_G$

$$p(\theta) = \overrightarrow{center} + r\cos(\theta)\overrightarrow{v_1} + r\sin(\theta)\overrightarrow{v_2}$$

$$\|p(\theta) - u\| > d_{col}$$

From infinite to finite positions:

- Compute valid angular intervals
- Sample a maximum of 12 positions in these intervals (minimum spacing of 15°)
- Keep only the 3 best candidates

Number of positions

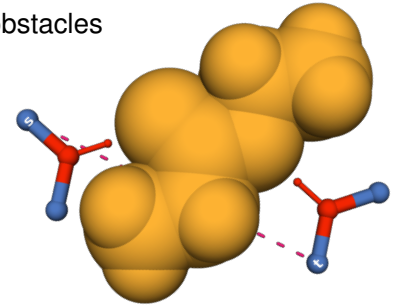
Instance	#Pos.	Path length	#Paths	NRMSD	Time (ms)
ABABELa	24	10	3	0.92	25
	12	10	10	0.53	24
	8	9	2	1.21	17
BAHSUYa	24	5	1	0.82	2.6
	12	5	2	0.82	2.1
	8	6	1	0.80	3.0

Number of best candidates

Instance	#BestC	Path length	#Paths	NRMSD	Time (ms)
ABABELa	2	10	1	0.87	6.3
	3	10	10	0.53	22
	4	10	67	0.33	99
BAHSUYa	2	*N/A*	0	*N/A*	1.7
	3	5	2	0.82	1.8
	4	5	3	0.82	10

Selecting the Best Candidates

Objective: Guide the path toward the target while avoiding obstacles



Distance heuristics

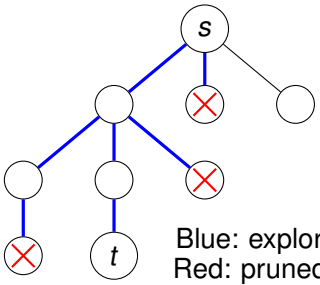
- Euclidean distance
- Grid shortest-path distance (A^*)
- Hybrid distance: $\begin{cases} \text{Euclidean, target visible} \\ \text{A*, otherwise} \end{cases}$

Instance	Distance	#Paths	NRMSD	Time (ms)
ABABELa	Euclidean	12	0.64	11
	A*	6	0.68	100
	HYBRID	10	0.53	23
BAHSUYa	Euclidean	2	0.79	0.32
	A*	2	0.90	10
	HYBRID	2	0.82	1.9
YARZUN03a	Euclidean	0	NA	0.24
	A*	29	0.39	78
	HYBRID	26	0.15	29

Size Pruning Strategies

Cut-off strategies:

- **Baseline:** fixed maximum path length $L_{\max}$
- **Min Length Cut:** update $L_{\max}$ with the length of the best path found so far
- **Projected Length Cut:** also prune if $\underbrace{|P|}_{\text{current}} + \underbrace{\left\lceil \frac{d(p_i,\, t)}{d_{\max}} \right\rceil}_{\text{estimated remaining}} > L_{\max}$



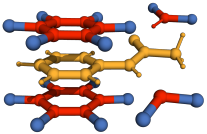
Blue: explored path
Red: pruned branches

Instance	Cut	#Paths	#Partial paths	Time (ms)
ABABELa	Baseline	45	30 839	145
	Min Length	10	7 009	35
	Projected Length	10	6 225	22
BAHSUYa	Baseline	5	138	1.8
	Min Length	5	120	1.8
	Projected Length	5	120	1.8
YILLAGc	Baseline	41	5 662	47
	Min Length	10	1 276	12
	Projected Length	10	1 134	8

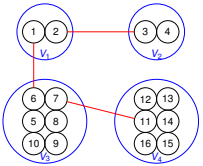
Molecular Path Overview

Objective: Obtain a connected molecular graph from the binding patterns.

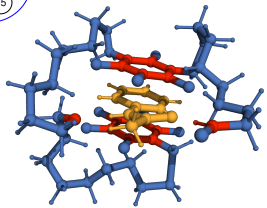
Input : Substrate (G_S) + Binding patterns (G)



Interconnection trees enumeration



For each selected pair of vertices: Generation of a molecular path

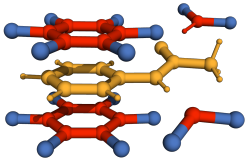


Interconnection Trees

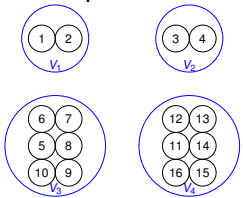
Objective: Find the vertex pairs that must be connected to make the graph connected.

Modeling:

Substrate + binding patterns

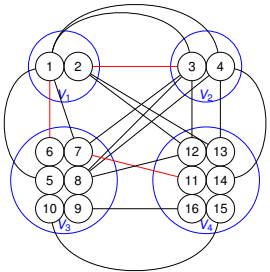


k independent sets

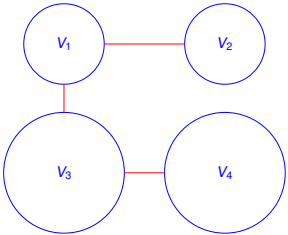


Interconnection tree:

Multipartite graph G



Quotient graph of G



$$T = \{(1,6),(2,3),(7,11)\}$$

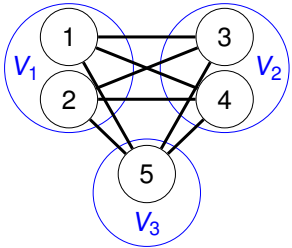
Complete Multipartite Graph

General case:

Interconnection Tree decision problem is NP-complete (Hamiltonian Path reduction)

Complete multipartite graphs:

Complete multipartite graph with k parts admits an interconnection tree iff $|V| \geq 2(k-1)$.



Intuition:

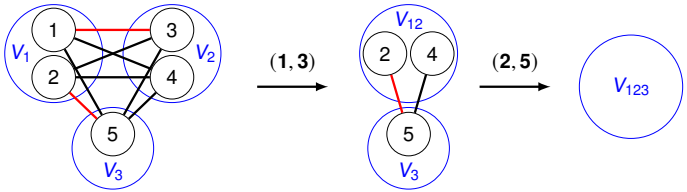
- A tree on k parts needs $k-1$ edges
- Each edge uses two distinct vertices
- Hence at least $2(k-1)$ vertices

Enumeration

Key idea: Recursive decomposition using edge contractions.

For complete multipartite graphs:

- All recursive subproblems remain complete multipartite
- Existence tested using $|V| \geq 2(k-1)$
- Enumeration with $O(k)$ worst-case delay $O(1)$ amortized delay



Instance	#Trees	Time (ms)	Delay (ns)
(3,6)	3 240	0.07	22
(5,3)	174 960	5.8	34
(5,6)	107 308 800	2 685	25

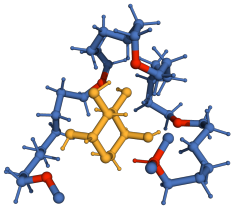
Generated Molecular Cages

Properties of the generated cages:

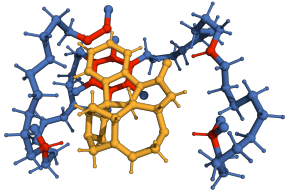
- Interaction points with the substrate
- Paths are shape-complementary
- Internal cavity remains unobstructed

Limitation:

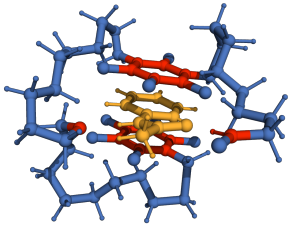
Static model — real cages are flexible.



(a) β-L-Arabinose



(b) Strychnine



(c) Acetanilide

Conclusion

Contributions:

- Computational framework for substrate-specific molecular cage generation
- Short molecular path construction via depth-first branch-and-bound
- Interconnection trees enumeration
- github.com/NoeDemange/MolecularCagesGeneration

Future work:

Rigidify cages by completing the structure

Thank You!

noe.demange@uvsq.fr

References

Bricage, M. (2018).
Modélisation et Algorithmique de graphes pour la construction de structures moléculaires.
phdthesis, Université Paris Saclay (COmUE).