Dührkop, Kai; Fleischauer, Markus; Ludwig, Marcus; Aksenov, Alexander A.; Melnik, Alexey V.; Meusel, Marvin; Dorrestein, Pieter C.; Rousu, Juho; Böcker, Sebastian

# SIRIUS 4

*Published in:*
Nature Methods

*DOI:*
10.1038/s41592-019-0344-8

Published: 01/04/2019

*Document Version*
Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

*Please cite the original version:*
Dührkop, K., Fleischauer, M., Ludwig, M., Aksenov, A. A., Melnik, A. V., Meusel, M., Dorrestein, P. C., Rousu, J., & Böcker, S. (2019). SIRIUS 4: a rapid tool for turning tandem mass spectra into metabolite structure information. *Nature Methods*, *16*(4), 299-302. https://doi.org/10.1038/s41592-019-0344-8

# SIRIUS 4: Turning tandem mass spectra into metabolite structure information

Kai Dührkop[1,6], Markus Fleischauer[1,6], Marcus Ludwig[1,6], Alexander A. Aksenov[2,3], Alexey V. Melnik[2,3], Marvin Meusel[1,4], Pieter C. Dorrestein[2,3], Juho Rousu[5], and Sebastian Böcker[1,7]

[1] Chair for Bioinformatics, Friedrich-Schiller-University, Jena, Germany
[2] Collaborative Mass Spectrometry Innovation Center, Skaggs School of Pharmacy and Pharmaceutical Sciences, University of California, La Jolla, San Diego, California, USA
[3] Skaggs School of Pharmacy and Pharmaceutical Sciences, University of California, La Jolla, San Diego, California, USA
[4] Department of Microbial Natural Products, Helmholtz-Institute for Pharmaceutical Research Saarland, Helmholtz Centre for Infection Research and Pharmaceutical Biotechnology, Saarland University, Saarbrücken, Germany
[5] Helsinki Institute for Information Technology HIIT, Department of Computer Science, Aalto University, Espoo, Finland
[6] These authors contributed equally to this work
[7] Corresponding author, sebastian.boecker@uni-jena.de

**Abstract.** Mass Spectrometry is one of the two predominant experimental techniques in metabolomics and related fields, but structural elucidation remains highly challenging. A "BLAST-like" computational tool for swiftly searching in structure databases, is currently missing but highly anticipated. We have developed a new computational approach that represents a milestone in identification performance, achieving identification rates of more than 70 % on challenging metabolomics datasets, and is also very swift in application.

## 1 Main Text

Identification of molecules remains a central question in analytical chemistry, in particular for natural products research, untargeted metabolomics, environmental research, and biomarker discovery. Due to its high sensitivity, Mass Spectrometry (MS) is well-suited for the high-throughput characterization of biomolecules. Automated interpretation of tandem mass spectra (MS/MS) is often limited to searching in spectral libraries, so that we can only dereplicate compounds for which a reference sample has been measured and stored. Manual interpretation is cumbersome and work-intense, as current MS technology can produce hundreds of thousands of MS/MS spectra per day on a single instrument.

Many existing tools in untargeted metabolomics are generalist tools for the complete metabolomics analysis pipeline [1–5], others concentrate on particular classes of metabolites. In contrast, SIRIUS 4 is a highly specialized tool addressing two inevitable and fundamental questions: What is the molecular formula of the query compound, among *all* molecular formulas, previously observed or unobserved? And, given a database of molecular structures, what is the structure that best explains the experimental data? Molecular formula identification using isotope pattern analysis was first addressed by SIRIUS version 1 released in 2009 [6]. In 2011, SIRIUS$^2$ added methods for the analysis of MS/MS data using fragmentation trees [7]. Version 3.0 of SIRIUS was released in May 2015; it was a complete rewrite of the software, and implemented the Maximum A Posteriori Estimation from [8], improving correct molecular formula identification rates from MS/MS data 2.5-fold. Different from previous versions, version 3.0 did not have a graphical user interface. CSI:FingerID [9] was introduced in Oct 2015, and is based on predicting a molecular fingerprint of a query compound from its fragmentation tree and spectrum [10]. Different to its ancestor FingerID [11], it integrates fragmentation trees in the prediction pipeline, resulting in a significant increase in correct identifications. In this first incarnation of CSI:FingerID, users had to submit queries through a web interface.

On the conceptual level, SIRIUS 4 integrates high-resolution isotope pattern analysis and fragmentation trees with structural elucidation, to provide a combined and coherent assessment of molecular structures from MS/MS data for large data sets. Users can now analyze full LC-MS datasets with the best-in-class tool, and not just a spectrum at the time; MS-driven annotations can be obtained for all detected features, not just those passing a preliminary statistical test, say, on fold change. This paves the road to more sophisticated data mining techniques, exploiting the structure of metabolic pathways [12] or structural similarity of molecules [13].

On the technical level, SIRIUS 4 and its web service include 213,071 lines of Java code, of which about 94% are new from SIRIUS 3.0. Novel functionalities were added to enable a true synergism that goes well beyond the sum of the two original parts: CSI:FingerID is now seamlessly integrated into SIRIUS 4 via a RESTful (Representational State Transfer) web service (Fig. 1a), allowing users to process full LC-MS datasets instead of individual compounds. SIRIUS 4 provides an intuitive graphical user interface with six views (Supplementary Fig. 1), including views for CSI:FingerID results and predicted fingerprints. Using extensive algorithm engineering and parallelization through a novel job scheduling system (Supplementary Fig. 2), SIRIUS 4 running times were reduced by more than two orders of magnitude, reducing the time needed to analyze a full LC-MS run from hours to minutes. SIRIUS 4 integrates a new model for scoring isotope patterns that combines absolute and relative noise for peak intensities, outperforming all previous scorings. Furthermore, MS/MS spectra with isotope peaks can be processed (Supplementary Fig. 3). Fragmentation tree computation is based on Maximum A Posteriori Estimation from [8] to choose the molecular formula that best explains the data. SIRIUS 4 also implements automated element detection from isotope patterns [14] using a Deep Neural Network. The CSI:FingerID web service uses ten novel kernels

(Supplementary Table 1) and predicts more than thousand additional molecular properties, mostly from combinatorially generated fingerprints. MS/MS data from 11,728 compounds have been added to the training data; furthermore, CSI:FingerID can now also analyze negative ion mode data. To score structure candidates, SIRIUS 4 implements the Bayesian network scoring from [15]. Between Jul 2016 and Aug 2018, the CSI:FingerID web service has processed 5.8 million queries submitted by users from 36 countries and six continents.

SIRIUS 4 can identify the molecular formula of a query compound with very high accuracy; no spectral or structural databases are required for this step of the analysis, as *all theoretically possible molecular formulas* are considered, allowing one to overcome limitations of current structure databases. Molecular formulas highly atypical for a biomolecule may be penalized (Supplementary Fig. 4) but are never discarded. SIRIUS 4 offers outstanding accuracy both for the identification of molecular formulas and, through the CSI:FingerID web service, for searching in molecular structure databases (see Supplementary Note 1 for related work): In an evaluation on independent data, correct molecular formulas and structures were ranked in first place (i.e., successfully identified) for more than 90 % and 70 % of the compounds, respectively. Using suspect databases, databases with hypothetical metabolites, or the predicted fingerprint of a query compound (Fig. 1b-f and Supplementary Fig. 5), SIRIUS 4 even allows to overcome boundaries of the "known chemical space".

CSI:FingerID provides structural information (namely, the predicted molecular fingerprint) without requiring any molecular structure database, or if no similar structure is present in any database today: The predicted fingerprint is now prominently featured in the user interface (Fig. 1b-f). When searching structure databases, these may (Supplementary Table 2) or may not [16] be restricted to molecules with known biological relevance, and may also comprise hypothetical metabolites [17]. In addition, user-defined structures ("suspects") can be added to the candidate lists. Results are seamlessly integrated into the SIRIUS 4 user interface, allowing users to examine CSI:FingerID's "reasoning" by a graphical display of predicted molecular properties. Links are provided for structure and spectral databases where a particular compound structure is included.

For the new isotope pattern scoring, we compare the performance of SIRIUS 3.0 and SIRIUS 4 using data from 3,965 compounds (Supplementary Fig. 6a, Supplementary Result 1). When using only isotope patterns, correct molecular assignments (top 1) improve by 74.3 % (26.6 percentage points). Using both MS1 and MS/MS data, the improvement is still 6.3 percentage points, and the number of wrongly assigned molecular formulas (792 for SIRIUS 3.0, 541 for SIRIUS 4) drops by 31.7 %. Improved rates can be observed throughout all ranks. For running time evaluation, we again compare SIRIUS 4 against SIRIUS 3.0. We use two datasets, one with 1,553 compounds (MS/MS only) and the above dataset with 3,965 compounds (MS1 and MS/MS). For the first dataset, we restrict the set of elements to CHNOPS, to avoid proliferating running times of particularly SIRIUS 3.0. Here, SIRIUS 4 reached a 332-fold speedup. For the second dataset, we also consider the halogens BrClFI. We ordered compounds by mass, and divided them into batches of 50 compounds each. SIRIUS 4 processed all compounds in 5 h 41 min. SIRIUS 3.0 ran into time and memory issues for the last two batches; it processed the 3850 lightest compounds in 4 d 1 h, compared to 25 min 4 s for SIRIUS 4, corresponding to a 231-fold speedup. See Supplementary Fig. 6b and Supplementary Result 2.

Preliminary versions of SIRIUS and CSI:FingerID were used to participate in the CASMI (Critical Assessment of Small Molecule Identification) 2016 contest [18]. For the 127 compounds with positive ion mode data, CSI:FingerID was the best-performing automated method by far, reaching 55.1 % correct and unambigious identifications of structures (ranking the correct structure on first place) when searching ChemSpider [19]. CSI:FingerID showed particularly good performance when independent MS/MS data for the same structure was present in the training data; removing MS/MS data of these structures from the training data (structure-disjoint training data, see Supplementary

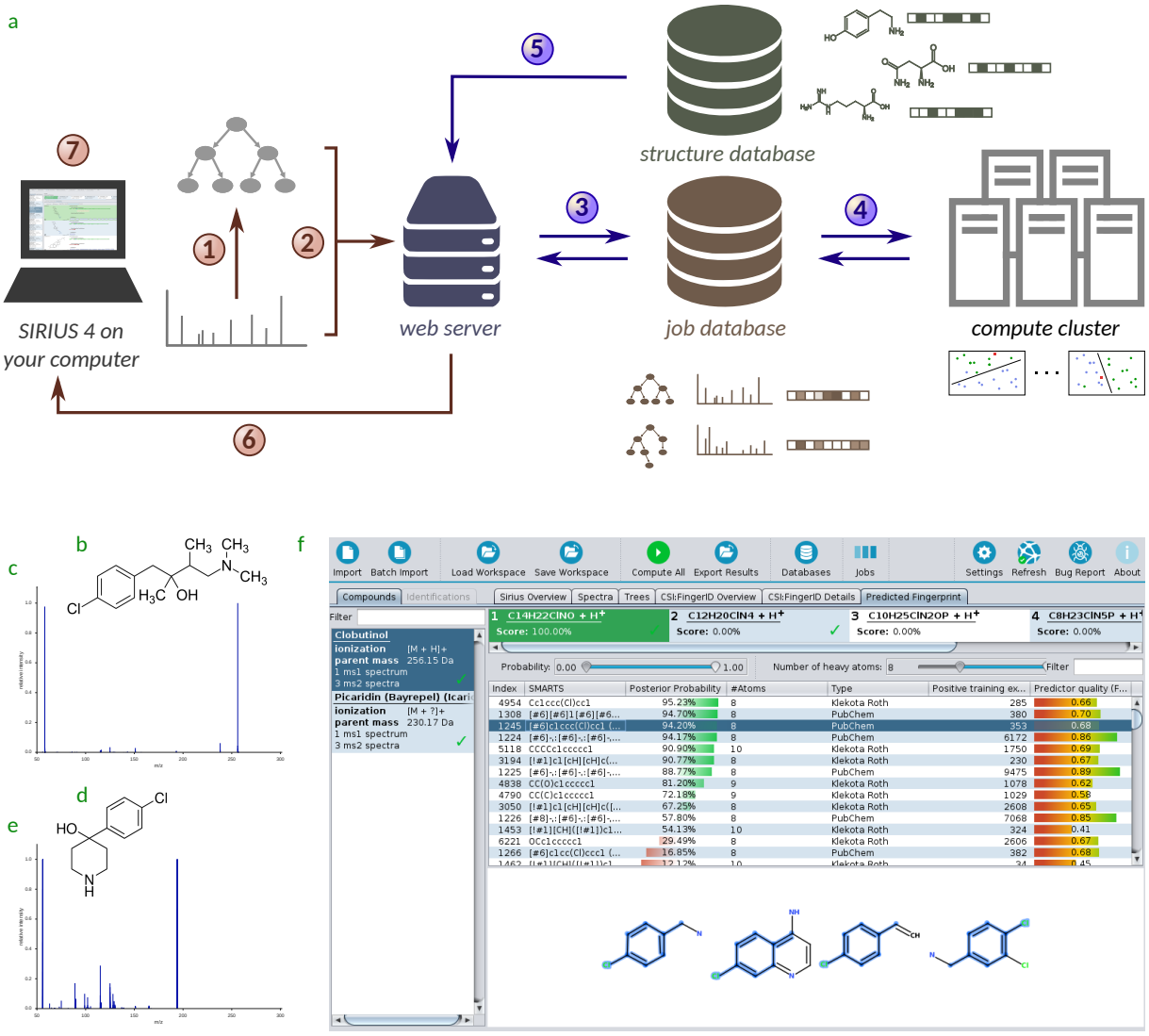


**Fig. 1. SIRIUS 4 embracing CSI:FingerID.** (a) Integration of CSI:FingerID into SIRIUS 4 via a RESTful web service: Fragmentation trees are computed locally with SIRIUS 4 (1). Fragmentation spectra and trees are uploaded to the CSI:FingerID web service (2). A job is created and temporarily stored in a relational database (3). The compute server fetches each job, predicts the fingerprint, and stores the result in the job database (4). Structures and fingerprints of candidates are retrieved from the structure database (5). The predicted fingerprint plus structures and fingerprints of candidates are returned via the web interface (6). The predicted fingerprint is locally scored against the candidate fingerprints; results are presented in the SIRIUS user interface (7). (b-f) Predicted Fingerprint Tab. (b) Structure of clobutinol, not known to SIRIUS 4 and CSI:FingerID. (c) MS/MS of query clobutinol; this, plus the precursor MS1, is the only input for SIRIUS 4 and CSI:FingerID. (d,e) Clobutinol is "novel", in the sense that no MS/MS data for this structure is present in the training data of CSI:FingerID (Supplementary Note 2). The structurally closest compound in the training data (measured by PubChem Tanimoto coefficient) is 4-(4-Chlorophenyl)-4-piperidinol (d), mass spectrum in (e). Cosine similarity (c,e) is below 0.01. (f) Predicted fingerprint of the query clobutinol. At this point, no structure database has been searched. Only molecular properties with at least 8 heavy atoms are displayed. One molecular property that is predicted to be present (indicated by green bars) has been selected; A few example structures that contain the corresponding property are displayed. For each property, SIRIUS 4 display the F1 score that this classifier reached in cross validation, and the number of positive examples in the training data. As expected, most molecular properties are predicted to be absent (indicated by red bars).

Note 2), the rate of unambiguous and correct identifications decreased to 27.6 %, which still outperformed all other methods participating in CASMI 2016 for positive ion mode data. We re-analyze the CASMI 2016 data with the version presented here (Supplementary Result 3): For 93.8 % of the compounds, SIRIUS 4 assigns the correct molecular formula using MS/MS and isotope pattern data (96.2 % in the top 2). When evaluating CSI:FingerID, we ensure structure-disjoint training data. Here, the rate of unambiguous and correct identifications in positive ion mode increases to 39.4 % (74.8 % in the top 10). For negative ion mode data, we unambiguously and correctly identify 28.4 % of the compounds (60.5 % in the top 10), again outperforming all other tools. Numbers are comparable or better than the 31.8 % correct identifications reported in Oct 2015 for cross-validation evaluation and positive ion mode [9]. In comparison, the current version reaches 40.4 % correct identifications in structure-disjoint cross-validation on GNPS, replicating the evaluation setup from ref. [9] (Supplementary Figs. 7 and 8, Supplementary Result 4). In practice, it is often reasonable to search in a database focusing on biomolecule structures: Searching CASMI 2016 data in a biomolecular structure database with 0.5 million structures (Supplementary Table 2) results in 74.0 % correct identifications (84.3 % in the top 3) for positive ion mode data, again using structure-disjoint cross-validation. For CASMI 2017 category 4 (automatic candidate ranking), SIRIUS and CSI:FingerID reached more than six times the number of correct identifications, compared to the best non-CSI:FingerID method (`http://www.casmi-contest.org/2017/results.shtml`).

In two biological case studies from human fecal and human skin data, we demonstrate how SIRIUS 4 can be used to derive knowledge otherwise unavailable (Fig. 2, Supplementary Fig. 9, Supplementary Results 5 and 6). Clearly, lack of structure annotations often impedes meaningful interpretation of the data and *in silico* structural elucidation becomes indispensable. In both studies, we investigated structures of molecules that arose as potentially important for understanding biological processes in these studies, but where no structural elucidations could initially be made. Application of SIRIUS 4 allowed determining structures of these key compounds, later confirmed by using standards, and thus provided a starting point for further exploration. We use spectral library search and propagation of annotation through molecular networks [1]; analogously, *in silico* annotations can be propagated through the network. Molecular networking and *in silico* annotation can be synergistic, but also allow us to verify *in silico* and network-propagated structure annotations, thus reducing the chance of misannotations.

## 2 Acknowledgments

*Statistics.* No statistic analysis or tests were performed.

*Life Science Reporting Summary.* Further information on research design is available in the Nature Research Reporting Summary linked to this article.

*Code availability.* SIRIUS 4 is written in Java, is open source under the GNU General Public License (Version 3), and works on Windows, OS X and Linux. In addition to the graphical front end, a comprehensive command-line version allows batch processing and integration into workflows; integration into GNPS [1], OpenMS [2] and MZmine [4] is ongoing. We also provides source code, executable binaries, documentation, support, example files and additional information on the SIRIUS website (`https://bio.informatik.uni-jena.de/sirius/`); a source copy is hosted on GitHub (`https://github.com/boecker-lab/sirius`).

*Data availability.* Data for the CASMI 2016 reevaluation are available from `https://bio.informatik.uni-jena.de/data` under the Creative Commons CC-BY license. Cross validation

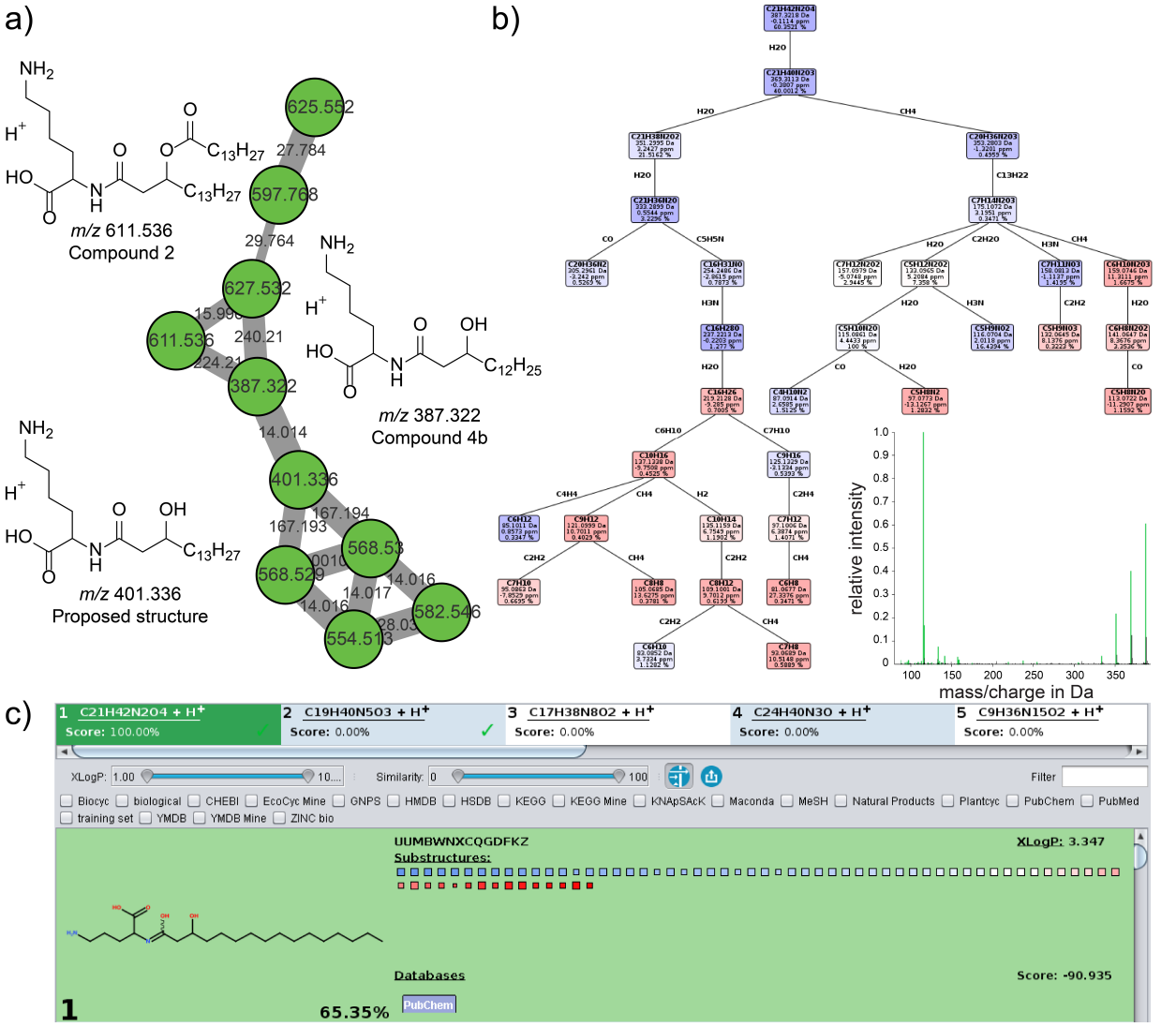


**Fig. 2. In silico annotation of novel N-acyl amide molecules.** (a) A network cluster containing m/z values matching to those of putative N-acyl amide compounds postulated in [20], none of the compounds could be annotated via library search. (b) Fragmentation tree that explains experimentally observed MS/MS fragmentation pattern of the ion with m/z 387.322. (c) Structure of the compound with the highest score, N-3-OH-palmitoyl ornithine, a compound reported in [20] (note enol tautomerism). This structure served as a starting point for annotation of other nodes in the cluster resulting in a discovery of several novel N-acyl amides not described previously using accurate molecular formula predictions.

data for the GNPS search reevaluation are available from `https://bio.informatik.uni-jena.de/data/` (Creative Commons CC0 1.0 Universal license). Data for the American Gut project are available from `https://massive.ucsd.edu/`, data sets MSV000080186 and MSV000080187 (Creative Commons CC0 1.0 Universal license). The analysis can be accessed via `http://gnps.ucsd.edu/ProteoSAFe/status.jsp?task=9bd16822c8d448f59a03e6cc8f017f43` and `http://gnps.ucsd.edu/ProteoSAFe/status.jsp?task=d26ae082b1154f73ac050796fcaa6bda`. Data for the clothing with antibacterial properties study are available at `https://massive.ucsd.edu/`, data set MSV000081379 (Creative Commons CC0 1.0 Universal license). Analysis at `https://gnps.ucsd.edu/ProteoSAFe/status.jsp?task=a5e8ca1b7a9c42cfb45fbb2855e36721`.

*Acknowledgments.* We gratefully acknowledge financial support by the Deutsche Forschungsgemeinschaft (BO 1910/20) and Academy of Finland (310107/MACOME). We thank the GNPS community, S. Stein, and F. Kuhlmann and Agilent Technologies, Inc. (Santa Clara, USA) for providing data that was used to estimate the hyperparameters of SIRIUS 4 and to train CSI:FingerID. We also thank F. Kuhlmann and Agilent Technologies for data used to evaluate the isotope scoring.

*Author Contributions.* K.D., P.C.D., J.R. and S.B. designed the research. K.D., M.F., M.L., J.R. and S.B. developed computational methods. K.D., M.F., M.L. and M.M. implemented computational methods and performed method evaluations, coordinated by S.B.. A.A.A. and A.V.M. performed the biological case studies, coordinated by P.C.D.. S.B. wrote the manuscript, to which K.D., M.F., M.L., A.A.A. and A.V.M. contributed, in cooperation with all authors.

*Competing Financial Interests Statement.* S.B. holds patents (Japanese patent 5559816, US patent 8263931) whose value might be affected by the publication.

## References


1. Wang, M. *et al.* Sharing and community curation of mass spectrometry data with Global Natural Products Social Molecular Networking. *Nat Biotechnol* 34, 828–837 (2016).
2. Röst, H. L. *et al.* OpenMS: a flexible open-source software platform for mass spectrometry data analysis. *Nat Methods* 13, 741–748 (2016).
3. Tsugawa, H. *et al.* MS-DIAL: data-independent MS/MS deconvolution for comprehensive metabolome analysis. *Nat Methods* 12, 523–526 (2015).
4. Pluskal, T., Castillo, S., Villar-Briones, A. & Oresic, M. MZmine 2: Modular framework for processing, visualizing, and analyzing mass spectrometry-based molecular profile data. *BMC Bioinf* 11, 395 (2010).
5. Smith, C. A., Want, E. J., O'Maille, G., Abagyan, R. & Siuzdak, G. XCMS: Processing mass spectrometry data for metabolite profiling using nonlinear peak alignment, matching, and identification. *Anal Chem* 78, 779–787 (2006).
6. Böcker, S., Letzel, M., Lipták, Zs. & Pervukhin, A. SIRIUS: Decomposing isotope patterns for metabolite identification. *Bioinformatics* 25, 218–224 (2009).
7. Böcker, S. & Rasche, F. Towards de novo identification of metabolites by analyzing tandem mass spectra. *Bioinformatics* 24. Proc. of European Conference on Computational Biology (ECCB 2008), I49–I55 (2008).
8. Böcker, S. & Dührkop, K. Fragmentation trees reloaded. *J Cheminform* 8, 5 (2016).
9. Dührkop, K., Shen, H., Meusel, M., Rousu, J. & Böcker, S. Searching molecular structure databases with tandem mass spectra using CSI:FingerID. *Proc Natl Acad Sci U S A* 112, 12580–12585 (2015).
10. Shen, H., Dührkop, K., Böcker, S. & Rousu, J. Metabolite Identification through Multiple Kernel Learning on Fragmentation Trees. *Bioinformatics* 30. Proc. of Intelligent Systems for Molecular Biology (ISMB 2014), i157–i164 (2014).
11. Heinonen, M., Shen, H., Zamboni, N. & Rousu, J. Metabolite identification and molecular fingerprint prediction via machine learning. *Bioinformatics* 28, 2333–2341 (2012).
12. Pirhaji, L. *et al.* Revealing disease-associated pathways by network integration of untargeted metabolomics. *Nat Methods* 13, 770–776 (2016).
13. Hatzimanikatis, V. *et al.* Exploring the diversity of complex metabolic networks. *Bioinformatics* 21, 1603–1609 (2005).
14. Meusel, M. *et al.* Predicting the presence of uncommon elements in unknown biomolecules from isotope patterns. *Anal Chem* 88, 7556–7566 (2016).
15. Ludwig, M., Dührkop, K. & Böcker, S. Bayesian networks for mass spectrometric metabolite identication via molecular fingerprints. *Bioinformatics* 34. Proc. of Intelligent Systems for Molecular Biology (ISMB 2018)., i333–i340 (2018).
16. Kim, S. *et al.* PubChem Substance and Compound databases. *Nucleic Acids Res* 44, D1202– D1213 (2016).
17. Jeffryes, J. G. *et al.* MINEs: open access databases of computationally predicted enzyme promiscuity products for untargeted metabolomics. *J Cheminform* 7, 44 (2015).
18. Schymanski, E. L. *et al.* Critical Assessment of Small Molecule Identification 2016: Automated Methods. *J Cheminf* 9, 22 (2017).
19. Pence, H. E. & Williams, A. ChemSpider: An Online Chemical Information Resource. *J Chem Educ* 87, 1123–1124 (2010).
20. Cohen, L. J. *et al.* Commensal bacteria make GPCR ligands that mimic human signalling molecules. *Nature* 549, 48–53 (2017).
21. Dührkop, K., Ludwig, M., Meusel, M. & Böcker, S. Faster mass decomposition in Proc. of Workshop on Algorithms in Bioinformatics (WABI 2013) 8126 (Springer, Berlin, 2013), 45–58.
22. Böcker, S. & Lipták, Zs. A fast and simple algorithm for the Money Changing Problem. *Algorithmica* 48, 413–432 (2007).
23. Böcker, S., Letzel, M., Lipták, Zs. & Pervukhin, A. Decomposing metabolomic isotope patterns in Proc. of Workshop on Algorithms in Bioinformatics (WABI 2006) 4175 (Springer, Berlin, 2006), 12–23.
24. Rauf, I., Rasche, F., Nicolas, F. & Böcker, S. Finding Maximum Colorful Subtrees in practice. *J Comput Biol* 20, 1–11 (2013).
25. White, W. T. J., Beyer, S., Dührkop, K., Chimani, M. & Böcker, S. Speedy Colorful Subtrees in Proc. of Computing and Combinatorics Conference (COCOON 2015) 9198 (Springer, Berlin, 2015), 310–322.
26. Dührkop, K., Lataretu, M. A., White, W. T. J. & Böcker, S. Heuristic algorithms for the Maximum Colorful Subtree problem in Proc. of Workshop on Algorithms in Bioinformatics (WABI 2018) 113 (Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2018), 23:1–23:14.
27. Senior, J. Partitions and Their Representative Graphs. *Amer J Math* 73, 663–689 (1951).
28. Pluskal, T., Uehara, T. & Yanagida, M. Highly accurate chemical formula prediction tool utilizing high-resolution mass spectra, MS/MS fragmentation, heuristic rules, and isotope pattern matching. *Anal Chem* 84, 4396–4403 (2012).

29. Dührkop, K., Hufsky, F. & Böcker, S. Molecular Formula Identification Using Isotope Pattern Analysis and Calculation of Fragmentation Trees. *Mass Spectrom* 3, S0037 (2014).
30. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* 521, 436–444 (2015).
31. Böcker, S. & Mäkinen, V. Combinatorial Approaches for Mass Spectra Recalibration. *IEEE/ACM Trans Comput Biology Bioinform* 5, 91–100 (2008).
32. Cortes, C., Mohri, M. & Rostamizadeh, A. Algorithms for learning kernels based on centered alignment. *J Mach Learn Res* 13, 795–828 (2012).
33. Shen, H., Szedmak, S., Brouard, C. & Rousu, J. in Discovery Science: 19th International Conference, DS 2016, Bari, Italy, October 19–21, 2016, Proceedings (eds Calders, T., Ceci, M. & Malerba, D.) 427–441 (Springer International Publishing, Cham, 2016).
34. Horai, H. *et al.* MassBank: A public repository for sharing mass spectral data for life sciences. *J Mass Spectrom* 45, 703–714 (2010).
35. Brodley, C. E. & Friedl, M. A. Identifying mislabeled training data. *J Artif Intell Res* 11, 131–167 (1999).
36. Rogers, D. & Hahn, M. Extended-connectivity fingerprints. *J Chem Inf Model* 50, 742–754 (2010).
37. Willighagen, E. L. *et al.* The Chemistry Development Kit (CDK) v2.0: atom typing, depiction, molecular formulas, and substructure searching. *J Cheminf* 9, 33 (2017).
38. Wang, R., Fu, Y. & Lai, L. A New Atom-Additive Method for Calculating Partition Coefficients. *J Chem Inf Comput Sci* 37, 615–621 (1997).
39. Wang, R., Gao, Y. & Lai, L. Calculating partition coefficient by atom-additive method. *Perspect Drug Discovery Des* 19, 47–66 (2000).
40. Steinbeck, C. *et al.* The Chemistry Development Kit (CDK): An Open-Source Java Library for Chemo- and Bioinformatics. *J Chem Inf Comput Sci* 43, 493–500 (2003).

## 3 OnlineMethods

*Running time improvements, algorithm engineering and parallelization.* Over the last decade, numerous algorithmic improvements and algorithmic engineering have resulted in the current fast speed of SIRIUS 4, be it with regards to the decomposition of masses [21, 22], the simulation of isotope patterns [6, 23], or the computation of fragmentation trees [24, 25]. Recall that finding the fragmentation tree that best explains the experimental data is an NP-hard problem [24], which forbids the existence of a polynomial-time algorithm for this problem, unless P = NP.

Beyond this, SIRIUS 4 uses additional algorithmic tricks to further cut down running times:

- If isotope pattern data are available and at least one molecular formula candidate receives a reasonable positive score, then SIRIUS 4 computes fragmentation trees only for those molecular formula candidates with isotope pattern score reasonably close to the best score.
- When interpreting the fragmentation spectrum, at most 60 peaks are considered, sorted by intensity; peaks that have no decomposition are excluded.
- To significantly speed up fragmentation tree computation, SIRIUS 4 proceeds in two rounds: In the first round, it uses a heuristic to get an estimate of the score of a molecular formula candidate. In the second round, only high-scoring molecular formula candidates are processed by an exact algorithm. We ensure that not only the best solution is computed exactly, but also a number of suboptimal solutions; this is necessary, for example, to make sure that for molecular formula candidates selected by soft-thresholding, the optimal fragmentation tree is also computed. Details can be found in [26].
- SIRIUS 4 uses hypothesis-driven recalibration, as described in [8]: For each molecular formula candidate, a fragmentation tree is computed; then, ideal masses from this tree are used to recalibrate the measured mass spectrum. To speed up the recalibration, SIRIUS 4 uses the heuristic fragmentation tree to recalibrate the measured spectrum.
- In case the user provides a "suspect list" of candidate structures, SIRIUS 4 uploads the InChI keys of all suspects to the web service, and retrieves molecular fingerprints for those present in the structure database. Only for suspects without precomputed fingerprints, SIRIUS 4 computes fingerprints locally. As computing thousands of molecular properties for thousands of candidates is very time-consuming, this procedure significantly reduces running times.

Beyond algorithm engineering, SIRIUS 4 now includes a powerful job scheduling system which further speeds up computations. Practically all present-day processors have multiple cores; SIRIUS 4 automatically detects the number of available cores and uses all of them. Whenever SIRIUS 4 wants to execute some step of the analysis pipeline (computation of fragmentation trees, both heuristically and exactly, web service submission, I/O jobs, and candidate scoring) it pushes the corresponding job to the job scheduling system. The job scheduling system decides if the job can be started immediately (for example, a web service submission where the fragmentation tree has been computed), needs to wait for the completion of other jobs, or is time-consuming and needs to wait for a free CPU core. This parallelization is effective across all compounds and analyzed molecular formula candidates of the current workspace. When using one of the commercial ILP solvers, this parallelization is in effect, and much more efficient than the parallelization which is offered as part of these ILP solvers: For example, parallelization through the commercial ILP solver on a 16-core CPU will result in a speedup of, say, 3-fold, whereas the job scheduler reaches a speedup close to the theoretical optimum as long as enough jobs are to be computed. But when using the free GLPK solver, the implementation of the solver prohibits to start several instances simultaneously. Users who rely on the GLPK solver will nevertheless benefit from the new parallelization, as most of the fragmentation trees are computed in parallel by the heuristics, and only few top-ranking trees have to be computed

exactly. The job scheduling system comes with a Job View in the the graphical user interface, see Supplementary Fig. 2. Individual jobs can be canceled through the Job View; dependent jobs will automatically be canceled from the system, too. Similarly, if a job fails then dependent jobs will not be started. The user can view log files for each of the (failed) jobs individually through the Job View. In full, the job scheduling results in a much better workload distribution, in particular on compute clusters with many cores, and is often able to reach CPU load beyond 90 %.

SIRIUS 4 now supports the CPLEX and Gurobi ILP solvers. To allow SIRIUS to be run without installing a third-party Integer Linear Programming solver, we have integrated the free and open-source GLPK (GNU Linear Programming Kit, `https://www.gnu.org/software/glpk`) solver. Alternatively, you can use one of the commercial ILP solvers Gurobi (Gurobi Optimization, Inc., Houston, USA; `http://www.gurobi.com`) and CPLEX (IBM, Armonk, USA; `https://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/`). Both are free-to-use for university members, and result in a speedup of up to tenfold for the time-intense exact fragmentation tree computation.

*Molecular formula determination.* SIRIUS 4 considers *all* molecular formulas that can possibly explain a certain monoisotopic mass. SENIOR rules [27] are used to filter out chemically infeasible molecular formulas; beyond this, no filtering is performed. SIRIUS 4 penalizes molecular formulas which are unlikely to correspond to a biomolecule using a linear Support Vector Machine. SIRIUS 4 *never discards* chemically feasible molecular formulas, and *never rewards* molecular formulas. SIRIUS 4 also includes a list of "outlier molecular formulas" which are atypical for a biomolecule but which are not penalized, such as $C_{10}HF_{19}O_2$ for perfluorodecanoic acid. SIRIUS 4 can consider numerous ion types; for ion type "unknown", SIRIUS will consider ion types protonation, sodium, and potassium for positive ion mode, and deprotonation and chlorine for negative ion mode. For a combined search with CSI:FingerID, SIRIUS 4 can be restricted to molecular formulas that are found in a structure database such as PubChem. See Supplementary Note 3 for details.

SIRIUS 4 uses the Maximum A Posteriori Estimation from [8] for the computation of fragmentation trees. Compared to SIRIUS versions before version 3.0, this resulted in massive improvements in molecular formula identification: Evaluations in [8] (Fig. 4) show that molecular formula identification rates practically double, compared to earlier versions, when relying solely on MS/MS data. Böcker and Dührkop [8] argue that the improved performance is due to an increase in fragmentation tree structural quality. High fragmentation tree quality is, in turn, required for a successful structural identification via CSI:FingerID.

*Isotope pattern analysis.* SIRIUS 4 implements a new scoring for comparing experimental and simulated (ideal, theoretical) isotope pattern. Mass deviations and mass difference deviations are still modeled by normal distributions [6]. For intensities, previous models only consider either relative errors [6] or absolute errors [28], but cannot deal with both types simultaneously. The model from [29] tries to overcome this limitation, but is not intuitive and requires many parameters.

To simultaneously model absolute and relative intensity deviations, we propose a simple maximum likelihood estimator that requires only two parameters, namely, absolute error $\sigma_{\text{abs}} > 0$ and relative error $\sigma_{\text{rel}} > 0$ of peak intensities. We statistically model the intensity error as

$$Y = x + D + E$$

where $x$ is the expected (theoretical) intensity, $Y$ is the random variable modeling the observed intensity, and $D, E$ are random variables for relative and absolute noise, respectively. We assume that both relative noise $D \sim x \cdot \mathcal{N}(0, \sigma_{\text{rel}}^2) = \mathcal{N}(0, x^2\sigma_{\text{rel}}^2)$ and absolute noise $E \sim \mathcal{N}(0, \sigma_{\text{abs}}^2)$ are

normally distributed: In detail, we assume that $D, E$ have densities

$$f_D(\delta) = \frac{1}{\sqrt{2\pi x^2 \sigma_{\text{rel}}^2}} \exp\left(-\frac{\delta^2}{2x^2\sigma_{\text{rel}}^2}\right) \quad \text{and} \quad f_E(\epsilon) = \frac{1}{\sqrt{2\pi\sigma_{\text{abs}}^2}} \exp\left(-\frac{\epsilon^2}{2\sigma_{\text{abs}}^2}\right)$$

for $\delta, \epsilon \in \mathbb{R}$. We are using the probability density function to estimate these probabilities, which can be interpreted as the limit of an arbitrary small interval around the values $\delta, \epsilon$, respectively. Note that this model can result in negative observed peak intensities; we found that this limitation is not relevant in application, where relatively weak noise is observed.

We further assume that relative and absolute noise are independent. Given an observed intensity $y$ and an expected intensity $x$, the likelihood of some model $\theta = (\delta, \epsilon)$ is

$$\mathcal{L}_{y|x}(\delta, \epsilon) = \frac{1}{\sqrt{2\pi x \sigma_{\text{rel}}}} \exp\left(\frac{-\delta^2}{2x^2\sigma_{\text{rel}}^2}\right) \cdot \frac{1}{\sqrt{2\pi\sigma_{\text{abs}}}} \exp\left(\frac{-\epsilon^2}{2\sigma_{\text{abs}}^2}\right) = \frac{1}{2\pi x \sigma_{\text{rel}}\sigma_{\text{abs}}} \exp\left(-\frac{\delta^2}{2x^2\sigma_{\text{rel}}^2} - \frac{\epsilon^2}{2\sigma_{\text{abs}}^2}\right). \quad (1)$$

We find that the Maximum Likelihood $\mathcal{L}_{y|x}(\delta_0, \epsilon_0)$ for model $(\delta_0, \epsilon_0)$ is

$$\mathcal{L}_{y|x}(\delta_0, \epsilon_0) = \frac{1}{2\pi x \sigma_{\text{rel}}\sigma_{\text{abs}}} \exp\left(-\frac{(y-x)^2}{2(\sigma_{\text{abs}}^2 + x^2\sigma_{\text{rel}}^2)}\right) \quad (2)$$

(Supplementary Note 4). But this implies that for the computation of the maximum likelihood, we actually do not have to compute $\delta_0$ or $\epsilon_0$; instead, it is sufficient to directly insert theoretical intensity $x$ and observed intensity $y$ into equation (2) to estimate the maximum likelihood of the data. We note that (2) is very similar to the probability density function of the random variable $D + E \sim \mathcal{N}(0, \sigma_{\text{abs}}^2 + x^2\sigma_{\text{rel}}^2)$.

The above model is mathematically sound, but has a conceptual disadvantage when scoring a set of candidate isotope patterns against one measured isotope pattern: The relative noise depends on the peak intensity in the theoretical (candidate) isotope pattern and, hence, each candidate pattern is scored differently. To this end, we have exchanged the role of $x$ and $y$ (expected/theoretical intensity vs. observed intensity) in our implementation of the scoring for SIRIUS 4. Likelihoods computed in this way are usually very large, positive values. SIRIUS 4 uses log odds, dividing values through the likelihood for $\delta_0 = 2\sigma_{\text{rel}}$ and $\epsilon_0 = 2\sigma_{\text{abs}}$. By default, SIRIUS 4 uses parameters $\sigma_{\text{abs}} = 0.01$ and $\sigma_{\text{rel}} = 0.08$.

For a candidate molecular formula, we simulate an isotope pattern with peak intensities and mean peak masses as described in [6,23]. We normalize both spectra using the first isotope peak. Assuming statistical independence [6,23], the likelihood of the candidate molecular formula is simply the product of the individual likelihoods for peak mass differences (modeled by normal distributions) and peak intensity differences (modeled above). For peaks which are not observed but expected in the theoretical pattern, we estimate no likelihood for mass deviation, but a likelihood for intensity deviation where the observed intensity is set to zero.

Isotope pattern are extracted from MS1 by searching for the most intensive peak around the precursor mass within the allowed mass deviation; then, we extend the pattern gradually by picking the next isotope peak that has a reasonable mass difference. If several such peaks exist, they are merged, using the weighted mean of their masses and the sum of their intensities. To prevent that we include a peak into the pattern which is not part of the isotope pattern (for example, a coeluting ion), we compute scores for all possible lengths of the patterns by successively removing the last peak of the pattern, and only report the maximum score.

*Isotope pattern analysis for MS/MS data.* In some experimental setups (All Ion Fragmentation etc.), isotope peaks and fragment peaks are measured together in the same spectrum. For such experiments, SIRIUS 4 offers a combined isotope and fragmentation pattern analysis. See Supplementary Note 5 for details.

*Automated detection of uncommon elements.* SIRIUS 4 integrates automated detection of "uncommon elements" from the isotope pattern of the query compound. These elements are added to the standard set of elements CHNOP when determining the molecular formula of the query compound. Even if a particular element is not excluded at this step, the subsequent analysis can still choose a molecular formula which contains zero atoms of this elements. Isotope pattern analysis can be used to exclude elements sulfur, chlorine, bromine, boron, selenium, as these have characteristic isotope patterns. There is also a predictor for silicon which is disabled by default, as it results in a relatively large number of false positives. Different from [14], we use a Deep Neural Networks [30] (DNN) for this task. By this, we reduce the memory requirements of the uncommon element prediction from more than 200 MB for Random Forests [14] to 75 KB for the DNN. See Supplementary Note 6 for details.

*SIRIUS 4 step-by-step computational workflow.* We now describe in detail the steps that SIRIUS 4 performs when analyzing the data from a single compound. For each compound, SIRIUS 4 inputs an MS/MS spectrum and, optionally, an isotope pattern in MS1. SIRIUS 4 aims to identify the molecular formula of the query compound and annotate the MS/MS spectrum with a fragmentation tree. Dührkop *et al.* [26] noted that fragmentation trees computed by heuristics can have structures that deviate notably from that of the optimal solution; to this end, we want to ensure that all fragmentation trees of the top-scoring candidate molecular formulas are computed exactly. This is based on the general assumption that the structure of the optimum solution is closer to the "true" structure than some arbitrary suboptimal solution.

In detail, SIRIUS 4 proceeds as follows:

1. If an isotope pattern is provided, we use the Deep Neural Network from Supplementary Note 6 to restrict the set of elements used.
2. Enumerate over all molecular formulas that explain the precursor ion peak, or (if provided) the monoisotopic peak of the isotope pattern using the Round Robin algorithm [21,22].
3. If an isotope pattern is given, score each of these molecular formulas using the Maximum Likelihood scoring described above. In case at least one isotope pattern receives a reasonable score, discard all molecular formula candidates with very low score.
4. For each molecular formula candidate we do the following: Create a fragmentation graph from the MS/MS data rooted in the molecular formula candidate. Edge weights in this graph are chosen as described in [8]. Different from there, we use the Deep Neural Network from Supplementary Note 3 to penalize outlier molecular formulas. If a MS1 isotope pattern is given, we modify the weight of the root by maximum likelihood score from the isotope pattern analysis. For the nodes of the graph, we decompose the sixty most intense peaks in the MS/MS spectrum, enumerating all molecular formula explanations.
5. If in-source fragments are detected in the MS1, we use their isotope score to modify the score of the corresponding nodes in the fragmentation graph.
6. For each molecular formula candidate and corresponding fragmentation graph, we compute a fragmentation tree (colorful subtree) using the Critical Path$^3$ heuristic from [26].
7. We sort the molecular formula candidates according to their scores.
8. For the top $k$ molecular formula candidate and corresponding fragmentation trees:
   (a) We use hypothesis-driven recalibration [8,31] to recalibrate the MS/MS spectrum, see [8].
   (b) We compute the fragmentation graph and edge weights as described above, but using the recalibrated MS/MS spectrum.
   (c) We compute the optimum fragmentation tree (maximum colorful subtree) using an exact algorithm; namely, the Integer Linear Program from [24].

9. We sort these $k$ fragmentation trees according to their new scores; top scoring trees are reported to the user.

By default, SIRIUS 4 computes $k = 20$ fragmentation trees exactly.

*Integrating CSI:FingerID as a web service.* The integration of CSI:FingerID into SIRIUS is realized using a RESTful (representational state transfer) web service via HTTPS, and implemented as a java servlet. The deliberately simple client-server architecture enables high performance, reliability, and scalability. Running CSI:FingerID as a web service avoids unnecessary maintenance, incompatibilities through upgrading, and does not require third-party libraries on the user side. Furthermore, CSI:FingerID is in a stage of rapid methodical progress, and the chosen architecture allows us to continuously integrate methodical upgrades without the user having to install new releases. No spectral libraries or compound structure databases have to be installed or updated on the user side; additional training data and structure databases can also be integrated continuously, without requiring upgrades. This is particularly relevant for integrating spectral libraries which are freely available for training computational methods, but not for download.

The workflow of integrating CSI:FingerID and SIRIUS is depicted in Fig. 1a. Fragmentation tree and spectrum are uploaded from the SIRIUS client to the web server, and temporarily stored in a relational job database. A compute cluster is regularly fetching jobs from the database, predicting fingerprints and storing results in the database. The SIRIUS client downloads these predicted fingerprints as well as a list of candidate structures and fingerprints via the REST application programming interface. Processing a single compound instance is performed by the web service in less than a second of wall-clock time, enabling an interactive analysis of the data without disturbing delays. Scoring, ranking and visualization of the candidate structures is done locally on the client side.

The strict separation between web interface and compute cluster allows us to easily scale the web service: In times of high workload (that is, when more than 50 jobs are stored in the job database), additional compute nodes are spawned on the compute cluster. Each node that has access to the job database, can become a compute node.

To ensure that the CSI:FingerID web service is available at all times, we have installed the complete setup on two independent physical machines. The two physical machines are permanently monitored; if one machine is no longer responding to requests of the CSI:FingerID web API, the second machine takes over. Furthermore, we now give users a time window to upgrade SIRIUS installations: Previously, any change to the web API required that all users immediately upgraded their SIRIUS installations, as the old web API was no longer responding. Starting with SIRIUS 4, we can run several versions of the CSI:FingerID API in parallel. For future updates, the outdated version will be working for at least a week, to give users time for upgrading.

*Negative ion mode support for CSI:FingerID.* Until recently, CSI:FingerID did not support the analysis of negative ion mode spectra; this was due to the lack of publicly available training data. With the integration of negative ion mode spectra from NIST (see below), CSI:FingerID 1.1 can now analyze negative ion mode spectra. Positive and negative ion mode spectra are trained separately. With the exception of the multiple kernel learning weights (Supplementary Note 7), integrating negative ion mode did not require any changes to the CSI:FingerID method.

*Novel kernels for CSI:FingerID.* We integrated ten novel kernels into the Support Vector Machine used for predicting fingerprints, and removed old ones that did not contribute to the search performance of CSI:FingerID. CSI:FingerID 1.1 uses the kernels described in Supplementary Table 1;

kernel weights are computed using the ALIGNF [32] and ALIGNF+ [33] multiple kernel learning methods.

*Additional training data for CSI:FingerID.* We have integrated new training data into the CSI:FingerID web service. Next to spectra from MassBank [34] and GNPS [1] we trained CSI:FingerID on 16,858 compounds from the NIST 2017 database (National Institute of Standards and Technology, v17). In full, CSI:FingerID 1.1 is trained on 16,083 structures, with 19,118 independent MS/MS measurements (compounds) in positive mode and 10,823 measurements in negative mode. We observe that the prediction performance of CSI:FingerID decreases when trained on all available spectra from GNPS and MassBank. Therefore, we train CSI:FingerID in a two-step approach removing outliers [35], using a 10-fold cross validation. In this way, we discard 757 structures. See Supplementary Note 8 for details.

*Novel molecular properties for CSI:FingerID.* For CSI:FingerID 1.1, we have added Extended Connectivity Fingerprints (ECFP) to the list of predicted molecular properties [36]. We use molecular properties from ECFP6 that are found sufficiently often in the training structures. Molecular properties are computed using the Chemistry Development Kit [37]. In total, 7,593 molecular properties are available for learning. We find that 2,937 molecular properties in positive and 1,996 in negative ion mode can be learned sufficiently well from the training data. See Supplementary Note 9 for details.

*Candidate retrieval and adduct types for CSI:FingerID.* SIRIUS 4 considers all structure candidates from the selected structure database which agree with the molecular formula of the parent ion determined in the previous analysis step. To prevent that candidates with slightly suboptimal molecular formula are not considered in this search, SIRIUS 4 uses a *soft threshold*: All molecular formula candidates of the parent ion with score above 0.75 of the optimal score are considered.

When retrieving candidates, SIRIUS 4 can consider numerous adduct types as well as in-source fragments: For example, $[M + H]^+$ for protonation, $[M]^+$ for intrinsically charged molecules, $[M-H_2O + H]^+$ for in-source fragments, and $[M + NH_4]^+$, $[M + CH_4O]^+$, $[M + C_2H_3N]^+$, $[M + C_4H_6N_2]^+$, etc. for adducts. Note that "ion types" (protonation, sodium, potassium, deprotonation, chlorine) have already been selected for molecular formula determination (Supplementary Note 3). In the graphical user interface, the user can select or unselect all of these adduct types and in-source fragments individually. For example, if SIRIUS 4 determined the molecular formula of the parent ion to be $C_{17}H_{25}N_2O_4{}^+$ and adduct type $[M + NH_4]^+$ (ammonium adduct) is selected, then candidate structures with molecular formula $C_{17}H_{21}NO_4$ are retrieved from the structure database; the union of these candidate structures is scored against the predicted fingerprint, sorted and displayed by SIRIUS 4. Furthermore, SIRIUS 4 can consider user-defined adduct types.

*Tree-based Maximum A Posteriori Estimation as a CSI:FingerID score.* For searching in molecular structure databases, we have integrated a posterior probability score [15] which outperforms both all scores from the original CSI:FingerID release [9]. The novel score models dependencies between molecular properties when scoring candidate structures. See Supplementary Note 10 and [15] for details.

*Structure databases.* We have updated the PubChem database used for searching; the current version (downloaded August 13, 2017) contains 93,859,798 compounds and 73,444,774 unique structures, constituting a 77.3 % increase in compounds and a 80.0 % increase in structures over [9]. As our *biomolecule structure database*, we combine compound structures from numerous public databases (Supplementary Table 2). The combined database contains 492,921 unique structures of biomolecules

and compounds that can be expected in biological samples. Furthermore, CSI:FingerID allows to search the MINE databases [17] (EcoCyc, YMDB, KEGG) of 651,824 *in silico* generated structures. Finally, users can provide a "suspect list" of structures as a text file of InChI strings, SMILES, or as SDF or MDL files. This suspect list can be searched in addition to, say, the biomolecule structure database. The numbers reported above represent only snapshots, as structure databases are continuously updated and extended.

*Graphical User Interface.* A usual analysis proceeds as follows (Supplementary Fig. 1): Mass spectrometry data can be loaded into SIRIUS 4 using drag-drop of one or more files. In the first step, isotope pattern and MS/MS data are used to determine the molecular formula of the compound, and to compute a fragmentation tree. In this step, SIRIUS 4 will usually not limit its computations to molecular formulas present in any database, but considers all molecular formulas that can possibly explain the observed precursor mass. Molecular formula candidates and their scores (isotope pattern, fragmentation pattern and total) are displayed in the *SIRIUS Overview* tab. For the total score, SIRIUS 4 reports the posterior probability of the molecular formula given the data; this *must not be mistaken for the probability that the molecular formula is correct.* A blue line indicates the soft score threshold; by default, only molecular formulas with score above this threshold are used for searching with CSI:FingerID. Two tabs allow an in-depth inspection of the spectra data, and the fragmentation tree for each molecular formula candidate.

The user can then initiate a molecular structure search: The corresponding fragmentation trees and spectra are uploaded to the CSI:FingerID web server, results are retrieved and, finally, displayed in the user interface. By default, SIRIUS 4 selects one or more molecular formulas for the downstream analysis based on a soft threshold, but the user can override this choice. The *CSI:FingerID Overview* tab summarizes results over all molecular formulas. As the predicted fingerprint differs depending on the candidate molecular formula, the *CSI:FingerID Details* tab allows to browse through molecular formula candidates individually, where for the best-scoring candidate, the molecular formula is highlighted in green. The tab also shows what predicted molecular features (presence or absence of certain substructures) support a particular candidate structure, and what predicted molecular features contradict that candidate. True positive predictions (molecular properties which are presented in the candidate, in agreement with the predicted fingerprint, blue) and false negative predictions (molecular properties which are presented in the candidate but not in the predicted fingerprint, red) are displayed as boxes. The size of a box indicates the prediction quality ($F_1$ score) for this molecular property *in cross validation*, whereas color and shade indicate how sure CSI:FingerID is about the prediction for this particular compound (Platt probability). A mouse-over displays the SMARTS string encoding the molecular property; clicking the square highlights the corresponding substructure in the candidate structure. Results can be filtered based on XlogP values [38, 39] that are predicted using the Chemistry Development Kit [37, 40]. SIRIUS 4 also displays all databases where each structure candidate can be found; clicking the database name directly opens the corresponding website in the browser.

Finally, the *Predicted Fingerprint* tab allows the user to view the fingerprint that was predicted by CSI:FingerID for the query compound, for each molecular formula candidate (Fig. 1 and Supplementary Fig. 5). For any property, the user interface displays graphical examples of the predicted substructures. Conceptually, fingerprint prediction is executed before searching in a structure database, and it is *not required that the query compound is contained in any structure database* to predict this fingerprint. In fact, predicted molecular properties can be used to hypothesize about the structure of a compound not present in any database.

The user interface offers additional filtering options, but also full text search for every output list to, say, search for a particular InChI string. See the SIRIUS manual (available from `https://bio.informatik.uni-jena.de/software/sirius`) for details.