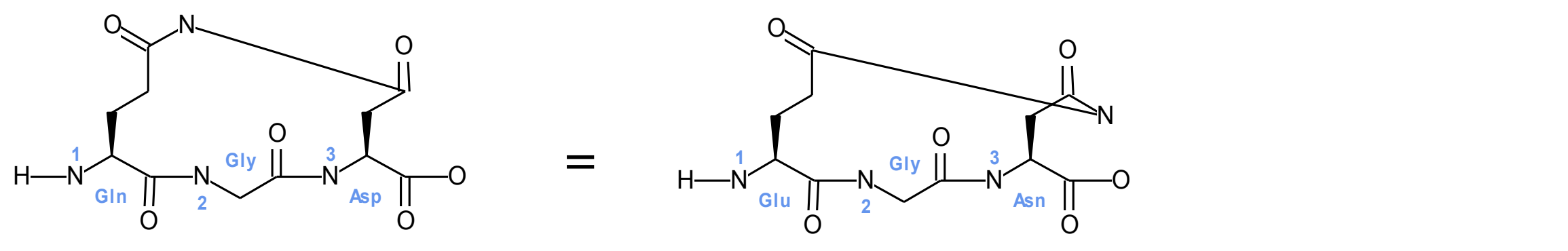
A computationally efficient structure key for large proteins

Jan Holst Jensen[1], Gerd Blanke[2]

[1] Biochemfusion ApS, Copenhagen, Denmark, [2] StructurePendium Technologies GmbH, Essen, Germany

Introduction - composite structure keys

Protein structures are conveniently described by their plain amino acid sequences. However, the plain sequence cannot capture sidechain and terminal modifications and it is therefore ill suited as a unique protein identifier. A further complication is crosslinks.



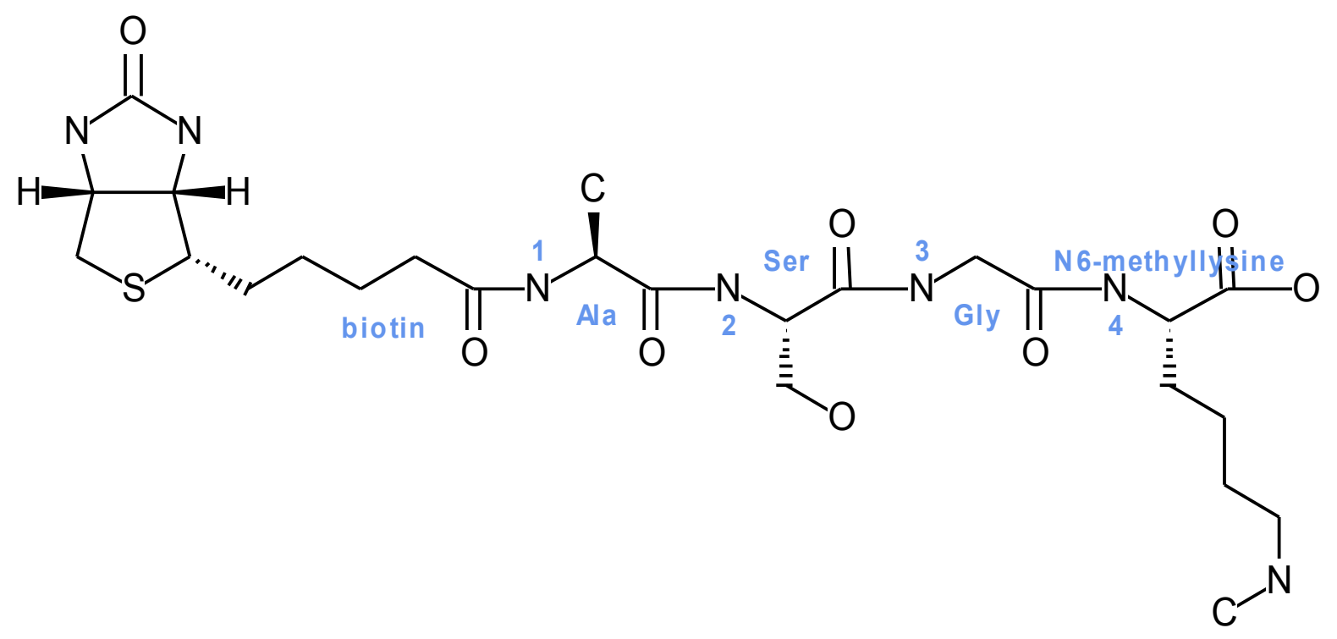
The crosslinked peptide above shows that lactam cyclization between Gln and Asp yields the same chemical structure as a lactam cyclization between Glu and Asn.

One solution is to use the InChI key for the entire chemical structure as an identifier. [1] However, this is not feasible (nor efficient) for structures having more than 1024 atoms, roughly corresponding to a protein with ~130 residues.

The approach pursued here is to use the protein plain sequence to the maximum extent possible, but replacing chemically modified parts of the protein by the InChI key of those chemical fragments. This resulting *composite key* is very efficient when the majority of the protein structure can be described by its plain sequence.

Simple chains - no crosslinks

Below is an example peptide expressed in PLN [2], where we have a terminally-modified residue and another residue with a sidechain modification.



```
PLN: [biotin]-Ala-Ser-Gly-[N6-methyllysine]-OH
Key: [JQGJUMKGKLLWLT-ZFLHDRRCSA-N]-Ser-Gly-[KLXPPJKQDSBAIA-FVMUQQIWSA-N]
```

The first InChI key *[JQG...]* in the composite key is the InChI key of the "[biotin]-Ala" fragment and the second key *[KLX...]* is for the "N6-methyllysine" residue.

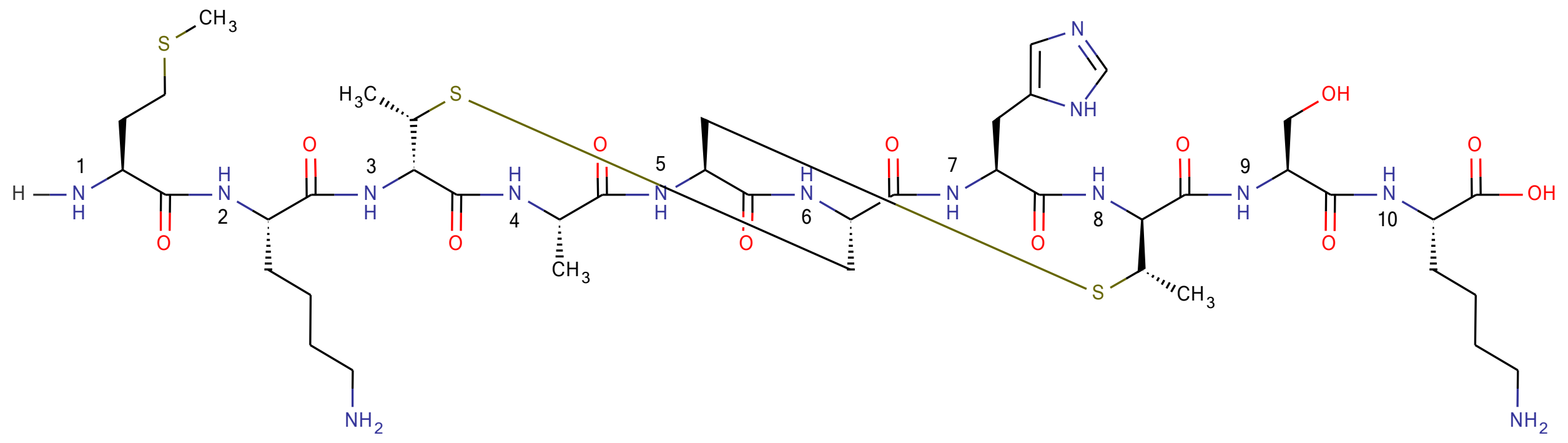
The InChI keys calculated for the fragments contain information about the fragment attachment points although InChI does not support attachment points. This attachment point information is crucial in order to unambiguously capture where the backbone is located and how its peptide bonds are oriented.

We have found that labelling each attachment point by fixed isotope complexes works reliably. The probability that these isotope complexes occur by accident in reality can be kept to practically zero when the isotope complexes are selected carefully.

Crosslinked residues

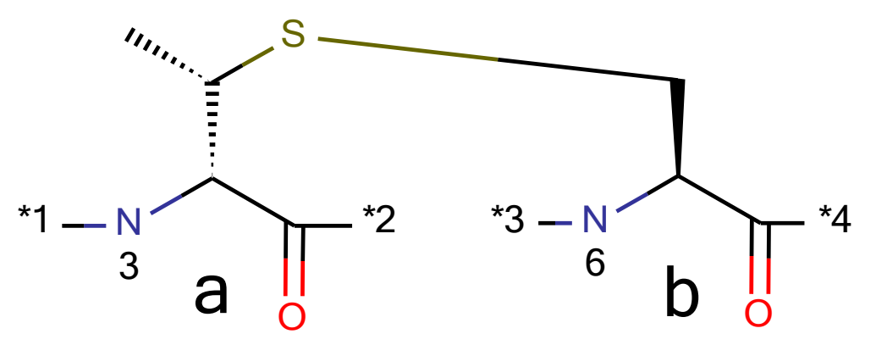
A real-life example of crosslinked residues is the 34-residue polycyclic peptide Nisin with five thioether crosslinks.[3]

The example peptide below is similar to the C-terminal end of Nisin, showing two Threonine-Cysteine thioether crosslinks, one going in the opposite direction of the other.

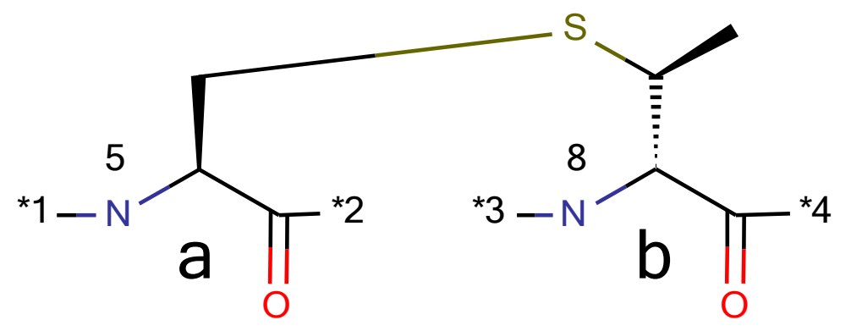


We could attempt to represent each crosslink by reverse-engineering its original residue structures and describe them as linked by a reaction, e.g. "Thr->Cys(thio)". This is difficult since multiple starting residues may potentially produce the same crosslink structure, depending on the reaction that produces the crosslink. As in the lactam example above.

Instead, we have chosen to represent each crosslink fragment by its structure key. The structure key will be normalized so it is direction-invariant - which is important when canonicalizing cyclic peptides - and additional data describes the direction of the crosslink, ensuring that the structure is described correctly by the final composite key.



Crosslink, residue 3 → 6, InChI key LXURMCIIVOJVNI-CDFSFGNOSA-N.



Crosslink, residue 5 → 8, InChI key HUFQJNDMEWMFMQ-CDFSFGNOSA-N.

The InChI key chosen as the direction-invariant key is the one with the lowest lexical sort order, here the *HUF...* key. The composite structure key for the example peptide is then

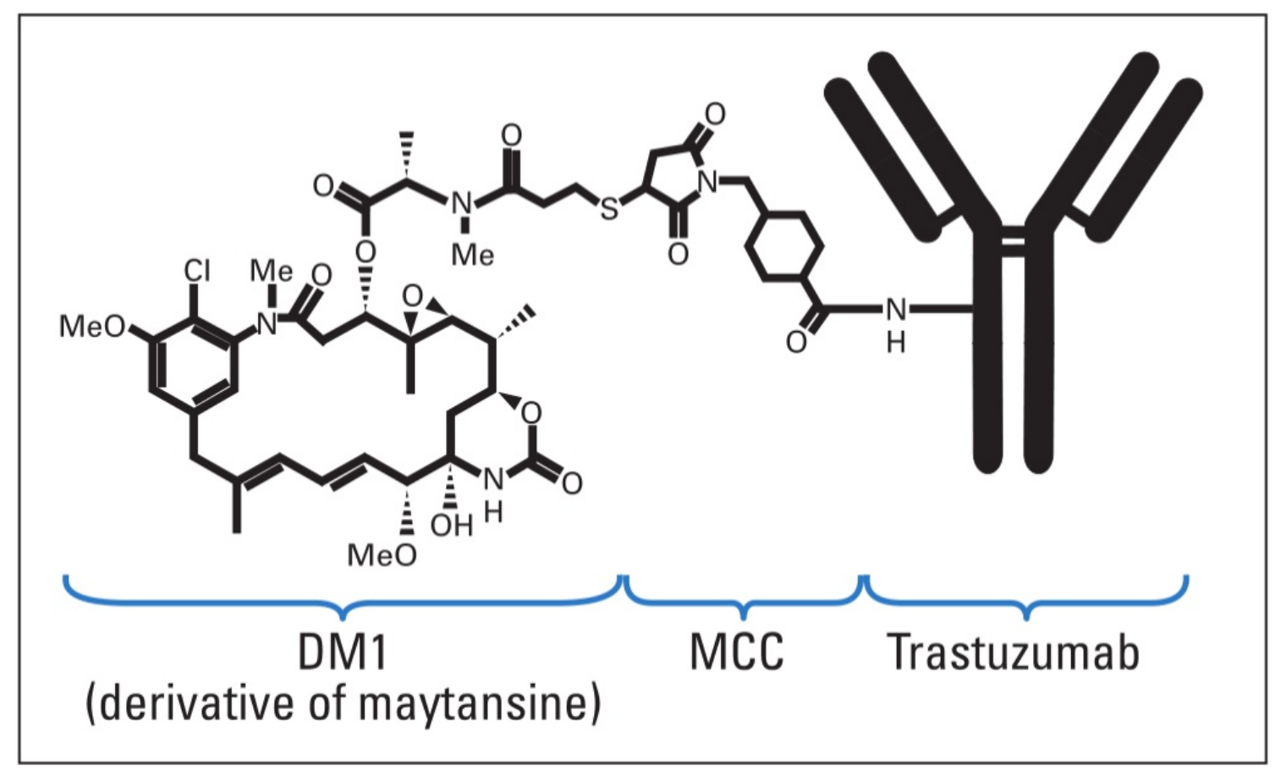
```
Met-Lys-
[HUFQJNDMEWMFMQ-CDFSFGNOSA-N](xl1,b)-Ala-
[HUFQJNDMEWMFMQ-CDFSFGNOSA-N](xl2,a)-
[HUFQJNDMEWMFMQ-CDFSFGNOSA-N](xl1,a)-His-
[HUFQJNDMEWMFMQ-CDFSFGNOSA-N](xl2,b)-Ser-Lys
```

The link from residue 3 to 6 is the reverse of the direction-invariant crosslink and therefore residue 3 is designated as the "b" endpoint, and residue 6 the "a" endpoint.

If the crosslink fragment were fully symmetric both residues would be marked as an "a" endpoint.

Scaling to antibody-size

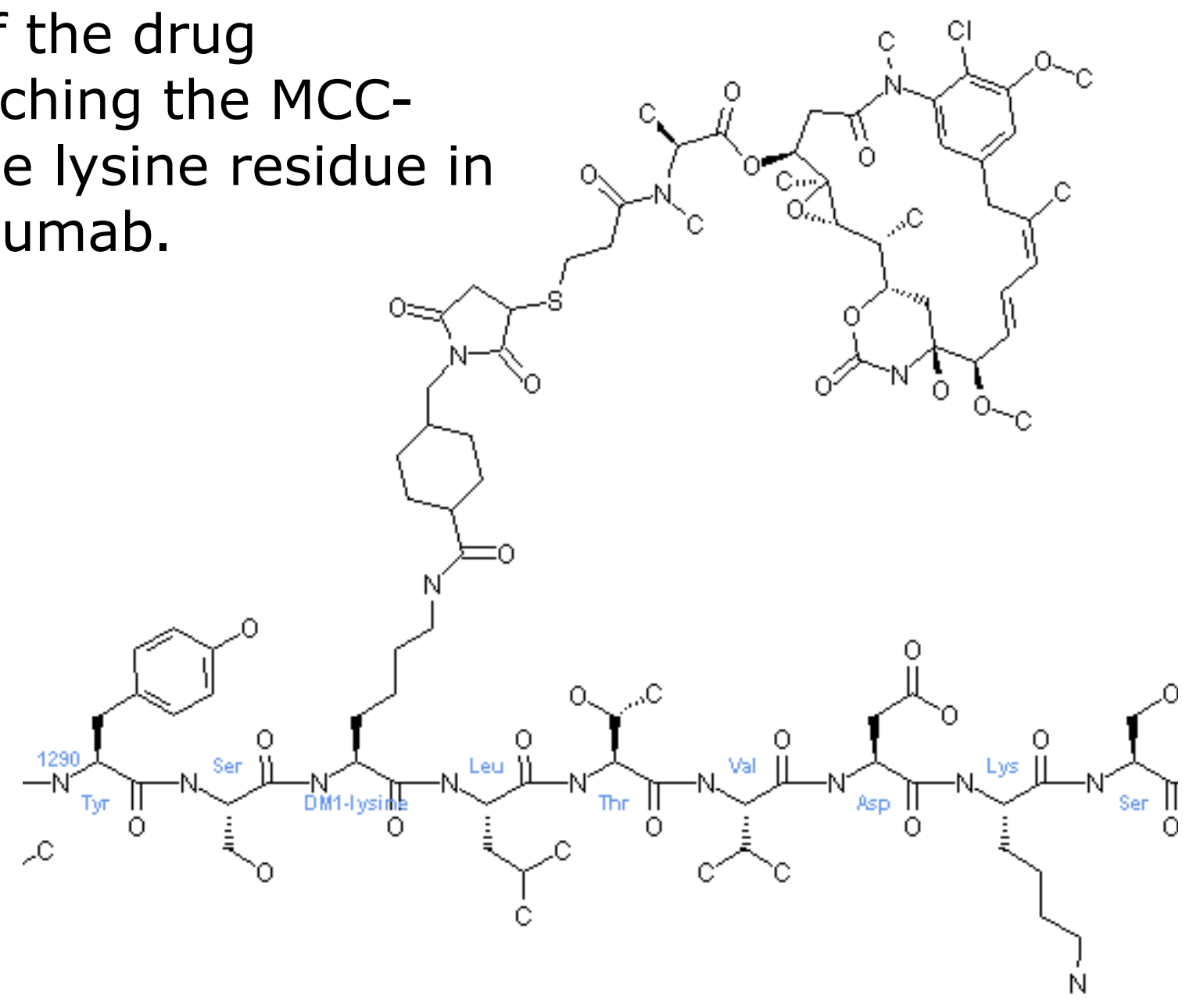
To demonstrate keys for large protein structures we used the following example from literature.



Antibody-drug conjugate as described in [4].

Sequence rendering of trastuzumab with lysine-linked DM1.

We emulate a version of the drug described above by attaching the MCC-DM1 structure to a single lysine residue in a heavy chain of trastuzumab.



Detail of antibody chemical structure - modified lysine residue.

We do not show the complete composite key of the antibody-drug conjugate since it is very long; the antibody sequence is more than 1300 residues.

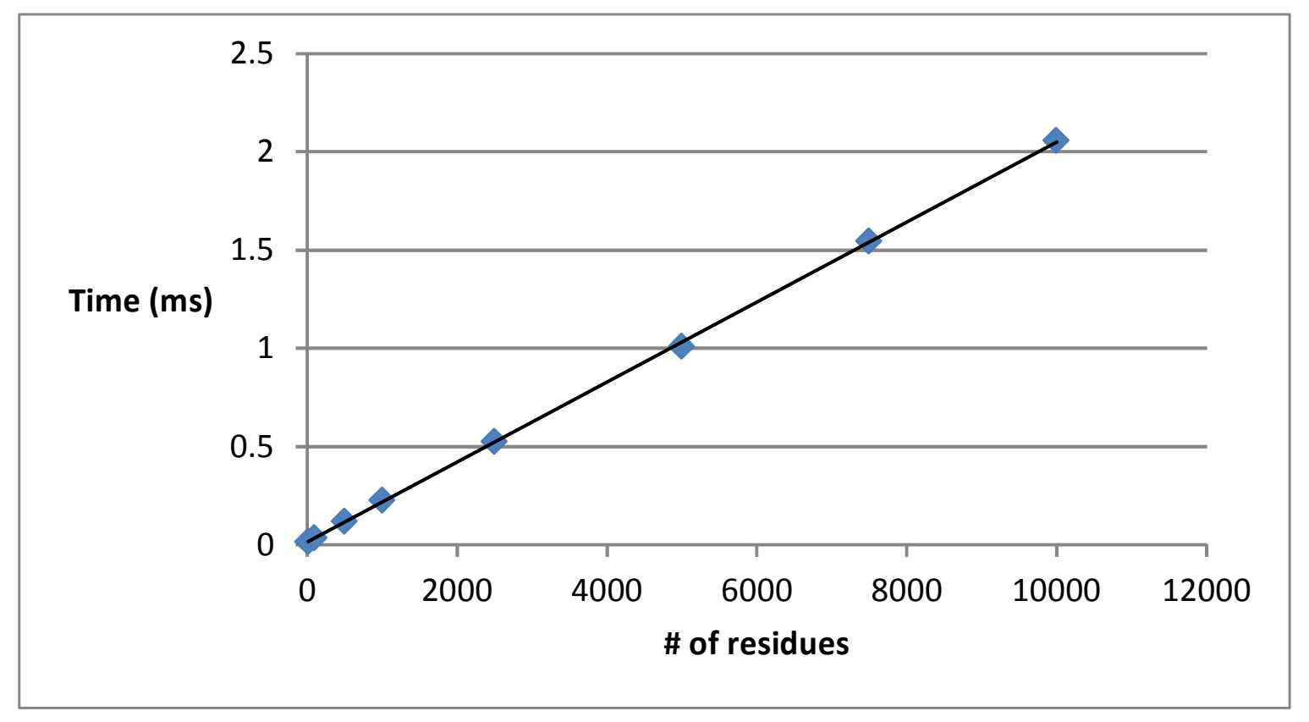
Very long keys are problematic to index efficiently and therefore we represent the key by a hash value. We have chosen to use the MD5 checksum of the complete composite key. The MD5 checksum of the composite structure key for the antibody-drug conjugate is

```
94e7c1e08a8381d3239afd43d0525f3d
```

This checksum is well-suited for indexing in a standard database system to support very fast duplicate checks.

Performance and scalability

The performance of the composite key checksum generation was tested for 4-chain proteins of varying chain lengths.



Timings obtained on 2.5 GHz 64-bit Intel CPU.

The chart to the left shows how long it takes to calculate the composite key checksum. The X-axis shows the total number of residues in the input protein and the Y-axis the processing time - including parsing the PLN input - in milliseconds.

When introducing modified residues, like the DM1-modified lysine residue above, the processing time increases since an InChI key has to be calculated for each unique modification present.

In the case of trastuzumab, the processing time goes from 0.33 ms (plain sequence antibody) to 2.16 ms (antibody with one modified DM1-lysine residue). However, the processing time stays constant regardless of whether the antibody has 1 or 10 DM1-lysines.

DM1-lysine count	Time (ms)
1	2.16
2	2.16
3	2.18
5	2.17
8	2.17
10	2.18

Discussion

The composite structure key is a very efficient way of describing well-defined proteins where you know the sequence and have a fair knowledge of its secondary structure and modifications. The key calculation scales well to proteins having several thousand residues and a modest amount of known modifications.

We have not described how tertiary or even more complex crosslinks are handled. The principles still apply although the performance of each crosslink structure key calculation scales badly as the ordinality of the crosslink increases: Tertiary crosslinks require 3 * 2 * 1 key calculations to normalize, quarternary crosslinks 4 * 3 * 2 * 1 key calculations, and so on. Time will tell how often crosslinks with high ordinalities occur in practice.

When the exact linking sites for modifications are not known, the composite structure key will describe the structure less reliably. In the case of trastuzumab with DM1, the cited article states that "*An average of 3.5 DM1 molecules are conjugated to the Fc region of trastuzumab.*".[4] To describe this scenario appropriately a structure key should be able to handle statistically distributed components. This is an area for further investigation.

Biochemfusion, Copenhagen, Denmark

StructurePendium, Essen, Germany

[1] InChI - The IUPAC International Chemical Identifier, http://www.iupac.org/home/publications/e-resources/inchi.html

[2] Biochemfusion PLN (Protein Line Notation), http://www.biochemfusion.com/doc/#Specifications

[3] Nisin, Wikipedia, https://en.wikipedia.org/wiki/Nisin

[4] Ian E. Krop et al., "Phase I Study of Trastuzumab-DM1, an HER2 Antibody-Drug Conjugate, Given Every 3 Weeks to Patients With HER2-Positive Metastatic Breast Cancer", *Journal of Clinical Oncology*, **2010**, Vol. 28, pp 2698-2704.