# Maximum Common Subgraph Finding and Dynamic Programming for Mechanistic Explanation in Mass Spectrometry

Akbar Davoodi* and Daniel Merkle*†

Joint work with Christoph Flamm, Marc Hellmuth, Johannes B. S. Petersen, Peter F. Stadler

*Department of Mathematics and Computer Science(IMADA)
University of Southern Denmark(SDU)

†Technical Faculty, Bielefeld University

$39^{th}$ TBI Winterseminar, Bled, Slovenia
Feb 11-16, 2024

# Maximum common substructures

Two approaches:

- Graph Alignments
- Graph Products

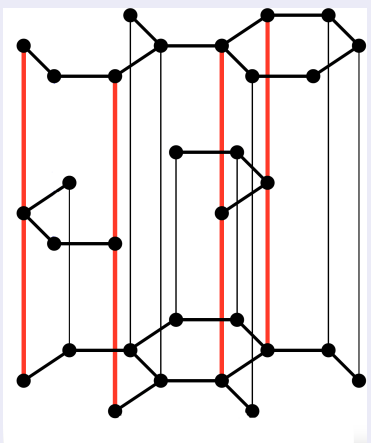
## Definition (Alignment)

An alignment of spaces $(X_\alpha, \mathscr{S}_\alpha)$, $\alpha \in S$, $|S| \geq 1$ is a space $(X, \mathscr{S})$ such that

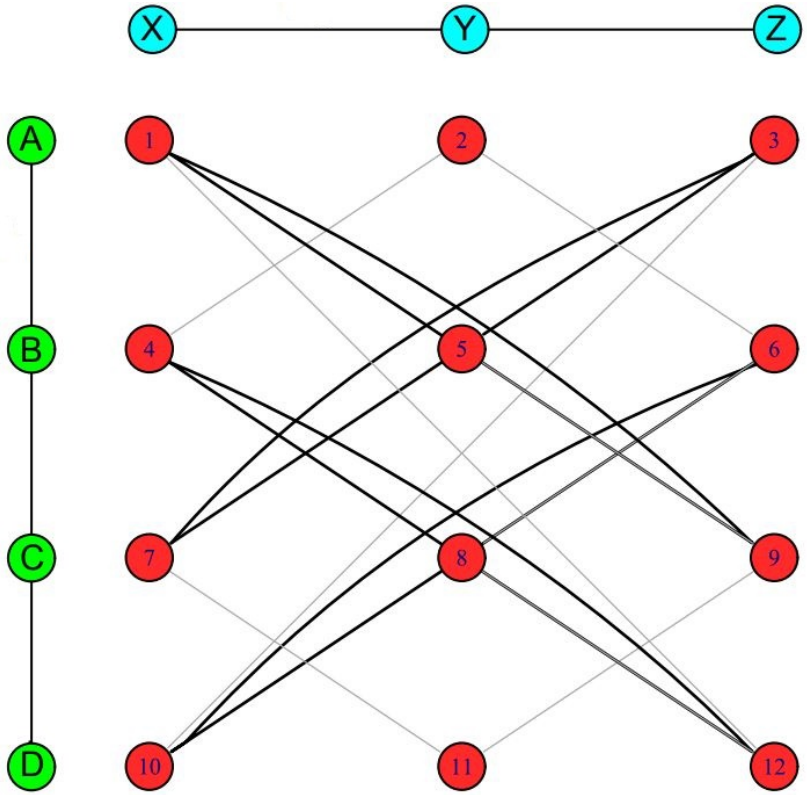
(i) there is a monomorphism $\mu_\alpha : X_\alpha \to X$ for every $\alpha \in S$;

(ii) for every $x \in X$, $\mu_\alpha^{-1}(x) \neq \emptyset$ for at least one $\alpha \in S$;

(iii) the restriction of $(X, \mathscr{S})[\mu_\alpha(X_\alpha)]$ is isomorphic to $(X_\alpha, \mathscr{S}_\alpha)$



# Modular product of two graphs



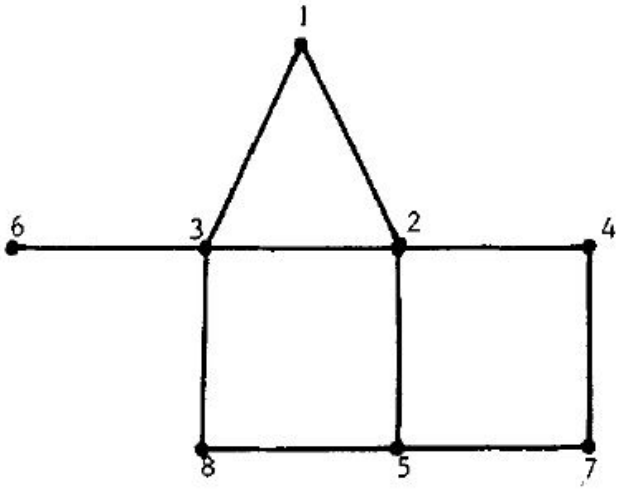

- Cliques in the modular product graph correspond to isomorphisms of induced subgraphs of $G$ and $G'$.
- The maximum common induced subgraph of two graphs corresponds to the maximum clique in their modular product.

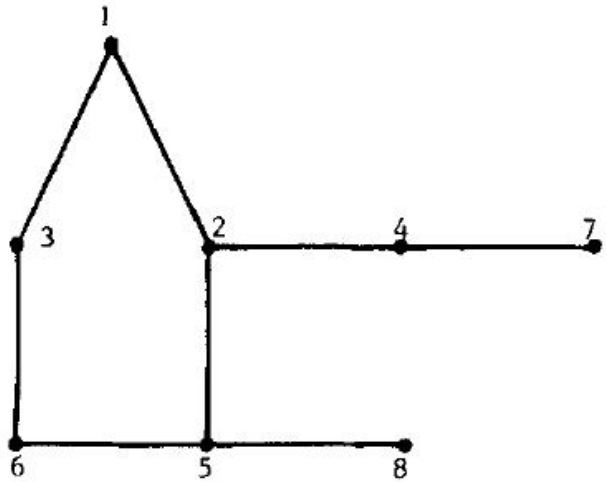
# What precisely do we require from a common substructure?

## Questions

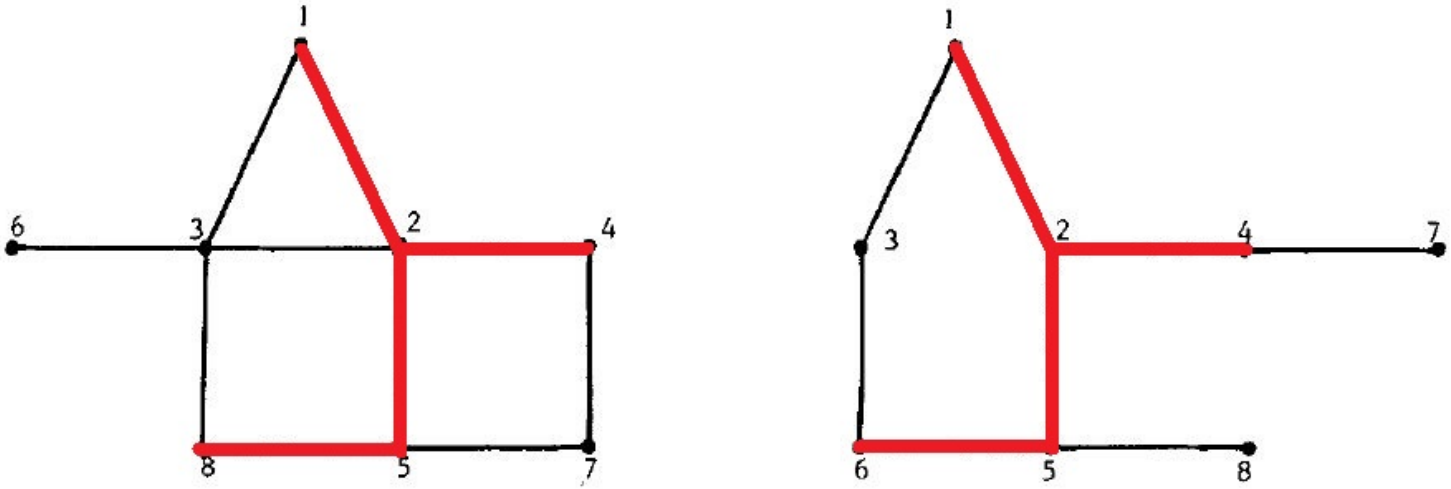
- Which properties need to be preserved for the common substructure?
  - Induced subgraph
  - Connectivity
  - ...
- How can we generalize each of the approaches for multiple graphs?
- Do we require an exact answer, or would an approximate one suffice?

# Subgraphs and vertex induced subgraphs



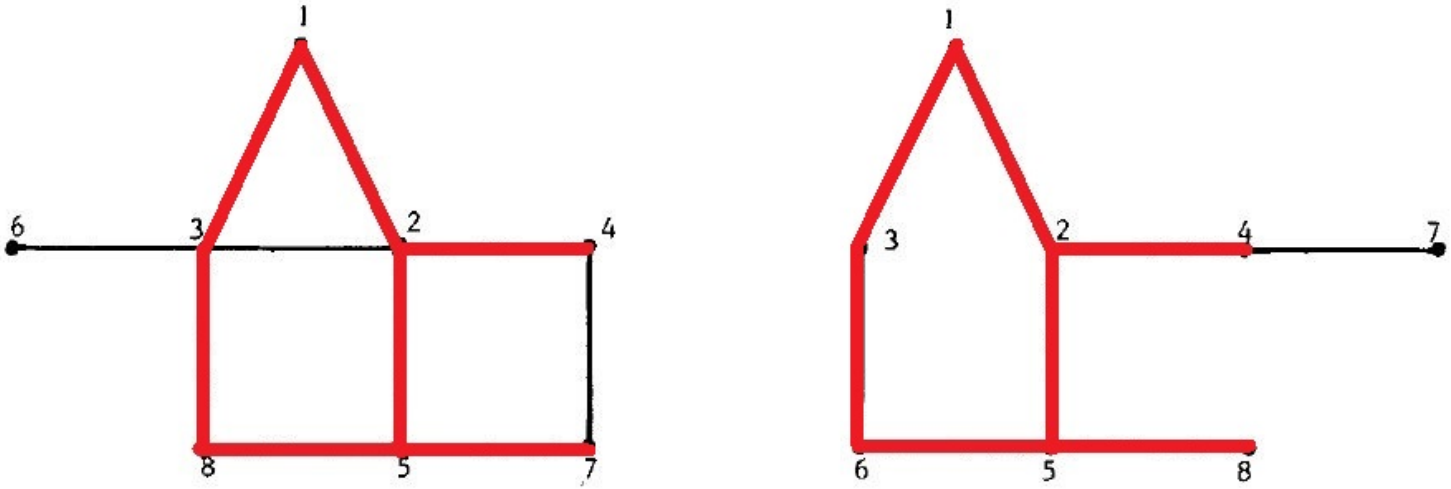



# Subgraphs and vertex induced subgraphs




5 vertices and 4 edges

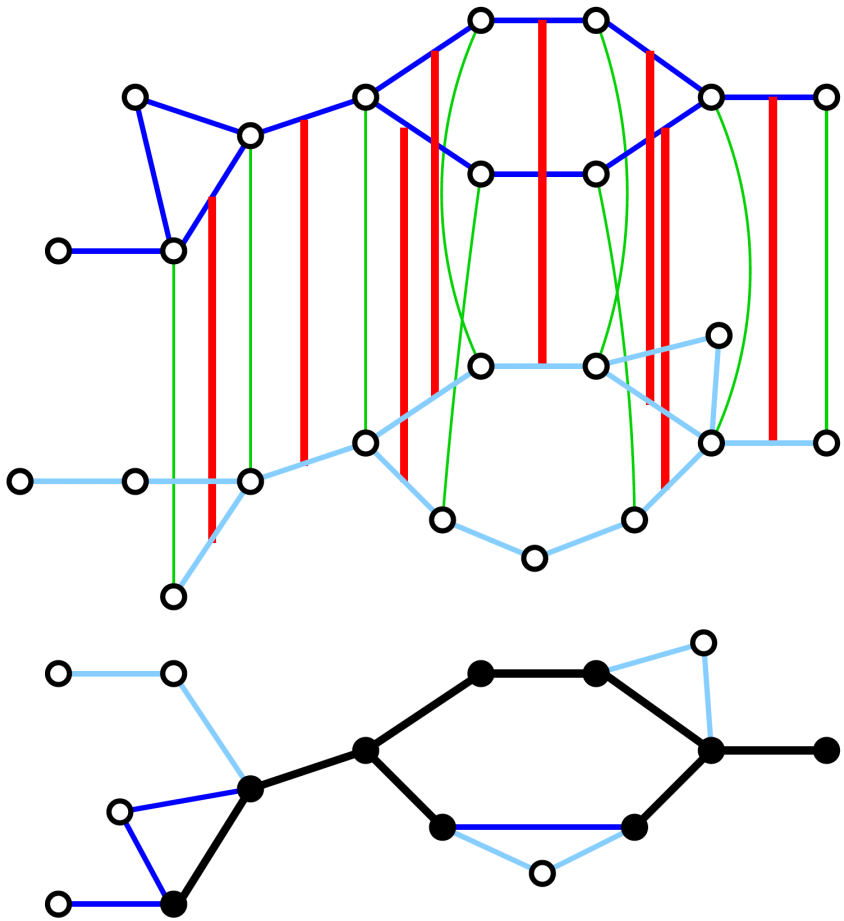
# Subgraphs and vertex induced subgraphs




7 vertices and 7 edges

# In graph alignments:

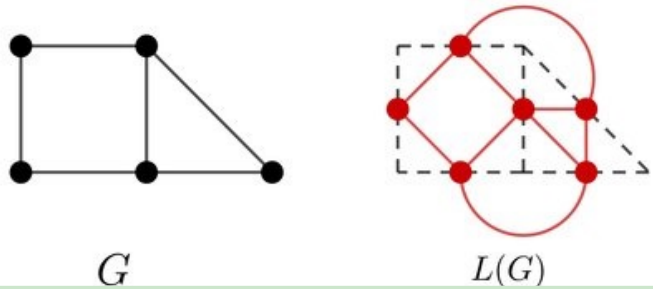
Solution: Edge-wise graph alignment:



# In graph products:

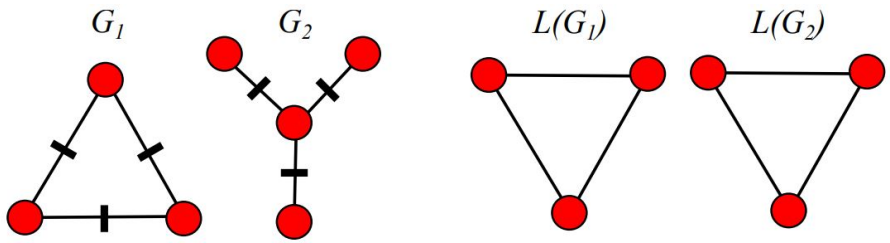
## Definition (Line graph)

Let $G = (V, E)$ be a simple graph. The line graph $L(G)$ is another simple graph. Each vertex of $L(G)$ represents an edge of $G$ and two vertices in $L(G)$ are adjacent iff the corresponding edges are adjacent in $G$.




## Whitney's Theorem (1932)

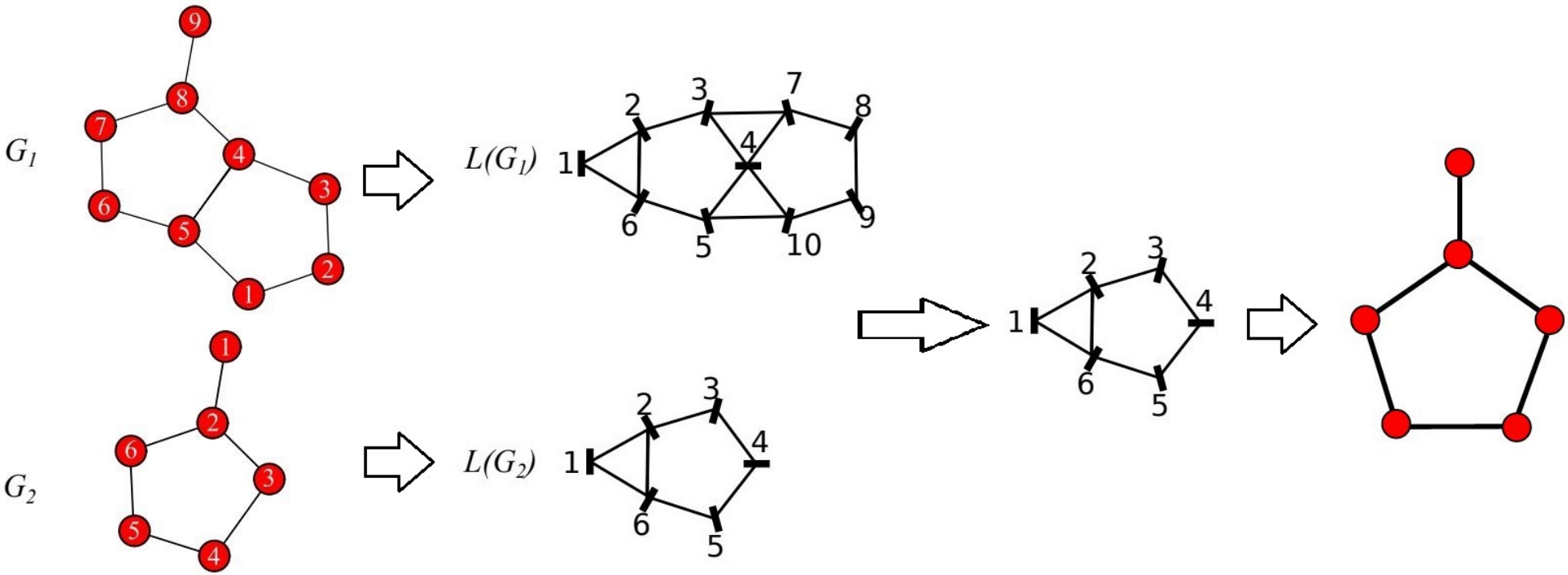
Every graph, except triangle or claw, is uniquely determined by its line graph.



# From MCS to MCES

$G$ and $G'$ $\overset{L}{\Longrightarrow}$ $L(G)$ and $L(G')$ $\overset{\text{vertex induced}}{\underset{\text{algorithm}}{\Longrightarrow}}$ MCS$(L(G), L(G'))$ $\overset{L^{-1}}{\Longrightarrow}$ MCES$(G, G')$

**Example:**

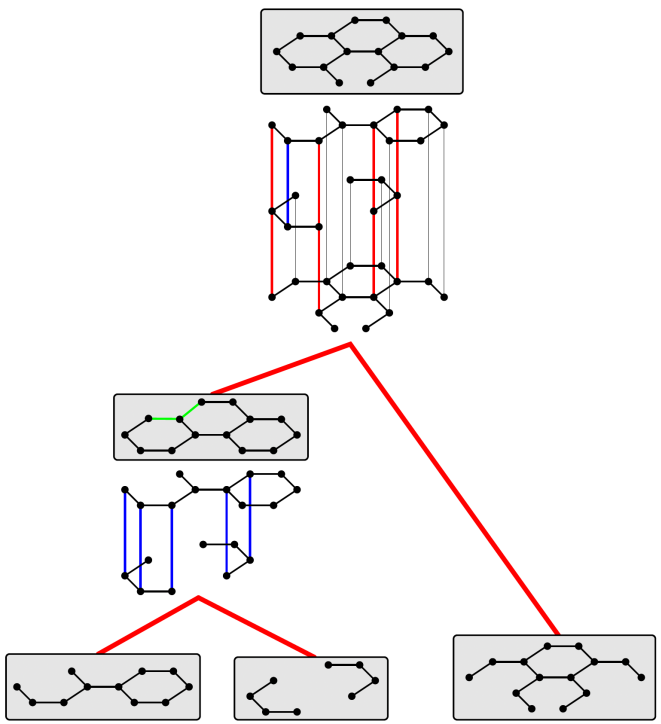


# How to find common subgraph of $\{H_1, H_2, \ldots, H_t\}$ ?

In graph product:

$$\overbrace{\underbrace{H_1 \times H_2}_{c_2} \times H_3}^{c_3} \times \cdots \times H_t$$

In graph alignment:



# Summary

- Both approaches can handle any structural property we wish to preserve for the common substructure.
- In the alignment approach, you cannot guarantee the optimality of the answer, but it is faster.
- In the product approach, you ensure that the answer is optimal, but it is slower in terms of time.
- Depending on the application, one may decide which of them to select.
- In the alignment approach, one has to deal with technical issues like ambiguous sets, whereas this is not the case in the product approach.

# Maximum Common Subgraph Finding and Dynamic Programming for Mechanistic Explanation in Mass Spectrometry

Akbar Davoodi[1], Daniel Merkle[2,1]

[1]University of Southern Denmark
[2]University of Bielefeld

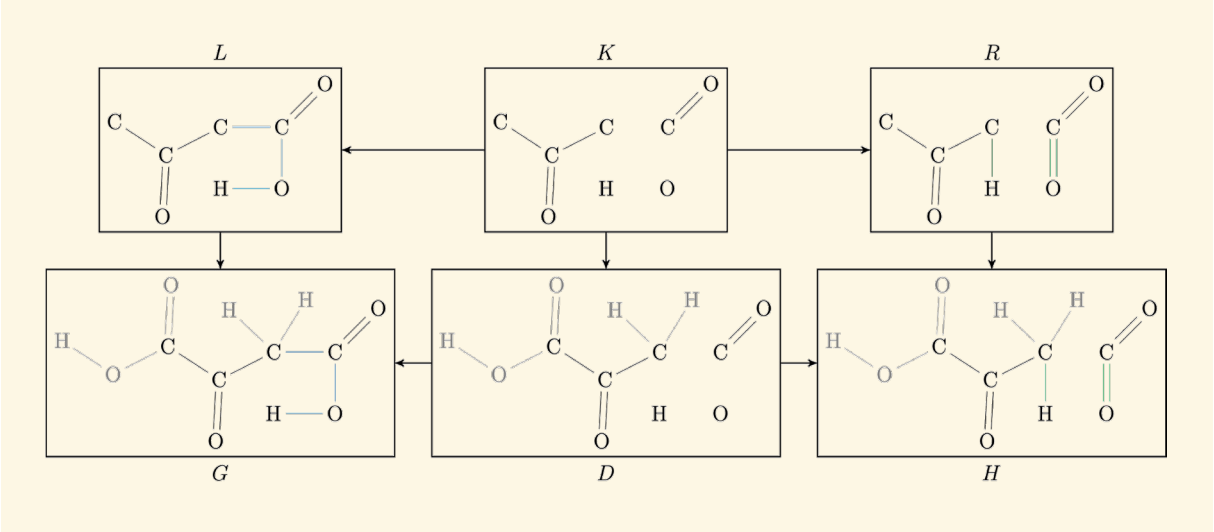
(Joint work with Christoph Flamm, Marc Hellmuth,
Johannes Borg Sandberg Petersen, Peter F. Stadler)

SDU

UNIVERSITÄT BIELEFELD

# Methodology: **Graph Transformations using the Double Pushout Approach**

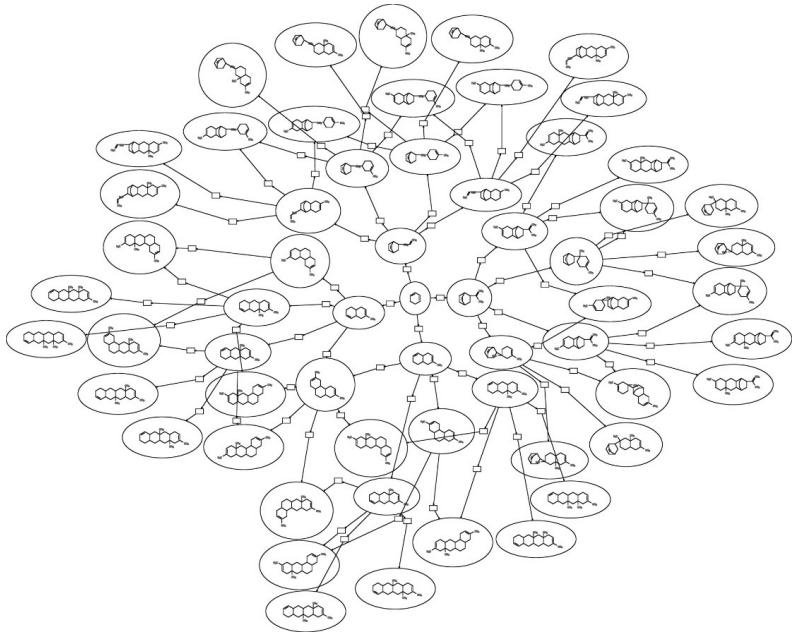
## Chemical reactions as mathematical rigorous graph transformations




**Atoms have identity**, allowing for:

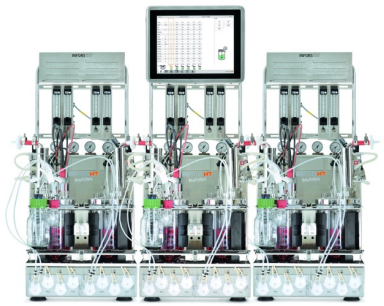
- direct wetlab validation
- atom tracing and isotope labelling experiment design
- automated coarse graining
- interfacing to (semi-empirical) quantum chemistry methods

## Generative chemistry



- reaction network as hypergraph
- inference of motifs as integer hyperflows (e.g., autocatalysis)
- causality analysis
- network completion

# Methodology: **SIHUMIx and Isotope Tracing**



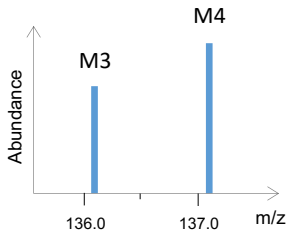
8 species

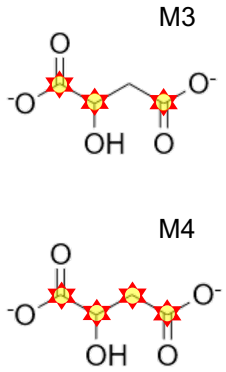
80-100 species

Sampling cells

Sampling medium

M/S-detection of isotopologues of metobolites (here: malate)




Continuous cultivation of an 8 species microbial community is established

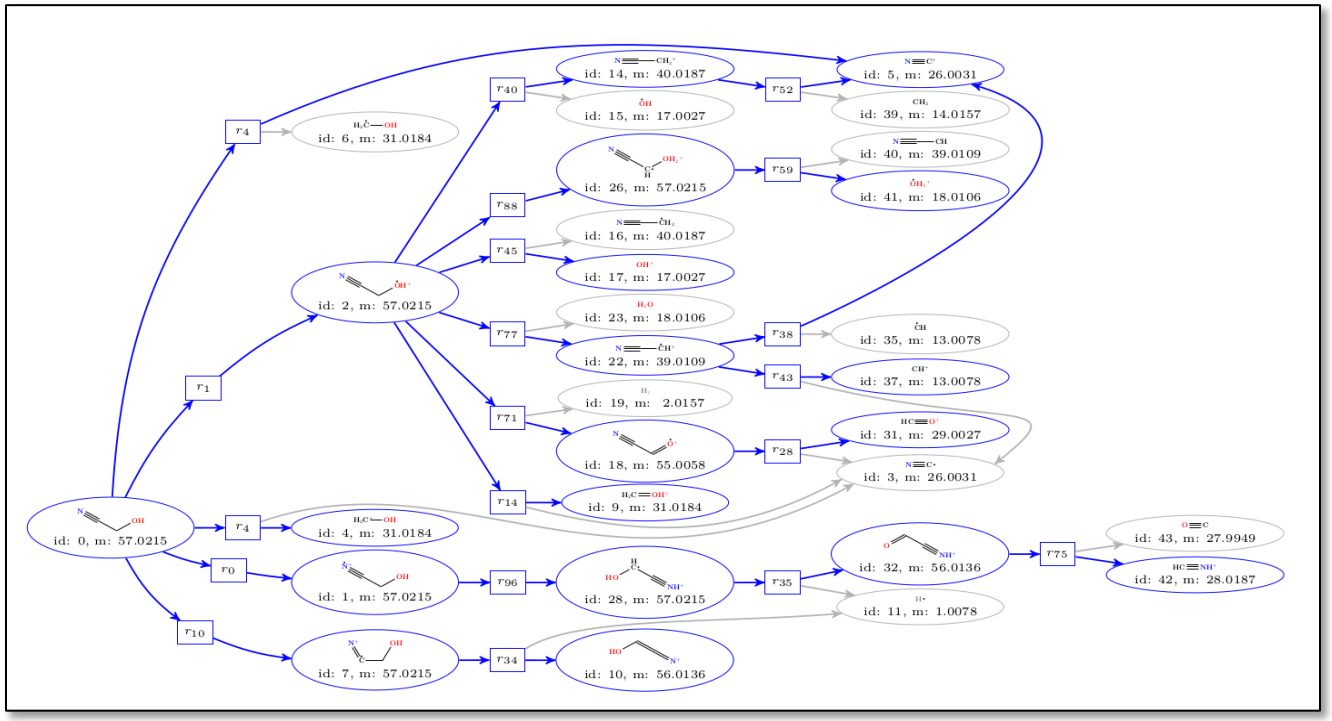
# MS using Graph Transformation

- Ionization
- Fragmentation

```
targetCompounds = [smiles("N#CCO")]

def hasCharge(g, gs, first):
   return sum(v.charge for v in g.vertices) != 0

strat = (
   ionizationRules
   >> filterSubset(hasCharge)
   >> repeat[4](
      fragmentationRules >> filterSubset(hasCharge)
   )
)
dg = dgRuleComp(inputGraphs, addSubset(targetCompounds) >> strat)
dg.calc()
dg.print()
```

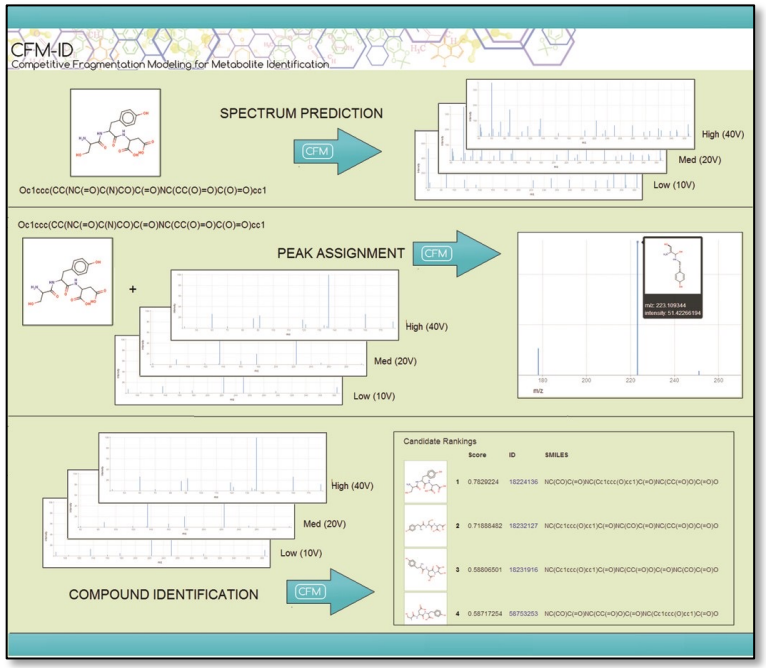



Andersen et al. 2018

# Black Boxes

- An **overapproximation of a fragmentation graph** for mechanistic explanations

  (e.g. CFM-ID, MØD, …)

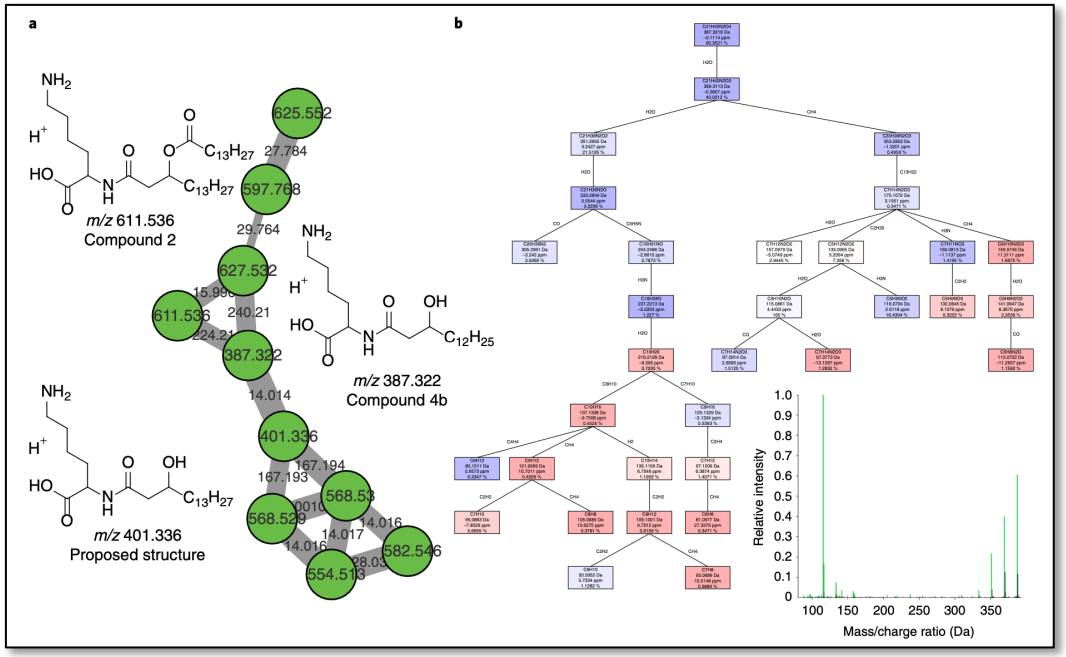



Wang et al., 2021

- creates huge fragmentation DAGs (ML)
- can be used for rules inference

- A (hopefully) **trustworthy fragmentation tree** (and more)

  (e.g. SIRIUS, QCxMS, … )

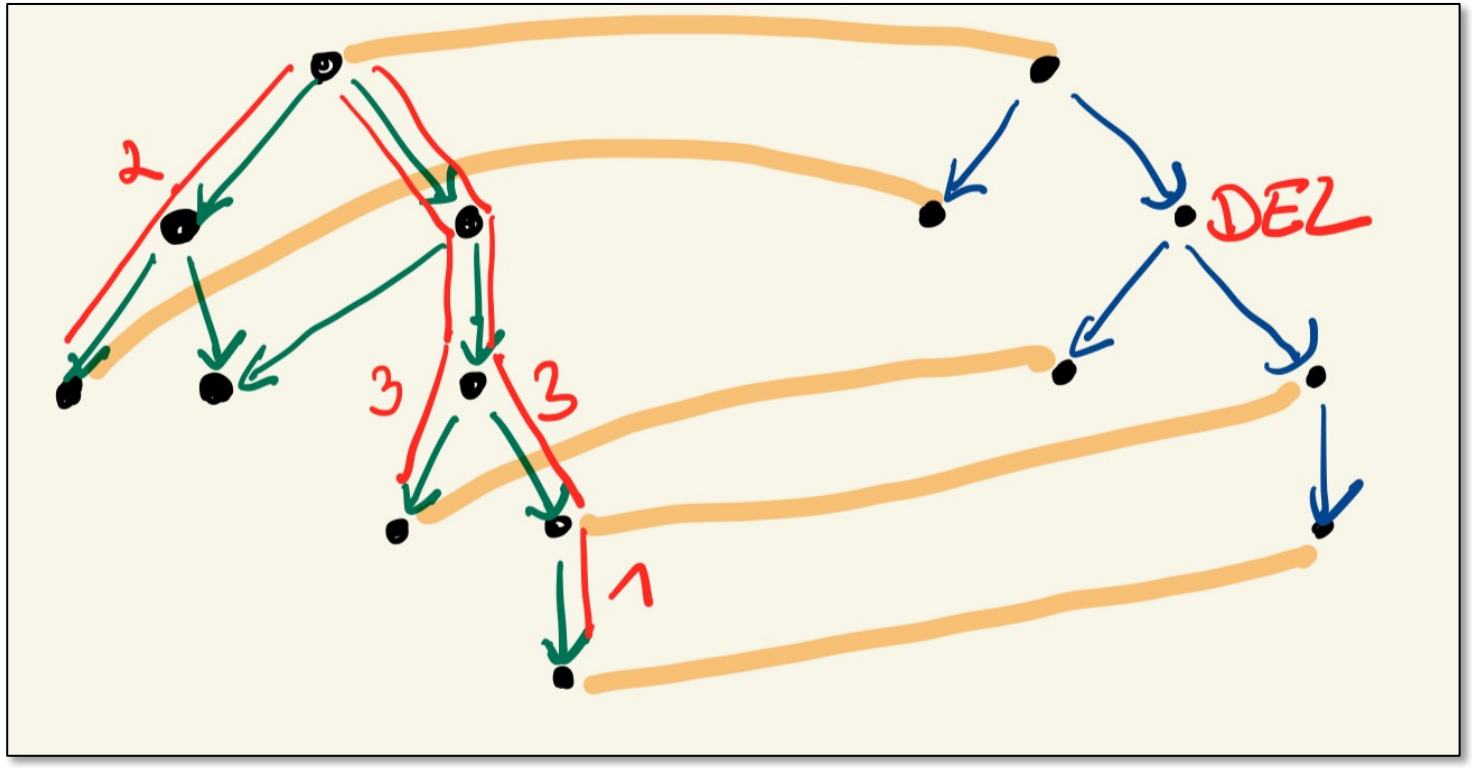



Dührkop et al., 2019

- no mechanistic explanation

# Dynamic Programming

MØD SIRIUS



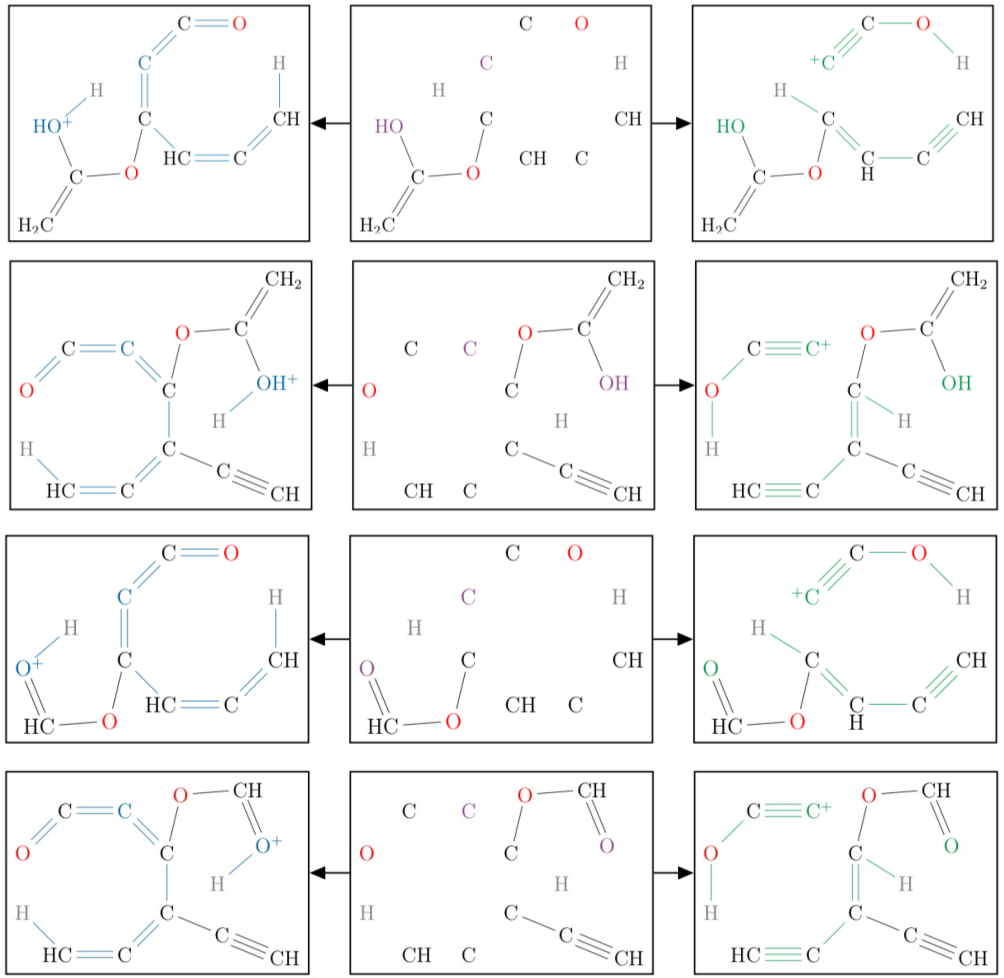

Map a tree into a DAG, under a certain cost measure

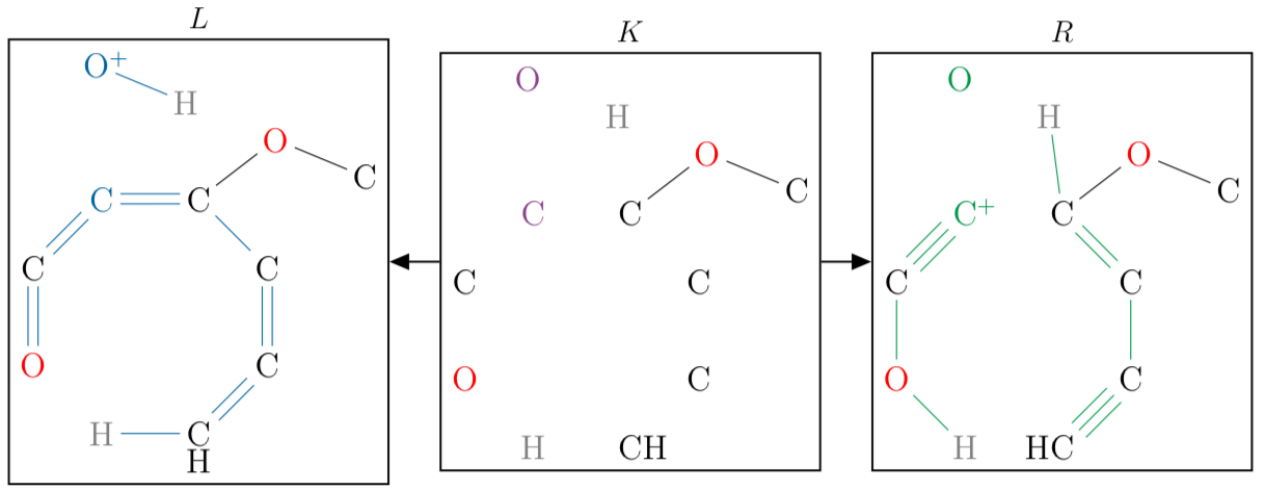
# Some numbers

| | | |
|---|---|---|
| Size of SIRIUS fragmentation trees : | approx. 1 – 20 vertices | |
| Size of graph transformation DAG (MØD derivation graph): | approx. 5000 – 100.000 vertices | (!) |
| Number of graph transformation rules: | approx. 10.000 | (!) |
| Succesfull application of graph transformation rules | approx. 1% - 2% | |

[ work in progress ]

# MCS





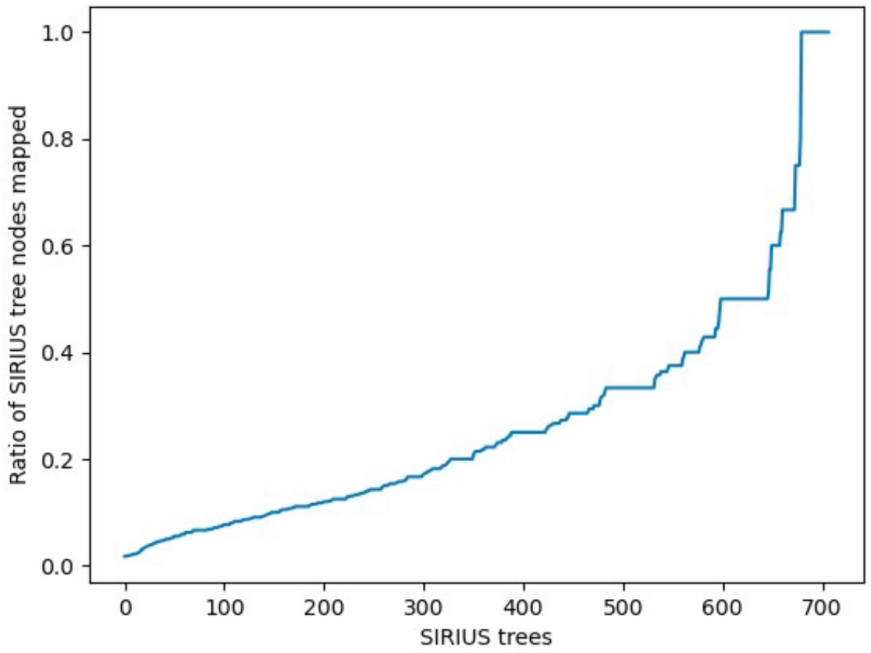

- here: one of 10000 rules (bin size 4)
- graph product based
- bin size: upto > 100

# DP Results

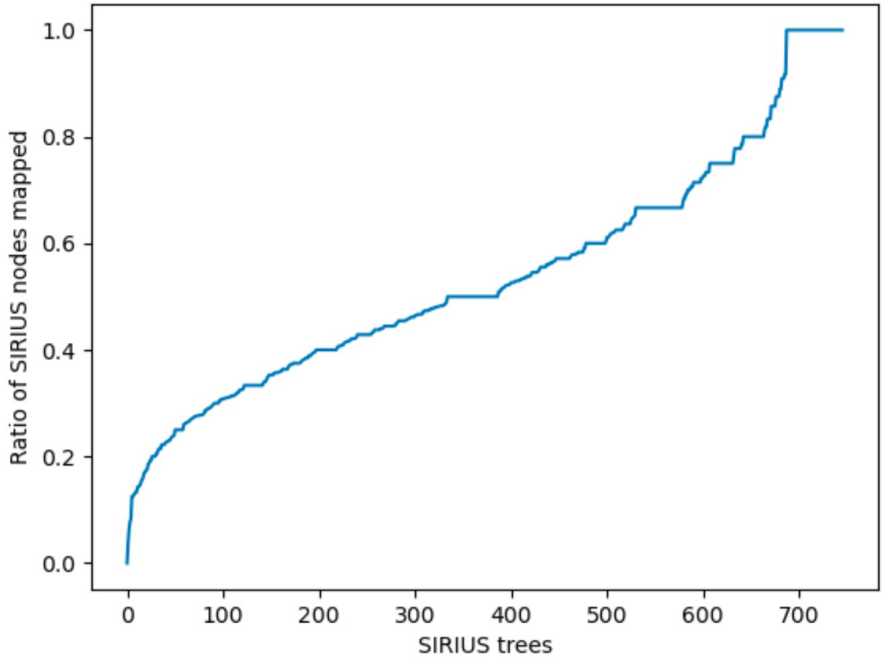
Approx. 700 SIRIUS trees, how many can be mapped, what is the quality of the mapping?

Sorted distribution qualities

Manually designed rule set

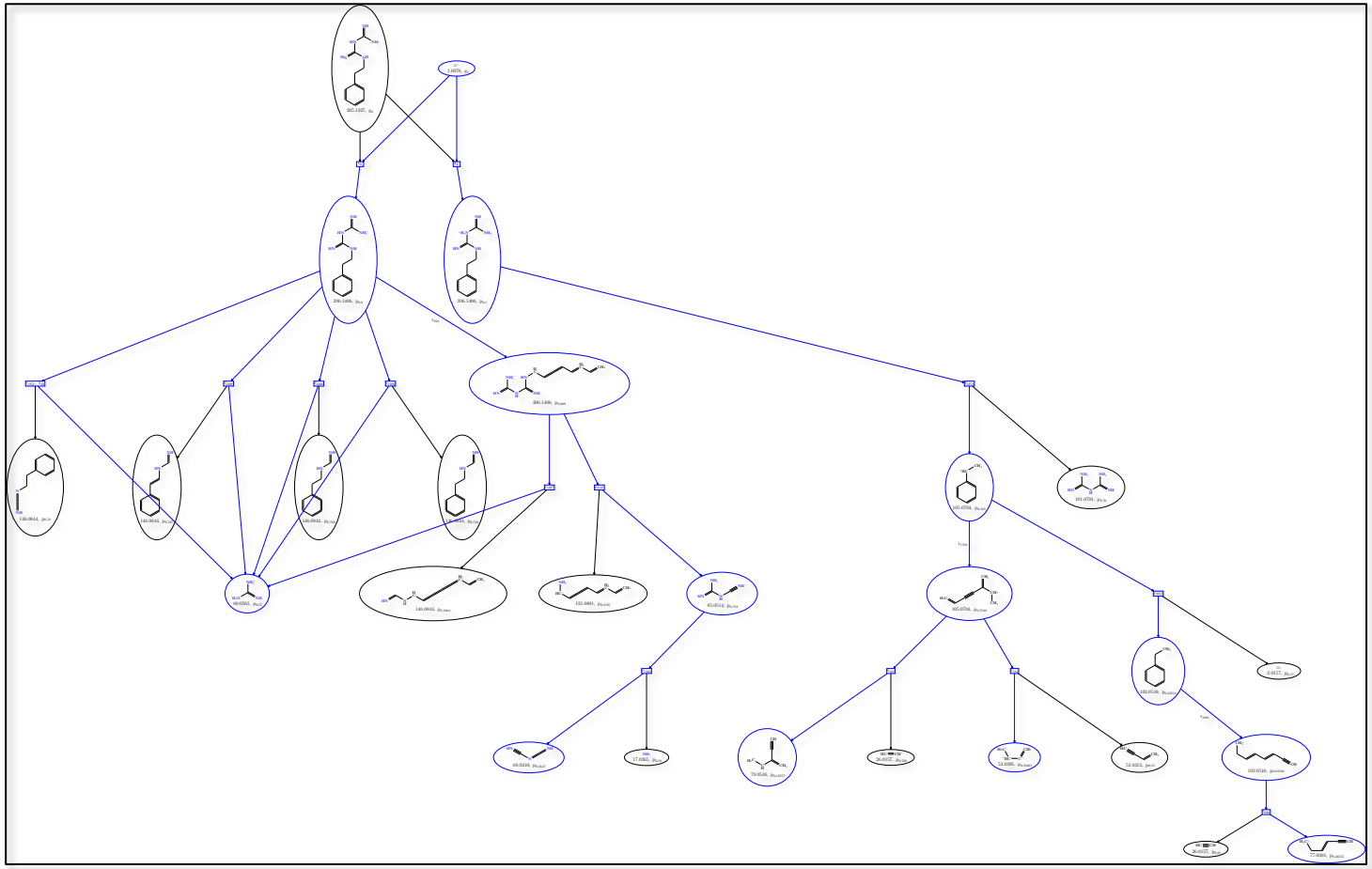



CFM-ID – based rule set (inferred)

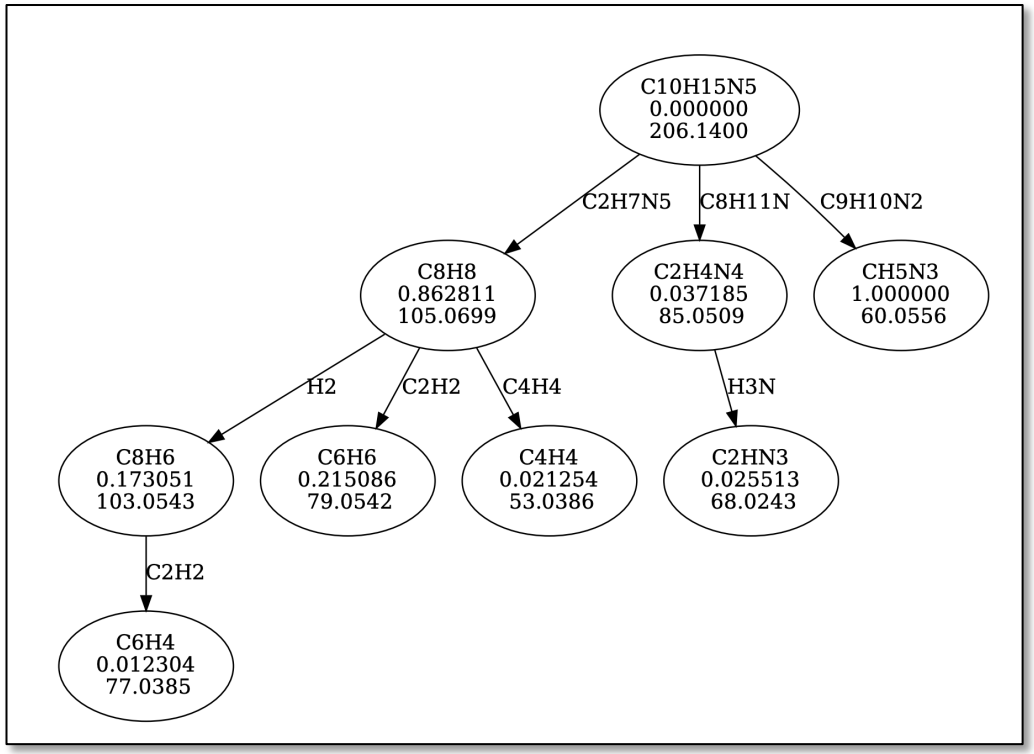


# Results (Examples)

MØD



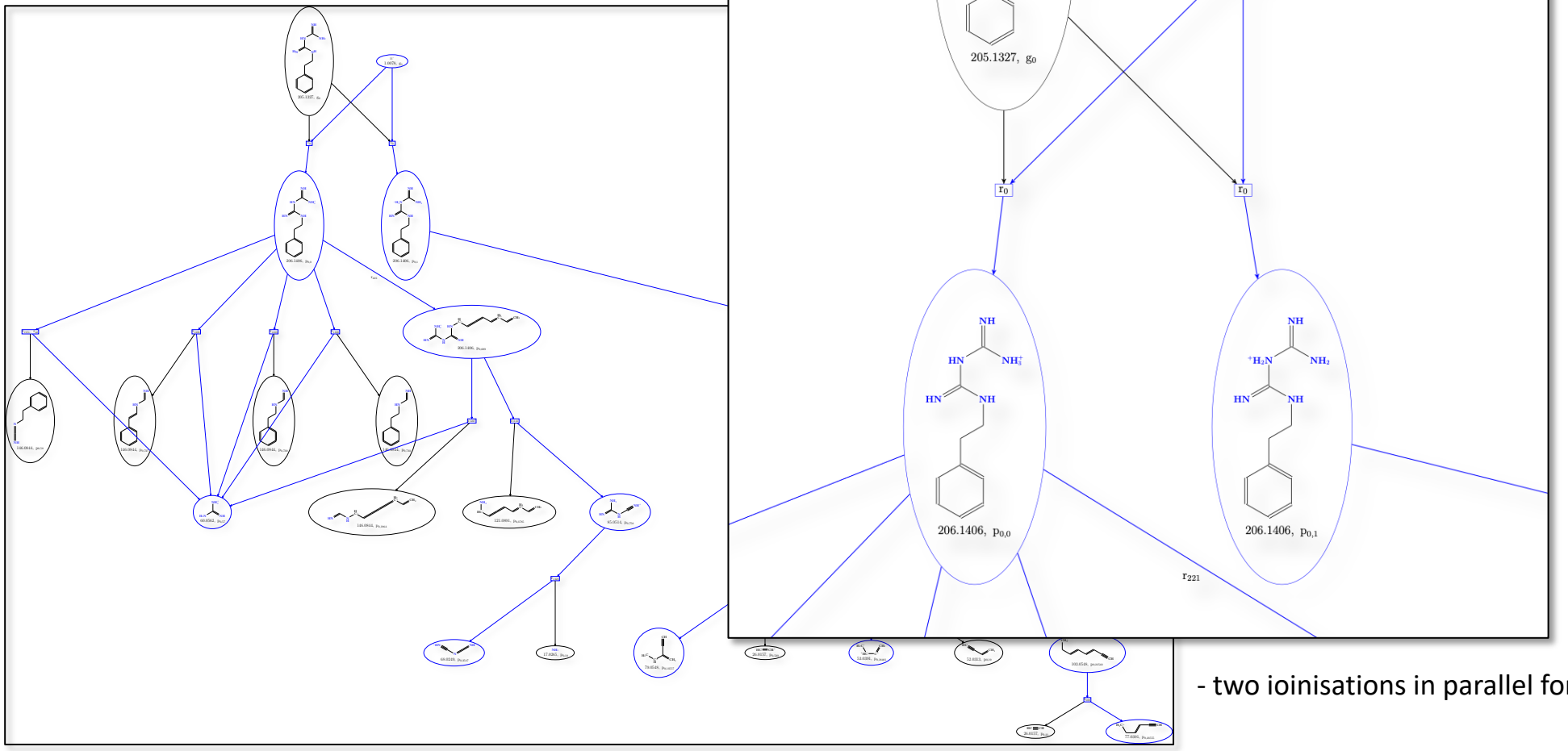
SIRIUS




- robust despite randomization of fragmentaion DAG generation (!)
- two ionised compounds for best explanation (!)

# Results (Examples)

MØD



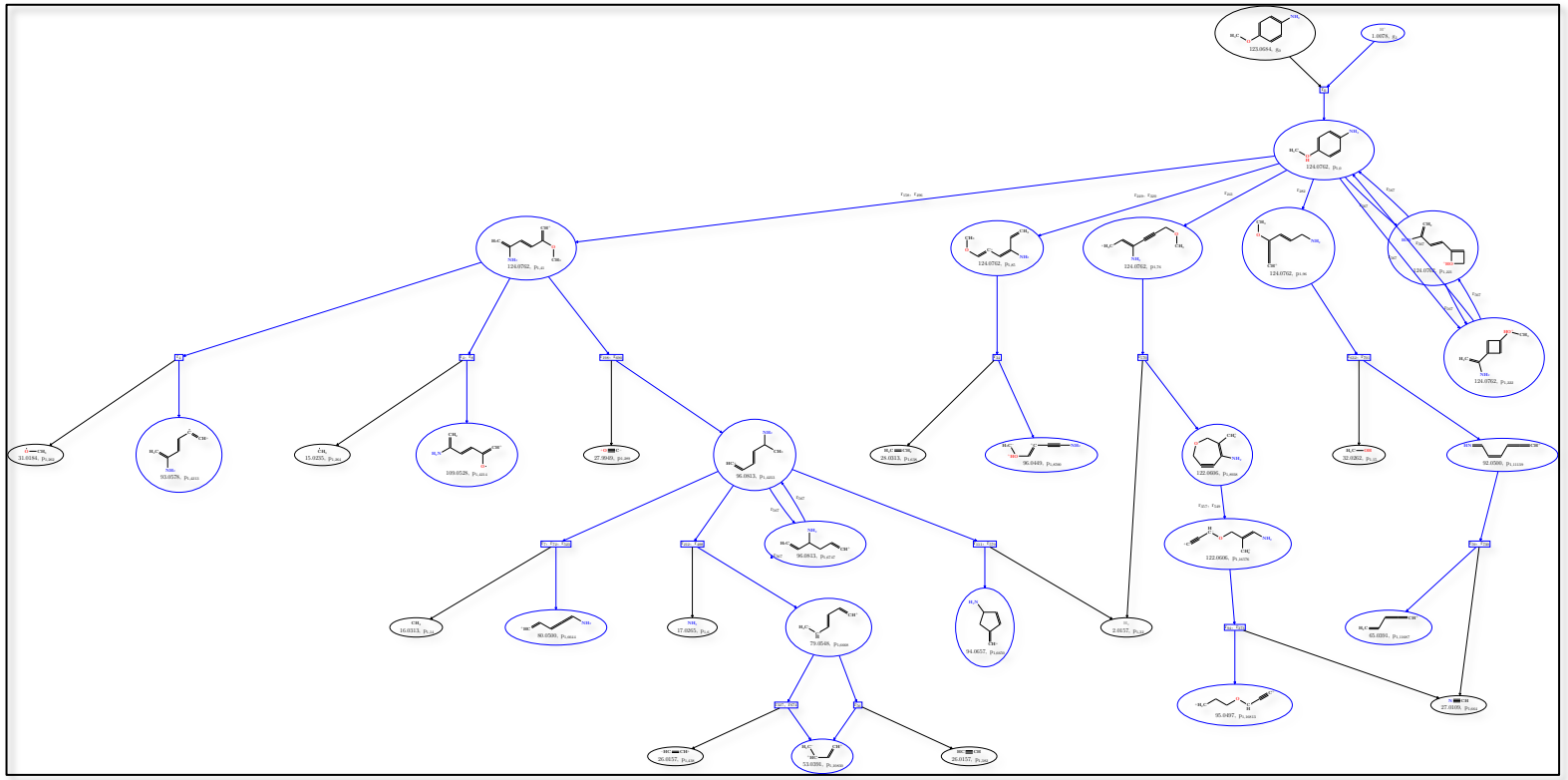

- two ioinisations in parallel for explanation

# Results (Examples)

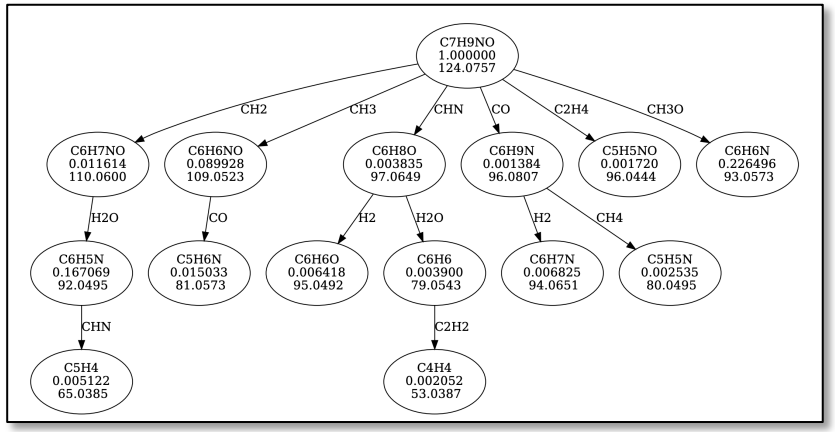
TACsy Training Alliance for Computational systems chemistry

MATONIC

MØD



SIRIUS

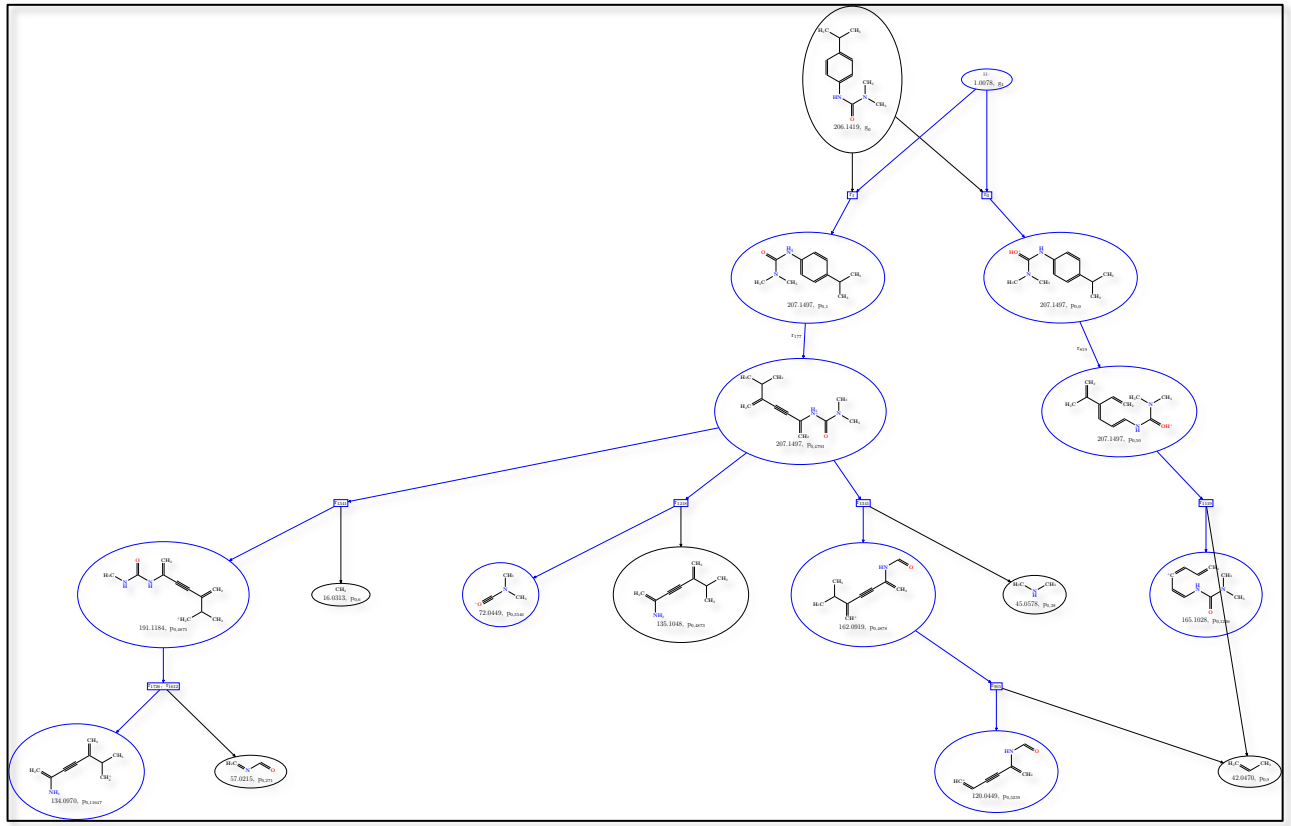


# Results (Examples)

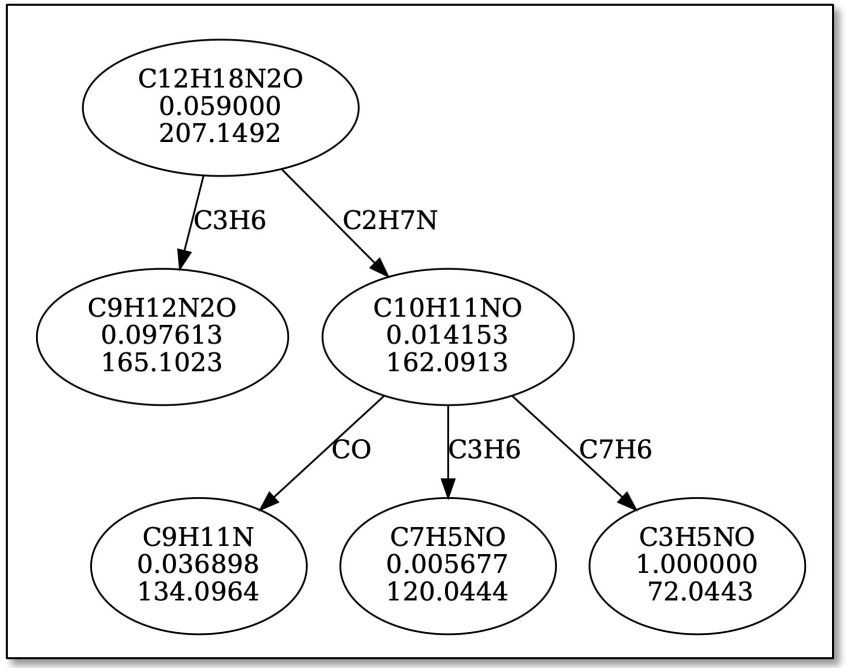
TACsy Training Alliance for Computational systems chemistry

MATOMIC

MØD

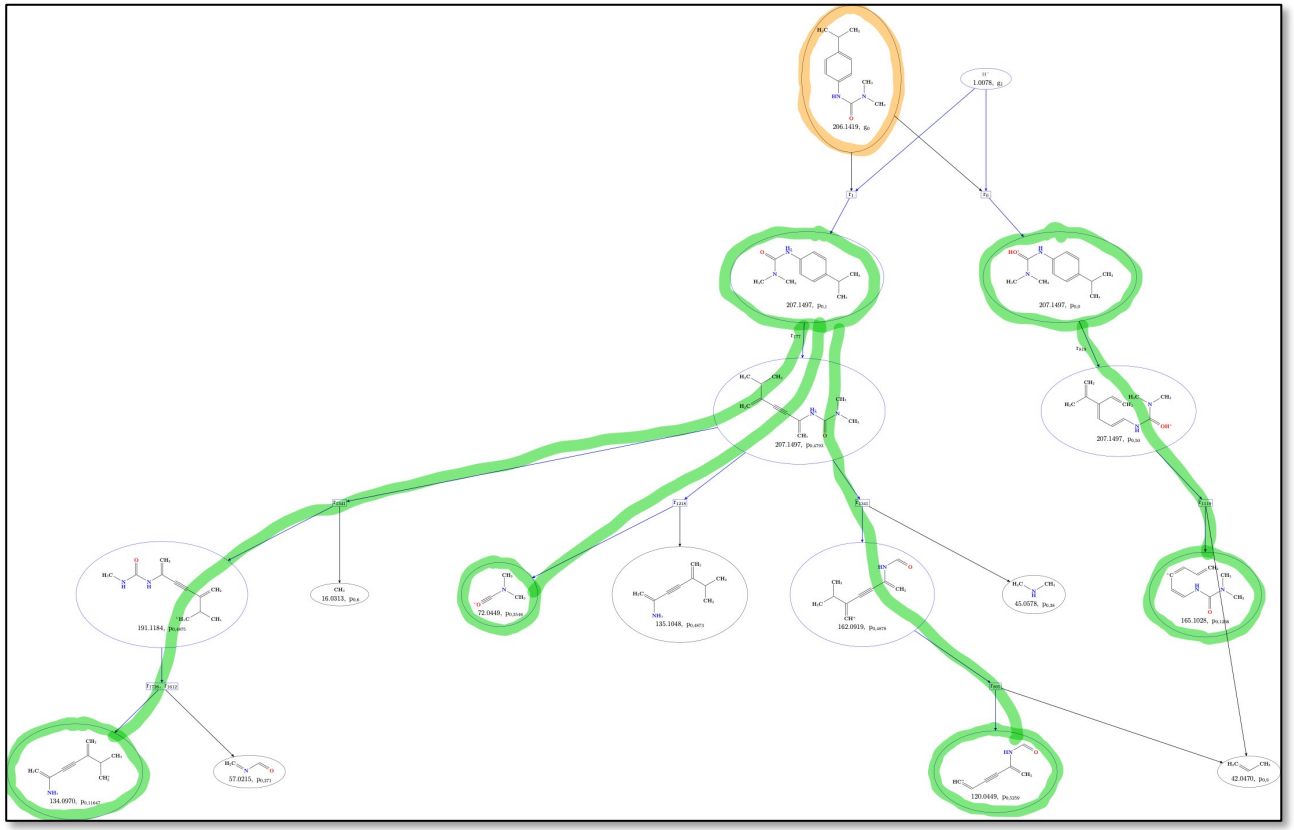


SIRIUS

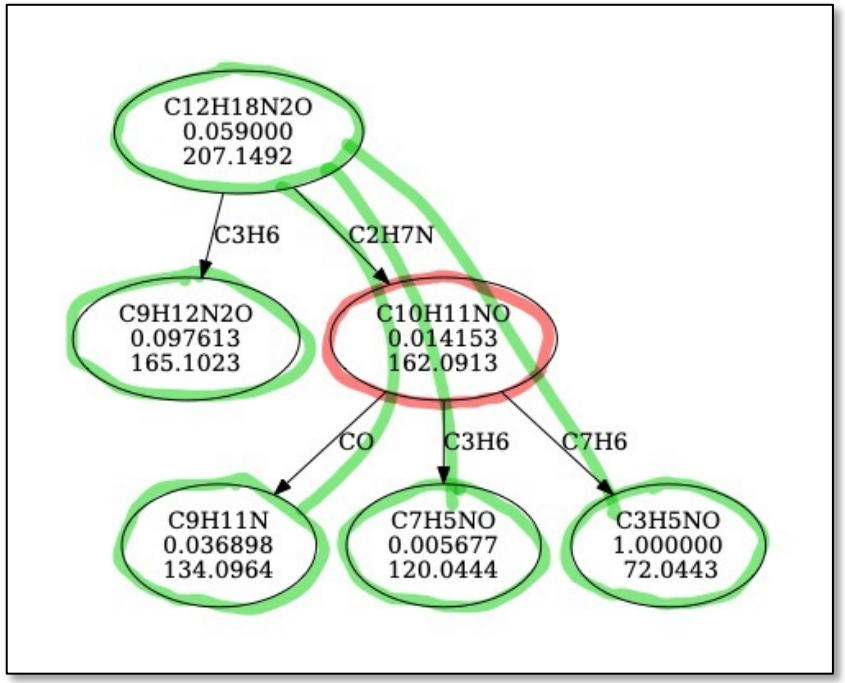


# Results (Examples)

MØD

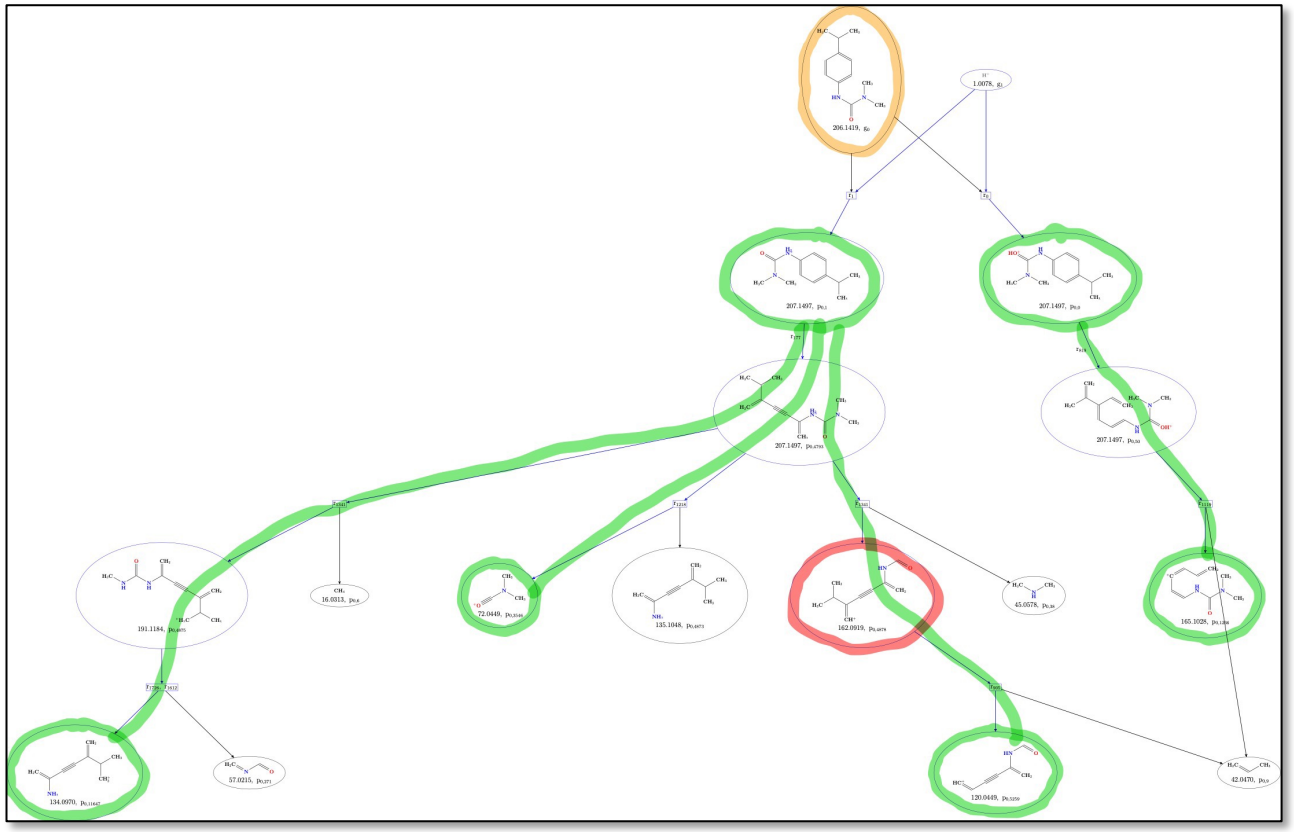


SIRIUS

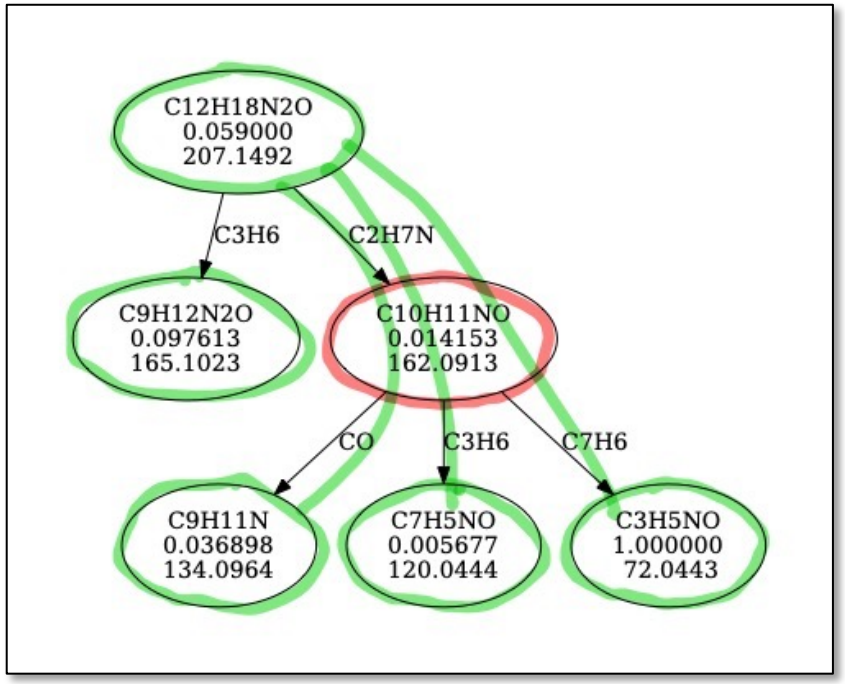


# Results (Examples)

MØD



SIRIUS




Potential SIRIUS correction

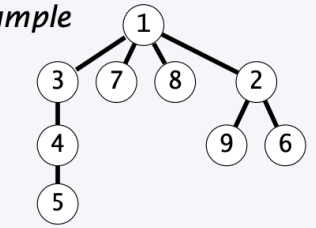
# Blackbox Replacement for SIRIUS

- Use (sampling of) increasing Cayley Trees (instead of SIRIUS fragmentation trees)

## Increasing trees

| | |
|---|---|
| *Class* | *Q*, the class of Cayley trees *whose labels increase on every path* |
| Construction | $Q = Z^{\square} \star SET(Q)$ |
| **EGF** equation | $Q'(z) = e^{Q(z)}$ |
| Solution | $Q(z) = \ln \frac{1}{1-z}$ |
| Counting sequence | $Q_N = N![z^N]Q(z) = (N-1)!$ |

*Example*




"Cayley" = "rooted, labelled, unordered"

# Conclusion

- Mechanisitc explanation for MS and MS/MS results
- (Overapproximated) rule set inferrence
- Rule set quality / black box quality

- Next steps:
  - Robustness
  - Isotopes
  - Application to lipids (Johannes in TACsy)
  - Rules inferrence (shadow size vs #rules, using progressive “anchored” MCS and ILP)
  - Application to metabolic networks (network completion)
  - Different black boxes
  - Increasing Cayley Trees

The TACsy project has received funding from the European Union’s Horizon 2021 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 101072930.