The knapsack algorithm in analytical chemistry

Myriam Guillevic, *Aurore Guillevic*, Martin K. Vollmer, Paul Schlauri, Matthias Hill, Lukas Emmenegger, Stefan Reimann

Laboratory for Air Pollution /Environmental Technology,
Empa, Swiss Federal Laboratories for Materials Science and Technology
Dübendorf, Switzerland
Université de Lorraine, CNRS, Inria, LORIA
Nancy, France

June 29, 2021

Empa Materials Science and Technology

Inria

https://members.loria.fr/AGuillevic/files/talks/21-Nancy-knapsack.pdf

The team, 2018–2021

- Myriam Guillevic, PhD in Climate Sciences with V. Masson-Delmotte,
  now at Group for Climate Gases, EMPA, ETH, Dübendorf, Switzerland
  `https://www.empa.ch/web/s503/climate-gases`
  Like LNE-CNAM in France (Laboratoire National de Métrologie et d'Essais, Conservatoire National des Arts et Métiers)
- Aurore Guillevic, computer scientist, Inria Nancy,
  in 2017–2020: adjunct assistant professor, Introduction to algorithms and programming in Java (INF411, J.-C. Filliâtre), Python (CSE103, H. Zhou, I. Mackie) at École Polytechnique, Palaiseau, France

Aims

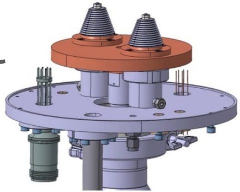
Trace gases: 0.066% of the (dry) atmosphere, gases other than nitrogen (78.1%), oxygen (20.9%) and argon (0.934%) (i.e. 99.934%, water vapor excluded).

- Measuring trace gases known to be present (e.g. CO_2, CH_4, $CFCl_3$):
  target screening (mass spectra in databases)
  e.g. banned or regulated substances (Montreal protocol)
- Searching for expected/potential pollutants (known to be used in industry) and start monitoring them before emissions to the atmosphere is rising:
  suspect screening (chemical compound known, e.g. HFO-1234yf)
- Searching for unexpected pollutants or unknown unknowns:
  non-target screening (detecting new subtances in the air)
  e.g. industrial disaster (Lubrizol fire in Rouen in 2019, Beyrouth harbour explosion in 2020)

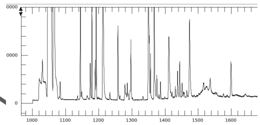
How to search for unknown unknowns?

Empa
Materials Science and Technology

Target screening : Aprecon – GC – ToF-MS



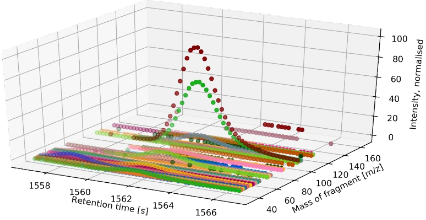
Pre-concentration (APRECON)

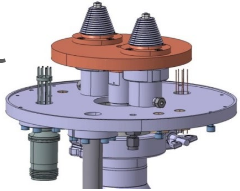


Gas Chromatography GasPro column

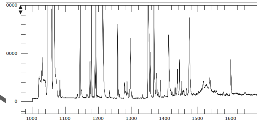
Target screening : Aprecon – GC – ToF-MS

Empa
Materials Science and Technology





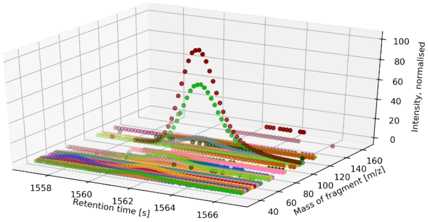
Pre-concentration (APRECON)

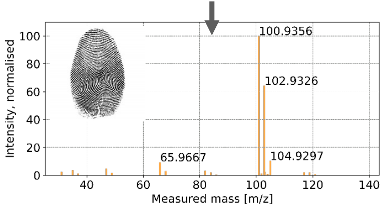


Gas Chromatography
GasPro column

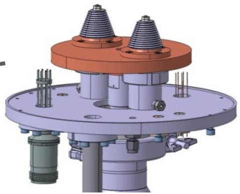
Electron impact (EI)
Time-of-Flight
Mass spectrometer
Tofwerk AG

Target screening : Aprecon – GC – ToF-MS

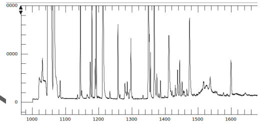




Electron impact (EI)
Time-of-Flight
Mass spectrometer
Tofwerk AG

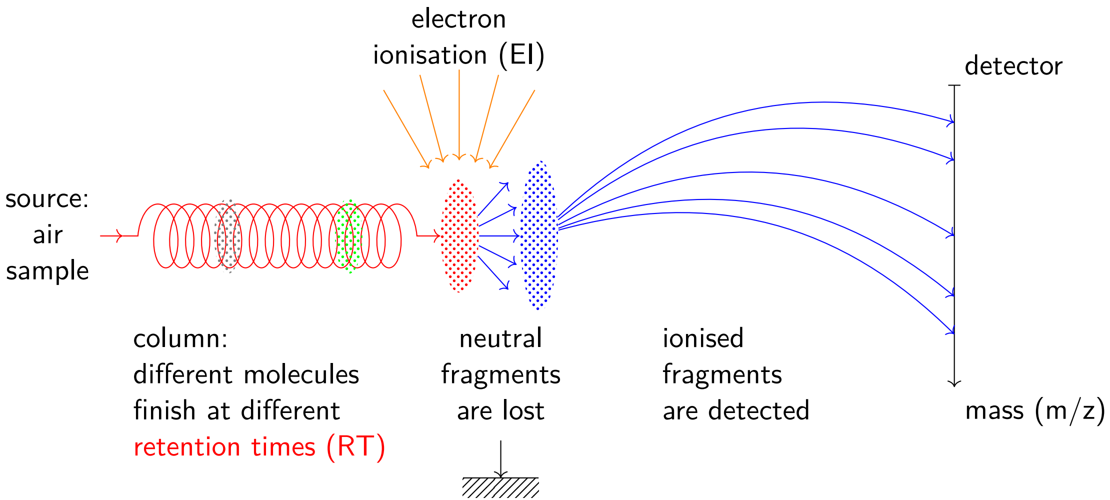


Pre-concentration (APRECON)

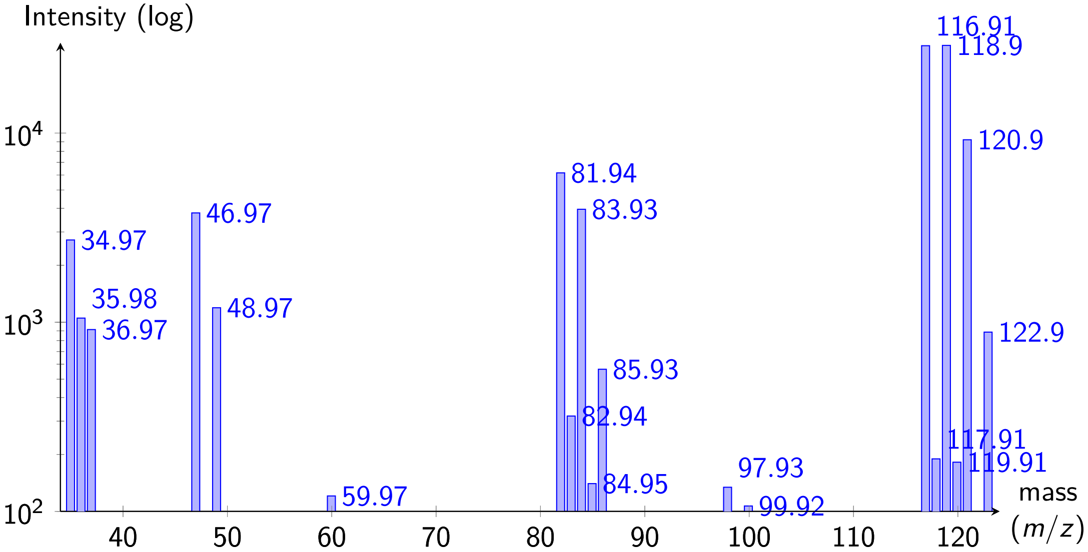


Gas Chromatography
GasPro column

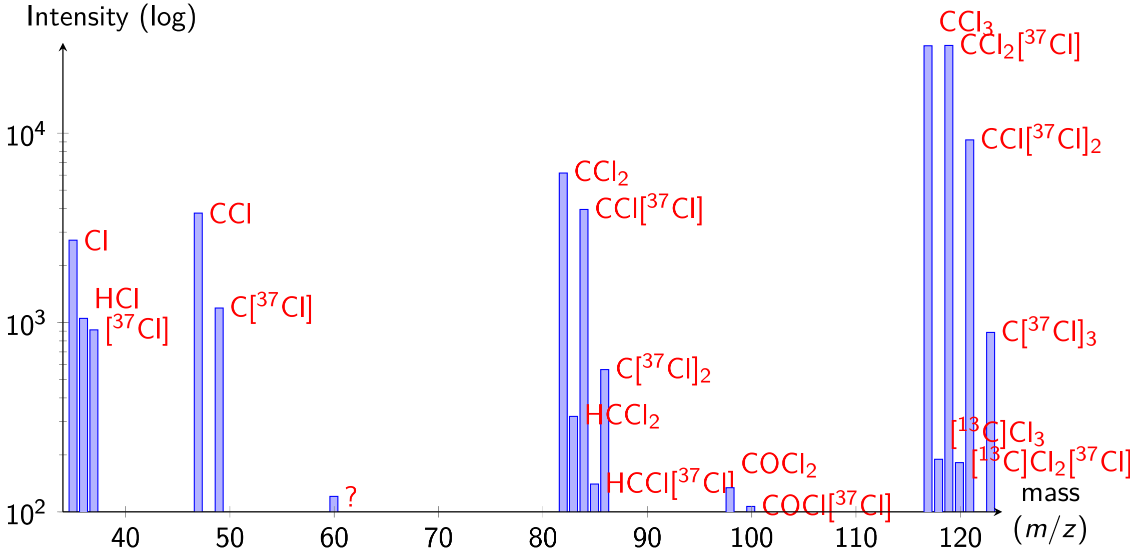
EI ToF MS: Electron Ionisation Time-of-Flight Mass Spectrometer



Input data: mass spectrum



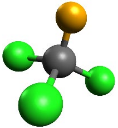
Aim: annotate the figure



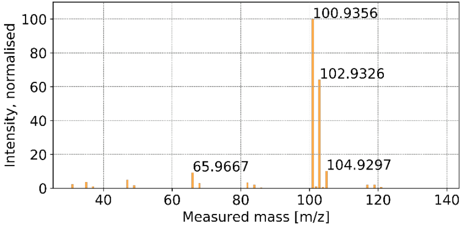
Target vs non-target screening

Empa
Materials Science and Technology

Target screening:



CFC-11
$CFCl_3$

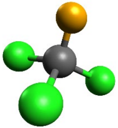


Instrumental fingerprint

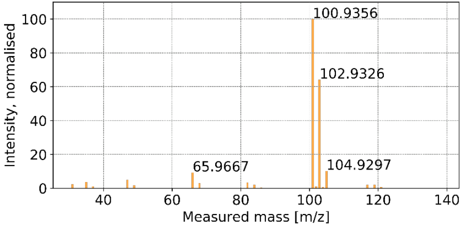
Target vs non-target screening

Empa
Materials Science and Technology

Target screening:



CFC-11
$CFCl_3$



Instrumental fingerprint

Non-target screening

Workflow: Knapsack algorithm

Weight: 100.936 ± 0.0002 g/mol
(U = 2 ppm)

Which **atoms** can be packed together to match the measured masses, ± u?

9 atoms: H, C, N, O, S, F, Cl, Br, I

The knapsack algorithm

Inputs:

- measured mass intervals
- IUPAC masses of atoms

Run the knapsack on each interval,
list all solutions.

H	1.0078250319
B	11.00930536
C	12.
N	14.0030740074
O	15.9949146223
F	18.99840316
P	30.973762
S	31.97207073
Cl	34.96885271
Br	78.9183376
I	126.9044719

mass min	mass max
34.96751071	34.97006625
35.97413780	35.97778836
36.96406557	36.96750599
46.96648952	46.97028744
48.96255817	48.96840119
59.96022019	59.97131577
81.93308641	81.93953315
82.93831759	82.95111397
83.93024931	83.93724265
84.92634272	84.97112964
85.92419323	85.93924313
97.92364853	97.93896563
99.90729357	99.94127719
116.90200574	116.90847942
117.89455759	117.92205637
118.89897942	118.90567754
119.89648859	119.91785117
120.89523104	120.90302932
122.88755350	122.90537266

mass min	mass max	knapsack
34.96751071	34.97006625	Cl
35.97413780	35.97778836	HCl
36.96406557	36.96750599	–
46.96648952	46.97028744	CCl
48.96255817	48.96840119	–
59.96022019	59.97131577	COS
81.93308641	81.93953315	CCl_2
82.93831759	82.95111397	S_2F, $HCCl_2$
83.93024931	83.93724265	–
84.92634272	84.97112964	H_2S_2F, H_2OSCl, $HNCl_2$, CF_2Cl
85.92419323	85.93924313	OCl_2
97.92364853	97.93896563	H_2S_3, $COCl_2$
99.90729357	99.94127719	HS_2Cl, HO_2SCl, $NOCl_2$
116.90200574	116.90847942	CCl_3
117.89455759	117.92205637	S_2FCl, $OSCl_2$, $HCCl_3$
118.89897942	118.90567754	–
119.89648859	119.91785117	C_2S_3
120.89523104	120.90302932	–
122.88755350	122.90537266	CSBr

mass min	mass max	knapsack	answer
34.96751071	34.97006625	Cl	Cl
35.97413780	35.97778836	HCl	HCl
36.96406557	36.96750599	–	$[^{37}Cl]$
46.96648952	46.97028744	CCl	CCl
48.96255817	48.96840119	–	$C[^{37}Cl]$
59.96022019	59.97131577	COS	–
81.93308641	81.93953315	CCl_2	CCl_2
82.93831759	82.95111397	S_2F, $HCCl_2$	$HCCl_2$
83.93024931	83.93724265	–	$CCl[^{37}Cl]$
84.92634272	84.97112964	H_2S_2F, H_2OSCl, $HNCl_2$, CF_2Cl	$HCCl[^{37}Cl]$
85.92419323	85.93924313	OCl_2	$C[^{37}Cl]_2$
97.92364853	97.93896563	H_2S_3, $COCl_2$	$COCl_2$
99.90729357	99.94127719	HS_2Cl, HO_2SCl, $NOCl_2$	$COCl[^{37}Cl]$
116.90200574	116.90847942	CCl_3	CCl_3
117.89455759	117.92205637	S_2FCl, $OSCl_2$, $HCCl_3$	$[^{13}C]Cl_3$
118.89897942	118.90567754	–	$CCl_2[^{37}Cl]$
119.89648859	119.91785117	C_2S_3	$[13C]Cl_2[^{37}Cl]$
120.89523104	120.90302932	–	$CCl[^{37}Cl]_2$
122.88755350	122.90537266	CSBr	$C[^{37}Cl]_3$

The knapsack algorithm in cryptography

Richard Schroeppel and Adi Shamir.
A $T = O(2^{n/2})$, $S = O(2^{n/4})$ Algorithm for Certain NP-Complete Problems.
SIAM Journal on Computing, 10(3):456–464, 1981.

A. M. Odlyzko.
The rise and fall of knapsack cryptosystems.
In Carl Pomerance, editor, *Cryptology and Computational Number Theory*, volume 42 of *Proceedings of Symposia in Applied Mathematics, August 6–7, 1989, Boulder, Colorado*, pages 75–88, Providence, Rhode Island, 1991. AMS.
http://www.dtc.umn.edu/~odlyzko/doc/arch/knapsack.survey.pdf.

The knapsack algorithm in computer science

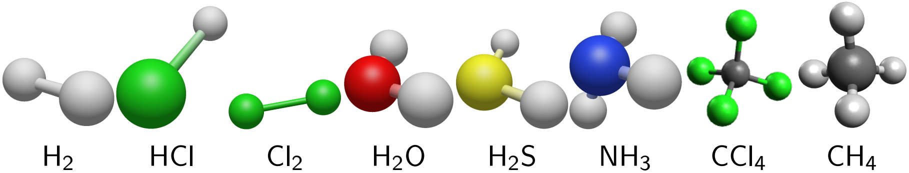
Divide-and-conquer method, target $[m_{\min}, m_{\max}]$:
(Thanks to Paul Zimmermann for pointing out the method)

1. Divide the element masses in two *balanced* sets A and B
2. In parallel:
   - List all possible sums in $[0, m_{\max}]$ of masses from set A, sort in increasing order $\rightarrow S_A$
   - List all possible sums in $[0, m_{\max}]$ of masses from set B, sort in increasing order $\rightarrow S_B$
3. Read onwards the masses from S_A and downwards the masses from S_B,
   find matches $m_{A,i} + m_{B,j} \in [m_{\min}, m_{\max}]$
   Linear complexity in $\text{length}(S_A) + \text{length}(S_B)$

Our knapsack algorithm in chemistry

Set A: masses of multi-valent atoms C (4), N (3), O (2), S (6)

Set B: masses of mono-valent atoms H, F, Cl, Br, I

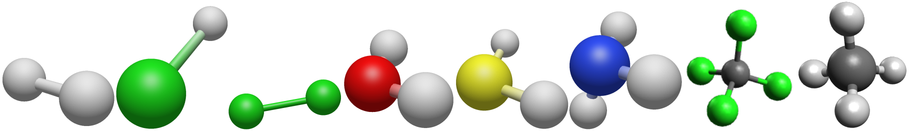


Our knapsack algorithm in chemistry

Set A: masses of multi-valent atoms C (4), N (3), O (2), S (6)

Set B: masses of mono-valent atoms H, F, Cl, Br, I

1. List all sums in $[0, m_{\mathrm{max}}]$ of masses from set A, sort in increasing order $\rightarrow S_A$
2. Compute maximum valence of each solution of $S_A \rightarrow \mathrm{valence}_{\mathrm{max}}$

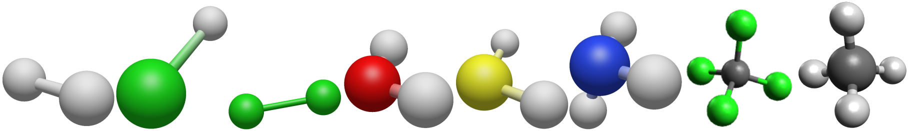


H_2 HCl Cl_2 H_2O H_2S NH_3 CCl_4 CH_4

Our knapsack algorithm in chemistry

Set A: masses of multi-valent atoms C (4), N (3), O (2), S (6)
Set B: masses of mono-valent atoms H, F, Cl, Br, I

1. List all sums in $[0, m_{\max}]$ of masses from set A, sort in increasing order $\rightarrow S_A$
2. Compute maximum valence of each solution of $S_A \rightarrow \text{valence}_{\max}$
3. List all sums in $[0, m_{\max}]$ of at most $\text{valence}_{\max}$ masses from set B, sort in increasing order $\rightarrow S_B$
4. Read onwards the masses from S_A and downwards the masses from S_B, find matches $m_{A,i} + m_{B,j} \in [m_{\min}, m_{\max}]$ and check DBE ≥ 0
   Linear complexity in $\text{length}(S_A) + \text{length}(S_B)$



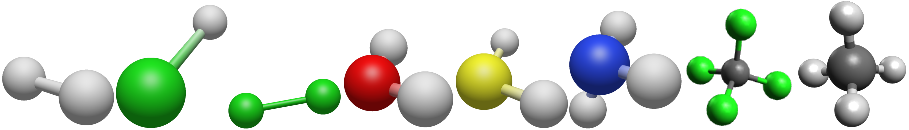
H_2 HCl Cl_2 H_2O H_2S NH_3 CCl_4 CH_4

Our knapsack algorithm in chemistry

Observe that:

- atoms in A are heavy (lightest one C of 12.0 m/z),
  $\text{valence}_{\text{max}} \leq 4 \cdot m_{\text{max}}/12.0 = m_{\text{max}}/3$
- lightest atom H $\in B$ of mass 1.0078250319, but #H bounded by $\text{valence}_{\text{max}}$

$\rightarrow$ in average, reduce the length of S_B by a factor 2



H_2 HCl Cl_2 H_2O H_2S NH_3 CCl_4 CH_4

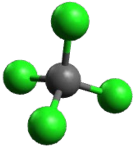
Other knapsack algorithm in analytical chemistry

SIRIUS software, AGPL `https://github.com/boecker-lab/sirius`
`https://bio.informatik.uni-jena.de`
Universität Jena, Germany, Bio-informatic group

- Sebastian Böcker and Zsuzsanna Liptak.
  A fast and simple algorithm for the money changing problem.
  Algorithmica, 48:413–432, 2007.

- Kai Dührkop, Marcus Ludwig, Marvin Meusel, and Sebastian Böcker.
  Faster mass decomposition.
  In Aaron Darling and Jens Stoye, editors, *Algorithms in Bioinformatics*, pages 45–58, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

Workflow: mimick fragmentation process



Molecular ion

CCl_4

e-

e-

e-

e-

e-

CCl_3^+

CCl^+

CCl_2^+

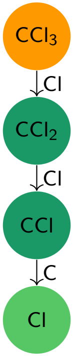
Cl^+

Measured fragments

More about graph algorithms: CCl_4

Singleton, Maximal, Node, Leaf.

Fragmentation graph of CCl_4

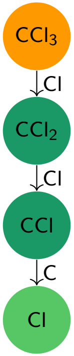


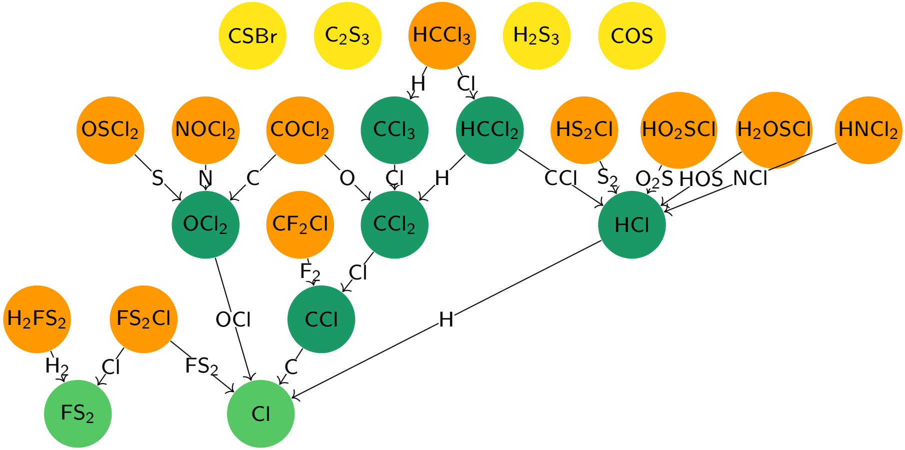
More about graph algorithms: CCl_4

Singleton, Maximal, Node, Leaf.

Pseudo-fragmentation graph of knapsack fragments w.r.t. partial order

Fragmentation graph of CCl_4





More about graph algorithms: CCl_4

Ouput of knapsack: a list of candidate fragment formulas

Define a partial order on the fragment formulas

Cl $\leq$ CCl $\leq$ CCl2 $\leq$ CCl3 $\leq$ HCCl3, but COS incomparable

Build a Directed Acyclic Graph according to the ordering

Setting edges has quadratic complexity in the number of vertices (nodes)

More about graph algorithms: CCl_4

Ouput of knapsack: a list of candidate fragment formulas

Define a partial order on the fragment formulas

Cl $\leq$ CCl $\leq$ CCl2 $\leq$ CCl3 $\leq$ HCCl3, but COS incomparable

Build a Directed Acyclic Graph according to the ordering

Setting edges has quadratic complexity in the number of vertices (nodes)

- Fragments in batches, per target interval mass, decreasing mass
- All fragments in one batch are incomparable (mass diff $\ll 1$)

More about graph algorithms: CCl_4

Ouput of knapsack: a list of candidate fragment formulas

Define a partial order on the fragment formulas

Cl $\leq$ CCl $\leq$ CCl2 $\leq$ CCl3 $\leq$ HCCl3, but COS incomparable

Build a Directed Acyclic Graph according to the ordering

Setting edges has quadratic complexity in the number of vertices (nodes)

- Fragments in batches, per target interval mass, decreasing mass
- All fragments in one batch are incomparable (mass diff $\ll 1$)

Much faster complexity, starting with the heaviest target mass

1. First batch: roots (maximal elements)

More about graph algorithms: CCl_4

Ouput of knapsack: a list of candidate fragment formulas

Define a partial order on the fragment formulas

Cl $\leq$ CCl $\leq$ CCl2 $\leq$ CCl3 $\leq$ HCCl3, but COS incomparable

Build a Directed Acyclic Graph according to the ordering

Setting edges has quadratic complexity in the number of vertices (nodes)

- Fragments in batches, per target interval mass, decreasing mass
- All fragments in one batch are incomparable (mass diff $\ll 1$)

Much faster complexity, starting with the heaviest target mass

1. First batch: roots (maximal elements)
2. Process one batch of fragments as a whole
   - 2.1 Compare each fragment to the roots
   - 2.2 If a subfragment, recursively visit the children until it is a leaf, set a new edge from the parent node
   - 2.3 if incomparable to any root, keep it aside

More about graph algorithms: CCl_4

Ouput of knapsack: a list of candidate fragment formulas

Define a partial order on the fragment formulas

$Cl \leq CCl \leq CCl2 \leq CCl3 \leq HCCl3$, but COS incomparable

Build a Directed Acyclic Graph according to the ordering

Setting edges has quadratic complexity in the number of vertices (nodes)

- Fragments in batches, per target interval mass, decreasing mass
- All fragments in one batch are incomparable (mass diff $\ll 1$)

Much faster complexity, starting with the heaviest target mass

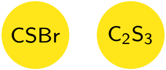
1. First batch: roots (maximal elements)
2. Process one batch of fragments as a whole
   - 2.1 Compare each fragment to the roots
   - 2.2 If a subfragment, recursively visit the children until it is a leaf, set a new edge from the parent node
   - 2.3 if incomparable to any root, keep it aside
3. update the list of roots with the incomparable fragments of the batch

More about graph algorithms: CCl_4

Singleton, Maximal, Node, Leaf.

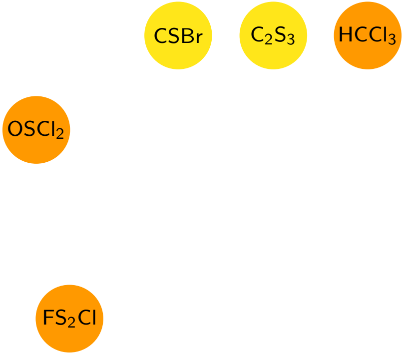
More about graph algorithms: CCl_4

Singleton, Maximal, Node, Leaf.



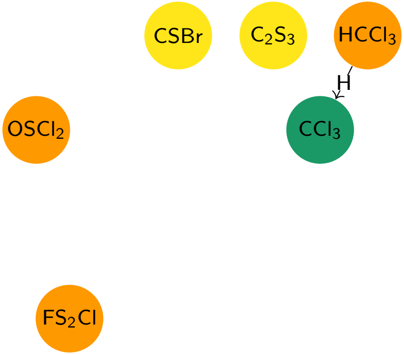
More about graph algorithms: CCl_4

Singleton, Maximal, Node, Leaf.



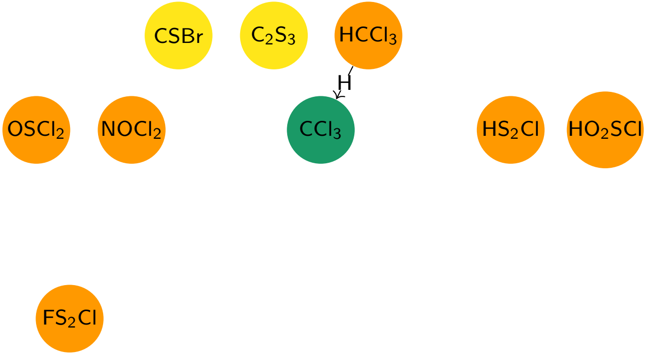
More about graph algorithms: CCl_4

Singleton, Maximal, Node, Leaf.

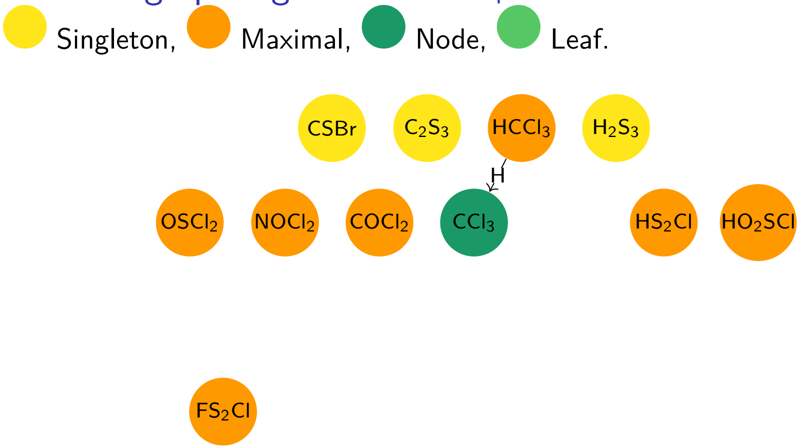


More about graph algorithms: CCl_4

Singleton, Maximal, Node, Leaf.

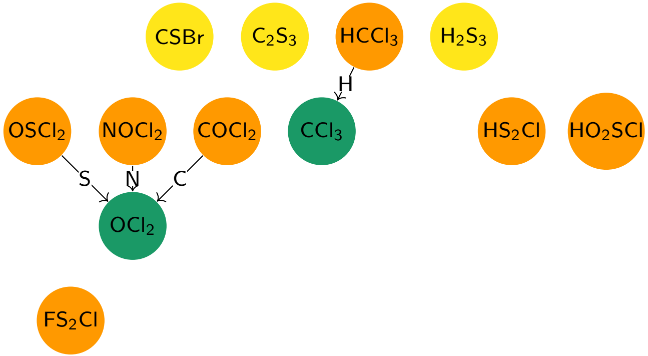


More about graph algorithms: CCl_4



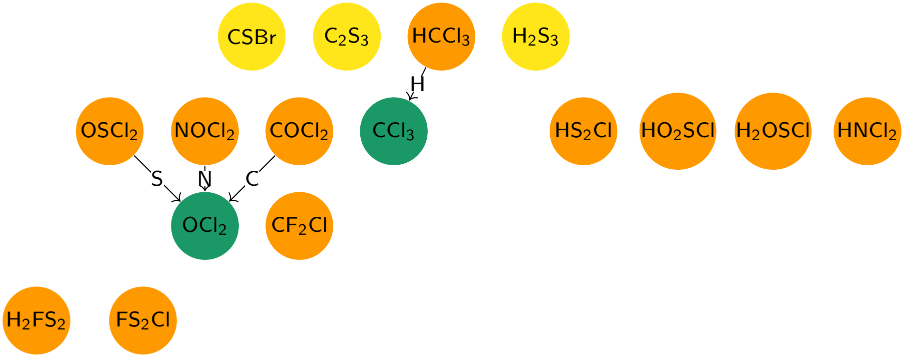
More about graph algorithms: CCl_4

Singleton, Maximal, Node, Leaf.



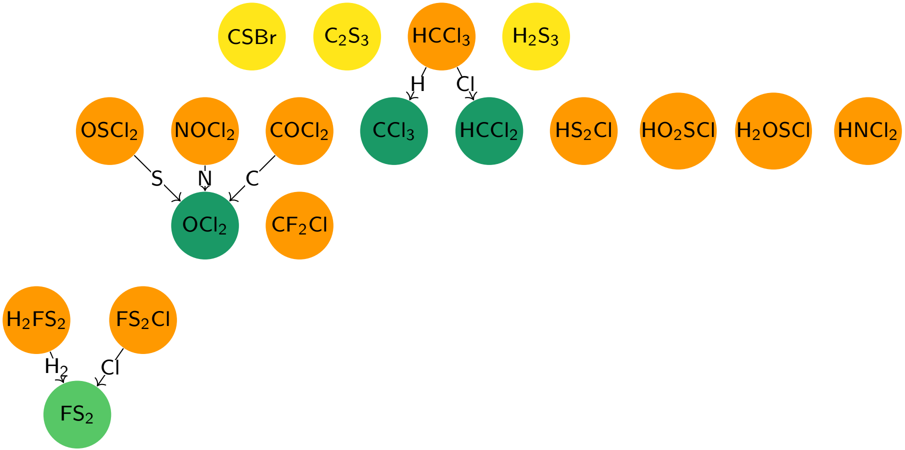
More about graph algorithms: CCl_4

Singleton, Maximal, Node, Leaf.



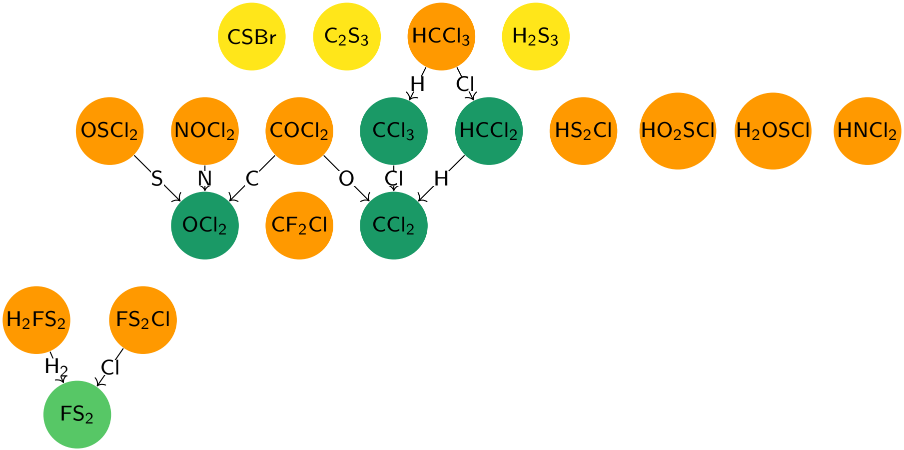
More about graph algorithms: CCl_4

Singleton, Maximal, Node, Leaf.



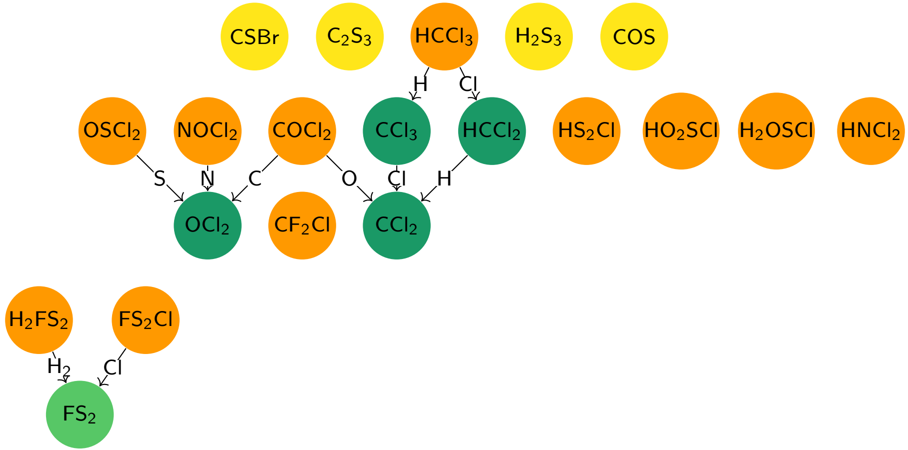
More about graph algorithms: CCl_4

Singleton, Maximal, Node, Leaf.



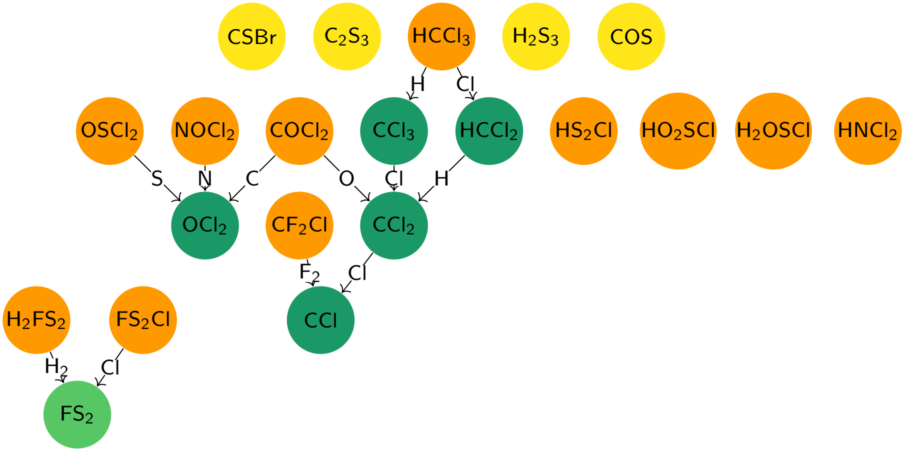
More about graph algorithms: CCl_4

Singleton, Maximal, Node, Leaf.



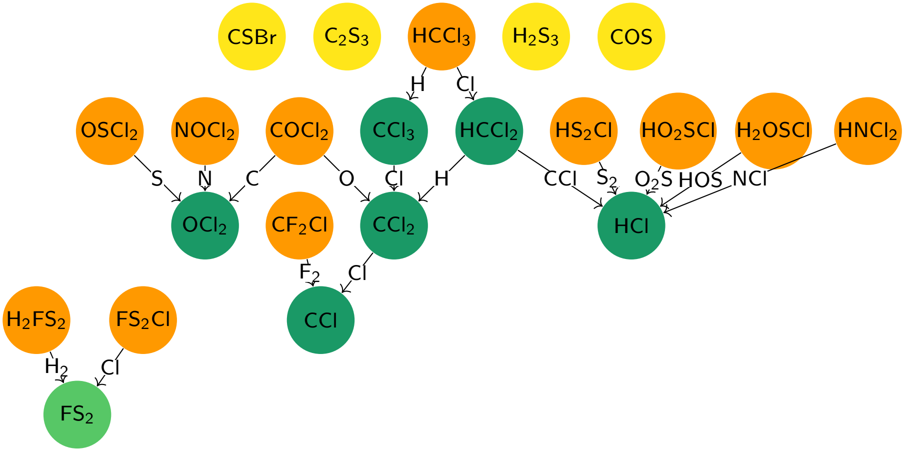
More about graph algorithms: CCl_4

Singleton, Maximal, Node, Leaf.



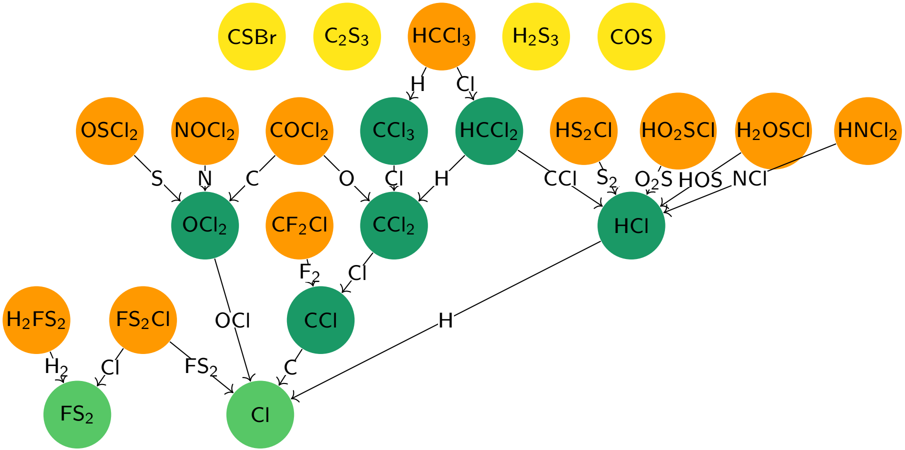
More about graph algorithms: CCl_4

Singleton, Maximal, Node, Leaf.



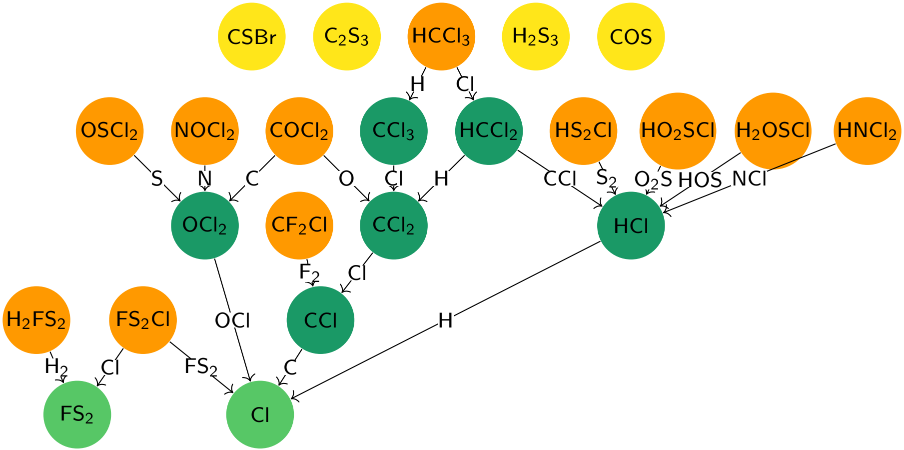
More about graph algorithms: CCl_4

Singleton, Maximal, Node, Leaf.



More about graph algorithms: CCl_4

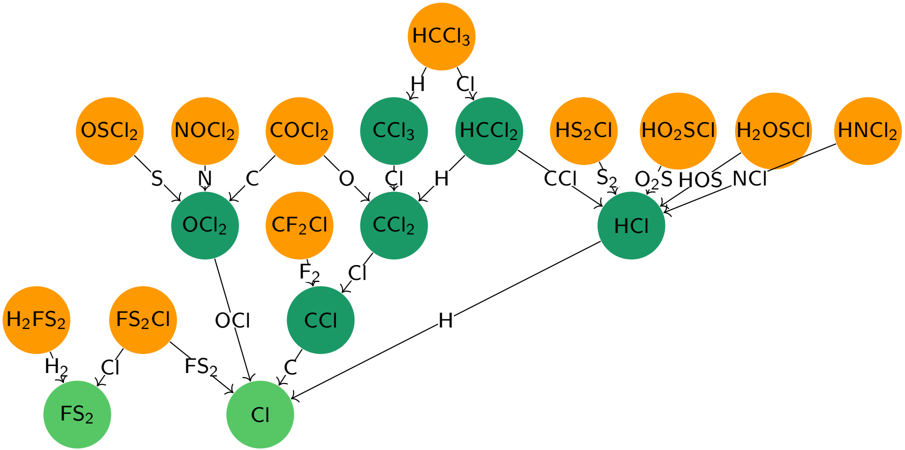
Singleton, Maximal, Node, Leaf.



Remove singletons

More about graph algorithms: CCl_4

Singleton, Maximal, Node, Leaf.

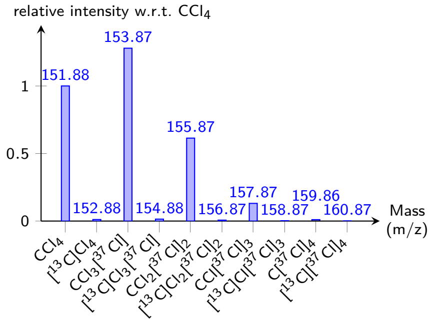


Remove singletons

Optimise isotopic profiles

Aim: eliminate all unlikely candidate formlas.

1. Compute the intensity profile of isotopocules →
2. Define a likelihood estimator
3. Fit the theoretic intensities to the measured signal (next slide)
4. Update pseudo-frag graph:
   Remove candidate formulas below LOD (limit of detection)
   Remove new singletons



Workflow: optimise isotopic 'profiles'

Empa
Materials Science and Technology

knapsack

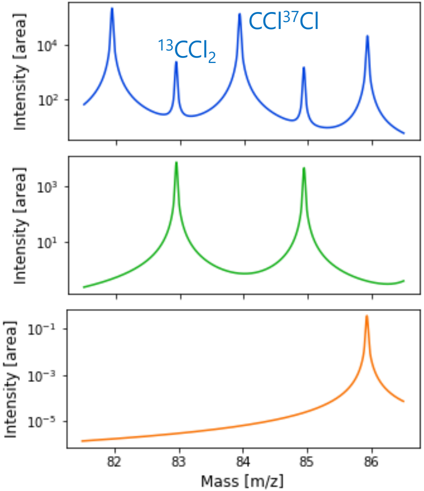
Isotopologue profiles

Python lmfit

CCl_2^+

$HCCl_2^+$

OCl_2^+



$\cdot k_1$

$\cdot k_2$

$\cdot k_3$

Workflow: optimise isotopic 'profiles'

Empa
Materials Science and Technology

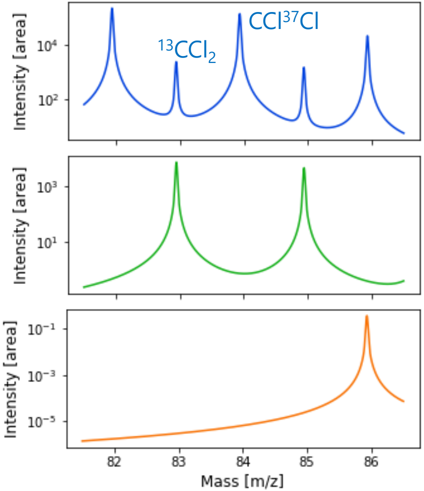
knapsack

CCl_2^+

$HCCl_2^+$

OCl_2^+

Isotopologue profiles



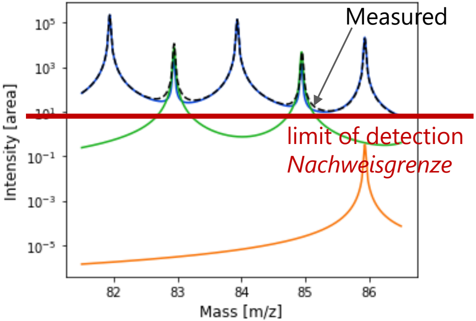
Python
lmfit

$\cdot\ k_1$

$\cdot\ k_2$

$\cdot\ k_3$

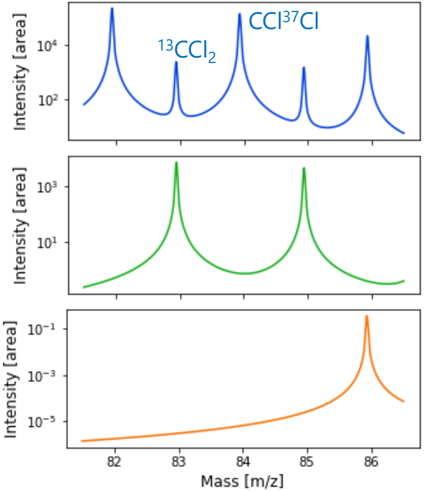
Optimised
contributions

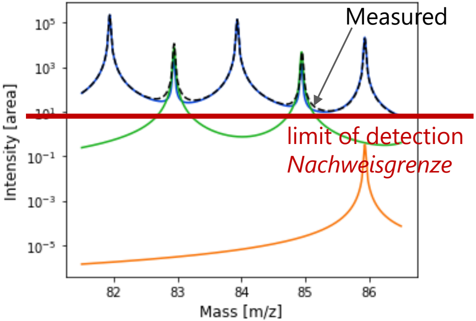


Workflow: optimise isotopic 'profiles'

Empa
Materials Science and Technology

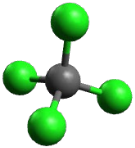
knapsack	Isotopologue profiles	Python lmfit	Optimised contributions
CCl_2^+		$\cdot k_1$	
$HCCl_2^+$		$\cdot k_2$	
~~OCl_2^+~~		$\cdot k_3$	

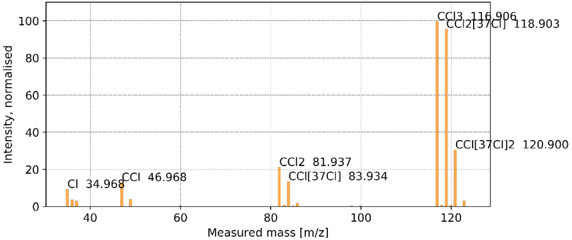




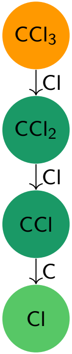
Results – example for CCl_4

- 19 measured masses
- 23 knapsack solutions using C, H, N, O, S, F, Cl, Br, I
- 3 singletons removed
- 2 solutions < LOD
- 98% correctly assigned signal
- Runtime: 4 s on a laptop

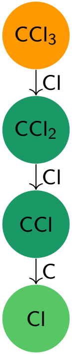




Result: final graph for CCl_4

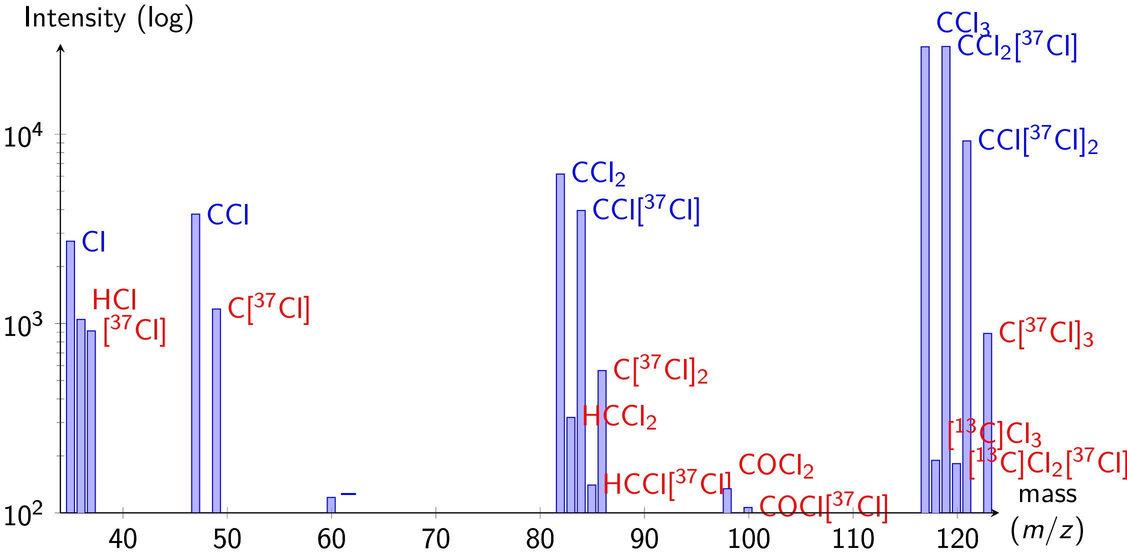


Result: final graph for CCl_4



Molecular Ion CCl_4 not in the graph!!!

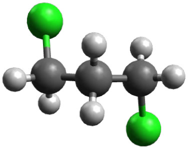
In 40% of measured samples, the molecular ion is not measured (due to the Electron Ionisation technique)


Intensity (log)
10^4
10^3
10^2
Cl
HCl
[^{37}Cl]
CCl
C[^{37}Cl]
CCl$_2$
CCl[^{37}Cl]
C[^{37}Cl]$_2$
HCCl$_2$
HCCl[^{37}Cl]
COCl$_2$
COCl[^{37}Cl]
CCl$_3$
CCl$_2$[^{37}Cl]
CCl[^{37}Cl]$_2$
C[^{37}Cl]$_3$
[^{13}C]Cl$_3$
[^{13}C]Cl$_2$[^{37}Cl]
40
50
60
70
80
90
100
110
120
mass
(m/z)

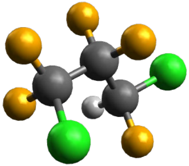
New in Dübendorf air...

Empa
Materials Science and Technology

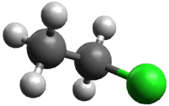
>75 newly found substances:



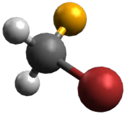
1,3-dichloropropane



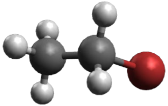
HCFC-225cb



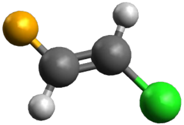
Chloroethane



CH_2FBr (~LOD)



Bromoethane (~LOD)



1-chloro-2-fluoroethene

Validation of the results

1. Buy the suspected substances on catalog
2. Measure with the same machine and same settings
3. Check Retention Time (RT), and mass spectra: peaks at same masses, same proprtion of peak intensities

24 subtances validated so far.

Issues:

- unavailable substances (exist, but cannot buy them on catalog)
- too toxic for shipping
- too costly ($1000/5g) (our threshold cost: $750/5g)
- all are banned substances
- special authorization from Ministry of Environment for customs department

Future Work

Solve the knapsack problem with LLL?

- François Morain says it will work
- Paul Zimmermann says it will work
- Léo Ducas says it will work

It's only a question of time and human resources...

Conclusion

Fruitful collaboration between computer-scientist and environmental science researchers and engineers

- learned about chemistry
- learned about how to teach CS to senior researchers and engineers
- co-authors learned about algorithms, Python programming, software architecture, and development tools (git)

Python source code upon release with LGPL license
Paper under review process at a computational chemistry journal
Preprint `https://hal.inria.fr/hal-03176025`

EMPA will be looking for hiring a computer scientist.