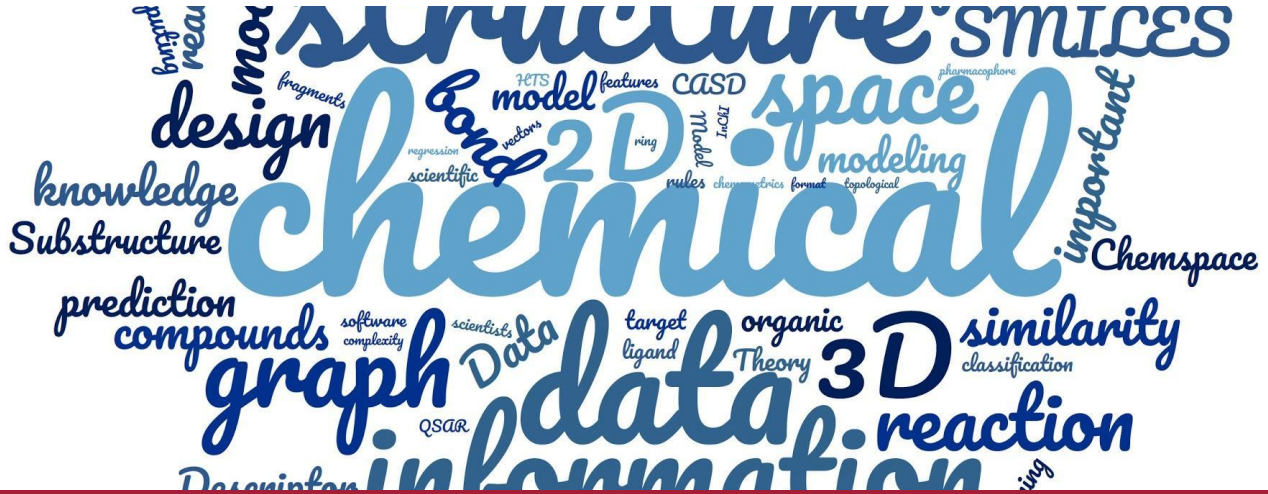
EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Institute for Bioinformatics and Medical Informatics



BIO-4372 **Cheminformatics**

L06 Topological Structure Comparison

Part III: Similarity Methods

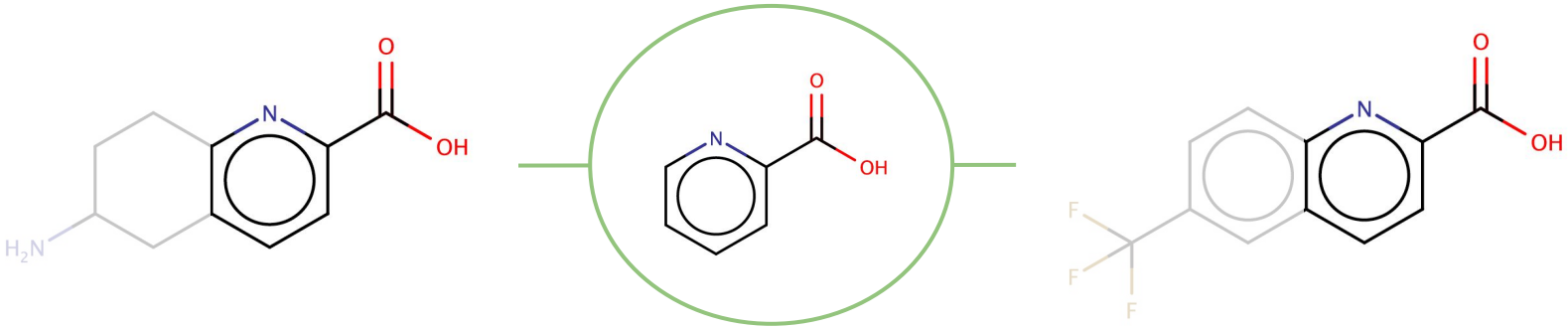
Winter Semester 2022-23

Philipp Thiel

- From MCS to topological similarity
- 2D Fingerprints: generalization of feature lists
- Similarity and distance measures
- Properties of important (dis)similarity coefficients
- Applications of topological similarity and dissimilarity
  - Similarity searching
  - Database clustering
  - Diversity analysis
- Speeding up 2D similarity calculations

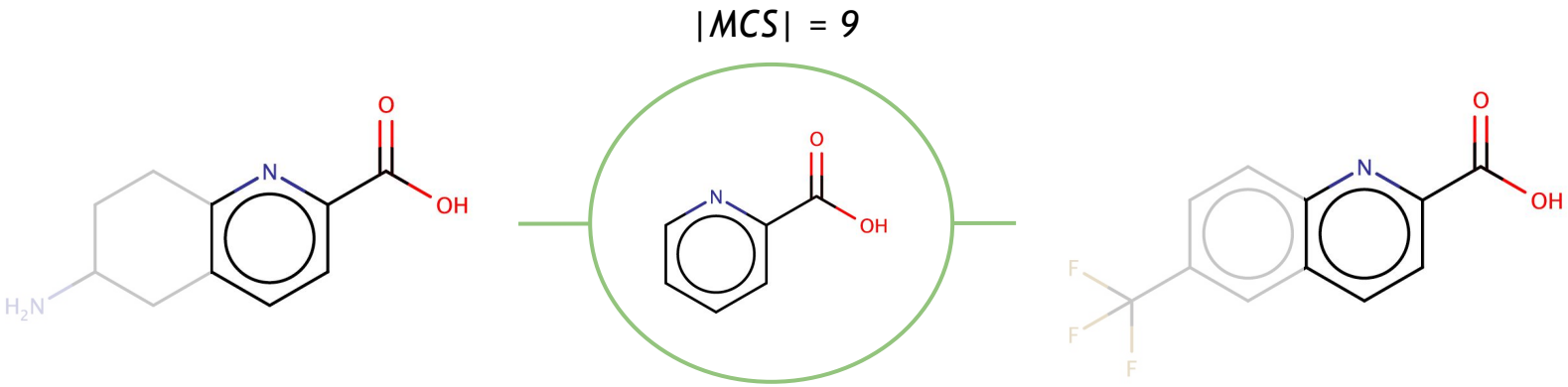
Motivation

- So far we focused on **(sub)structural identity**
- We effectively tested if either
  1. Two structures are identical
  2. A structure is contained in another structure
  3. Multiple structures share a substructure
- In case of MCS we tested for **locally conserved regions**



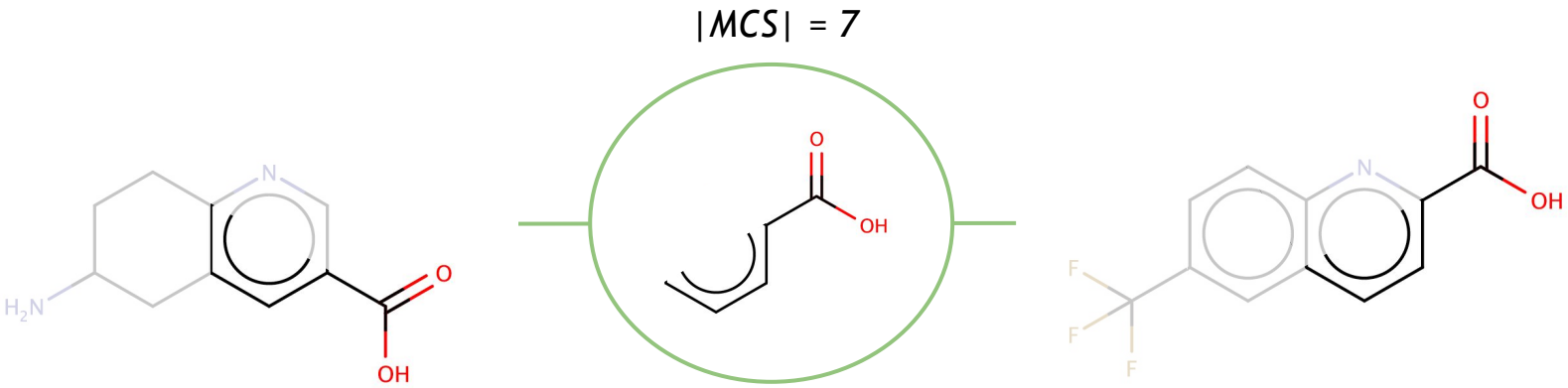
Motivation

- Cardinality of MCS correlates somehow with similarity
- Problem:
  - **Similarity depends on molecule sizes**
  - No normalization



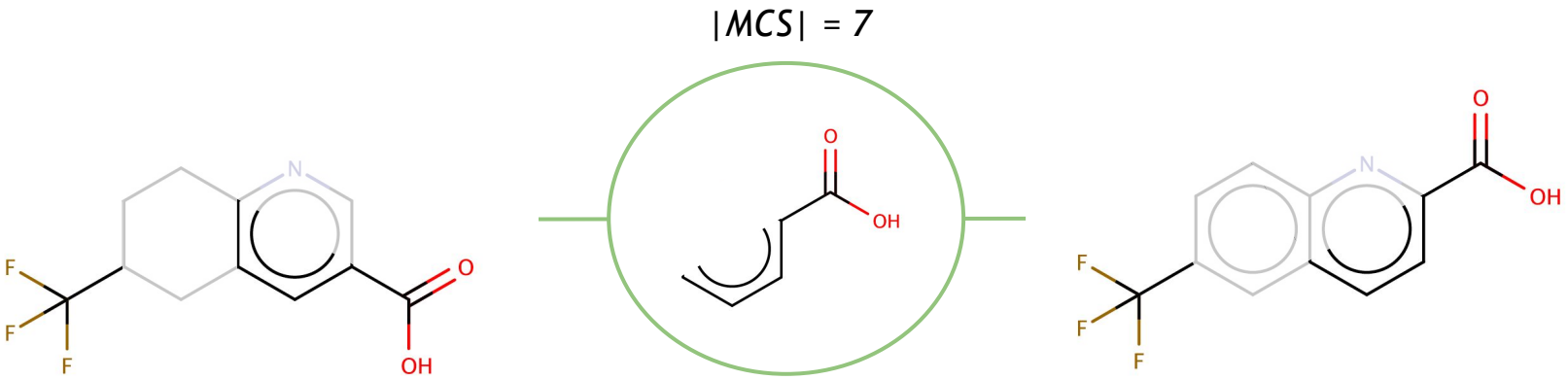
Motivation

- Cardinality of MCS correlates somehow with similarity
- Consider the following case
- Interesting substructures do not contribute to similarity any more



Motivation

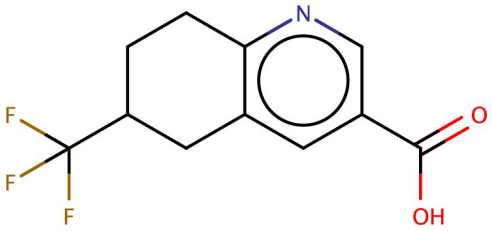
- Cardinality of MCS correlates somehow with similarity
- Consider the following case
- Interesting substructures do not contribute to similarity any more
- Disconnected fragments neglected
- Idea: **A global measure of similarity is required**

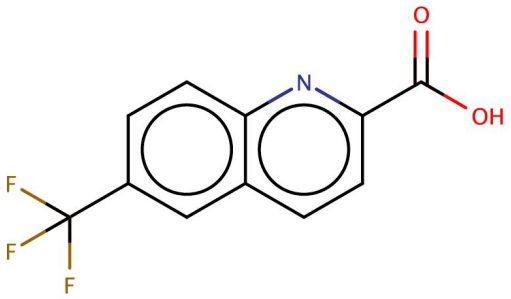


Motivation

- Idea:

  Use shared fragments as a measure of similarity





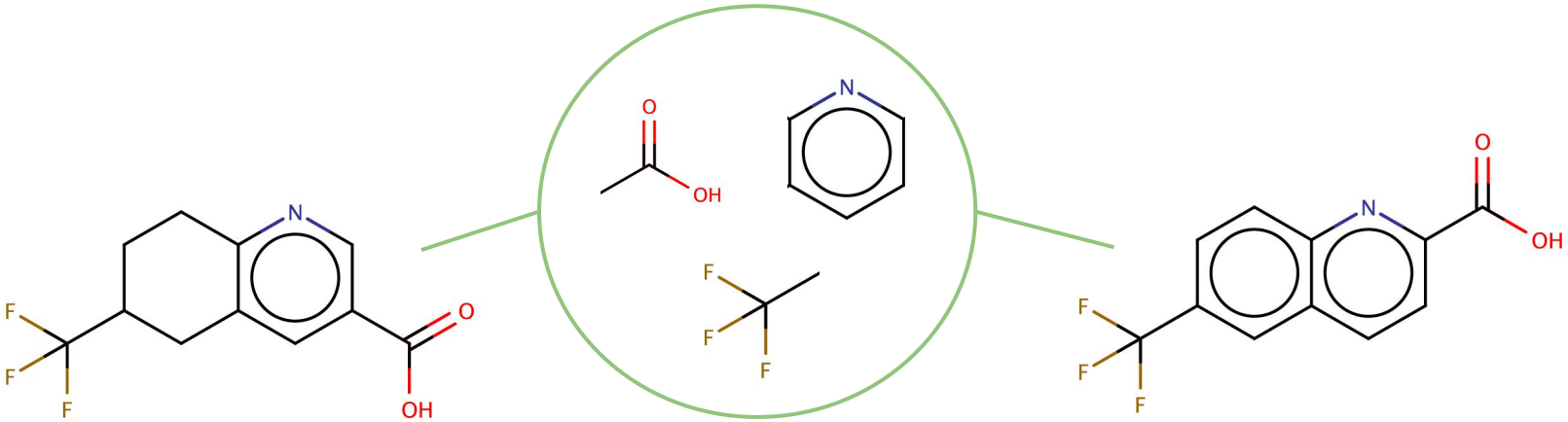
Motivation

- Idea:

  Use shared fragments as a measure of similarity

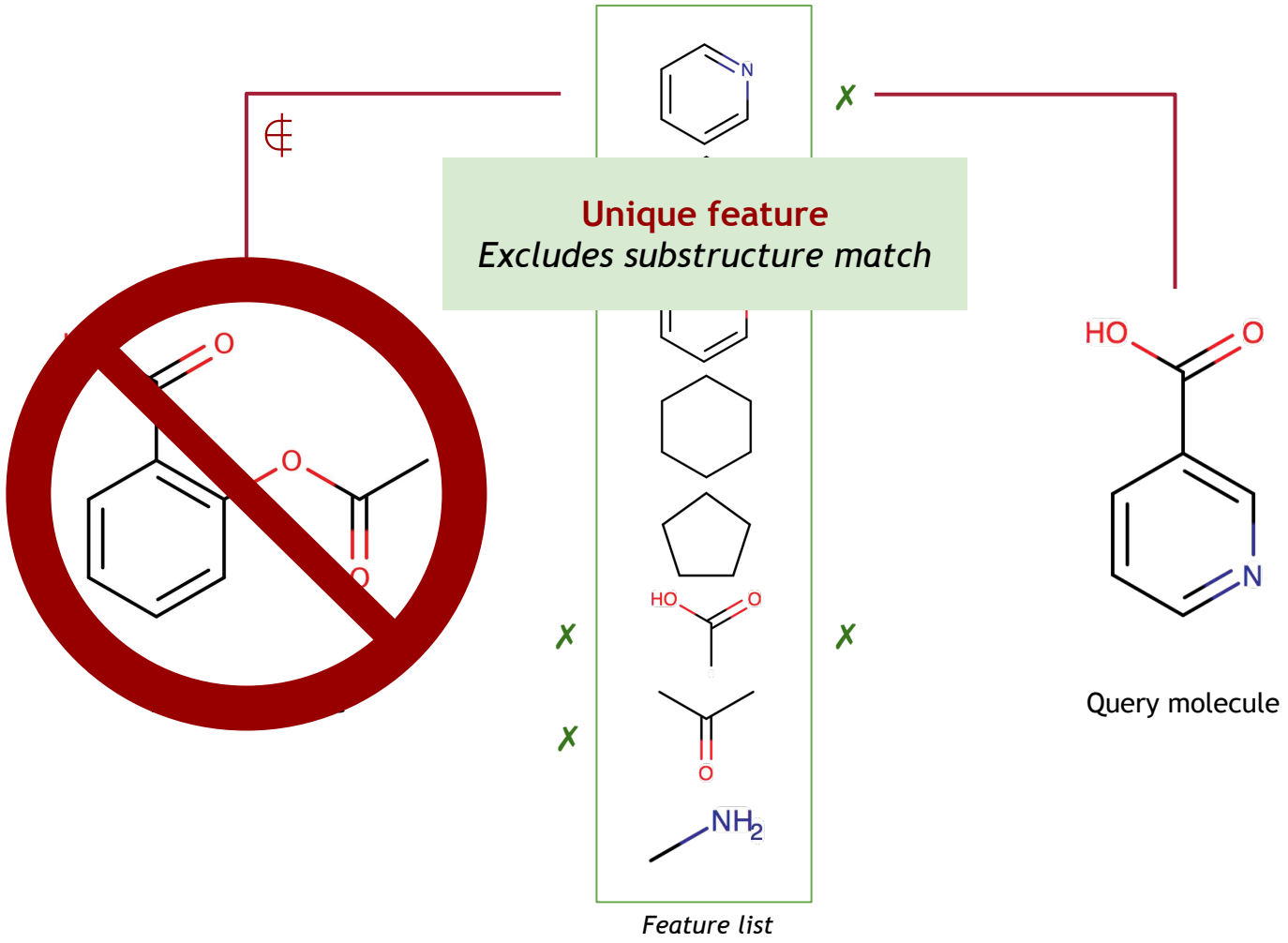
- We have already used a similar concept:

  Feature lists for fast elimination in substructure searching



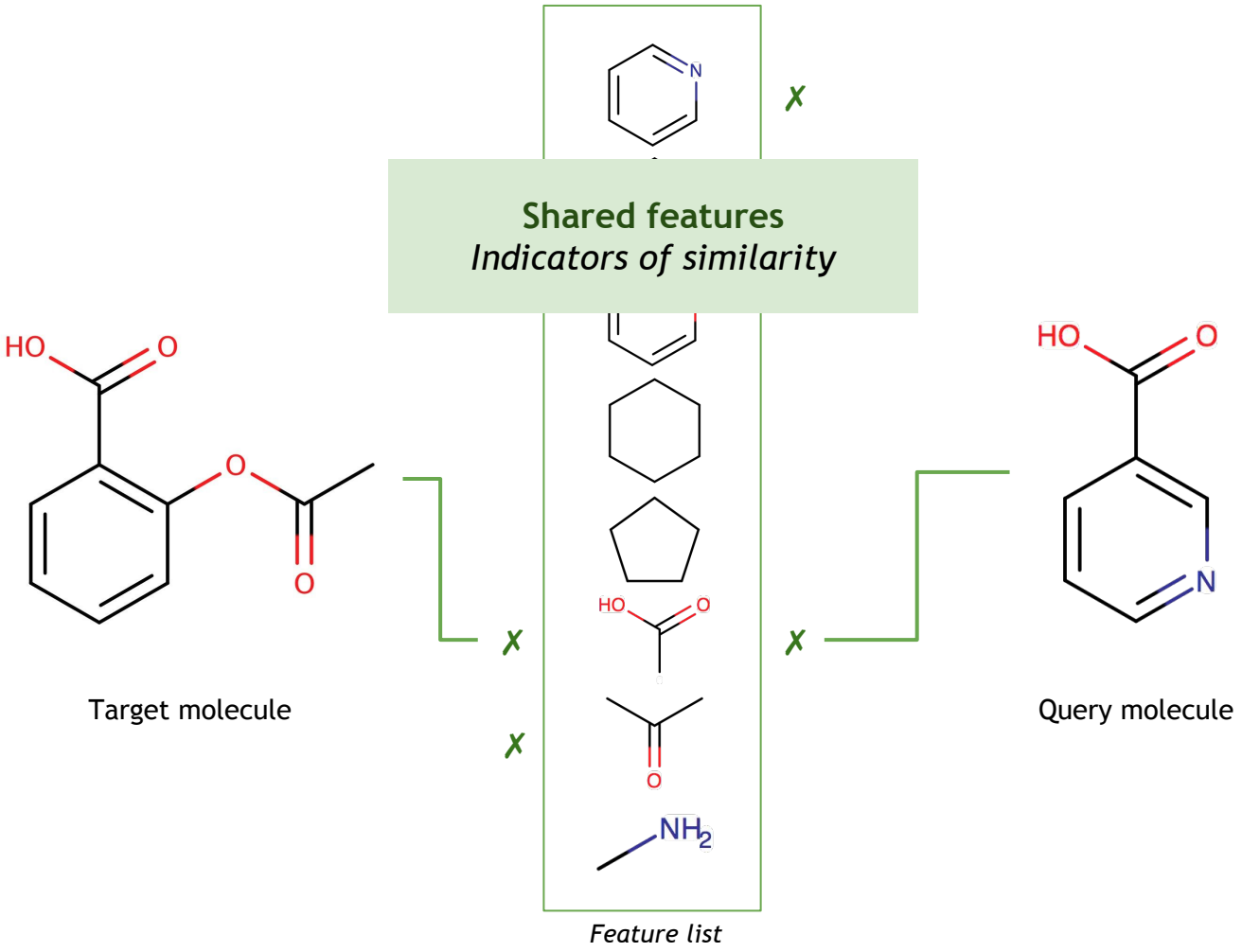
Feature Lists

Indicators of Impossible Substructure Matching



Feature Lists

Overlap as a Measure For Similarity



2D Fingerprints

Structural Keys

- Feature lists encoded as **bit-vectors**
  - Each position corresponds to a predefined fragment
  - 0-bit→ corresponding fragment is absent
  - 1-bit→ corresponding fragment is present

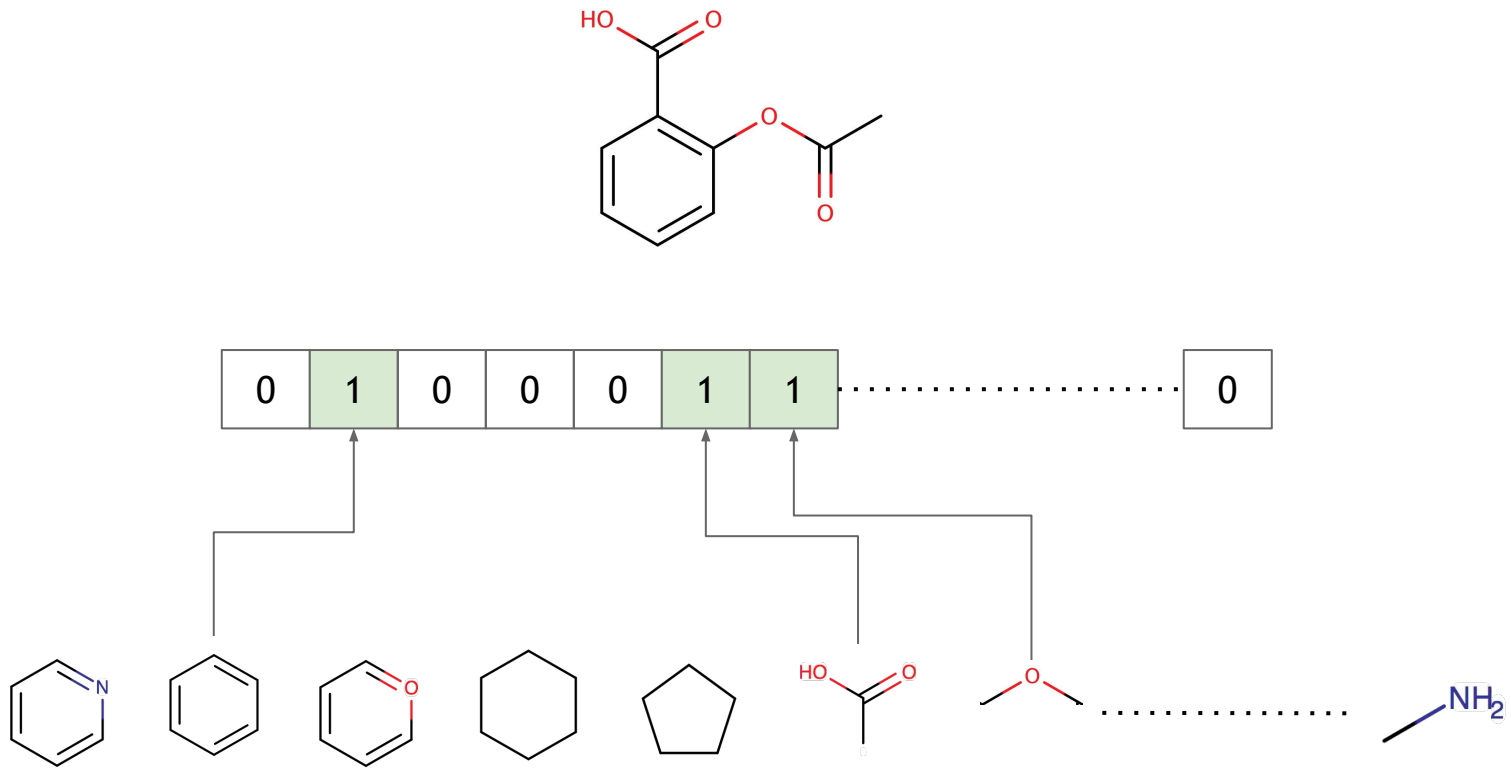
- 2D Fingerprints (FPs) can easily be precalculated

- Fragments can be defined e.g. as SMARTS patterns
  - This additionally allows fuzzy matching

2D Fingerprints

Structural Keys

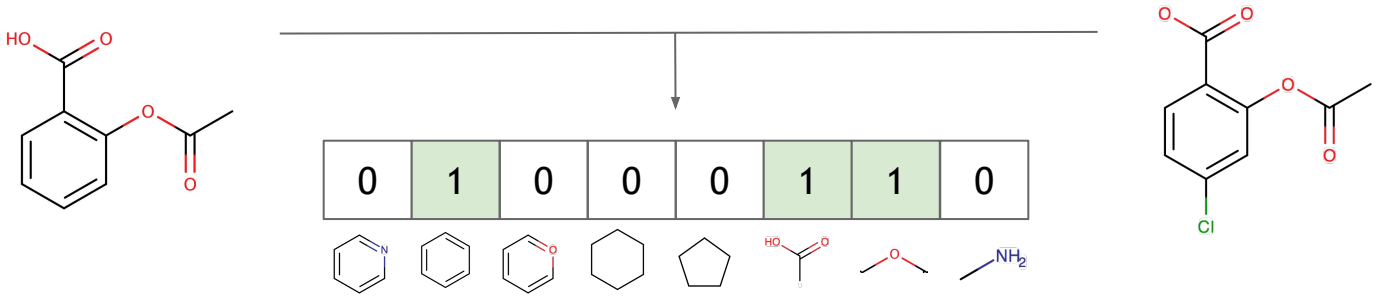
- This special type is known as **structural keys**



2D Fingerprints

Structural Keys

- **Pros**
  - Very compact representation
  - Efficient and fast comparison possible
- **Cons**
  - We test only for the **existence** of substructures
    - No frequency counts
    - No relative orientation in topology
    - **Information loss!**
  - Different structures can have identical fingerprints



2D Fingerprints

Hashed Fingerprints

- **Structural keys problem**:

  Represent only predefined fragments

- Possible solution:

  Systematically enumerated substructures

- Approaches followed

  1. Enumerate **linear paths** [1]
  2. Enumerate **radial atom environments** [2]

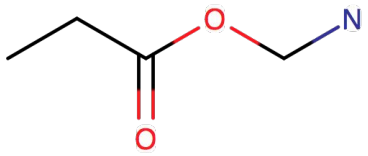
1. http://www.daylight.com/dayhtml/doc/theory/theory.finger.html
2. Bender A. et al. (2004) *J. Chem. Inf. Model.*, 44, 170-8

2D Fingerprints

Hashed Fingerprints

- Example: **path-based fingerprints**
- Enumerate all linear paths of lengths 2 to *N* in molecular graph
  - Typically *N* = 7
  - Often *N* is an adjustable parameter
  - For small molecules this leads to ~10^5 different fragments
- Thus, each fragment can be assigned a **unique ID**

Example: paths of length 3



2D Fingerprints

Hashed Fingerprints

- Substructure IDs can be stored as
  1. **Bit-vectors** like structural keys
  2. **Feature lists** that store IDs of present substructures

- Fingerprints often **sparsely populated** = small number of 1-bits
- In bit-vectors most space is wasted on 0-bits

Bit-vector: <00000000...1...1...1...1...1...1...0000000000000>
(1-bits at positions 37, 501, 713, 7657, 21017, 37221)

Feature list: 37,501,713,7657,21017,37221

2D Fingerprints

Hashed Fingerprints

- Idea:

  Recycling, that is **spread out information**

  of many substructures over a **shorter bit-vector** by **hashing**

- Map substructure IDs onto bit-vectors using a **hash function *f***
- Usually the parameters for *f* are:
  1. Length of bit-vector, typically 2^n with n = 9,10,11,12
  2. Number of 1-bits created per ID, typically 2-5

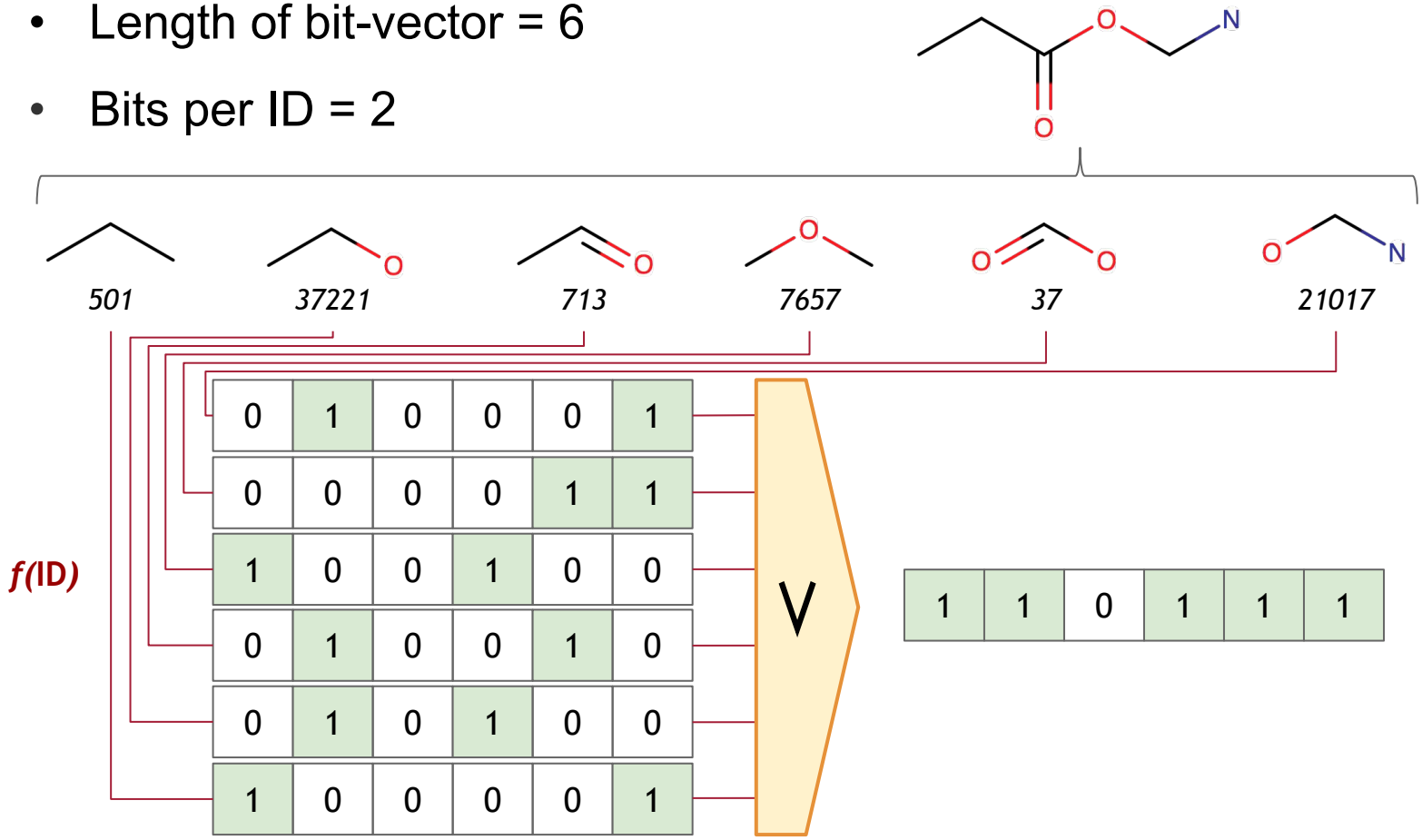
     ⇒ low probability of bit collisions

- Fingerprint obtained by **adding all bit-vectors using logical OR**

2D Fingerprints

Hashed Fingerprints: Example

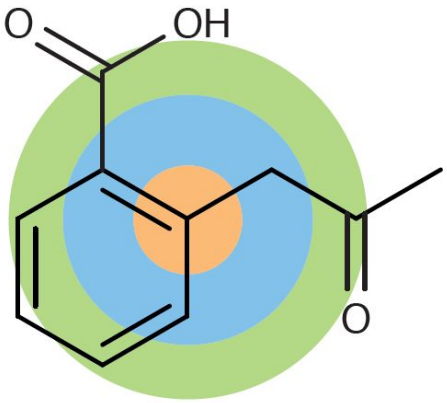
- Length of bit-vector = 6
- Bits per ID = 2



2D Fingerprints

Hashed Fingerprints

- Example: **radial atom environments**
- Enumerate all radial substructures with radius 1 to *N*
  - Typically *N* in range [2, 6]
  - Usually, *N* is an adjustable parameter



- Remaining steps similar to path-based fingerprints

Bender A. et al. (2004) *J. Chem. Inf. Model.*, 44, 170-8

2D Fingerprints

Common Fingerprint Examples

- Structural keys
  - PubChem with 881 bits [1]
  - MACCS keys: two variants with 160 (public) and 960 bits [2]
- Path-based
  - Daylight fingerprints with 1024 or 2048 bits, hashed [3]
- Radial atom environments
  - Molprint 2D, feature lists [4]
  - Extended-connectivity fingerprints (ECFP), hashed [5]
- RDKit implements a lot of these types and performance measures [6]

1: ftp://ftp.ncbi.nlm.nih.gov/pubchem/specifications/pubchem_fingerprints.txt
2. MDL Information Systems, now BIOVIA
3. http://www.daylight.com/dayhtml/doc/theory/theory.finger.html
4. Bender A. et al. (2004) *J. Chem. Inf. Model.*, 44, 170-8
5. Rogers D. and Hahn M. (2010) *J. Chem. Inf. Model.*, 50, 742-54
6. Riniker S. and Landrum G.A. (2013) *J. Cheminform.*, 5, 26

2D Fingerprint Similarity

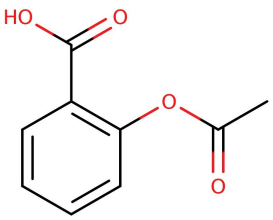
Similarity on Bitvectors

- Comparison of two molecules **A** and **B** reduces to the comparison of their corresponding bit-vectors **x** and **y** of length *N*:

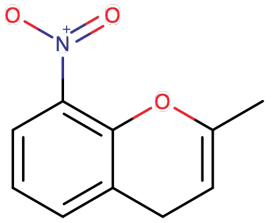
$$\mathbf{x} = (x_1, ..., x_N) \text{ and } \mathbf{y} = (y_1, ..., y_N)$$

- Required:
  Similarity measure *S*(x, y) yielding values in [0, 1]

A



B



x	0	1	0	0	0	1	0	1

y	0	1	0	1	0	1	1	0

2D Fingerprint Similarity

Tanimoto Coefficient

- Most popular similarity measure in cheminformatics: **Tanimoto Coefficient** [1] or Jaccard Coefficient [2]

- The *Tanimoto* for a pair of molecules **A** and **B** is calculated from their corresponding 2D fingerprints **x** and **y** by:

$$S_{Tan}(\mathbf{x}, \mathbf{y}) = \frac{\sum_{i=1}^{N} x_i y_i}{\sum_{i=1}^{N} (x_i^2 + y_i^2 - x_i y_i)}$$

1. Tanimoto T.T. (1957) *IBM Internal Report*
2. Jaccard P. (1901) *Bull. Soc. Vaud. sci. nat.*, 37, 547-79

2D Fingerprint Similarity

Tanimoto Coefficient

- Upon closer inspection, the Tanimoto coefficient divides the number of shared one-bits of **x** and **y** by the total number of one-bits (unique features) present in **x** or **y**:

$$S_{Tan}(\mathbf{x},\mathbf{y}) = \frac{\sum_{i=1}^{N} x_i y_i}{\sum_{i=1}^{N}(x_i^2 + y_i^2 - x_i y_i)} = \frac{c}{a+b-c}$$

$$\begin{cases} a = \sum_{i=1}^{N} x_i & \text{Number of 1-bits in } \mathbf{x} \\ b = \sum_{i=1}^{N} y_i & \text{Number of 1-bits in } \mathbf{y} \\ c = \sum_{i=1}^{N} x_i y_i & \text{Number of shared 1-bits between } \mathbf{x} \text{ and } \mathbf{y} \end{cases}$$

- Property values *a*, *b*, and *c* can be used to define a huge variety of similarity measures on bit-vectors

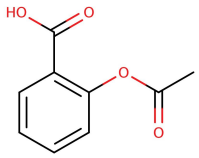
1. Tanimoto T.T. (1957) *IBM Internal Report*
2. Jaccard P. (1901) *Bull. Soc. Vaud. sci. nat.*, 37, 547-79

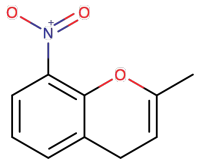
2D Fingerprint Similarity

Tanimoto Coefficient: Example

$$S_{Tan}(\mathbf{x},\mathbf{y}) = \frac{\sum_{i=1}^{N} x_i y_i}{\sum_{i=1}^{N}(x_i^2 + y_i^2 - x_i y_i)} = \frac{c}{a+b-c}$$

$$\begin{cases} a = \sum_{i=1}^{N} x_i & \text{Number of 1-bits in } \mathbf{x} \\ b = \sum_{i=1}^{N} y_i & \text{Number of 1-bits in } \mathbf{y} \\ c = \sum_{i=1}^{N} x_i y_i & \text{Number of shared 1-bits between } \mathbf{x} \text{ and } \mathbf{y} \end{cases}$$





x	0	1	0	0	0	1	0	1	$a = 3$
		↕				↕			$c = 2$
y	0	1	0	1	0	1	1	0	$b = 4$

$$S_{Tan}(\mathbf{x},\mathbf{y}) = \frac{c}{a+b-c} = \frac{2}{3+4-2} = 0.4$$

1. Tanimoto T.T. (1957) *IBM Internal Report*
2. Jaccard P. (1901) *Bull. Soc. Vaud. sci. nat.*, 37, 547-79

2D Fingerprint Similarity

Further Similarity Measures

- Numerous similarity measures are in use
  - Applicable also to continuous property vectors (middle)
  - Given ranges for binary forms

Hodgkin Index

- *Dice*
- *Czekanowski*
- *Sørensen*

$$S_{HI}(\mathbf{x},\mathbf{y}) = \frac{2 \times \sum_{i=1}^{N} x_i y_i}{\sum_{i=1}^{N} (x_i^2 + y_i^2)} = \frac{2c}{a+b}$$

[0, 1]

Cosine

- *Ochiai*
- *Carbó*

$$S_C(\mathbf{x},\mathbf{y}) = \frac{\sum_{i=1}^{N} x_i y_i}{\sqrt{\sum_{i=1}^{N} x_i^2 \times \sum_{i=1}^{N} y_i^2}} = \frac{c}{\sqrt{ab}}$$

[0, 1]

Willett et al. (1998) *J. Chem. Inf. Comput. Sci.*, 38, 983-96

2D Fingerprint Similarity

Distance Measures

- Besides similarity, **distance** is also interesting

Euclidean Distance

$$D_E(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{N}(x_i - y_i)^2} = \sqrt{a + b - 2c}$$

[0, *N*]

Hamming Distance

$$D_H(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{N}|x_i - y_i| = a + b - 2c$$

[0, *N*]

- *Manhattan*
- *City-Block*

Soergel Distance

$$D_S(\mathbf{x}, \mathbf{y}) = \frac{\sum_{i=1}^{N}|x_i - y_i|}{\sum_{i=1}^{N} max(x_i, y_i)} = \frac{a + b - 2c}{a + b - c}$$

[0, 1]

Willett et al. (1998) *J. Chem. Inf. Comput. Sci.*, 38, 983-96

2D Fingerprint Similarity

Similarity and Distance Measures

- Presented similarity measures yield values in range [0, 1]
- Trivial **conversion to distance** possible by:

  $D = 1 - S$

- Presented distance measures yield values in [0, 1] or [0, N]
- Also trivial **conversion to similarities** possible by:

  $S = N - D$ (Euclidean, Hamming)

  $S = 1 - D$ (Soergel)

Willett et al. (1998) *J. Chem. Inf. Comput. Sci.*, 38, 983-96

2D Fingerprint Similarity

Similarity and Distance Measures

- A distance coefficient is a **metric** if it satisfies:
  - Non-negativity: $D(\mathbf{x}, \mathbf{y}) \geq 0$
  - Definiteness: $D(\mathbf{x}, \mathbf{y}) > 0 \Leftrightarrow \mathbf{x} \neq \mathbf{y}$
  - Symmetry: $D(\mathbf{x}, \mathbf{y}) = D(\mathbf{y}, \mathbf{x})$
  - Triangle inequality: $D(\mathbf{x}, \mathbf{y}) \leq D(\mathbf{x}, \mathbf{z}) + D(\mathbf{z}, \mathbf{y})$

- D_E and D_H are metrics
- D_S is a metric for non-negative values
  - Otherwise doesn't obey triangle inequality
- Complements of S_{HI} and S_C do not fulfill the triangle inequality

Willett et al. (1998) *J. Chem. Inf. Comput. Sci.*, 38, 983-96

2D Fingerprint Similarity

Similarity and Distance Measures: Some Properties

- For binary data: D_S is the complement S_{Tan}

- Tanimoto contains a sort of **size normalization** by its denominator

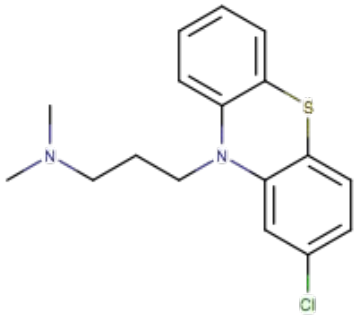
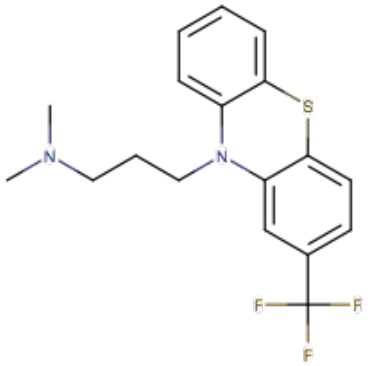
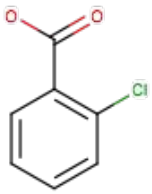
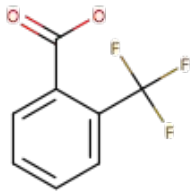
- S_{Tan}, S_{HI}, S_C directly depend on shared feature count
  - Similarity tends to increase with shared feature count
  - Smaller molecules can appear less similar (fewer 1-bits)

- S_H and S_E consider shared zero-bits as indicator of similarity
  - Absence of common features
  - Small molecules can appear closer

[LG] pp. 103 ff.
Willett et al. (1998) *J. Chem. Inf. Comput. Sci.*, 38, 983-96

2D Fingerprint Similarity

Similarity and Distance Measures: Some Properties

	Hamming	***Soergel***	
	78	0.33	
	41	0.41	
	24	0.86	

[LG] pp. 103 ff.
Willett et al. (1998) *J. Chem. Inf. Comput. Sci.*, 38, 983-96

Applications of Molecular Similarity

Overview

1. Similarity searching

- Given a query molecule, e.g. with known bioactivity
- Search a database for highly similar molecules (**SPP**!)

2. Cluster analysis

- Given a set of molecules
- Identify groups of similar molecules
  - Group molecules with common properties
  - Select representatives from groups to generate a diverse library

3. Diversity analysis

- Given a set of molecules
- Directly generate a diverse library

Applications of Molecular Similarity

Common Challenge

- **Chemical Space** (**Chemspace**)
  - Space spanned by all possible molecules
  - It is infinite!

- Focus on organic compounds with MW ≤ 500 Da
  - Subspace containing the druglike molecules
  - Simple estimates assume 10^{60} molecules [1]
  - GDB-17 [2]: 166.4 billion molecules (10^9)
    - Systematic enumeration of all organic molecules under certain constraints
    - ≤ 17 atoms of C, N, O, S, and halogens

1. Bohacek R.S. et al. (1996) *Med. Res. Rev.*, 16, 3-50
2. Ruddigkeit L. et al. (2012) *J. Chem. Inf. Model.*, 52, 2864-75

Similarity Searching

Problem

- **Given**: A **query molecule m_q** with a desired property
  for example colour, smell, biological activity, …

- **Problem**: Search structure database for highly similar molecules.
  According to the *Similar Property Principle* these molecules have a good chance to also possess the desired property.

- This approach is the most basic form of **Virtual Screening**

Similarity Searching

Problem

- Problem has two major variants

- **Fixed Threshold Search**
  - Given a user-defined lower similarity threshold S^t
  - Find all molecules database molecules $\mathbf{m}_i$ with $S_{Tan}(\mathbf{m}_q, \mathbf{m}_i) \geq S^t$

- ***k*-Nearest Neighbour Search (*k*-NN)**
  - Find database molecules $\{\mathbf{m}_1, \ldots, \mathbf{m}_k\}$ that are most similar to $\mathbf{m}_q$

Similarity Searching

Efficient Searching

- Standard cheminformatics application
- Core service of online structure databases

  ⇒ **Interactive user experience necessary**

  ⇒ **Highly efficient search strategies required**

- This can be implemented as a two-step procedure:

  1. **Search space pruning**
  2. Speedup of similarity calculation (discussed later)

Similarity Searching

Efficient Searching: Search Space Pruning

- Question:

  Is it possible to efficiently exclude database compounds from *expensive* similarity calculation?

- Swamidass and Baldi revisited similarity coefficients [1]
- Can we define an **upper bound** for similarities to $\mathbf{m_q}$

$$S_{Tan}(\mathbf{x},\mathbf{y}) = \frac{\sum_{i=1}^{N} x_i y_i}{\sum_{i=1}^{N}(x_i^2 + y_i^2 - x_i y_i)} = \frac{c}{a+b-c}$$

1. Swamidass S. J. and Baldi P. (2007) *J. Chem. Inf. Model.*, 47, 302-17

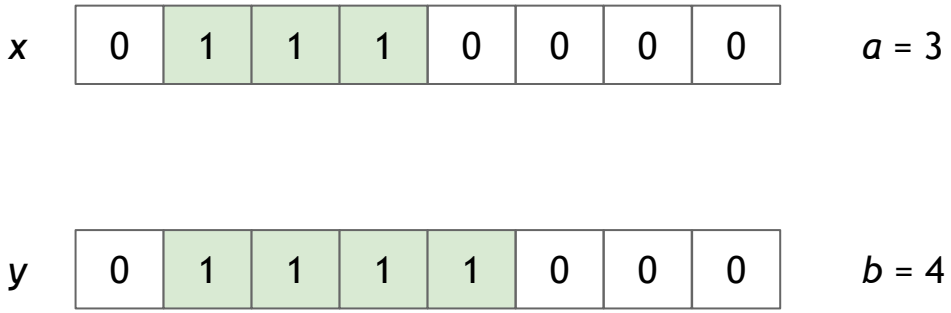
Similarity Searching

Efficient Searching: Search Space Pruning

- Given:

  2D Fingerprints **x** and **y** with property values *a*, *b*, and *c*

- Property values *a* and *b* are fixed



Swamidass S. J. and Baldi P. (2007) *J. Chem. Inf. Model.*, 47, 302-17

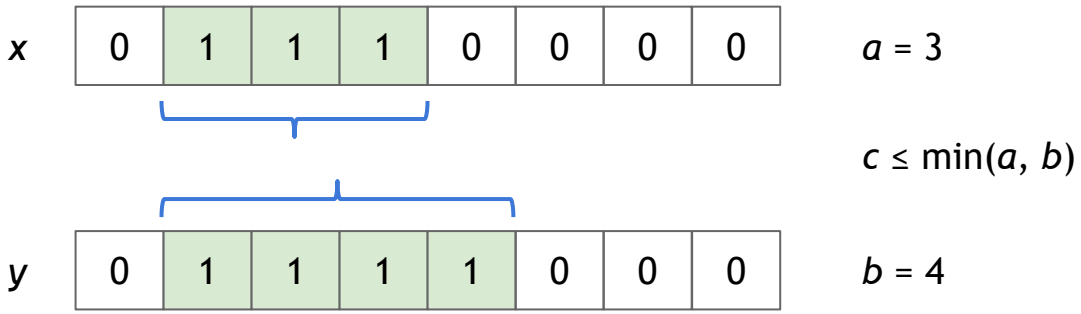
Similarity Searching

Efficient Searching: Search Space Pruning

- Given:

  2D Fingerprints **x** and **y** with property values *a*, *b*, and *c*

- **What is the maximum possible value of *c*?**



Swamidass S. J. and Baldi P. (2007) *J. Chem. Inf. Model.*, 47, 302-17

Similarity Searching

Efficient Searching: Search Space Pruning

- Given:

  2D Fingerprints **x** and **y** with property values *a*, *b*, and *c*

- **Upper similarity bound S^{UB} given by**

$$S_{Tan}(\mathbf{x},\mathbf{y}) \leq S_{Tan}^{UB}(\mathbf{x},\mathbf{y}) = \frac{c}{a+b-c}$$

$$= \frac{min(a,b)}{a+b-min(a,b)}$$

$$= \frac{min(a,b)}{max(a,b)}$$

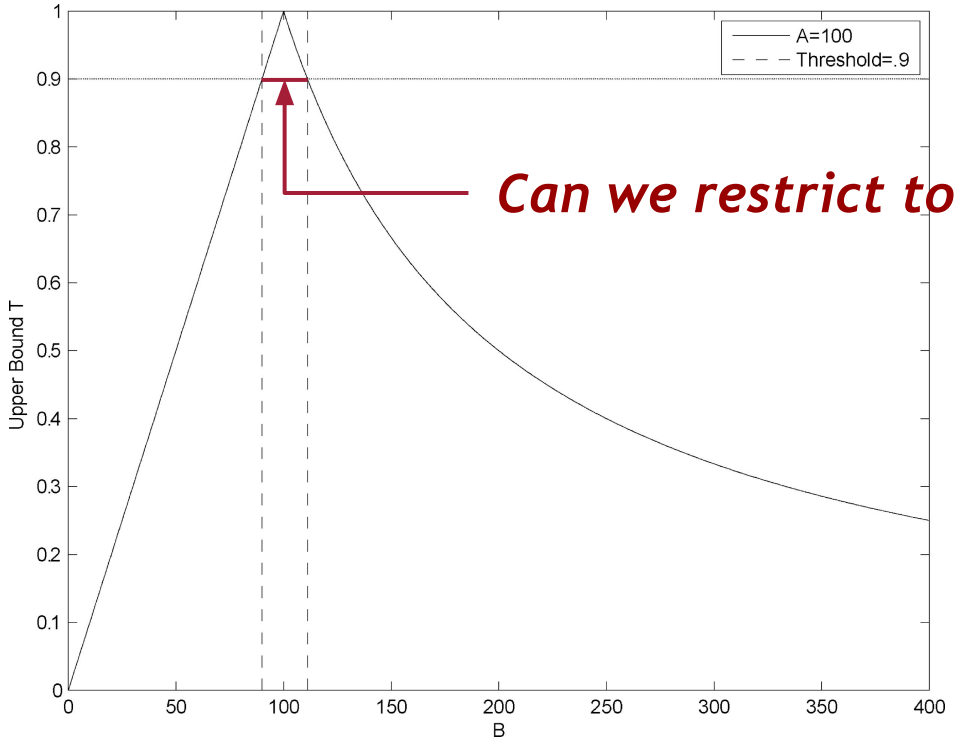
How does this help us?

Swamidass S. J. and Baldi P. (2007) *J. Chem. Inf. Model.*, 47, 302-17

Similarity Searching

Efficient Searching: Search Space Pruning

- Consider the following plot
  - Here, S^{UB} shown as a function of *b* for given a



Can we restrict to that region?

Swamidass S. J. and Baldi P. (2007) *J. Chem. Inf. Model.*, 47, 302-17

Similarity Searching

Efficient Searching: Search Space Pruning

- **Given**:
  - Similarity threshold S^t for database searching
  - Query compound $\mathbf{m}_q$ with fingerprint $\mathbf{f}_m$
  - Arbitrary database compound $\mathbf{m}_i$ with fingerprint $\mathbf{f}_i$
  - Number of 1-bits in $\mathbf{f}_m = a$ and $\mathbf{f}_i = b$

- **Goal:**

  Test if $\mathbf{m}_i$ can be discarded without calculating c

Swamidass S. J. and Baldi P. (2007) *J. Chem. Inf. Model.*, 47, 302-17

Similarity Searching

Efficient Searching: Search Space Pruning

- We can discard $\mathbf{m_i}$ if the following is true

$$S^t > S^{UB}_{Tan}(\mathbf{m_q}, \mathbf{m_i}) = \frac{min(a,b)}{max(a,b)}$$

- According to this observation we can state the following

$$\text{If } b \leq a \implies S^{UB}_{Tan} = \frac{b}{a}$$

$$\implies \text{all } \mathbf{m_i} \text{ with } \frac{b}{a} < S^t \text{ can be discarded}$$

$$\text{If } b \geq a \implies S^{UB}_{Tan} = \frac{a}{b}$$

$$\implies \text{all } \mathbf{m_i} \text{ with } \frac{a}{b} < S^t \text{ can be discarded}$$

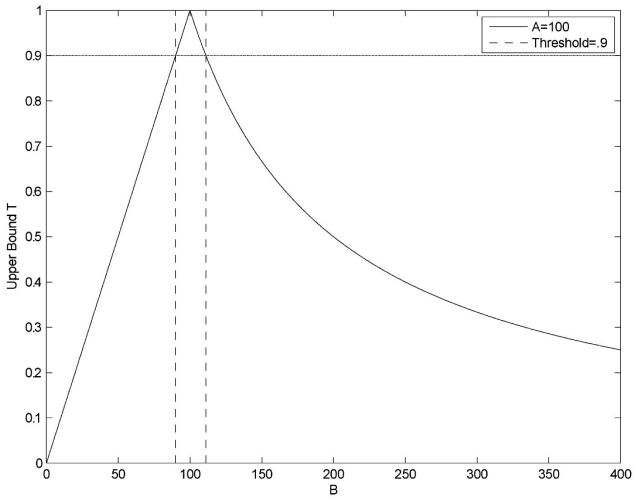
Swamidass S. J. and Baldi P. (2007) *J. Chem. Inf. Model.*, 47, 302-17

Similarity Searching

Efficient Searching: Search Space Pruning

- For a given query molecule $\mathbf{m}_q$ we can restrict the search to all molecules $\mathbf{m}_i$ that satisfy

$$aS^t \leq b \leq \frac{a}{S^t}$$



- Required values can be pre-calculated and stored
- For a given query they can simply be looked up

Swamidass S. J. and Baldi P. (2007) *J. Chem. Inf. Model.*, 47, 302-17

Similarity Searching

Closing Remarks

- **Pros**
  - Only a single query molecule required
  - Highly efficient searching possible

- **Cons**
  - Purely topological
  - Choice of 2D fingerprint and similarity coefficient not obvious

Cluster Analysis

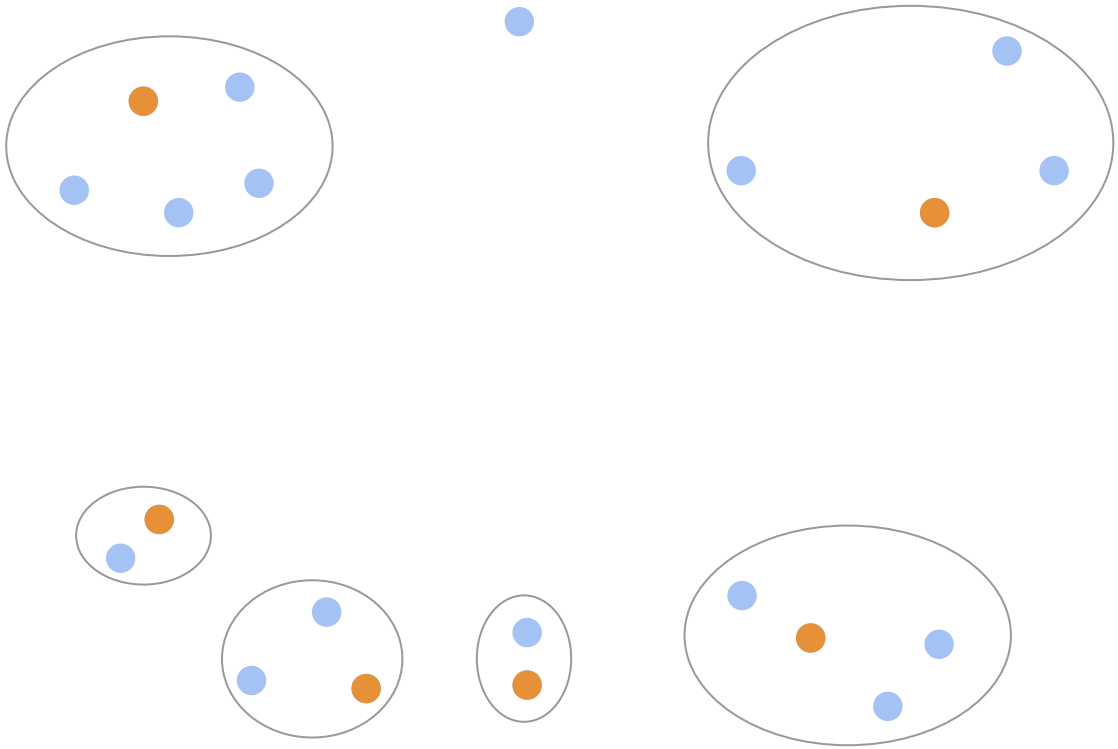
Overview

- Clustering is used a lot in cheminformatics
  - **Analysis of HTS results**
  - **Scaffold hopping**
  - **Library generation**
  - ...

- Depending on the problem size different approaches are used
  - Problem size is the number of molecules to be clustered

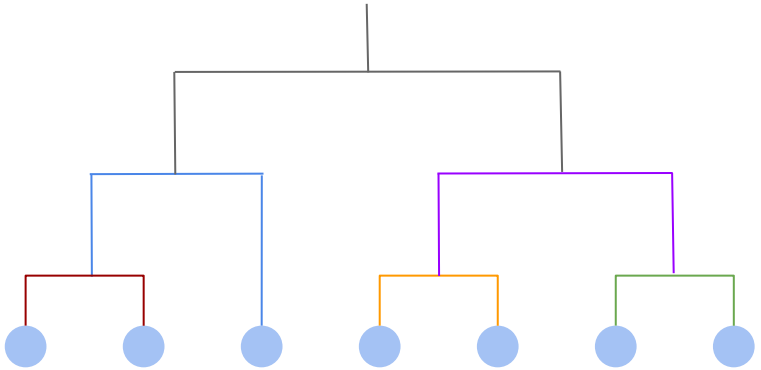
Cluster Analysis

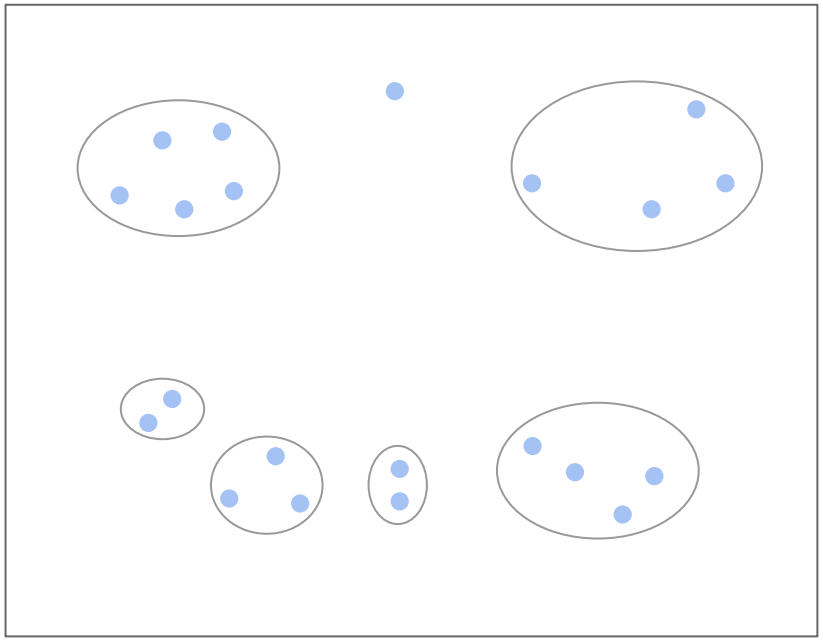
Overview

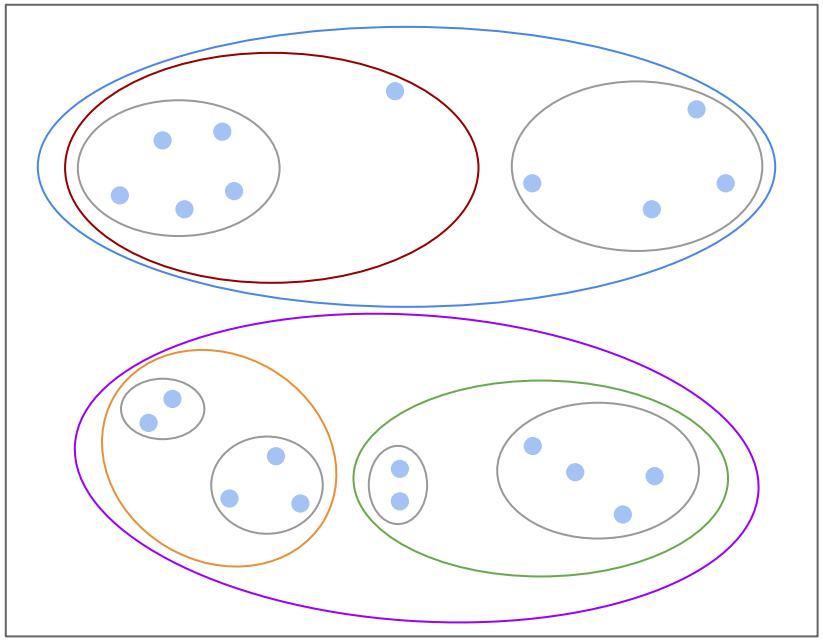


Cluster Analysis

Overview: Strategies







Non-hierarchical clustering methods

Hierarchical clustering methods

Cluster Analysis

Hierarchical Methods

- Two variants

1. **Agglomerative**:
    a. Initially, each structure forms its own cluster
    b. Iterative merging of closest pair until one cluster is left

2. **Divisive:**
    a. Initially, all structures assigned to a single cluster
    b. Iterative splitting of clusters

Cluster Analysis

Agglomerative Hierarchical Methods

- Variants:
  - Linkage methods: single, complete, and average
  - Ward's minimum variance method
  - Differ in their **similarity update formula** [1]

- **Generic algorithm**

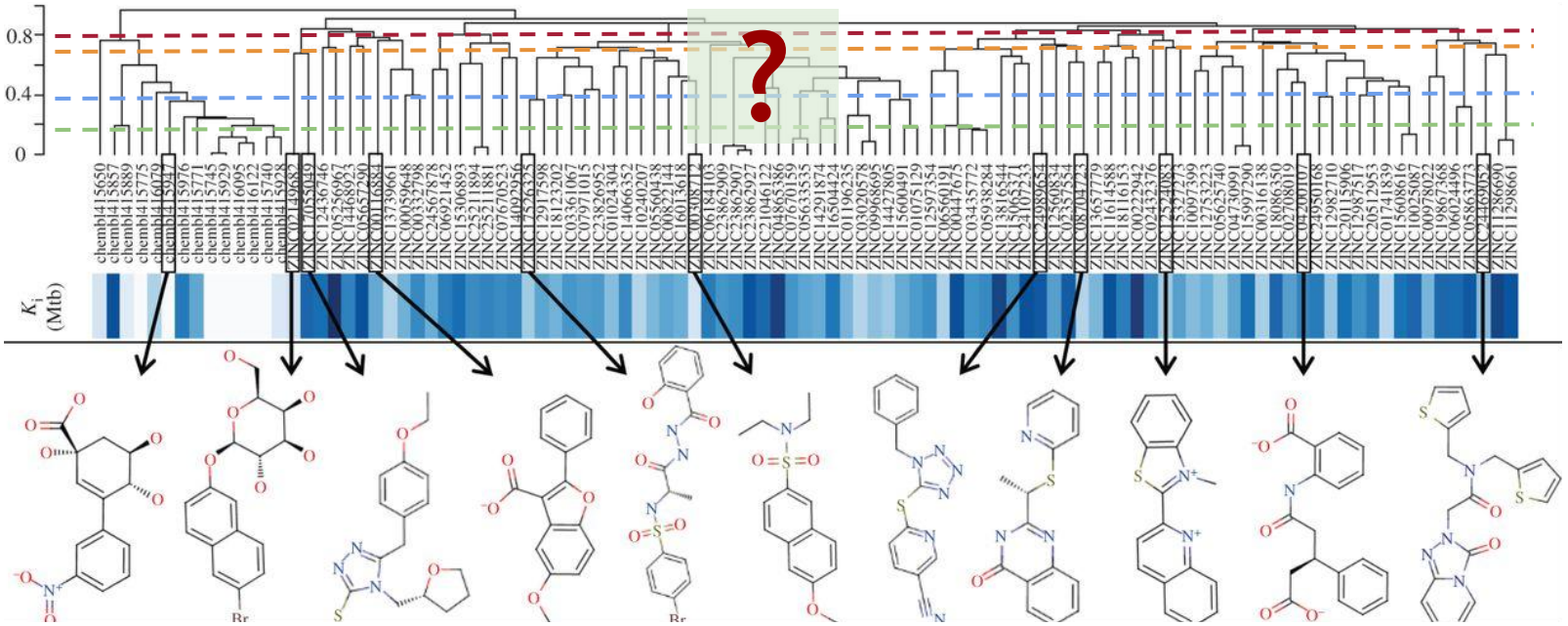
1. Calculate all pairwise similarities
2. While |Clusters| > 1:
   a. Merge most similar cluster pair
   b. Update similarities

1. Lance G. and Williams W. A. (1967) *Comput. J.*, 9, 373-80

Cluster Analysis

Agglomerative Hierarchical Methods

- Resulting cluster tree does not yield clusters directly
- Methods for cluster level selection have to be applied
  - Not necessarily straightforward



Ballester P.J. et al. (2012) *J. R. Soc. Interface.*, 9, 3196-207

Cluster Analysis

Agglomerative Hierarchical Methods

- Most important approach for smaller datasets
- For library generation representative selection has to be performed

- **Pros**
  - Produces **very homogeneous clusters** (Ward and Average)
  - Hierarchy provides useful information
- **Cons**
  - **Expensive**: $\mathcal{O}(n^2)$ space and $\mathcal{O}(n^3)$ time complexity in general
  - **Cluster level selection** required to generate clusters

Cluster Analysis

Non-Hierarchical Methods

- Three major strategies used in cheminformatics

1. **Single Pass**
   - A single scan through the set of molecules
   - E.g. Leader clustering [1]
2. **Relocation**
   - Select initial cluster centers and refine them iteratively
   - E.g. *k*-Medoids [2]
3. **Nearest Neighbour (NN)**
   - Based on sets of nearest neighbours for all compounds
   - E.g. Jarvis-Patrick [3]

1. Hartigan J. A. (1975) *Clustering Algorithms*, John Wiley & Sons, NY
2. Vinod H. D. (1969) *J. Am. Stat. Assoc.*, 64, 506-19
3. Jarvis R. A. and Patrick E. A (1973) *IEEE Trans. Comput.*, C22 1025-34

Cluster Analysis

Jarvis-Patrick

- Frequently used in cheminformatics
- Applicable to large datasets

- **Generic algorithm**

Input: Parameters k and $j \leq k$

1. For each molecule $m_i \in \{M_1, \ldots, M_n\}$ calculate N_i, the list of its k NN
2. Molecules m_r and m_s cluster together if conditions a and b are true:
   a. $m_r \in N_s \quad and \quad m_s \in N_r$
   b. $|N_r \cap N_s| \geq j$

Cluster Analysis

Jarvis-Patrick

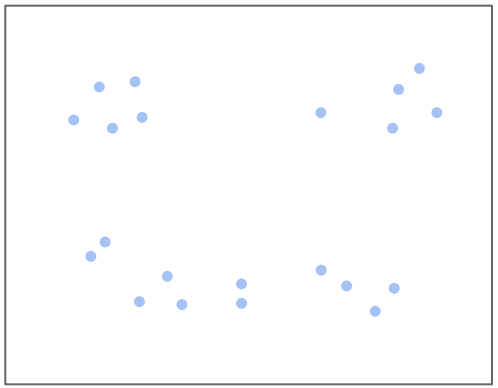
- **Pros**
  - Fast implementation possible
  - Applicable to very large data sets
  - Not dependent on input ordering

- **Cons**
  - Parameter choice of *k* and *j* often difficult
  - Can hield heterogeneous clusters

Diversity Analysis

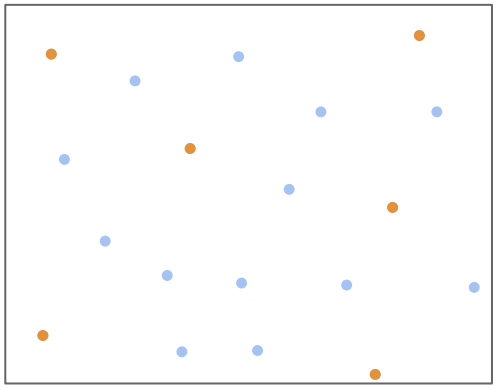
Overview

- Dissimilarity in the focus



- **Library comparison**
  - Which library is more diverse

- **Library generation**
  - Select *k* molecules as diverse as possible



- **How to measure diversity?**

Diversity Analysis

Measures for Library Comparison

- For example based on Tanimoto

  1. **Mean Inter-Molecular Similarity (MIMS)**:

$$MIMS = \frac{1}{N^2} \sum_{i,j}^{N} S_{Tan}(\mathbf{m}_i, \mathbf{m}_j)$$

  2. **Mean Inter-Molecular Dissimilarity (MIMD)**:

$$MIMD = 1 - MIMS$$

- MIMD is a measure of **relative diversity**
  - I.e. how strongly molecules differ from each other
- It is **not an absolute measure** of covered chemspace

Diversity Analysis

Library Generation

- Select *k* molecules from a library as diverse as possible
- In contrast to clustering a **direct method**
- Number of possible subsets of size *k* is large

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

- Exact approach: **maximum diversity problem**
  - MIMD as a target function to be maximized
  - Combinatorial optimization problem
  - NP-hard [1]

1. Kuo C.C. et al. (1993) *Dec. Sci.*, 24, 1171-85

Diversity Analysis

Library Generation

- Heuristics are used to solve problem in acceptable time
- **Dissimilarity-Based Compound Selection** (**DBCS**) [1]
  - Basic algorithm, often varied
  - Often referred to as MaxMin method

Input: Library *M* of size *n*, parameter *k*

Output: Set *S* containing *k* diverse molecules

1. Select initial seed molecule and move it to *S*
2. Repeat *k-1* times:
   a. For each $m \in M$:
      Calculate $d_m = \min(D(m,s))$ for all $s \in S$
   b. Select $m \in M$ with largest d_m and move to *S*

1. Lajiness M.S. (1990) *Computational chemical graph theory*, Nova Science Publishers, 299-316

Speeding Up Similarity Calculation

Overview

- A lot of effort has been spent on this in the recent years
  - After a rather long and calm period
- Reasons
  1. **Steady growth of accessible compounds**
  2. **Often full similarity matrix required (cf. clustering)**

- Naive Tanimoto implementations are too slow
- Property values *a* and *b* are calculated once ⇒ cheap
- **Shared feature count *c* is the expensive part**
  - Has to be calculated for every molecule pair

Speeding Up Similarity Calculation

Overview

- Naive approach for shared feature count *c*

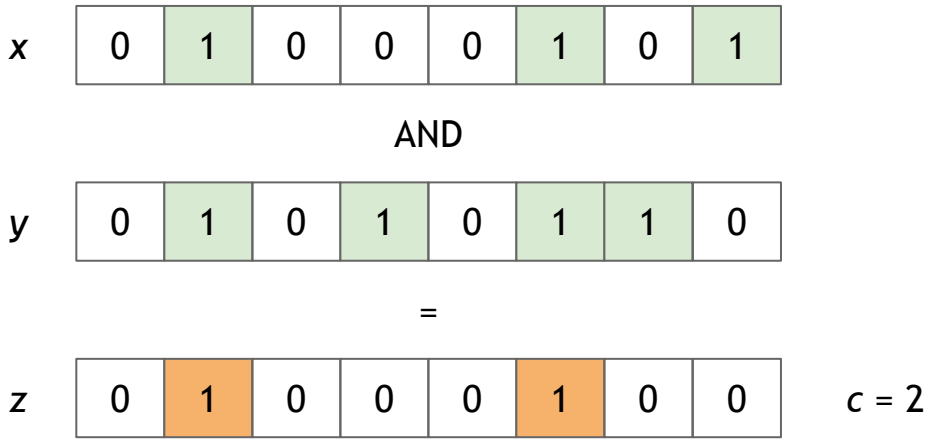
In : Fingerprints **x** and **y**
Out: Shared feature count c
$c = 0;$
for $i = 0$ **to** $n - 1$ **do**
    $c \mathrel{+}= x_i \times y_i$;
end
return c

- Approaches to speed up this step can be grouped into:
  1. **Hardware speedup of similarity calculation**
  2. **Algorithmic speedup of similarity calculation**

Similarity Calculation: Hardware Speedup

Population Count

- Shared feature count *c* of bitvectors **x** and **y**:
  1. **z = x** ∧ **y**
  2. Calculating 1-bits in **z**



1. Haque I. et al. (2011) *J. Chem. Inf. Model.*, 51, 2345-51

Similarity Calculation: Hardware Speedup

Population Count

- Shared feature count *c* of bitvectors **x** and **y**:
  1. **z** = **x** ∧ **y**
  2. Calculating 1-bits in **z**
- Step 2 is known as **popcount** (population count)

- In modern CPUs and GPUs it is available as hardware instruction
  - **Single-cycle throughput**
  - Popcount on full 64-bit words
- Speedup of factor 20-40 over naive CPU implementations [1]

1. Haque I. et al. (2011) *J. Chem. Inf. Model.*, 51, 2345-51

Similarity Calculation: Algorithmic Speedup

Overview

- When calculating e.g. all pairwise similarities like this ...

```
In  : Molecule set M with n fingerprints of length l
Out: Matrix of pairwise shared feature counts C
C = 0;
for i = 1 to n do
    for j = i + 1 to n do
        for k = 0 to l − 1 do
            c_ij += m_ik × m_jk ;
        end
    end
end
```

… one very often

1. **revisits the same fingerprint positions**
2. **also the 0-bits have to be iterated**

Similarity Calculation: Algorithmic Speedup

Inverted Index Method

- Prevented by using **inverted index** (**InvID**) data structure
  - A technique developed for information retrieval

- Idea: Molecule fingerprints are lists of features.

  An **InvID** is a **list of molecules**.

  An **InvID** represents a **single feature**.

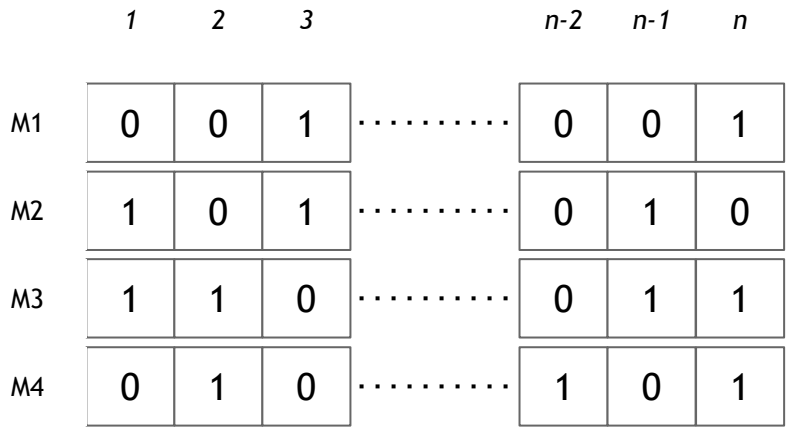
  Molecules in an **InvID** possess the corresponding feature.

  Effectively a **reordered data structure** without 0-bits.

Nasr R. et al. (2012) *J. Chem. Inf. Model.*, 52, 891-900
Thiel P. et al. (2014) *J. Chem. Inf. Model.*, 54, 2395-401

Similarity Calculation: Algorithmic Speedup

Inverted Index Method: Example

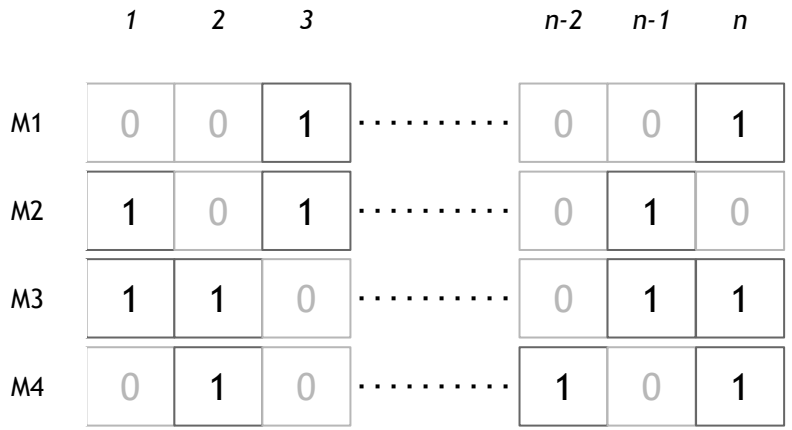


Fingerprints

Nasr R. et al. (2012) *J. Chem. Inf. Model.*, 52, 891-900
Thiel P. et al. (2014) *J. Chem. Inf. Model.*, 54, 2395-401

Similarity Calculation: Algorithmic Speedup

Inverted Index Method: Example

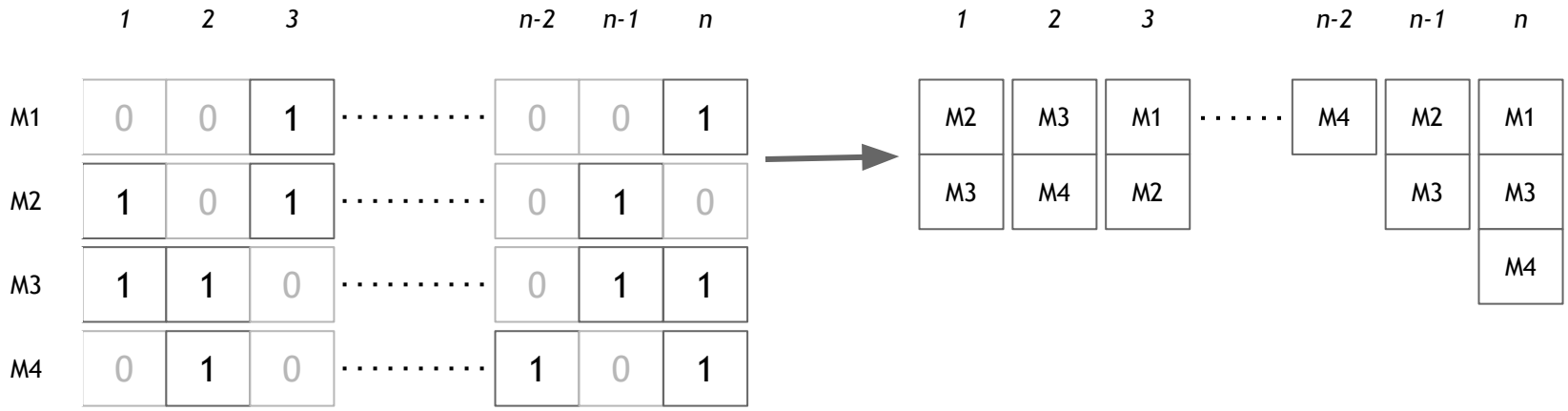


Fingerprints

Nasr R. et al. (2012) *J. Chem. Inf. Model.*, 52, 891-900
Thiel P. et al. (2014) *J. Chem. Inf. Model.*, 54, 2395-401

Similarity Calculation: Algorithmic Speedup

Inverted Index Method: Example

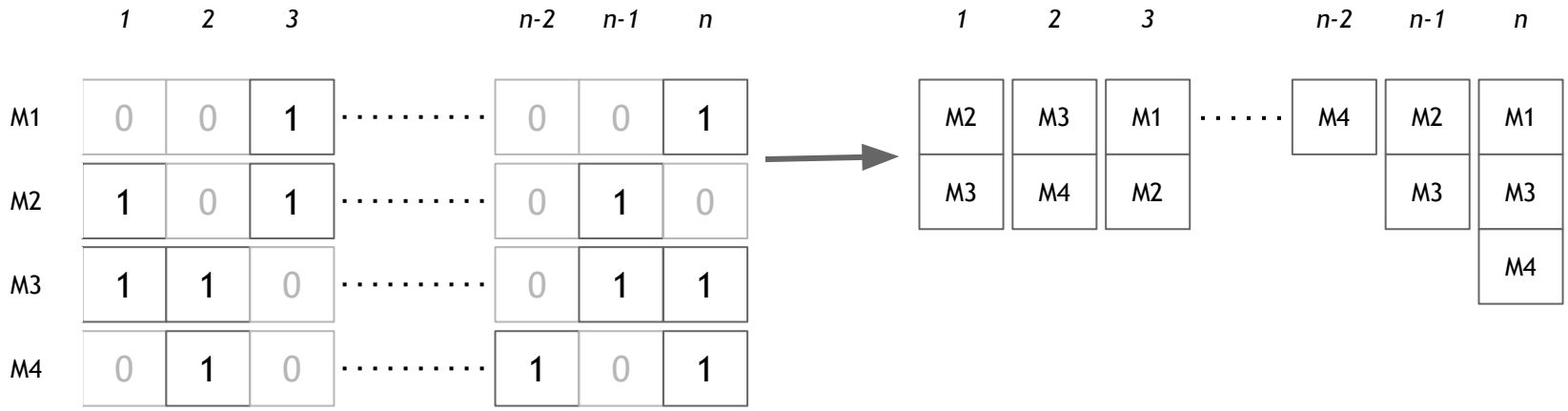


Fingerprints *Inverted Index*

Nasr R. et al. (2012) *J. Chem. Inf. Model.*, 52, 891-900
Thiel P. et al. (2014) *J. Chem. Inf. Model.*, 54, 2395-401

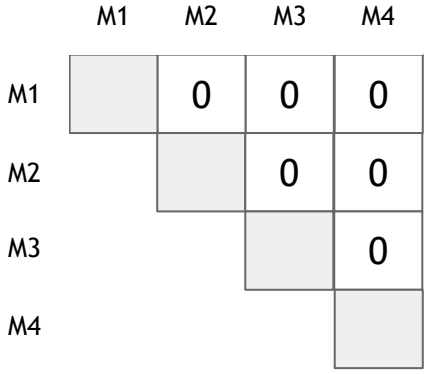
Similarity Calculation: Algorithmic Speedup

Inverted Index Method: Example



Fingerprints *Inverted Index*

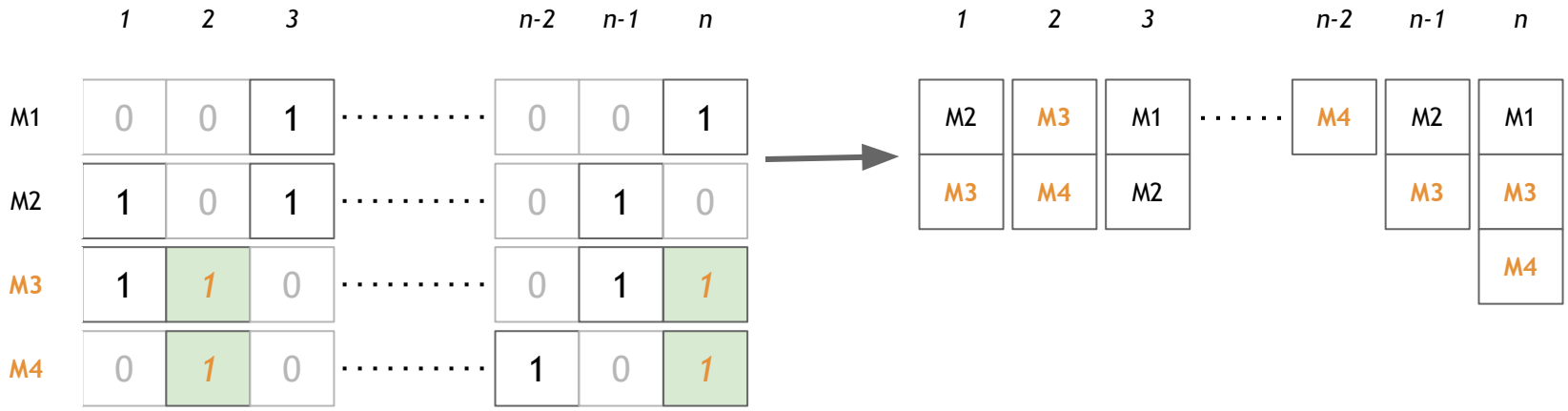
Shared feature count matrix



Nasr R. et al. (2012) *J. Chem. Inf. Model.*, 52, 891-900
Thiel P. et al. (2014) *J. Chem. Inf. Model.*, 54, 2395-401

Similarity Calculation: Algorithmic Speedup

Inverted Index Method: Example



Fingerprints *Inverted Index*

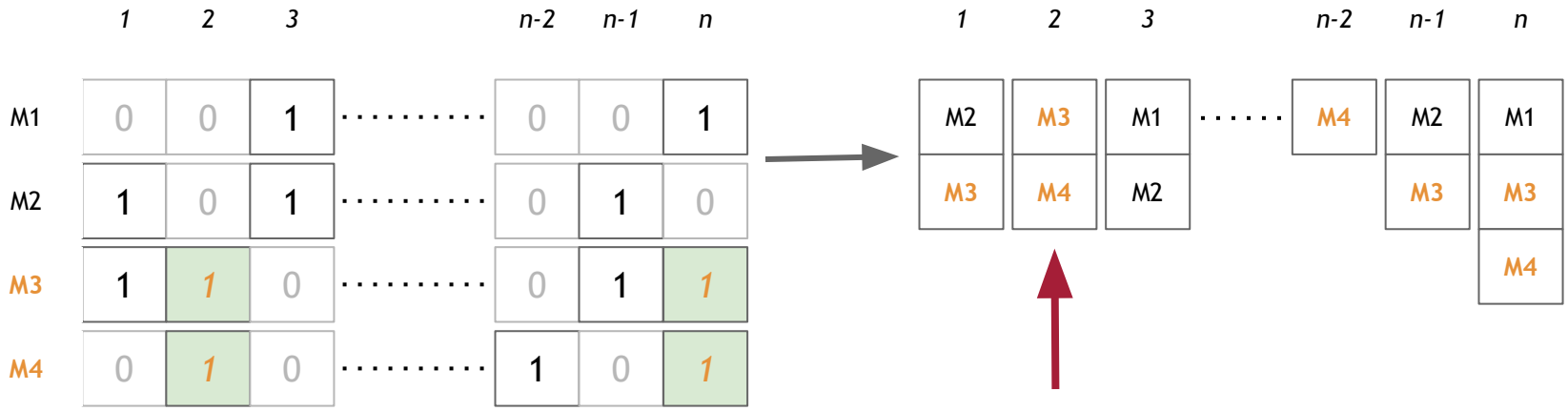
Shared feature count matrix

	M1	M2	M3	M4
M1		0	0	0
M2			0	0
M3				0
M4				

Nasr R. et al. (2012) *J. Chem. Inf. Model.*, 52, 891-900
Thiel P. et al. (2014) *J. Chem. Inf. Model.*, 54, 2395-401

Similarity Calculation: Algorithmic Speedup

Inverted Index Method: Example



Fingerprints *Inverted Index*

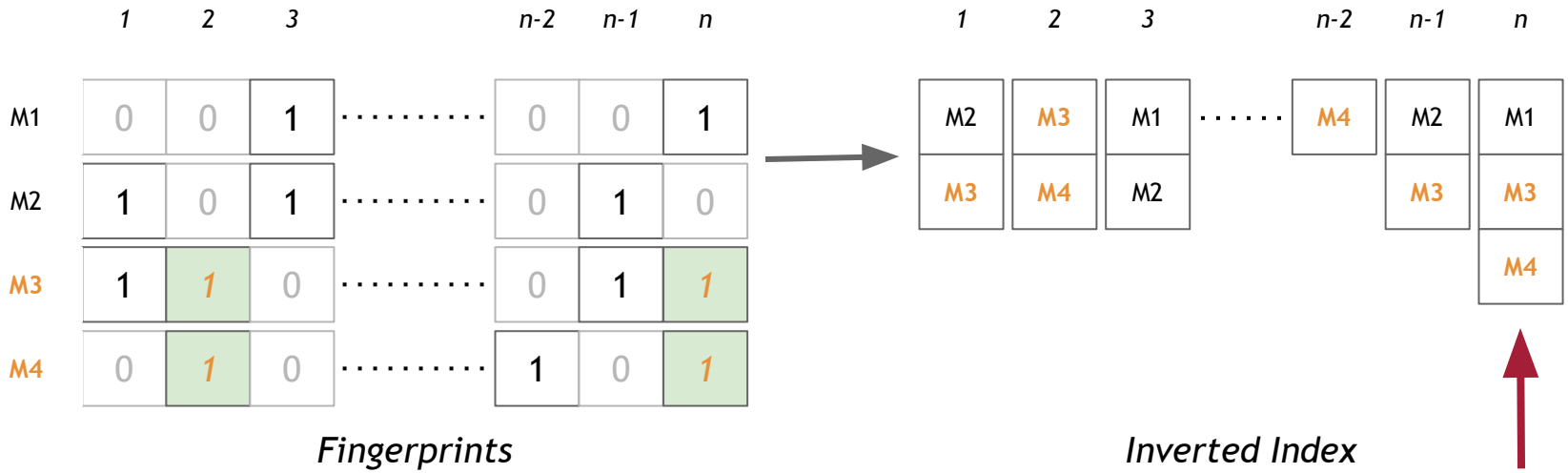
Shared feature count matrix

	M1	M2	M3	M4
M1		0	0	0
M2			0	0
M3				1
M4				

Nasr R. et al. (2012) *J. Chem. Inf. Model.*, 52, 891-900
Thiel P. et al. (2014) *J. Chem. Inf. Model.*, 54, 2395-401

Similarity Calculation: Algorithmic Speedup

Inverted Index Method: Example



Fingerprints *Inverted Index*

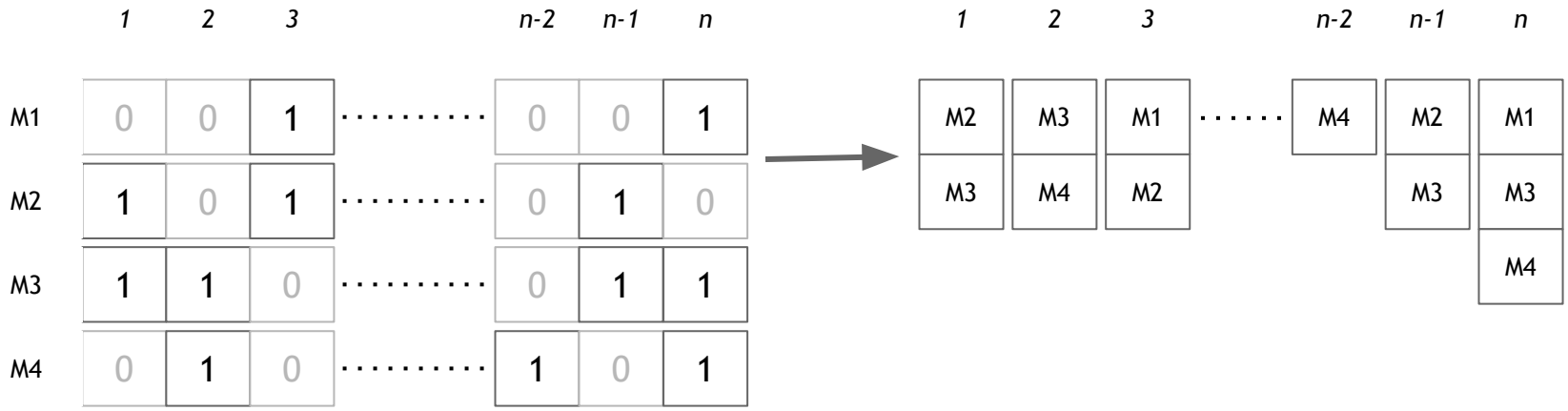
Shared feature count matrix

	M1	M2	M3	M4
M1		0	0	0
M2			0	0
M3				2
M4				

Nasr R. et al. (2012) *J. Chem. Inf. Model.*, 52, 891-900
Thiel P. et al. (2014) *J. Chem. Inf. Model.*, 54, 2395-401

Similarity Calculation: Algorithmic Speedup

Inverted Index Method: Example



Fingerprints *Inverted Index*

Shared feature count matrix

	M1	M2	M3	M4
M1		1	1	1
M2			2	0
M3				2
M4				

Nasr R. et al. (2012) *J. Chem. Inf. Model.*, 52, 891-900
Thiel P. et al. (2014) *J. Chem. Inf. Model.*, 54, 2395-401

Similarity Calculation: Algorithmic Speedup

Inverted Index Method Tanimoto

- **Generic algorithm:**

Input: Fingerprints $F = \{f_1, \ldots, f_n\}$ and associated 1-bit counts $K = \{k_1, \ldots, k_n\}$

Output: Tanimoto for all pairs of fingerprints

1. Generate **InvID** from *F*
2. Generate 0-initialized shared feature counts matrix *S*
3. Iterate **InvID** and fill *S*
1. For each s_{ij} (upper triangular matrix only!)
   a. Calculate Tanimoto for fingerprint pair f_i and f_j:

$$S_{Tan}(f_i, f_j) = s_{ij} \div (k_i + k_j - s_{ij})$$

Nasr R. et al. (2012) *J. Chem. Inf. Model.*, 52, 891-900
Thiel P. et al. (2014) *J. Chem. Inf. Model.*, 54, 2395-401

Speeding Up Similarity Calculation

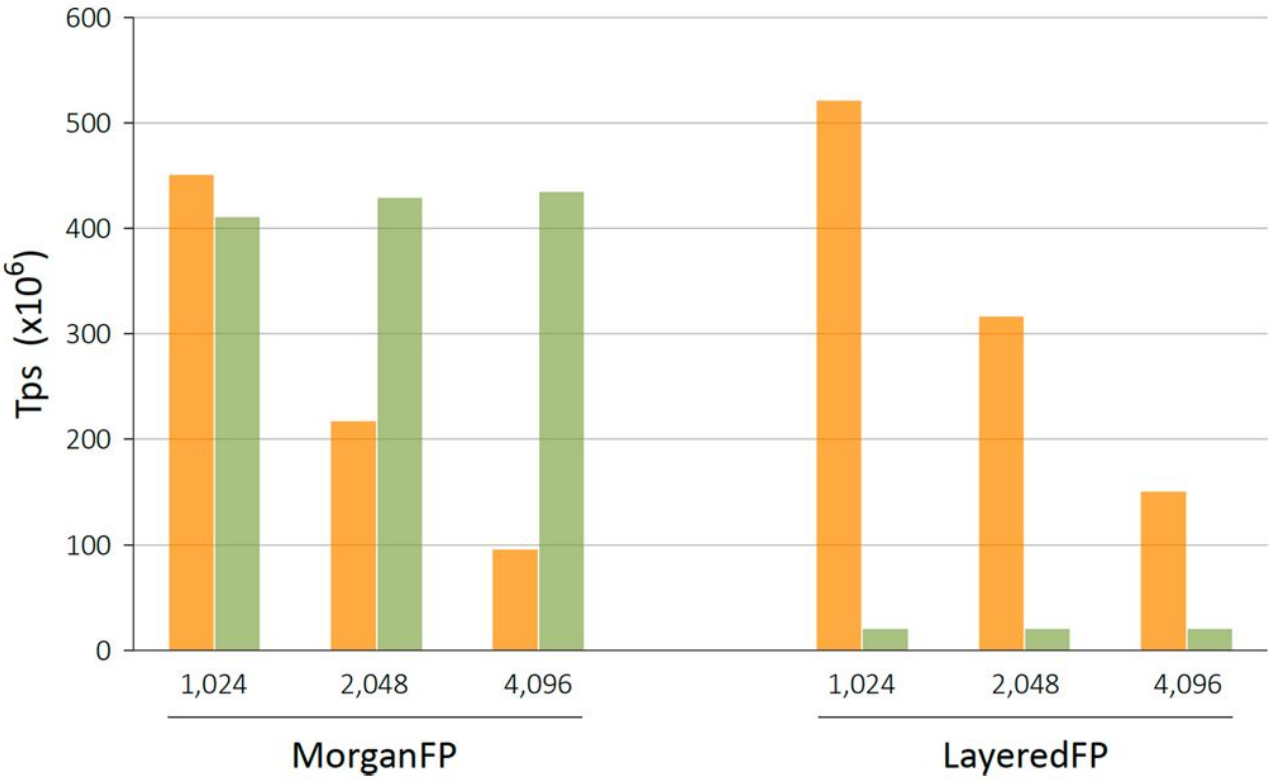
Comparison

- Naive Tanimoto calculates < 1M similarities per second
- Optimized hardware Tanimoto highly efficient
  - CPU: ≥ 200M Tan / sec
  - GPU: ≥ 1000M Tan / sec
  - Sensitive to fingerprint length
  - Insensitive to fingerprint density
- Optimized InvID Tanimoto also very efficient
  - ≥ 100M Tan / sec on a single CPU core
  - Insensitive to fingerprint length
  - Sensitive to fingerprint density

Haque I. et al. (2011) *J. Chem. Inf. Model.*, 51, 2345-51
Nasr R. et al. (2012) *J. Chem. Inf. Model.*, 52, 891-900
Thiel P. et al. (2014) *J. Chem. Inf. Model.*, 54, 2395-401

Speeding Up Similarity Calculation

Benchmark on 4-Core CPU



Green bars	InvIND
Orange bars	Hardware
MorganFP	Sparse FPs
LayeredFP	Dense FPs

http://chemfp.com
Haque I. et al. (2011) *J. Chem. Inf. Model.*, 51, 2345-51
Thiel P. et al. (2014) *J. Chem. Inf. Model.*, 54, 2395-401

Summary

- MCS correlates with local structural similarity
- Overlap of feature lists can indicate global similarity
- 2D Fingerprints are bitvector representations of feature lists
- Similarity and Distance measures can be defined on bitvectors
- Tanimoto is arguably most similarity measure in cheminformatics
- Similarity searching most basic application
- Bounds on maximum similarity can speedup database search
- Clustering is used for data analysis and library generation
- Diversity analysis is used for library comparison and generation
- Naive Tanimoto is slow
- Significant speedup by hardware and algorithmic techniques
- Popcount and inverted index

Text Books:

- LG — Leach A. and Gillet V., Revised Edition, Springer, 2007
  An Introduction to Chemoinformatics
- GE — Gasteiger J. and Engel T. (Eds.), 1st Ed., Wiley-VCH, 2003
  Chemoinformatics - A Textbook
- KA — Kerber A. et al.
  Mathematical Chemistry and Chemoinformatics, De Gruyter, 2014

Acknowledgments:

- 2D structure drawings were generated with ChemAxon **MarvinSketch**
  - `https://www.chemaxon.com/products/marvin/marvinsketch`
- 3D structures were generated with **BALLView**
  - `http://www.ball-project.org`
  - Hildebrandt A. et al. (2010) *BMC Bioinformatics*, 11, 531
  - Moll A. et al. (2006) *Bioinformatics*, 22, 365-6