frontiers | Frontiers in Earth Science

TYPE Technology and Code
PUBLISHED 09 August 2023
DOI 10.3389/feart.2023.1232256

OPEN ACCESS

EDITED BY
Ali Abedini,
Urmia University, Iran

REVIEWED BY
Reneé González-Guzmán,
National Autonomous University of Mexico, Mexico
William F. McDonough,
University of Maryland, United States

*CORRESPONDENCE
Tom Buckle,
t.buckle@exeter.ac.uk

RECEIVED 31 May 2023
ACCEPTED 25 July 2023
PUBLISHED 09 August 2023

CITATION
Buckle T, Williams M, Nathwani CL and Hughes HSR (2023), WebNORM: a web application for calculating normative mineralogy.
Front. Earth Sci. 11:1232256.
doi: 10.3389/feart.2023.1232256

WebNORM: a web application for calculating normative mineralogy

Tom Buckle[1]*, Morgan Williams[2], Chetan L. Nathwani[3] and Hannah S. R. Hughes[1]

[1]Camborne School of Mines, University of Exeter, Cornwall, United Kingdom, [2]CSIRO Mineral Resources, Kensington, WA, Australia, [3]Institute of Geochemistry and Petrology, Department of Earth Sciences, ETH Zürich, Zürich, Switzerland

Normative mineralogy is a valuable tool for interpreting the mineralogical composition of rocks based on their bulk rock geochemical data. Despite the development of various computational tools for normative mineralogy calculations, the accessibility and continued usage of many existing tools is limited by the use of older languages, licensing constraints, and restricted access to the source code. In this paper, we introduce a Python-based implementation of the normative mineralogy algorithm described by Verma et al. (Schweizerische Mineralogische und Petrographische Mitteilungen, 2003, 83, 197–216), which is accessible through a user-friendly web application *webNORM*, or as part of the *pyrolite* Python package. The algorithm uses major, minor, and trace element oxides as input and returns up to 31 normative minerals. It offers two methods for estimating the Fe_2O_3:FeO ratio. The web application offers a convenient interface for users without programming experience, while the *pyrolite* integration allows more advanced users to leverage the Python ecosystem for end-to-end geochemical data analysis. Comparison between our implementation of the Verma et al. (Schweizerische Mineralogische und Petrographische Mitteilungen, 2003, 83, 197–216) normative calculation and a previous implementation shows that our version is accurate and reliable. *webNORM* offers the geoscience community an accessible solution for calculating normative mineralogy. The algorithm and web application are open-source, and we encourage contributions and adaptations by the community for specific needs and improvements.

KEYWORDS

normative mineralogy, CIPW, mineralogy, web app, python

1 Introduction

Normative mineralogy is an idealised assemblage of minerals in a sample calculated from its bulk geochemical composition and typically an assumption on the specific order in which minerals form. The concept of calculating normative mineralogy for igneous rocks was first introduced by Cross et al. (1902), with the commonly used acronym of "CIPW" normative mineralogy being derived from names of the authors on the paper. Several revisions and modifications have since been made to the method in an attempt to improve the accuracy (Washington, 1918; Kelsey, 1965), to simplify the computational procedure (Pruseth, 2009) and to make it applicable to estimate mineralogy for sedimentary or metamorphic rocks (Imbrie and Poldervaart, 1959; Rosen et al., 2004). The algorithmic nature of the normative calculation means that the procedure is well suited to implementation in computer code and there have

been many such developments in a number of languages (Stuckless, 1983; Cohen and Ward, 1991; Räisänen et al., 1995; Verma et al., 2002; Verma and Rivera-Gómez, 2013; González-Guzmán, 2016). These tools have provided geochemists a convenient way to easily calculate normative mineralogy, however some suffer a number of technical and accessibility shortfalls, including requiring a paid license, being written in older or outdated languages, being closed source with unmaintained code bases, or suffering performance issues particularly when being used on large datasets. Further to this, some require users to be familiar with working in code. Existing domain-focused software tools in the geochemistry and petrology space such as *GeoPyTool* (Yu et al., 2018), *GCDKit* (Janoušek et al., 2006), and *shinyNORRRM* (González-Guzmán et al., 2023) have embraced the use of graphical user interfaces (GUIs) to enhance user experience. Nevertheless, apart from *GeoPyTool*, these tools still require some understanding of the R programming language, as their installation and operation are managed via the R Console. González-Guzmán et al. (2023) includes a comparative study on normative mineralogy outputs from these and other implementations, and showed that *shinyNORRRM* delivered the most consistent results.

Here we present a new implementation of the Verma et al. (2003) normative mineralogy algorithm written in Python, which may be used through a GUI in the form of a web application or in a coding environment with the Python package *pyrolite* (Williams et al., 2020). The algorithm requires 11 major element oxides as the minimum input, and can also accept a set of minor and trace element oxides. It returns a set of up to 31 normative minerals. Two methods for estimating the Fe_2O_3:FeO ratio are also included within the algorithm to account for the fact that geochemical analyses rarely provides both values and the ratio varies depending on the rocks composition.

2 Methods

2.1 Algorithm and python implementation

The Verma et al. (2003) version of the normative mineralogy calculation was used as the basis for this work. Verma et al. made a number of revisions to previous normative calculations, including i) the inclusion of minor and trace elements; ii) correction of errors in formulae and molecular weights for apatite that had existed since Cross et al. (1902); iii) inclusion of the concept of variable molecular weights to account for substitution of trace elements and calculation of free or unused oxides from remaining mass balance of elements. Their revisions resulted in very consistent results compared to previous calculations, with the absolute difference between the sum of the normative minerals and the bulk chemical analyses typically less than 0.002% (Verma et al., 2003). This version of the normative calculation was initially distributed as part of the "Standard Igneous Norm and Volcanic Rock Classification System" (SINCLAS) (Verma et al., 2002). Subsequently, *IgRoCS* incorporated the standard igneous norm procedure with some minor corrections to the normative calculation, and added the ability to process multiple samples in one run (Verma and Rivera-Gómez, 2013).

Python is an interpreted high-level programming language which has become one of the most widely used and popular programming languages (Perez et al., 2011). Over multiple decades a large ecosystem of open-source third-party libraries has been developed, providing a vast range of functionality. Python also works cross-platform, allowing use within Mac, Windows or Linux operating systems. These factors made it a good choice for use in developing a new, easy to use implementation of the normative mineralogy algorithm.

The algorithm steps described by Verma et al. (2003) were translated to Python code and where possible, written to take advantage of the high performance of vectorized functions in *NumPy* (Harris et al., 2020). Our implementation requires the major oxides as a minimum (SiO_2, TiO_2, Al_2O_3, Fe_2O_3, FeO, MnO, MgO, CaO, Na_2O, K_2O, P_2O_5 as weight percent), and optionally minor and trace elements (CO_2, SO_3, S, F, Cl, Ba, Co, Cr, Cs, Li, Ni, Rb, Sr, V, Zr as parts per million). The major oxides are first adjusted to 100% on anhydrous basis, and then a second adjustment to 100% including the minor and trace elements is performed, as per Verma et al. (2003). Trace elements are added to certain major oxides and variable molecular weights calculated. A generalised overview of the algorithm is shown in Figure 1. For further information on the specifics of the normative mineralogy algorithm we refer the reader to the details in Verma et al. (2003). The code developed in this work also leverages many of the functions available within the geochemistry-focused Python package *pyrolite* (Williams et al., 2020), and the final algorithm has itself been included as a submodule function within the latest release of *pyrolite*.

The web application utilises the Python based web framework library *Streamlit*, which is designed to allow for rapid development of web applications. In deploying the implementation as a web application, users are not required to install anything in order to access it (requiring only a modern web browser), updates for bugfixes and new features are easily distributed (users always have the most up to date version), and the maintenance burden for the developer is minimised.

2.2 Iron oxidation ratio

The majority of whole-rock geochemical analyses report the iron content of a sample as total Fe_2O_3 or FeO depending on the method of analysis (XRF, ICP, etc.) and on the convention for the type of rocks analysed and the laboratory undertaking the analysis. However the normative mineralogy calculation requires both to be specified. Verma et al. (2003) discuss a number of different methods to apportion the total Fe to Fe_2O_3 and FeO, and in *IgRoCS* (Verma et al., 2002) three methods are implemented: Le Maitre (1976), Middlemost (1989) and a user defined adjustment for cases where the ratio is known (for example, based on an assumption of lithology or if a titration for Fe has been conducted to measure the ratio of Fe_3+ to Fe_2+). The Middlemost (1989) method defines a Fe ratio based on the total alkali-silica (TAS) classification of each sample. The Le Maitre (1976) method uses an adjustment based on the SiO_2, Na_2O, and K_2O content of the sample, with separateformulas for volcanic (Eq. 1) and plutonic (Equation 2)

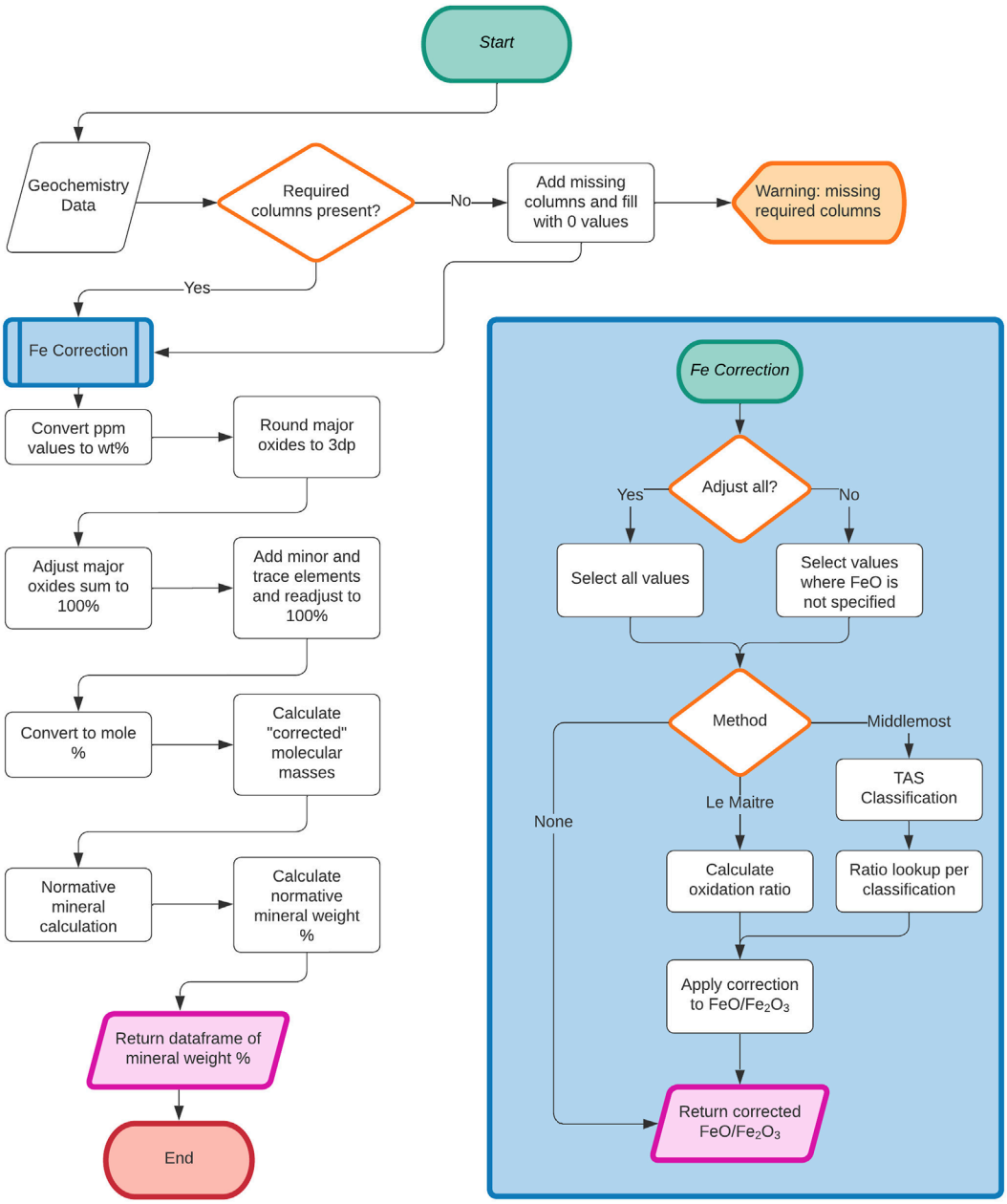

FIGURE 1
A simplified overview of the algorithm used in *webNORM*.

rocks, though only the former is used in *IgRoCS*.

$$\frac{FeO}{(FeO + Fe_2O_3)} = 0.93 - 0.0042 \times SiO_2 - 0.022\,(Na_2O + K_2O) \quad (1)$$

$$\frac{FeO}{(FeO + Fe_2O_3)} = 0.88 - 0.0016 \times SiO_2 - 0.027\,(Na_2O + K_2O) \quad (2)$$

Both the Le Maitre (1976) and Middlemost (1989) method have been implemented in this work, with an additional option in the web app to specify a column in the user's uploaded data that contains a specific Fe adjustment ratio for each sample or a fixed value for all samples.

2.3 Algorithm validation

To confirm that the algorithm is performing as expected, the output from *webNORM* was compared to the output from *IgRoCS*. A test dataset of 551 bulk rock geochemical samples with major oxide values is provided with the *IgRoCS* software and was used as the basis for our comparison. The Fe Adjustment was calculated using both the Le Maitre (1976) and Middlemost (1989) methods for all samples. As *IgRoCS* only uses the volcanic variant (Equation 1) of the Le Maitre (1976) method, the plutonic variant could not be compared.

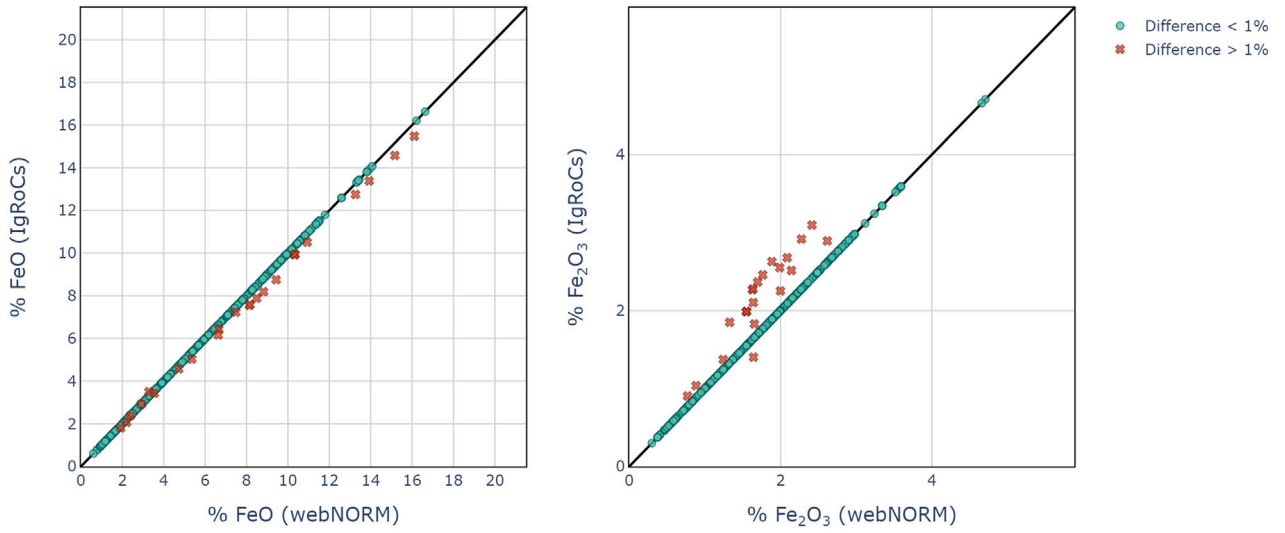

FIGURE 2
Comparison between the *IgRoCS* and *webNORM* implementation of the Middlemost Fe Adjustment method. The red markers indicate samples where the relative error between greater than 1%.

The Le Maitre (1976) adjusted data was used for the input to the normative mineralogy calculations. Normative mineralogy was calculated in both *IgRoCS* and *webNORM* twice: initially with only the major element data, and subsequently incorporating major, minor, and trace elements. The normative algorithm as described by Verma et al. (2003) should result in a normative mineralogy that sums to within ±0.002% of 100%. Herein we use this as a general indicator that our implementation of the Verma et al. (2003) algorithm is performing as intended. For a number of samples, *IgRoCS* does not calculate the normative mineralogy where *webNORM* does, and these samples are excluded from the comparison.

A comparison of the normative sum between *webNORM* and *shinyNORRRM* (González-Guzmán et al., 2023) was also made. Comparison of individual normative mineralogy between *webNORM* and *shinyNORRRM* proved difficult due to *shinyNORRRM* not including minor and trace elements when readjusting to 100%. A dataset of 7,019 samples provided with *shinyNORRRM* ('Deccan.csv'; González-Guzmán et al. (2023)) was used for this comparison.

3 Results

3.1 Fe adjustment

When using the Le Maitre (1976) adjustment, the FeO and Fe_2O_3 values from *IgRoCS* and *webNORM* show a very good agreement (Figure 2). When rounded to 3 decimal places, only one sample had a difference of <0.002% for Fe_2O_3, and none for FeO. The maximum difference between the two calculations for Fe_2O_3 was 0.003%.

For the Middlemost (1989) adjustment method, there were more discrepancies between *IgRoCS* and *webNORM* (Figure 2). When rounded to 3 decimal places, 22 samples had a difference of <0.002% for both Fe_2O_3 and FeO. The maximum difference for Fe_2O_3 and FeO was 0.675% and 0.741% respectively. Further investigation showed that the samples that show the errors all fall close to the TAS class boundaries (Figure 3).

3.2 Normative mineralogy

When using the Le Maitre (1976) Fe adjusted data with only major oxides as the input, the results from *IgRoCS* and *webNORM* generally show a good match. The results from *webNORM* show

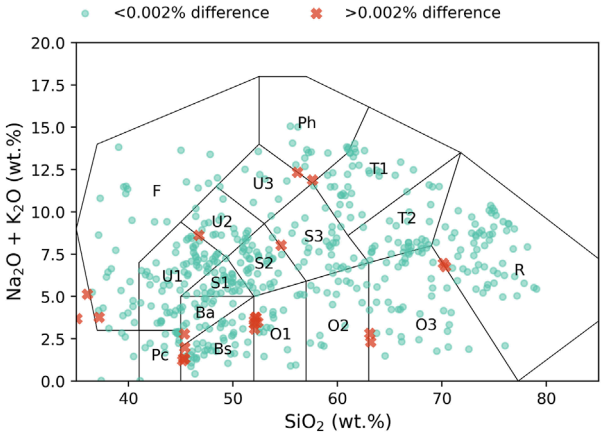

FIGURE 3
Test data plotted on the TAS diagram. Red X markers indicate samples with a relative error between *webNORM* and *IgRoCS* is greater than 1% for Fe_2O_3. Note that these samples fall on the boundaries between the TAS classes.

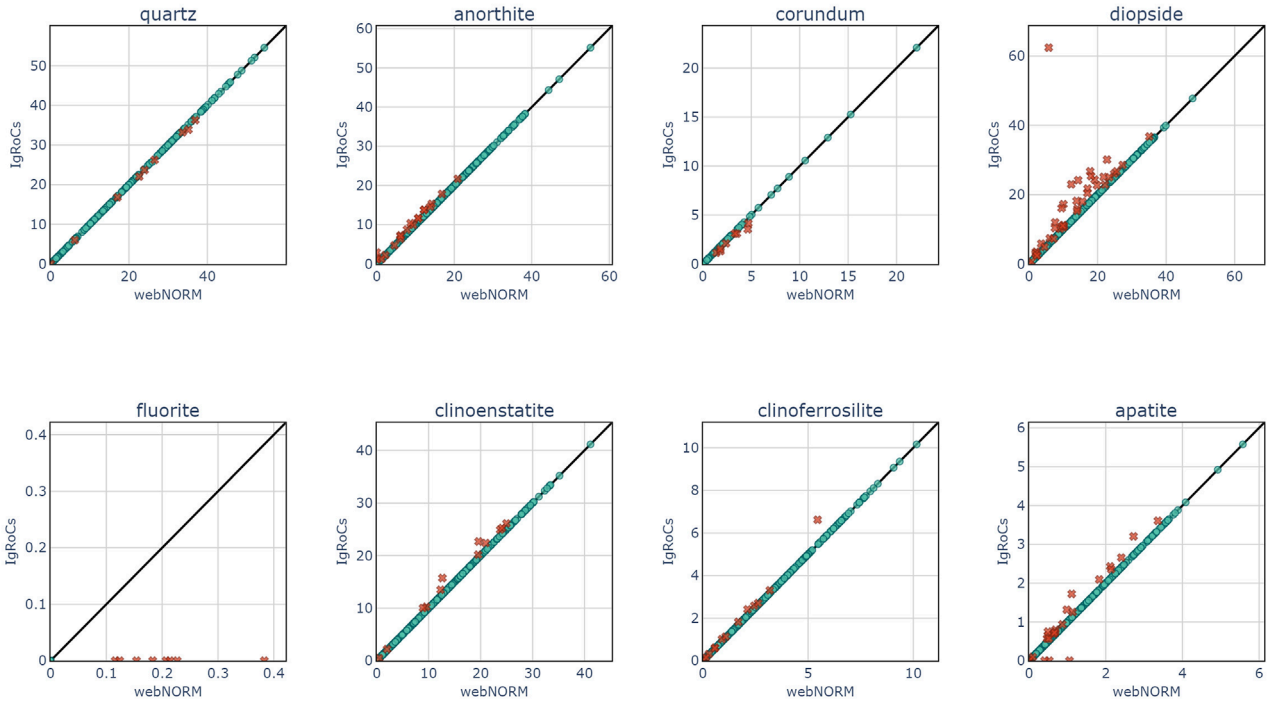

FIGURE 4
Scatter plots for normative mineral percentages as calculated by *webNORM* (*x*-axis) and *IgRoCS* (*y*-axis), using the major, minor and trace element input data. Only minerals which showed a discrepancy between *IgRoCS* and *webNORM* are plotted. A 1:1 line is shown for reference. Red points show samples that deviate from the line by >0.01%.

a very consistent normative sum, with 20 samples deviating from 100% by more than ±0.002%, and a minimum sum of 99.998% and a maximum of 100.003%. This is compared to 58 samples deviating from 100% by more than ±0.002% in the results from *IgRoCS*. For normative diopside, there is a notable discrepancy between *IgRoCS* and *webNORM* in 47 samples. When looking at the deviation from 100% in the normative sum for just these samples, all show a deviation of ±0.002% from 100% in the *webNORM* results compared to a mean of ±3.2% (with a maximum of ±56.7%) for *IgRoCS*.

When calculating normative mineralogy using major and trace elements (with the Le Maitre (1976) Fe adjusted data), 28 samples show a normative sum deviating by more than ±0.002% (−0.09% to +0.003%) compared to 56 samples from *IgRoCS*. The previous issue with diopside still remains, and 120 samples show discrepancy between *IgRoCS* and *webNORM* of >0% in either corundum, apatite, anorthite or fluorite (Figure 2).

The mean deviation of the normative sum from 100% is lower in the *webNORM* results compared to *IgRoCS* for the erroneous samples for apatite (±0.033% vs ±0.691%), fluorite (±0.092% vs ±1.418%) and anorthite (±0.101% vs ±0.598%). For samples showing a discrepancy in normative corundum, the mean deviation of the normative sum from 100% is larger in *webNORM* (±0.081%) than in *IgRoCS* (±0.046%).

When comparing *webNORM* and *shinyNORRRM*, both show a very consistent normative sum that averages 100.000%. The standard deviation of the sum is marginally lower from *shinyNORRRM* (0.001% vs 0.003%). Out of the 7,019 samples tested, the normative sum is > ±0.002% for 67 samples from *shinyNORRRM* compared to 107 from *webNORM*. The minimum and maximum deviation from 100% is −0.004% and +0.004% from *shinyNORRRM* compared to −0.087% and +0.003% from *webNORM*.

4 Discussion

The *webNORM* implementation provides an accurate and convenient means for calculating normative mineralogy, addressing several limitations of existing tools. In general, the outputs from *webNORM* closely match those from *IgRoCS*, suggesting that it accurately reproduces the Verma et al. (2003) normative algorithm.

The results show that the Le Maitre (1976) adjustment in *webNORM* is performing the same as in *IgRoCS*. The discrepancies in the calculation of the Middlemost (1989) Fe ratio occur in samples that plot near the TAS class boundaries (Figure 3). The TAS classifications returned by pyrolite are found to be consistent with the boundaries defined by Middlemost (1994) and we are therefore satisfied that *webNORM* is calcuating the $FeO:Fe_2O_3$ ratio correctly.

When using only major elements for calculations, *webNORM* performs well, with the primary discrepancy between *webNORM* and *IgRoCS* occurring in the normative diopside. Further investigation into the subdivision of diopside into clinoenstatite and clinoferrosilite reveals that their sum does not match the diopside value reported by *IgRoCS*. When the sum of clinoenstatite and clinoferrosilite from *IgRoCS* is used to compare with diopside from *webNORM*, we observe a perfect match. This along with the fact that the normative sum from *webNORM* is closer to 100% for samples

where there is a discrepancy suggests that the issue lies with how *IgRoCS* is handling diopside vs clinoenstatite and clinoferrosilite.

Incorporating minor elements in the calculation reveals discrepancies in corundum, apatite, anorthite, and most noticeably fluorite, where for some samples *webNORM* reports normative fluorite when *IgRoCS* does not (Figure 4). The source of this error is thought to begin with the calculation of apatite which is one of the first normative minerals to be calculated and has the potential to consume fluorine. After this, normative fluorite is calculated, which can consume CaO, so any error in the amount of normative fluorite (caused from an error in normative apatite consuming the available fluorine) will result in an incorrect amount of CaO remaining. Anorthite is the next normative mineral after fluroite to be determined in the algorithm that requires CaO. Further analysis of normative sums for samples containing normative fluorite reveals that the values from *webNORM* are closer to 100% (99.908%) compared to *IgRoCS* (101.297%). This suggests that the issue may lie with the *IgRoCS* implementation. Corundum is calculated in the same step as anorthite, and so whilst it would at first make sense to assume the errors are again carried through, in this case, the deviation of the normative sum from 100% is greater in the *webNORM* results. No obvious errors in the *webNORM* code can be found when referencing the algorithm steps in Verma et al. (2003). The precision of the molecular masses that *webNORM* uses is higher than in *IgRoCS*, so it is possible that these slight variations could propagate through and cause some discrepancy, though it cannot account entirely for the difference.

The comparison of normative sums between *webNORM* and *shinyNORRRM* suggests good consistency. A minority of samples, only 107 out of 7,019 from *webNORM*, did not sum to 100% ±0.002% (maximum deviation of −0.087%), compared to 67 samples from *shinyNORRRM* (maximum deviation of ±0.004%). The level of agreement between *IgRoCS* and *shinyNORRRM* (González-Guzmán et al., 2023) is very good, although we do not know which Fe adjustment method was used and if minor and trace elements were included in the calculation.

Despite the marginal discrepancy in the sum from *webNORM* only existing for a very limited set of samples, it would be prudent to find the source of this minor error in future work. Given that the code is available and open to improvements, any modifications required can be easily implemented, enabling both Python and R users to access highly comparable implementations of the CIPW norm. The latest version of the normative algorithm would then be automatically updated in *webNORM*, underlining one of the benefits of using a web application.

5 Conclusion

webNORM offers the geoscience community an accessible, open-source solution for calculating normative mineralogy for a large number of samples. Comparisons of the output from *webNORM* with the *IgRoCS* implementation gives confidence that the Fe adjustment methods and the normative algorithm is performing as intended. The web application provides a user-friendly interface, requiring only an up to date browser and needs no installation or user updates. The code for both the algorithm and the web application are open source, and improvements, contributions and adaptations for specific use cases are encouraged.

Data availability statement

The code for the web application is available at https://github.com/bomtuckle/webNORM. The code for the normative mineralogy algorithm and the test dataset used in this study is available at https://github.com/morganjwilliams/pyrolite. Documentation for pyrolite is available at https://pyrolite.readthedocs.io.

Author contributions

TB: Conceptualisation (equal), investigation (lead), software (lead), writing—original draft preparation (lead), validation (equal). MW: Conceptualisation (equal), investigation (supporting), validation (equal), software (supporting), writing—review and editing (equal). CN: Conceptualisation (equal), investigation (supporting), validation (equal), software (supporting), writing—review and editing (equal). HH: Supervision, writing—review and editing (supporting). All authors contributed to the article and approved the submitted version.

Funding

A majority of this work has been undertaken by TB during his PhD that is funded by Anglo American Plc. The funder was not involved in the study design, collection, analysis, interpretation of data, the writing of this article or the decision to submit it for publication.

Acknowledgments

Thanks to Jens Andersen for encouraging the development of this paper, and to Chris Yeomans for his mentoring and reviews of initial drafts.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

References

Cohen, D., and Ward, C. R. (1991). SEDNORM-a program to calculate a normative mineralogy for sedimentary rocks based on chemical analyses. *Comput. Geosciences* 17, 1235–1253. doi:10.1016/0098-3004(91)90026-A

Cross, W., Iddings, J. P., Pirsson, L. V., and Washington, H. S. (1902). A quantitative chemico-mineralogical classification and nomenclature of igneous rocks. *J. Geol.* 10, 555–690. doi:10.1086/621030

González-Guzmán, R., Elizondo-Pacheco, L. A., González-Roque, A., Sánchez-Torres, C. E., Cárdenas-Muñoz, K. S., et al. (2023). shinyNORRRM: A cross-platform software to calculate the CIPW norm. *Math. Geosci.* 55, 563–577. doi:10.1007/s11004-023-10052-2

González-Guzmán, R. (2016). Norrrm: A free software to calculate the CIPW norm. *Open J. Geol.* 6, 30–38. doi:10.4236/OJG.2016.61004

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., et al. (2020). Array programming with NumPy. *Nature* 585 (7825), 357–362. doi:10.1038/s41586-020-2649-2

Imbrie, J., and Poldervaart, A. (1959). Mineral compositions calculated from chemical analyses of sedimentary rocks. *J. Sediment. Res.* 29, 588–595. doi:10.1306/74D709A2-2B21-11D7-8648000102C1865D

Janoušek, V., Farrow, C. M., and Erban, V. (2006). Interpretation of whole-rock geochemical data in igneous geochemistry: introducing geochemical data toolkit (GCDkit). *J. Petrology* 47, 1255–1259. doi:10.1093/petrology/egl013

Kelsey, C. H. (1965). Calculation of the C.I.P.W. norm. *Mineralogical Mag. J. Mineralogical Soc.* 34, 276–282. doi:10.1180/MINMAG.1965.034.268.23

Le Maitre, R. W. (1976). Some problems of the projection of chemical data into mineralogical classifications. *Contributions Mineralogy Petrology* 56 (2), 181–189. doi:10.1007/BF00399603

Middlemost, E. A. (1989). Iron oxidation ratios, norms and the classification of volcanic rocks. *Chem. Geol.* 77, 19–26. doi:10.1016/0009-2541(89)90011-9

Middlemost, E. A. (1994). Naming materials in the magma/igneous rock system. *Earth Sci. Rev.* 37, 215–224. doi:10.1016/0012-8252(94)90029-9

Perez, F., Granger, B. E., and Hunter, J. D. (2011). Python: an ecosystem for scientific computing. *Comput. Sci. Eng.* 13, 13–21. doi:10.1109/MCSE.2010.119

Pruseth, K. L. (2009). Calculation of the CIPW norm: new formulas. *J. Earth Syst. Sci.* 118, 101–113. doi:10.1007/s12040-009-0010-0

Räisänen, M. L., Tarvainen, T., and Aatos, S. (1995). Norma - a program to calculate a normative mineralogy for glacial tills and rocks from chemical analysis. *GFF* 117, 215–224. doi:10.1080/11035899509546219

Rosen, O. M., Abbyasov, A. A., and Tipper, J. C. (2004). Minlith - an experience-based algorithm for estimating the likely mineralogical compositions of sedimentary rocks from bulk chemical analyses. *Comput. Geosciences* 30, 647–661. doi:10.1016/j.cageo.2004.03.011

Stuckless, J. S. (1983). A program in Hewlett-Packard BASIC for calculation of CIPW normative minerals using HP-Series 80 computers and VISICALC electronic worksheet. Tech. rep.

Verma, S. P., and Rivera-Gómez, M. A. (2013). Computer programs for the classification and nomenclature of igneous rocks. *Episodes J. Int. Geoscience* 36, 115–124. doi:10.18814/EPIIUGS/2013/V36I2/005

Verma, S. P., Torres-Alvarado, I. S., and Sotelo-Rodríguez, Z. T. (2002). Sinclas: standard igneous norm and volcanic rock classification system. *Comput. Geosciences* 28, 711–715. doi:10.1016/S0098-3004(01)00087-5

Verma, S. P., Torres-Alvarado, I. S., and Velasco-Tapia, F. (2003). A revised CIPW norm. *Schweiz. Mineral. Petrogr. Mittl.* 83, 197–216. doi:10.5169/seals-63145

Washington, H. S. (1918). *Chemical Analyses of igneous rocks*. Published from 1884 to 1913 inclusive. With a critical discussion of the character and use of analyses. By henry stephens Washington. U. S. Geological survey, professional paper 99, Washington, 1917. *Science* 47, 368–370. doi:10.1126/science.47.1215.368

Williams, M., Schoneveld, L., Mao, Y., Klump, J., Gosses, J., Dalton, H., et al. (2020). pyrolite: python for geochemistry. *J. Open Source Softw.* 5, 2314. doi:10.21105/joss.02314

Yu, Q.-Y., Bagas, L., Yang, P.-H., and Zhang, D. (2018). GeoPyTool: A cross-platform software solution for common geological calculations and plots. *Geosci. Front.* 10, 1437–1447. doi:10.1016/j.gsf.2018.08.001