BEILSTEIN ARCHIVES

This open access document is posted as a preprint in the Beilstein Archives at https://doi.org/10.3762/bxiv.2026.45.v1 and is considered to be an early communication for feedback before peer review. Before citing this document, please check if a final, peer-reviewed version has been published.

This document is not formatted, has not undergone copyediting or typesetting, and may contain errors, unsubstantiated scientific claims or preliminary data.

Preprint Title	*checkHNMR*: an open-source tool for checking the consistency of ^{1}H NMR descriptions before publication
Authors	Artur V. Eshtukov-Shcheglov
Publication Date	13 Nov. 2026
Article Type	Full Research Paper
Supporting Information File 1	checkHNMR_1_9445-9C84-5432.json; 13.5 KB
Supporting Information File 2	checkHNMR_Supporting_Information.pdf; 191.0 KB
Supporting Information File 3	methyl_findings_JOC_V2.1.3-beta.xlsx; 25.5 KB
Supporting Information File 4	records_JOC_development_descriptions.xlsx; 1.3 MB
ORCID® iDs	Artur V. Eshtukov-Shcheglov - https://orcid.org/0000-0002-8975-9958

License and Terms: This document is copyright 2026 the Author(s); licensee Beilstein-Institut.

This is an open access work under the terms of the Creative Commons Attribution License (https://creativecommons.org/licenses/by/4.0). Please note that the reuse, redistribution and reproduction in particular requires that the author(s) and source are credited and that individual graphics may be subject to special legal provisions. The license is subject to the Beilstein Archives terms and conditions: https://www.beilstein-archives.org/xiv/terms.
The definitive version of this work can be found at https://doi.org/10.3762/bxiv.2026.45.v1

checkHNMR: an open-source tool for checking the consistency of ^{1}H NMR descriptions before publication

*Artur V. Eshtukov-Shcheglov**

Independent Researcher, Moscow, Russian Federation

*Corresponding Author

E-mail: nemtcev.94.nmr@gmail.com

Abstract: A ^{1}H NMR description in the Supporting Information is usually the only place where the values of a spectrum are stated. Such descriptions serve as models for students and researchers and as a reference for reviewers and editors, and they are used by databases, prediction programs and language models, so an error that passes peer review spreads. Here I describe *checkHNMR*, an open-source, rule-based tool that checks before publication whether the values of a ^{1}H NMR description of a small organic molecule agree with one another and with the proposed structure. It applies the items of a published checklist that can be checked from the text and the structure alone: the number of protons, a partner for every coupling constant, multiplicities consistent with the number of constants and of possible neighbors, and first-order notation only where it is valid. Each alert states its reason, and the decision is left to the author, the referee or the editor. The tool runs in a web browser at checkhnmr.org, where the data are not sent anywhere, and as a local version that reads Supporting Information files as published; the two versions give the same results. Each report carries a verification code. A reviewer can repeat a check by entering the entries saved with the report into the same version of the tool and comparing the code with the one in the report. The paper describes this procedure step by step, with an example. In stress tests with synthetic descriptions, the tool gave no false findings on correct descriptions, found most of the introduced errors, and misread 0.03% of descriptions written in the styles of published ones; these tests do not measure accuracy on new articles. A description that passes every check can still be wrong or invented: only the raw data (the free induction decay, FID) can show what was measured.

Keywords: ^{1}H NMR spectroscopy; data quality; open-source software; organic chemistry; peer review

Introduction

A ^{1}H NMR description (the line of shifts, multiplicities, coupling constants and integrals in the Supporting Information) is usually the only place where the values of a spectrum are stated. The spectrum itself typically reaches the reader as an image, often reduced to fit a page, from which coupling constants cannot be read. Raw data are rarely shared: in a survey of 240 papers from 12 organic chemistry journals, only 3 of the 223 papers that reported NMR data shared primary NMR data (as MestReNova or JCAMP-DX files), and none provided the raw FID data [1]. In 2026, such descriptions are written, read and checked by software as well as by chemists: processing programs and language models produce them, databases and machine-learning models are built from them, and AI systems are used to look for errors in papers.

Published descriptions are also how students and young researchers learn to describe spectra. A description that has passed peer review reads as an accepted model, and its errors are reproduced in the next manuscript. Such errors do pass peer review: in my earlier work, I analyzed six retracted papers in which the NMR analysis led to incorrect structures, and a paper in which the structure is correct but its spectrum is described incorrectly [2].

Software learns from the same descriptions. Prediction programs trained on published data inherit the habits of published descriptions, including their errors [2]. Collections of literature NMR data have to filter them out: NMRexp, built from nearly 200,000 Supporting Information documents, discarded about 10% of the extracted entries as malformed or non-standard (for example, "d, *J* = 1H" or "*J* = 7..2 Hz") and encouraged "careful checking for numeric consistency and completeness of NMR parameters" [3]; the authors of a shift-prediction model trained on these data note that "the literature data may contain reporting errors or typos" [4]. Validated reference sets are built for the same reason: the authors of one such set of experimental NMR parameters

note that typographical and assignment errors in reported chemical shifts are "relatively common" and that, for ^{1}H–^{1}H coupling constants, "the multiplicity, individual *J* values and/or their assignments are often not reported at all" [5]. Language models reason about NMR spectra unreliably: in ChemBench, predicting the number of signals in a ^{1}H NMR spectrum from a SMILES string proved difficult even for the best models (22% correct answers for o1-preview) [6], and in a test of structure elucidation from ^{1}H and ^{13}C peak lists, the best language model gave the correct structure as its first answer in 28% of cases [7]. AI systems that check papers for errors are not yet reliable either: on a benchmark of 83 published papers with 91 errors that had led to errata or retractions, no model found on average more than 21% of the errors in a single run, and no more than 6% of the problems a model reported matched a known error [8]; in the words of a *Nature* report, such tools "seem to be better suited to drawing human attention to possible problems, rather than deciding themselves that a paper is wrong" [9]. A check of a description should therefore give the same answer every time, be open to inspection, and leave the final decision to a person.

Crystallography has had such a check for many years. *checkCIF*, provided by the International Union of Crystallography, checks crystal structure reports for completeness and internal consistency and returns alerts, which authors are expected to act on before submission [10]. For other characterization data, Christmann checked the high-resolution mass spectrometry data in 3,028 Supporting Information files with a script and found that only about 40% of the files with readable accurate-mass data were internally consistent [11]. Kutateladze and co-workers test proposed structures against published chemical shifts and coupling constants by computation, both for natural products [12,13] and for the products of synthetic reactions [14]. Similar checks of descriptions are already used to clean databases: NMRexp, for example, required the reported

number of protons to match the molecule (with *OH* and *NH* protons optional in $CDCl_3$) and excluded entries with ^1H–^1H coupling constants greater than 30 Hz in molecules composed of C, H, O and N [3]. Such checks serve to filter a database. I am not aware of an open tool that applies them, together with checks of multiplicities and coupling constants, to a description before publication.

How often published descriptions are inconsistent, and whether such inconsistencies are occasional or systematic, is the subject of a separate study that is in progress. That question concerns work already published. A tool can act earlier, before publication, while a description can still be corrected. Here I describe *checkHNMR*, an open-source tool that checks a ^1H NMR description against items 1–7 of the checklist proposed in my earlier work [2], as far as they can be checked from the description and the structure alone. It runs in a web browser at https://checkhnmr.org without sending data anywhere, and as a local version for larger sets; both give identical results. The tool is rule-based: it gives the same answer for the same input, shows the reason for every alert, and leaves the decision to the author, the referee or the editor.

Results and Discussion

This section first describes what *checkHNMR* checks and what it does not, with the reasons for each decision. It then describes how the tool is built, both as a local version and as a website, and what the report shows. The last two parts describe how the tool was developed and tested and the limits of what such a check can show.

What is checked and what is not

checkHNMR applies items 1–7 of the checklist proposed in my earlier work [2], with the criteria defined in a registered protocol [15], except where noted below. It checks a description as published: whether its values agree with one another and with the spin system of the proposed

structure. Apparent typing errors are not corrected, and signals are not assigned to protons. The input is the ^{1}H NMR description and, optionally, the structure as a SMILES string, the ^{19}F and ^{31}P NMR descriptions recorded without proton decoupling, and the HRMS line. Checks that need the structure are made only when a structure is available: a SMILES string or, in the local version, a compound name converted with OPSIN. For item 1, the neutral molecular formula from the HRMS line can be used instead, if it reproduces the reported calculated m/z. Equivalent notations are accepted (for example, br s, bs and brs; app, pseudo and "~t"; sept and hept). Table 1 lists what is checked for each item and why.

Item 1 is checked in part. The sum of the reported integrals is compared with the number of protons in the structure or in the molecular formula. Integrals that match one half, a third or a quarter of a symmetric molecule are reported for a person to check, with a note on the common factor. If the difference does not exceed the number of exchangeable protons (*NH*, *OH*, *SH*, CO_2H) and the description does not state that they were not observed, the finding is reported separately and is not counted as an inconsistency; in protic deuterated solvents such as CD_3OD or D_2O, no statement is needed. With a formula alone, exchangeable protons cannot be identified. The other half of item 1, whether every signal has a place in the structure, needs an assignment and is not checked.

Item 2 asks whether the chemical shifts are consistent with the structure. *The current version does not check it: it contains no reference data on chemical shifts*. A check would need, for each proton of the structure, the range of shifts expected for its environment. Such ranges can be obtained with HOSE codes [16], which describe the environment of an atom sphere by sphere. In the open database NMRShiftDB, the shifts stored for atoms with the same HOSE code give an estimate of the shift and, when ten or more values are found, its range [17]; the database now

provides such predictions for ^{1}H as well as ^{13}C, and an unusually wide range has been used to detect a wrong assignment in a published paper [18]. Without an assignment, a description could then be checked in two directions: a proton of the structure with no reported signal in its range, and a reported signal outside the ranges of all protons. Because such databases are themselves built from published assignments, and a shift outside the usual range can have a structural reason, such as the anisotropy of a nearby aromatic ring or an intramolecular hydrogen bond, these findings would be reported for a person to check and not counted as inconsistencies. This check is planned for the release version.

Item 3 is checked for H–H couplings. Each reported coupling constant must appear in the signal of another proton. The two values are measured independently, each from the line spacings of its own multiplet, so they can differ even in a correct description. Following the protocol, two constants are taken as the same coupling if they differ by no more than the larger of 0.3 Hz and the step in which they are reported (0.5 Hz for values given as 7.0 and 7.5; 1 Hz for integers); 0.3 Hz is about the resolution of a routine ^{1}H spectrum with an acquisition time (AQ) of 3–4 s [19,20]. A constant without a partner is reported for a person to check if the description contains a multiplet or a broad signal without constants, which may hide the partner. A constant larger than 25 Hz cannot be an H–H coupling and is reported as impossible if the compound contains neither fluorine nor phosphorus nor another nucleus that can give a large coupling to ^{1}H, for example ^{11}B or ^{103}Rh, or ^{119}Sn and ^{195}Pt, whose couplings are read from satellites. For H–F and H–P couplings, only the presence of a splitting in the ^{19}F or ^{31}P description is checked, because the resolution of these spectra cannot be judged from the text. In a compound that contains fluorine or phosphorus, if the ^{19}F or ^{31}P description is absent, a constant without a ^{1}H partner is reported as a likely inconsistency for a person to check, because it cannot be distinguished from an H–F or H–P coupling.

Item 4(a) needs only the text: the number of coupling constants must match the multiplicity (one for d, t or q; two for dd, dt or td; three for ddd), so “dd, J = 8.0 Hz” and “d, 1H” without a constant are mismatches. Item 4(b) asks whether the multiplicity needs more coupling partners than the structure provides. Without an assignment, the tool does not know which proton a signal belongs to, so it checks a weaker, necessary condition: whether any proton of the structure, of a kind that fits the shift, has enough possible partners. Partners are counted generously from the number of bonds between the nuclei, with longer paths allowed for small constants (*meta*, allylic and W-type couplings) [21] and for ^{19}F and ^{31}P. A multiplicity that no proton of the structure can show is reported: for example, a tt in an unsymmetrically 1,2-disubstituted benzene, a four-spin system (ABCD, or close to AMPX at high field) in which each proton has only one meta neighbor. A multiplicity that is impossible for the proton it belongs to, but possible for another proton of the molecule, is not found: a triplet reported for an OCH_3 group passes if the molecule also contains an ethyl group. Signals reported as “m”, “app” or “br” are not checked under 4(b), and a multiplicity showing fewer partners than the structure provides is not a finding, because a small coupling can remain unresolved. Since V2.1.3-beta, item 4(b) also has a lower bound for methyl groups. A CH_3 group bonded to a carbon that carries a proton is split by a vicinal coupling of about 7 Hz and is not expected to give a singlet. Without an assignment, the tool checks whether each singlet of 3, 6, 9 … H below 4.5 ppm can be matched to a different group of the structure that can give a singlet, such as a CH_3 group on a quaternary or aromatic carbon, on a carbonyl group or on a heteroatom. If not, and the structure has a methyl group that must be split, the singlet is reported for a person to check. Diastereotopic methyl groups ($SiMe_2$, acetonides), which may give two singlets, are allowed for. Broad singlets are not counted, and neither are methyl groups next to a

CH or CH_2 bonded to silicon (TIPS, TES), which are often reported as one singlet with the CH or CH_2 protons; a singlet of 6 H or more may also come from the CH_2 groups of a chain.

Item 5 compares the reported couplings with the resolution of the spectrum, judged from the line width of the residual solvent signal: *a coupling smaller than the line width cannot have been measured, and a coupling that should be resolved but is missing needs an explanation.* Like item 2, item 5 is not checked in the current version, because the line width is not stated in a description. In the release version, the full width at half maximum (FWHM) will be accepted as an optional input, either for the residual solvent signal or for each signal of the description; every reported coupling constant can then be compared with the line width of its own signal.

Item 6 is checked in two ways. (a) Some spin systems are not first-order at any field, because chemically equivalent protons couple differently to the same nucleus [19,20,22,23]; examples are the AA′XX′ systems of para-disubstituted benzenes and 4-substituted pyridines and the AA′BB′C system of a monosubstituted phenyl group. The tool finds such rings in the structure. If every aromatic proton of the molecule belongs to them, an aromatic signal reported with a first-order multiplicity and a coupling constant can only come from such a ring and is counted as inconsistent. If other protons can also appear in the aromatic region, the signals are listed for a person to check. CH_2–CH_2 fragments of saturated rings, such as those of piperidines, piperazines, morpholines, pyrrolidines, tetrahydrofurans and tetralins, are not first-order either: although fast ring inversion averages the axial and equatorial positions, a vicinal coupling constant depends on the dihedral angle [24,25], so each proton remains coupled to the two protons of the neighboring CH_2 group with two different mean constants, which makes the system AA′XX′. CH_2 signals reported with coupling constants in a molecule with such a fragment are listed for checking, because the tool cannot tell whether a signal belongs to the ring or to an open chain. Open-chain X–CH_2–CH_2–Y

fragments, which the protocol records in a separate subcategory, are not checked in the current version. (b) For two signals that share a coupling constant, the separation Δν in hertz is calculated from the shifts and the spectrometer frequency; for a signal reported as a range, the midpoint is used. If $\Delta\nu/J < 6$, first-order notation for the pair is inconsistent; between 6 and 8, the pair is reported as borderline; at 8 or more, it is accepted [22,23,26]. If the frequency is not stated, the pair cannot be assessed, and the report notes this. Isolated AB systems are exempt, because in them the line spacing equals J at any $\Delta\nu/J$, and so are signals that the author has described together as an AB system. Protons coupled to two strongly coupled ^{19}F nuclei ($\Delta\nu/J_{F–F} < 8$) are the X part of an ABX-type system, and first-order notation with J is reported for them in the same way. Signals reported as multiplets, or with the splitting given as an apparent value (J_{app}), are consistent under both parts; this includes the notation of Tang, Xu and Hoye, in which non-first-order doublets, triplets and multiplets are reported as “nfod”, “nfot” and “nfom” with J_{app} [27].

Item 7 asks whether a signal whose appearance differs from how it arises is labeled as apparent. A triplet of a proton with two different neighbors can arise only if the two couplings happen to be equal, and it should then be reported as “app t” with an apparent splitting, or as a dd with both constants. To recognize such a case, the tool would need to know which proton gives the signal and whether its neighbors are chemically equivalent. *This needs an assignment, and item 7 is not checked*.

A finding states only that values disagree. Whether the description or the structure is wrong cannot be decided from the text, and the alerts are worded accordingly. Where a rule cannot decide without an assignment, the case is reported for a person to check and is not counted as an inconsistency.

Table 1: Items 1–7 of the checklist [2] in *checkHNMR* V2.1.3-beta.

Item	Question (short)	Status	What limits the check
1	Proton count; every signal has a place in the structure	In part	*Proton count only; needs a structure or an HRMS formula*. Placing every signal needs an assignment
2	Shifts consistent with the structure	Not checked; planned	Needs reference shift ranges (HOSE codes, NMRShiftDB)
3	Every coupling constant has a partner	Checked	*H–F and H–P: only the presence of the splitting in the $^{19}F/^{31}P$ description*
4	Multiplicity consistent with the number of neighbors	Checked; 4(b) in part	4(b): without an assignment, only a multiplicity that no proton of the structure can have, and a methyl singlet that only a CH_3 next to a C–H could give
5	Couplings consistent with the resolution	Not checked; planned	Needs the line width (FWHM), not stated in a description
6	First-order notation only where valid	Checked	6(a) needs a structure; 6(b) needs the spectrometer frequency; open-chain CH_2–CH_2 not checked
7	Apparent multiplicities labeled as apparent	Not checked	Needs an assignment

How the tool is built

checkHNMR is a set of Python scripts that can be run in two ways: as a local version, on the user's computer or in Google Colab, and as a website in which the same scripts run in the browser. In both, the checks are made by the same code, with RDKit [28] for the structure, and give the same answer for the same input.

The local version reads Supporting Information files and articles in PDF format as they were published, in the layouts of different journals. It was developed mainly on files from four journals of three publishers: *J. Org. Chem.* (Supporting Information and articles), *Beilstein J. Org. Chem.* and *ACS Omega* (Supporting Information), and *Molecules* (articles, which include the experimental data). The lists of articles are given in the repository (see **Development and testing**). The tool finds the ^{1}H NMR descriptions in the text together with the compound labels; for each description, it reads the compound name, the HRMS line and any ^{19}F or ^{31}P description. Structures

are made from the names with OPSIN [29] and compared with the neutral formula from the HRMS line. A structure is not used if it has fewer carbon atoms than the formula, which usually means that the name of another compound, such as a reagent, was read; item 1 is then checked against the formula. Structures can also be added or corrected by hand in the intermediate table, and a corrected structure is never changed by the program. The results are written as an Excel table and as a Word report with one compound per page, the structure, the description with the flagged signals highlighted and the findings for every item. Without an internet connection, OPSIN cannot be reached, and the descriptions are checked from the text alone (items 3, 4(a) and 6(b), and item 1 against the HRMS formula). The local version requires Python 3.10 or newer and has been tested on Windows, on Linux and in Google Colab; a launcher for macOS is included but has not been tested. It has no limit on the number of compounds. In Google Colab it runs without installation, but the files are processed on Google servers, which is suitable for published Supporting Information but not for unpublished data.

The website (https://checkhnmr.org) runs the same Python scripts in the browser with Pyodide [30], a port of Python to WebAssembly. RDKit is used there through RDKit.js, the JavaScript build of the same RDKit release, with a thin compatibility layer that gives the scripts the functions they call, so the scripts themselves are not changed. PDF files are read with PyMuPDF [31], version 1.26.3 on the website (the latest built for Pyodide) and 1.28.2 in the local version; on 251 PDF files with 7,745 descriptions, the two versions gave the same descriptions, labels, names, HRMS formulas and ^{19}F and ^{31}P descriptions, and the extracted text differed only in four files, by a few characters outside the descriptions. Pyodide, RDKit.js and PyMuPDF are loaded from the website itself, about 20 MB at the first check, and then run on the user's computer: nothing is installed, no request is sent to other servers, and the descriptions and files are not sent anywhere.

The website accepts one compound entered in a form (label, SMILES string, ^{1}H, ^{19}F and ^{31}P descriptions, HRMS line) or a PDF file of up to 20 compounds written in the format of a template, in which each compound has a "SMILES:" line; Supporting Information files in their published form are checked with the local version. On the website, structures are taken only from SMILES strings: OPSIN is a web service, and converting names would send them to another server. A separate layer of the website turns the verdicts of the scripts into the report (see **The report**) and does not change them.

That the two versions give the same results was tested directly. For 11,291 distinct structures, every property of the molecule that the checks use (atoms, bonds, rings, hydrogen counts, aromaticity, distances, symmetry classes and substructure matches) was computed with RDKit in Python and with RDKit.js; they agreed except in two cases that do not affect any check. For one compound with a *cis*-1,4-disubstituted cyclohexane ring, the atoms were assigned to different symmetry classes, but the checks use symmetry classes only for aromatic rings. For one aziridine, RDKit.js does not show the stereocenter at the nitrogen atom in the displayed SMILES. The checks were then run on 24,463 descriptions, 14,994 of them with structures, both in Python and in Pyodide with RDKit.js: the results were identical byte for byte. These descriptions came from the development data and from the *Org. Lett.* Supporting Information files described in **Development and testing**. This comparison was made with V2.1-beta; V2.1.2-beta changed only the reading of descriptions, which is pure Python and runs unchanged in both versions. V2.1.3-beta added the lower bound for methyl groups, which uses RDKit, so the comparison was repeated on the 5,136 development descriptions from *J. Org. Chem.* that have structures: the verification codes and all verdicts were identical.

The report

For each compound, the website shows a report (**Figure 1**). Its header gives the version of the tool, the date of the check and a verification code, the structure drawn from the SMILES string, and the formula used for item 1. A summary line gives the number of alerts of each level. The description is shown as entered, with the signals named in the alerts highlighted in the color of their level. It is followed by an "As read" table of the shifts, multiplicities, coupling constants and integrals as the tool read them. This table is the first thing to check: if a signal was read wrongly, the findings about it concern the reading by the tool and say nothing about the description. The report ends with the items that were not checked and with the statement that only the raw data can show whether the values were measured. When a PDF file is checked, the reports are preceded by a table with the number of alerts of each level for every compound.

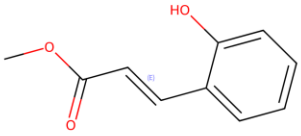
(a)

Report for compound 1

checkHNMR V2.1.3-beta · checked 31 October 2026 at 02:15
verification code 9445-9C84-5432

SMILES COC(=O)/C=C/c1ccccc1O · $C_{10}H_{10}O_3$ · 10 H

HRMS: [M + H]+, $C_{10}H_{11}O_3$ → neutral $C_{10}H_{10}O_3$



Alerts: A 1 inconsistent B 1 to check C 1 format or note

(b)

Description

^{1}H NMR (400 MHz, $CDCl_3$) δ 8.01 (d, J = 16.1 Hz, 1H), 7.47 (dd, J = 7.7 Hz, 1H), 7.27–7.21 (m, 1H), 6.92 (t, J = 7.5 Hz, 1H), 6.85 (d, J = 8.1 Hz, 1H), 6.63 (d, J = 16.1 Hz, 1H), 3.81 (s, 3H).

As read

#	δ, ppm	Multiplicity	J, Hz	nH
1	8.01	d	16.1	1
2	7.47	dd	7.7	1
3	7.27–7.21	m	–	1
4	6.92	t	7.5	1
5	6.85	d	8.1	1
6	6.63	d	16.1	1
7	3.81	s	–	3

Reported 9 H; structure 10 H. If a signal was read wrongly, say so in your response to the alert.

(c)

Alerts and responses

☐ I am a reviewer or an editor

A **Item 4(a). Multiplicity and coupling constants do not agree**

The number of coupling constants does not fit the multiplicity (for example, a dd with one J, or a t with two different J).

▼ Details from the check

- #2 7.47 (dd): 1 J reported, 2 expected

Author's response (not checked by checkHNMR)

Typing error: the signal is 7.47 (dd, J = 7.7, 1.6 Hz, 1H); corrected in the manuscript.

B **Item 3. Coupling partner cannot be verified**

The partner of this coupling constant may be a multiplet without constants, a broad signal or a heteronucleus; the description does not make it possible to verify it.

▼ Details from the check

- #5 6.85: J = 8.1 - no partner; partner may be an m/br signal

Author's response (not checked by checkHNMR)

The partner of J = 8.1 Hz (H-4) is in the multiplet at 7.27–7.21 ppm.

C **Item 1. Exchangeable proton not reported**

Fewer protons than the structure, by the number of exchangeable protons (OH, NH): a subcategory, not counted as an inconsistency.

▼ Details from the check

- reported 9 H; structure 10 H (1 exchangeable)

(d)

Not checked

Items 2, 5 and 7 are not checked; item 1 and item 4(b) only partly. What is checked and what is not

checkHNMR checks that the numbers in a description agree with each other and with the structure. It cannot tell whether they were actually measured: a description can pass every check and still be wrong or invented.

Only the raw data (FID) can show that. Authors: deposit your FIDs. Journals and reviewers: ask for them.

Save report (HTML) Save data (JSON)

Print / Save as PDF

Figure 1: Report of the *checkHNMR* website (V2.1.3-beta) for an *invented description* of methyl (*E*)-3-(2-hydroxyphenyl)acrylate. (a) Header with the version, the date, the verification code, the structure and the alert counts. (b) The description with the signals named in the alerts highlighted, and the "As read" table. The signal at 6.92 ppm is an apparent triplet [2] and should be reported as "app t, J_{app} = 7.5 Hz"; the tool does not report it because item 7 is not checked in the current version. (c) Alerts and responses: A, the signal at 7.47 ppm is reported as a "dd" with one coupling constant; B, the partner of J = 8.1 Hz may be the multiplet at 7.27–7.21 ppm; C, the OH proton is not reported. The responses are examples written by an author and are not checked by the tool. (d) Items not checked, the statement on raw data, and the options to save the report. The four parts are taken from one report and arranged side by side; the content is not changed.

The levels display the verdicts of the scripts and do not change them. Level A marks values that break the rule of an item, such as a number of protons that does not agree with the structure, a coupling constant without a partner, a multiplicity with the wrong number of constants, or a first-order description of a system that is not first-order. Level B marks cases that need a person, such as a partner that may be hidden in a multiplet, a multiplicity that may be impossible (item 4(b)), integrals that may be given for part of a symmetric molecule, a signal that may belong to a non-first-order system, or a pair close to the limit of strong coupling. Level C marks problems of format and notes, such as a missing spectrometer frequency, Cyrillic letters typed instead of Latin ones, or an exchangeable proton that is not reported. Each alert gives the reason and, where it helps, the details from the check. A report without alerts states that this does not mean the description is correct.

Below each alert of level A or B, the author can write a response, as for the alerts of *checkCIF* [10]*:* for example, that a signal was read wrongly or that the partner of a coupling is in a multiplet. The responses are not checked by the tool. A reviewer or an editor does not need to fill in anything and can switch the report to a mode in which the responses can be read but not edited; this prevents accidental changes and does not protect the text. The report can be saved as an HTML file, which opens without the website and keeps the responses, saved as a JSON file with the entries and the results, or printed and saved as PDF.

The verification code is the first twelve characters of a SHA-256 hash of the version of the tool, the entries, the verdicts and the findings; the date, the responses and the highlighting do not enter the code, so writing a response does not change it. The same entries checked with the same version give the same code. To use the code, a reviewer needs the entries and the version of the tool; both are in the report that the authors of the manuscript save as an HTML or JSON file and submit with

it. The reviewer opens the website of that version (the current version at checkhnmr.org, earlier versions under Archived versions), copies each entry into the field of the same name in the One compound form, runs the check, and compares the code shown with the code in the report. The same code shows that the report was produced by this version from exactly these entries, and that its verdicts and findings are reproduced; whether the entries match the description in the manuscript can be checked only by comparing the two. A different code shows that an entry or the version differs; a single changed character, such as a hyphen typed instead of an en dash, is enough. The report of **Figure 1** saved as a JSON file is **Supporting Information File 1**, and **Supporting Information File 2** (**Section S1**) gives this procedure step by step for it, with the entries and the code they give. The website version described here, V2.1.3-beta, is kept unchanged at https://checkhnmr.org/v2.1.3-beta/, so that these checks can be repeated after the main website is updated.

The local version gives the same verdicts and findings in its Word report and Excel table (see **How the tool is built**).

Development and testing

checkHNMR is a rule-based tool, developed iteratively on a development set. Each time the tool read a description wrongly or gave a wrong finding, I decided what the correct reading was; a rule was then written for the whole class of such texts and a unit test was added, and each change is listed in the development log of the repository. The code was written together with Claude (Anthropic), mainly Claude Opus 5.5, a large language model: I decided what the tool should check and how each text should be read, and checked the results; the model wrote and ran the code and the tests (see Acknowledgments). The development set consists of open-access articles with CC BY licenses from PubMed Central: *J. Org. Chem.* 2024–2025 (300 articles, Supporting

Information and article PDFs), *Beilstein J. Org. Chem.* 2025 (109 articles), *ACS Omega* 2025 (170 articles), and *Molecules* 2024–2025 (1,004 articles), with 15,723 ^{1}H NMR descriptions in all (15,354 distinct: 369 *J. Org. Chem.* descriptions appear both in the article and in its Supporting Information); the PMCID lists are in the repository (docs/development_data). Earlier versions were also corrected on the Supporting Information of six other articles, and the item-4(b) check was tuned on 982 descriptions of *Org. Lett.* 2026 used in a separate study; the *Org. Lett.* files were later used only to check that a change did not alter other results and to calibrate a marker of reading quality. For this reason, none of these data can be used to measure accuracy. The lower bound for methyl groups in item 4(b), added in V2.1.3-beta, was checked on the 5,136 development descriptions from *J. Org. Chem.* that have structures: 34 descriptions received a finding of the lower bound for methyl groups (**Supporting Information File 3**), 10 others changed from "nothing to check" to "consistent" under item 4(b), and no other result changed. All 8,540 records of these articles are given in **Supporting Information File 4**, as read, with the errors of reading. The program marks the problems it detects in its own columns, for example names that gave no structure, structures whose formula differs from the HRMS formula, and labels that were not found; a record without such a mark has not been checked by hand. I made this table on 8 November 2026 with the local version V2.1.3-beta (run_all.py, with the package versions listed in requirements.txt) from the PDF files of the articles listed in docs/development_data. The same code and package versions read the same files into the same table, and check_descriptions of V2.1.3-beta gives the results above from it; only the structures depend in addition on the OPSIN web service at the time of the run.

The tool is tested in two ways. The 90 unit tests check single readings and rules. The stress tests generate descriptions for which the correct answer is known and compare it with the output of the

tool (**Table 2**). The first test writes known signals with format noise (dashes, spaces, the order of the items within the parentheses, artifacts of copying from PDF). The second builds first-order spin systems in which every constant appears in both partners and then introduces one known error: a changed constant, a missing or an extra constant, or two coupled signals moved close together. The third writes simplified descriptions from real structures of the development set and changes one integral, writes the signal of a CH_3 group next to a carbon with a proton as a singlet, or adds partners that the proton cannot have. The fourth, the realistic reading test, writes the content of one published description in the style of another, with the spellings and their frequencies taken from the development set. All four tests run with one command (*python run_all.py --selftest -n 100000*), and the summary of this run is in the repository (docs/selftest_V2.1.3-beta_100k.txt).

Table 2: Stress tests of *checkHNMR* V2.1.3-beta (command *python run_all.py --selftest -n 100000*; 100,000 synthetic descriptions per test, seed 20261023, unless stated otherwise). In the format-noise test, each description is written both without and with noise; in the spin-system and structure tests, the descriptions are divided at random among the cases listed. **n** is the number of descriptions of each case. Result is the number of these descriptions that were misread, received a false finding, or in which the introduced error was found, with the share of **n** in parentheses. In the spin-system test, a system to which the chosen error could not be applied (for example, an added constant when no signal is a singlet) was skipped, so the values of **n** add up to fewer than 100,000.

Test	Case	n	Result
Format noise	clean descriptions, misread	100,000	0
	descriptions with format noise, misread[a]	97,330	68 (0.07%)
Spin systems	correct descriptions, false finding in items 3, 4(a) or 6(b)	19,858	0
	one constant changed, found (item 3)	19,133	19,133 (100%)
	one constant missing, found (item 4(a))	15,663	15,663 (100%)
	one constant added, found (item 4(a))	14,662	14,662 (100%)
	two coupled signals moved close, found (item 6(b))	19,094	18,067 (94.6%)
Structures	correct descriptions, false finding in items 1 or 4(b)	28,730	0
	one integral changed, found (item 1)	23,056	21,628 (93.8%)
	CH_3 next to a C–H written as a singlet, found (item 4(b))	19,156	18,311 (95.6%)
	partners added that the proton cannot have, found (item 4(b))	29,058	14,463 (49.8%)[b]
Realistic reading	descriptions misread	100,000	30 (0.03%)
	descriptions misread, seed 260626: V2.1.3-beta / V2	100,000	39 (0.04%) / 7,017 (7.02%)

[a] 2,670 descriptions written with a decimal comma, which the tool does not read, are not counted.
[b] An upper-bound check; misses are expected.

The checks gave no false findings on correct descriptions: none in 19,858 spin systems for items 3, 4(a) and 6(b), and none in 28,730 descriptions from structures for items 1 and 4(b) (**Table 2**). Known errors were found in all cases for items 3 and 4(a) and in 94.6% of cases for item 6(b); the missed pairs are isolated AB pairs, which the rule exempts. A wrong integral was found in 93.8% of cases; the rest are descriptions in which one proton fewer than the structure is reported and the structure has an *NH* or *OH* group, which the tool reports as an exchangeable proton not reported. A methyl singlet next to a carbon with a proton was found in 95.6% of cases; in the rest, another group of the structure that can give a singlet could account for it. Item 4(b) found 49.8% of the added partners: it flags a multiplicity only when no proton of that kind in the structure can have so

many neighbors, so it is an upper-bound check and misses are expected. In the realistic reading test, V2.1.3-beta misread 30 of 100,000 descriptions (0.03%); in a separate run with another seed, it misread 39 (0.04%) and the earlier version V2 misread 7,017 (7.02%).

For the website, I also tested how it reads text that was typed by hand or pasted from another program. Such text often differs from the original in characters that look the same on screen. A script made five copies of each of the 15,723 development descriptions, each with one kind of such change:

i. Cyrillic letters that look like Latin ones, as typed with a Russian keyboard layout (for example, “1H” or “2H” with the Cyrillic letter H (U+041D));

ii. the changes made by AutoCorrect in Microsoft Word: a hyphen between spaces replaced by an en dash, straight quotation marks by curly ones;

iii. the minus sign (U+2212) instead of the hyphen-minus typed from the keyboard;

iv. no-break and thin spaces, double spaces and line breaks, as left by copying from a PDF;

v. all of these together.

All copies were checked with the website code. For every copy, the tool read the same signals and gave the same verdicts and findings as for the original text, and all 149,041 signals were found in the text so that they could be highlighted in the report.

Stress tests are not accuracy measurements. They show how robustly the tool reads the forms that occur in the literature and whether the rules find the errors they were written for. They do not show how accurate the tool is on new articles. *checkHNMR* contains no language model or other machine learning: *it reads only the forms of notation for which a rule has been written*. *A form that it has not met, such as a decimal comma, will be read wrongly or not at all, and a finding may then be about the reading rather than the description.* This is why the report shows the “As read”

table and marks descriptions that were not read completely. The accuracy of each version is measured only on articles that were not used in its development; for the current version, this is part of the separate study that is in progress (OSF registration doi:10.17605/OSF.IO/EQ9AH).

checkHNMR is still in beta. While this paper was being written, the tool went from V2.1-beta to V2.1.3-beta, and I continue to improve both the website and the local version alongside the separate study of how often published descriptions are inconsistent and whether such inconsistencies are occasional or systematic. Each change is recorded in the development log of the repository and in the version history on the website, and from V2.1.3-beta on, each version is also kept unchanged under Archived versions. Errors reported by users will be considered and, where they are confirmed, corrected in the next version.

Limitations

checkHNMR checks only whether the numbers in a description agree with each other and with the structure. It cannot tell whether they were measured. A description copied from an analog, adjusted by hand or invented can pass every check, and a description that passes is not shown to be correct. Only the raw data can show what was measured, yet raw NMR data are rarely shared and FIDs almost never [1]. For this reason, the website and every report state that authors should deposit their FIDs and that journals and reviewers should ask for them. A consistency check is useful alongside the raw data and does not replace them.

The same checks that help an honest author can help someone who fabricates data to make an invented description look consistent. Crystallography has met this problem. In 2010, Acta Crystallographica Section E reported at least 70 falsified structures, in which genuine diffraction data had been reused with the metal or other atoms replaced [32]. The problem was first noticed through alerts of the journal's own checking programs and was confirmed by comparing the

deposited structure-factor files, and the editors noted that it "could not have been easily discovered without the availability of the structure-factor files" [32]. For NMR, the FID plays the role of the structure-factor file. I decided to make *checkHNMR* open because the checks it performs can be done by hand by anyone who knows the rules, so keeping the code closed would not stop such use, while openness lets authors, reviewers and editors see exactly what is checked.

A check that becomes a requirement can also lose its value as a measure; in Strathern's formulation of Goodhart's law, "When a measure becomes a target, it ceases to be a good measure" [33]. If a report without alerts were demanded, the easiest way to obtain one would be to report every signal as "m" without coupling constants, and the current version accepts such a description, since a multiplet with no constants gives nothing to check. In a later version, I plan to report a signal written as "m" when none of the protons that could give it has coupling partners. More generally, the report is meant to be read: an alert with a reasoned response from the author, for example that a coupling constant was obtained by spectral simulation, is a better outcome than a description rewritten to avoid the alert.

Like the checklist [2], *checkHNMR* is meant for small organic molecules; it was developed on descriptions from organic chemistry journals. Descriptions of polymers, in which signals are broad and integrals refer to a repeat unit, and of coordination, organometallic or paramagnetic compounds, in which couplings to metal nuclei, exchange or paramagnetic broadening change the spectrum, were not part of the development data, and the findings for them may not be meaningful.

The findings also depend on the structure. A finding states that the description and the structure disagree; when the structure is wrong, for example because a compound name was converted into the structure of another compound, the finding is about the structure. Of the 34 development descriptions in which the lower bound for methyl groups in item 4(b) flagged a methyl singlet, 5

are not included in **Supporting Information File 3**: in 3, the local version matched the description with another compound, and 2 were the same descriptions in the Supporting Information and in the article. Of the other 29, 16 also had a number of protons that did not agree with the structure (item 1), which points to a structure that does not match the description, to a mixture or to an integral error. Of all 29, 12 were errors of the description in which a CH_3 group next to a C–H was written as a singlet (for example, "4.37 (q, *J* = 7.1 Hz, 1H) ... 1.44 (s, 3H)"), 1 was an integral error, 10 came from a compound name that does not correspond to the structure drawn in the publication, 5 were mixtures of diastereomers, and 1 described the enol form of a compound whose name gives the keto form (**Supporting Information File 3**). Names are converted literally: "trimethyl(methyl propionylthio)silane", for example, becomes an *S*-silyl thioester instead of the drawn methyl 3-(trimethylsilylsulfanyl)propanoate. The program had marked all 10: the formula of the structure made from the name differs from the HRMS formula (opsin_status "FORMULA MISMATCH" in **Supporting Information File 4**), a mark that none of the other 19 records has.

These records come from published Supporting Information read with the local version, in which the structure is converted from the compound name and each description is matched to a compound heading found in the text; a wrong name, or a label that is not read, gives a description the structure of another compound. Authors who check their own descriptions on the website write them in the template, in which each compound has its own block with the label on a separate line and a "SMILES:" line, so the structure is the one they enter. A wrong SMILES string still gives findings about the structure, not about the description. The structure therefore has to be checked by a person along with the "As read" table.

Published descriptions are now also read by machines. It has been proposed to look for new reactions by analyzing NMR spectra of crude reaction mixtures with an AI platform, with the

proposed structures then confirmed by a chemist [34]. BLIND, a model that proposes structures from ^{1}H and ^{13}C NMR descriptions, was developed on 5.5 million descriptions extracted from the literature, and its authors note that "manual transcription of spectral data introduces a non-negligible risk of human error" [35]. On descriptions in which the number of signals or the integrals do not agree with the structure, the share of correctly proposed structures fell to 2.4%, compared with 45.0% on their main test set [35]. *checkHNMR* does not correct such data; it can only help authors find inconsistencies before a description is published and becomes a source for others.

Conclusion

checkHNMR checks a ^{1}H NMR description against items 1–7 of the checklist proposed in my earlier work [2]: the number of protons (item 1, in part), coupling partners (item 3), multiplicities and the number of neighbors (item 4, with 4(b) in part) and first-order notation (item 6). Items 2 and 5 are not checked in the current version and are planned; item 7 needs an assignment. The website runs the checks in the browser for up to 20 compounds and sends nothing anywhere; the local version reads Supporting Information files as published and gives the same results. In stress tests with known answers, the tool read synthetic descriptions written in the styles of published ones almost without errors, found most of the errors that its rules were written for, and gave no false findings on correct descriptions. These tests show how robustly the tool works; its accuracy on new articles will be reported with the separate study in progress. The tool checks only whether the numbers in a description agree with each other and with the structure: *a description can pass every check and still be wrong or invented, and only the raw data (FID) can show what was measured*.

Mistakes are part of science, and they can be found before publication or after it. checkHNMR lets authors find them before, on their own, while the description can still be corrected.

Supporting Information

Supporting Information File 1: File name: checkHNMR_1_9445-9C84-5432.json; Format: JSON; Website report of **Figure 1** saved as a JSON file. *Description*: Saved from the website V2.1.3-beta with "Save data (JSON)": the entries, the signals as read, the verdicts, findings and alerts, and the verification code. The entries can be copied from this file to repeat the check.

Supporting Information File 2: File name: checkHNMR_Supporting_Information.pdf; Format: PDF; Entries for repeating the check shown in **Figure 1**. *Description*: The entries of the check in **Figure 1**, field by field, the alerts and the verification code that the website V2.1.3-beta gives for them, and what changes the code.

Supporting Information File 3: File name: methyl_findings_JOC_V2.1.3-beta.xlsx; Format: Excel; Development descriptions flagged by the lower bound for methyl groups (item 4(b)), with classification. *Description*: The 29 records that remain of the 34 *J. Org. Chem*. development descriptions with this finding, classified by hand against the publications, including records in which the compound name in the publication is wrong; with the findings of items 1 and 3 and the verification code of each record on the website.

Supporting Information File 4: File name: records_JOC_development_descriptions.xlsx; Format: Excel; Development descriptions from *J. Org. Chem.* read by the local version, with names and structures. *Description*: 8,540 ^{1}H NMR descriptions read on 8 November 2026 with V2.1.3-beta from the Supporting Information and article PDFs of 300 articles (descriptions were found for 251 of them), 5,136 of them with structures made from the compound names with OPSIN; the input of check_descriptions, with the list of descriptions that appear both in the Supporting Information and in the article.

Acknowledgments

I thank Prof. Mikhail S. Baranov (Shemyakin–Ovchinnikov Institute of Bioorganic Chemistry, Russian Academy of Sciences, Moscow), my former supervisor, for his support and for discussions of the plans for this work. I thank Dr. Áron Péter (Max-Planck-Institut für Kohlenforschung, Mülheim an der Ruhr) for his comments on my earlier work, from which this tool grew. I thank Prof. Scott J. Miller (Yale University) for discussions of whether a tool such as *checkHNMR* is needed and of how NMR descriptions should be viewed today. I thank Dr. Oleg I. Afanasyev and Dr. Denis A. Chusov (both D.Sc.; A. N. Nesmeyanov Institute of Organoelement Compounds of the Russian Academy of Sciences, Moscow) for sharing their preprints and for discussions. Papers by Prof. Thomas R. Hoye (University of Minnesota), Prof. Andrei G. Kutateladze (University of Denver) and Prof. Craig P. Butts (University of Bristol), including those cited here, were among the reasons I wrote this paper, and I thank them for their contributions to NMR spectroscopy. I thank Vlad Koselovsky for his help with the English of the manuscript.

checkHNMR was built in conversation with Claude (Anthropic), mainly Claude Opus 5.5, a large language model. I proposed the checklist and decided what the tool checks, how each form of description should be read, which data were used for development, and which results were accepted. Claude wrote and ran the code, the tests and the website, checked numbers and references against the sources, and wrote first drafts of parts of the English text of this paper from my outline, which I then revised. Claude also classified the records in Supporting Information File 3, which I checked against the publications, and assembled Supporting Information Files 3 and 4 from the output of the tool. I also ran the code on my own computer and examined it in a code editor. All decisions are mine, and I take full responsibility for the tool and the paper.

References

(1) Bloodworth, S.; Willoughby, C.; Coles, S. J. *Beilstein J. Org. Chem.* **2025**, *21*, 864–876. doi:10.3762/bjoc.21.70

(2) Eshtukov-Shcheglov, A. The Problem of NMR Data Analysis in Synthetic Chemistry: Why Is It Important to Understand Spectra? ChemRxiv 2026. doi:10.26434/chemrxiv.15010060/v1

(3) Wang, J.-J.; Jin, Y.; Zhi, C.-Y.; Liu, Y.-J.; Huang, X.-H.; Xu, F.; Ji, X.; Fang, X.; Tao, H.; E, W.; Zhang, L.; Ke, G.; Zhu, R. *Sci. Data* **2025**, *12*, No. 1954. doi:10.1038/s41597-025-06245-5

(4) Jin, Y.; Wang, Y.; Wang, J.-J.; Zhu, R.; Ke, G.; E, W. From Human Labels to Literature: Semi-Supervised Learning of NMR Chemical Shifts at Scale. arXiv 2026. doi:10.48550/ARXIV.2601.18524

(5) Dickson, C. L.; Mohammed, S.; Jones, L. S. L.; Crick, D. J.; Augustyniak, W.; Beaumont, M.; Westerman, J.-C.; Henshaw, S.; Johnston, S. T.; Blundell, C. D.; Watson, M.; Takacs, Z.; Butts, C. P. *Analyst* **2025**, *150*, 2901–2906. doi:10.1039/D5AN00240K

(6) Mirza, A.; Alampara, N.; Kunchapu, S.; Ríos-García, M.; Emoekabu, B.; Krishnan, A.; Gupta, T.; Schilling-Wilhelmi, M.; Okereke, M.; Aneesh, A.; Asgari, M.; Eberhardt, J.; Elahi, A. M.; Elbeheiry, H. M.; Gil, M. V.; Glaubitz, C.; Greiner, M.; Holick, C. T.; Hoffmann, T.; Ibrahim, A.; Klepsch, L. C.; Köster, Y.; Kreth, F. A.; Meyer, J.; Miret, S.; Peschel, J. M.; Ringleb, M.; Roesner, N. C.; Schreiber, J.; Schubert, U. S.; Stafast, L. M.; Wonanke, A. D. D.; Pieler, M.; Schwaller, P.; Jablonka, K. M. *Nat. Chem.* **2025**, *17*, 1027–1034. doi:10.1038/s41557-025-01815-x

(7) Kliuev, F.; Smirnov, I.; Losev, M. A.; Afanasyev, O. I.; Chusov, D. Can AI Help Chemists To Solve NMR? ChemRxiv 2026. doi:10.26434/chemrxiv.15006195/v2

(8) Son, G.; Hong, J.; Fan, H.; Nam, H.; Ko, H.; Lim, S.; Song, J.; Choi, J.; Paulo, G.; Yu, Y.; Biderman, S. When AI Co-Scientists Fail: SPOT-a Benchmark for Automated Verification of Scientific Research. arXiv 2025. doi:10.48550/ARXIV.2505.11855

(9) Stokel-Walker, C. *Nature* **2026**, *656*, 278–279. doi:10.1038/d41586-026-02235-8

(10) Spek, A. L. *Acta Crystallogr., Sect. E: Crystallogr. Commun.* **2020**, *76*, 1–11. doi:10.1107/S2056989019016244

(11) Christmann, M. *Org. Lett.* **2025**, *27*, 4–7. doi:10.1021/acs.orglett.4c03458

(12) Novitskiy, I. M.; Kutateladze, A. G. *J. Org. Chem.* **2022**, *87*, 4818–4828. doi:10.1021/acs.joc.2c00169

(13) Beduru, S.; Kutateladze, A. G. *Nat. Prod. Rep.* **2026**. doi:10.1039/D5NP00088B

(14) Beduru, S.; Asimbisa, E.; Novitskiy, I. M.; Kutateladze, A. G. *J. Org. Chem.* **2026**, *91*, 4130–4153. doi:10.1021/acs.joc.5c02863

(15) Eshtukov-Shcheglov, A. Assessment Protocol for a Survey of Published ^{1}H NMR Descriptions (Pass 3, Version 1.2). OSF Registries 2026. doi:10.17605/OSF.IO/EQ9AH

(16) Bremser, W. *Anal. Chim. Acta* **1978**, *103*, 355–365. doi:10.1016/S0003-2670(01)83100-7

(17) Steinbeck, C.; Krause, S.; Kuhn, S. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 1733–1739. doi:10.1021/ci0341363

(18) Kuhn, S.; Johnson, S. R. *ACS Omega* **2019**, *4*, 7323–7329. doi:10.1021/acsomega.9b00488

(19) Keeler, J. *Understanding NMR Spectroscopy*, 2nd ed.; John Wiley and Sons: Chichester, U.K., 2010

(20) Claridge, T. D. W. *High-Resolution NMR Techniques in Organic Chemistry*, 3rd ed.; Elsevier: Amsterdam, 2016
(21) Pretsch, E.; Bühlmann, P.; Badertscher, M. *Structure Determination of Organic Compounds: Tables of Spectral Data*, 5th ed.; Springer: Berlin, 2020. doi:10.1007/978-3-662-62439-5
(22) Günther, H. *NMR Spectroscopy: Basic Principles, Concepts and Applications in Chemistry*, 3rd ed.; Wiley-VCH: Weinheim, 2013
(23) Silverstein, R. M.; Webster, F. X.; Kiemle, D. J.; Bryce, D. L. *Spectrometric Identification of Organic Compounds*, 8th ed.; Wiley: Hoboken, NJ, 2015
(24) Karplus, M. *J. Chem. Phys.* **1959**, *30*, 11–15. doi:10.1063/1.1729860
(25) Karplus, M. *J. Am. Chem. Soc.* **1963**, *85*, 2870–2871. doi:10.1021/ja00901a059
(26) Hoye, T. R.; Hanson, P. R.; Vyvyan, J. R. *J. Org. Chem.* **1994**, *59*, 4096–4103. doi:10.1021/jo00094a018
(27) Tang, R.; Xu, Q.; Hoye, T. R. *Molecules* **2025**, *30*, No. 2610. doi:10.3390/molecules30122610
(28) RDKit: Open-Source Cheminformatics Software, Release 2026.09.1, 2026
(29) Lowe, D. M.; Corbett, P. T.; Murray-Rust, P.; Glen, R. C. *J. Chem. Inf. Model.* **2011**, *51*, 739–753. doi:10.1021/ci100384d
(30) The Pyodide development team. Pyodide/Pyodide, Version 0.29.0, 2021. doi:10.5281/zenodo.5156931
(31) Artifex Software. PyMuPDF, Version 1.28.2, 2026
(32) Harrison, W. T. A.; Simpson, J.; Weil, M. *Acta Crystallogr., Sect. E: Struct. Rep. Online* **2010**, *66*, e1–e2. doi:10.1107/S1600536809051757
(33) Strathern, M. *Eur. Rev.* **1997**, *5*, 305–321. doi:10.1002/(SICI)1234-981X(199707)5:3<305::AID-EURO184>3.0.CO;2-4
(34) Afanasyev, O. I.; Kliuev, F.; Balalaeva, A. I.; Chusov, D. Rethinking Reaction Discovery: From Serendipity to Data-Driven Chemistry. ChemRxiv 2025. doi:10.26434/chemrxiv-2025-cxhdq
(35) Zlobin, I.; Bespalov, I.; Shandybo, M.; Rekut, N.; Nelyubina, Y.; Chusov, D.; Afanasyev, O.; Novikov, V. Bimodal Learning from Imperfect NMR Data (BLIND): A Chemistry-Ignorant Model for Autonomous Structure Elucidation from NMR. ChemRxiv 2026. doi:10.26434/chemrxiv.15006080/v1

Data Availability Statement

The code of checkHNMR V2.1.3-beta described in this article is openly available on Zenodo at https://doi.org/10.5281/zenodo.23823029; all versions of the code are available at https://doi.org/10.5281/zenodo.23749627. The code is also available on GitHub at https://github.com/ArturEshtukov-Sheglov/checkHNMR. The website version described here is available at https://checkhnmr.org/v2.1.3-beta/. The lists of development articles (PMCIDs) are in the repository (docs/development_data). The entries and the saved report of the check in Figure 1 are given in Supporting Information Files 1 and 2. The development descriptions flagged by the lower bound for methyl groups, with their classification, are given in Supporting Information File 3, and all development records from *J. Org. Chem.* as read by the tool are given in Supporting Information File 4.