



Geochemical Data Toolkit for Windows written in R language

© 2000–2018

Vojtech Janousek, Czech Geological Survey;

Colin M. Farrow, ex-University of Glasgow;

Vojtech Erban, Czech Geological Survey

Jean-Francois Moyen, Université St. Etienne

<http://www.gcdkit.org>

User's Guide to version 5.0 (March 2018)



GCDkit is a system for handling and recalculation of whole-rock analyses from igneous rocks. It is fully menu driven but, at the same time, can be used in an interactive regime (power users only).

Using *GCDkit*, the data can be loaded, individual analyses grouped into coherent groups or searched according to various criteria. They can be plotted into commonly used classification diagrams (e.g. Harker plots, TAS, AFM, R_1 - R_2) as well as a variety of user-defined plots (binary and ternary plots, multiple plots, spider plots).

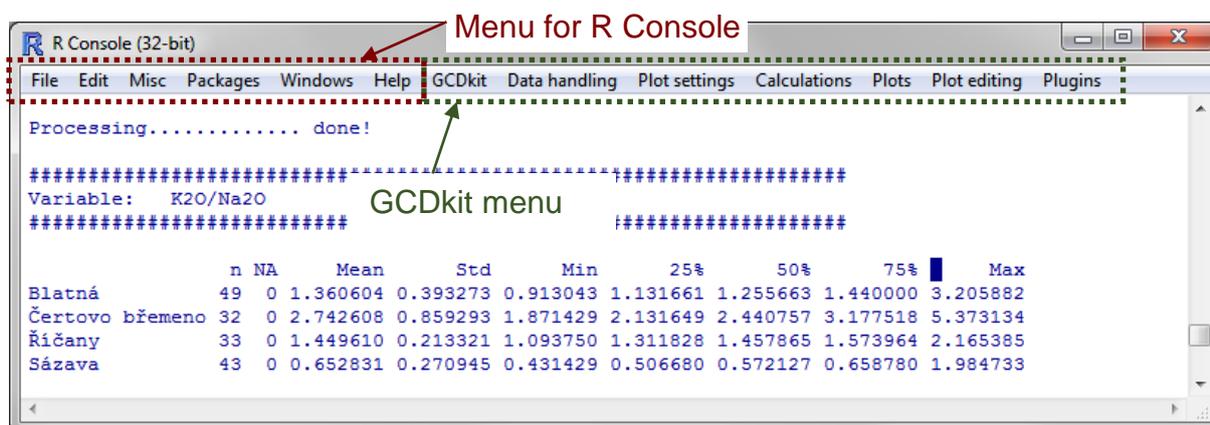
Moreover, some basic statistical methods are implemented (both for the whole data set and individual groups) using a fraction of the power built in R language. Included are the descriptive statistics, boxplots, histograms, principal component and cluster analysis. *GCDkit* also makes possible the most common normative calculations to be performed.

The system is easy to expand by means of the user-defined diagrams or so-called plugins that provide a simple method of adding effortlessly new items to the menus of *GCDkit*. Seven of the latter are included in the current distribution, namely modules for interpretation of the radiogenic isotope data (*SrNd.r*: calculating Sr and Nd initial compositions, Nd model ages, plotting isochrons etc.), modules estimating saturation temperatures of accessory phases (*saturation.r*: zircon, apatite and monazite; *Rt_saturation.r*: rutile), temperatures of granitic magmas production (*JungAlTitemp.r*), isocon/wedge plots (*isocon.r*), a module assessing the magnitude of the REE tetrad effect (*tetrad.r*) and *clr* transformation of (closed) compositional data (*disclosure.r*).

Important!

This *GCDkit* was built in **R version 3.4.3** for Windows. Also this manual applies in its entirety only to the Windows (GUI) version of *GCDkit*. Due to the quick developments in the R project, the correct function with any other version cannot be guaranteed!

The platform-independent (i.e. non-Windows) version of *GCDkit* is still highly experimental. Many features are not available (most importantly much of the graphical user interface, GUI, including the menus), others may work unpredictably. Please use sensibly, at your own risk. For further information, see <http://blog.gcdkit.org>.



TECHNICAL NOTES

- After double clicking the icon *GCDkit* opens “*R Console*”, a text window serving for entry of commands as well as display of textual output. In addition, during the session are typically opened one or more graphical windows.
- Please note that the *GCDkit* menu appears right of the menus of the R system itself (see Figure above).
- The interactions built in many plots (most commonly data point identification) can be stopped from the menu that appears after pressing the right mouse button.
- If experiencing problems, the computation/plotting can be interrupted anytime by hitting *Escape* key or from the menu (*Misc/Stop current computation*).
- The errors encountered while running the *GCDkit* system and displayed in the Console are (mostly) not fatal. In most cases it means that the command can be re-run (one can simply press the up/down arrows in the Console to scroll through the commands history), entering/modifying the parameters that caused the crash. If the problem persists, record the details (including the version of *GCDkit*, R and your operation system) and send the data file to us, so we can track down and fix the bug.
- *GCDkit* requires the R packages *base*, *stats*, *methods*, *utils*, *graphics*, *MASS*, *grid*, *foreign* and *lattice* to be installed (but this is normally the case).
- Highly recommended are also the package *RODBC* enabling import and export into the MS Excel, Access and dBase (DBF) formats. Moreover, the package *sp* has to be present as it is used for classification purposes, and the package *R2HTML* that provides output to HTML format. The package *XML* provides access need to read input from online databases, such as EartChem. All the three can be installed from our binary distribution of *GCDkit* (i.e. that with the Windows installer; Internet connection required). But if need be, they can be uploaded via the menu *Packages/Install package(s)...* (you need to be online). Offline installation is also possible, though: zip file downloaded from the CRAN site (<http://www.r-project.org>) can be installed via menu *Packages/Install package(s) from local zip files...*
- **Note that recommended systems** to run *GCDkit* are Windows 10/8/7/Vista/XP.
- It is always a good idea to close the unnecessary graphical windows, for instance using the function *graphicsOff()* or the corresponding item from the menu *GCDkit*.
- More immediate response is obtained when print buffering (*Misc/Buffered Output*) is disabled.
- *PDF* or *PostScript* output should be given preference over *WMF* (Windows metafiles) or copying via clipboard, as the latter two methods lead sometimes to distortions of the

graphs. *PDF* has an extra advantage as it preserves the semitransparency information if used for plotting symbols or fields. For saving simply right-click the graphical window and select the desired format.

- The use of special symbols or accented characters (such as in some East European languages) is mostly okay but it may also sometimes cause unexpected problems. So please be sensible, especially in the variable names.

Quick guide to data files

GCDkit requires plain text data files, delimited by tabs, commas or semicolons (delimiter is recognised automatically, as is a decimal point/comma). See the testing files *sazava.data* and *cbp_iso.data* that come with the *GCDkit* distribution for examples of valid data files with whole-rock geochemical and Sr–Nd isotopic data, respectively.

As an (less forgiving) alternative, the data can be pasted, via clipboard, from any Windows-based software, including Excel. Be careful, though, as the precision of the data imported will be that *displayed* in the original Excel file!

If library *RODBC* has been installed, and R is run in 32-bit mode, the *GCDkit* first attempts to establish ODBC connection to the selected file, and open it as a dBase III/IV (*.DBF), Excel (*.XLS, *.XLSX) or Access (*.MDB) format. The DBF files are used to store data by other popular geochemical packages, such as *IgPet* (Carr, 1995) or *MinPet* (Richard, 1995) as well as output of the *NAVDAT* online database (<http://www.navdat.org>).

Without ODBC support, the *GCDkit* can read plain text files, including the variants represented by the *NewPet* (*.ROC) (Clarke et al., 1994) and *PetroGraph* (*.PEG) (Petrelli et al., 2005) as well as the outputs from WWW-based databases such as *GEOROC* (<http://georoc.mpch-mainz.gwdg.de/georoc>) or *PETDB* (<http://www.petdb.org>).

New feature is a possibility of searching online the *EarthChem* database (<http://www.earthchem.org>) directly from within the *GCDkit* menu.

As a rule of thumb, if encountering any problems with importing data, please use import from text file or *via* clipboard, which are much more robust than any other method.

For text files and spreadsheets, the first row should contain names of the data columns (except for the first column that is automatically assumed to contain the sample names). Hence the first line of the text file may (or may not) have one item less than the following ones.

Missing values (NA) are allowed anywhere in the body of the data file; values equalling to zero, or any of 'NA', 'N.A.', '-', 'b.d.', 'bd', 'b.d.l.', 'bdl' and 'n.d.' are also treated as such. While loading Excel files, the values *#WHATEVER!* (Excel error messages) are also replaced by 'NA'.

If desired, negative numbers and values '< x' (used by some authors to indicate items below detection limit) can be either replaced by their half (i.e. interpreted as a half of the detection limit) or 'NA'. User is prompted which of these options he prefers.

Alternatively, negative values can be imported in original form (as may be desirable e.g. for $\delta^{18}\text{O}$ values).

The data rows start with sample name and do not have to be all of the same length (the rest of the row is filled by 'NA' automatically).

A column named "Symbol" (if any) is taken as containing plotting symbols. They can be specified either as single character strings or numeric codes, whose overview is given below or can be obtained by invoking the menu item *Data handling/Show available symbols*.

Another column, whose name is “Colour” or “Color” (if any, capitalization does not matter) may contain codes (1–49) or English names for the plotting colours. There are 657 of the latter, see Help, invoke menu *Data handling/Show available colours* or type *colours()* into the Console.

If specifications of both the plotting symbols and colours are missing completely, and at least one non-numeric variable is present, the user is prompted whether he does want to have the symbols and colours assigned automatically, according to the individual values (levels) of the selected label. Otherwise default symbols (empty black circles) are used.

Column named “Size” or “cex” may contain relative size for each of the plotting symbols (use 1 for standard size, 2 for double and 0.5 for half the size).

Apostrophes and other special characters should be avoided in the sample/variable names. The data files are practically freeform, i.e. no specified oxides/elements are required and no exact order of these is to be adhered to. Analyses can contain as many numeric columns as necessary; the names of oxides and trace elements are self-explanatory (e.g. ‘SiO2’, ‘Fe2O3’, ‘Rb’, ‘Nd’). Total iron, if given, should be expressed as ferrous oxide (‘FeOt’, ‘FeOT’, ‘FeOtot’, ‘FeOTOT’ or ‘FeO*’) or ferric oxide (‘Fe2O3t’, ‘Fe2O3T’, ‘Fe2O3tot’, ‘Fe2O3TOT’ or ‘Fe2O3*’). Structurally bound water can be named ‘H2O.PLUS’, ‘H2O.P’, ‘H2O+’, ‘H2OPLUS’ or ‘H2O_PLUS’.

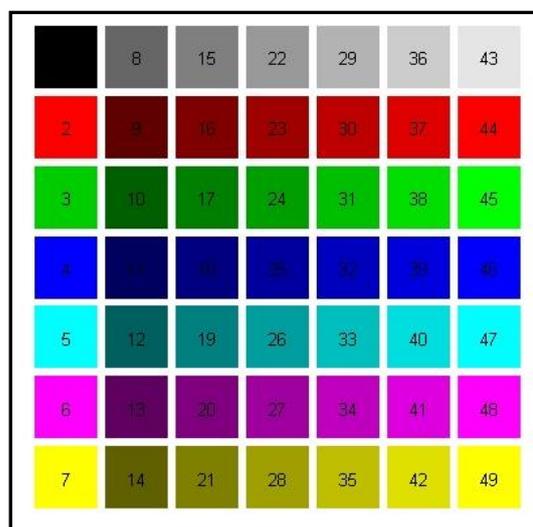
For isotopic ratios, please use simple forms such as ‘87Rb/86Sr’, ‘87Sr/86Sr’, ‘147Sm/144Nd’, ‘143Nd/144Nd’ or ‘delta18O’.

Note that names of variables are **case sensitive** in R. However, any of the fully upper case names of the oxides/elements that appear in the following list are translated automatically to the appropriate capitalization:

SiO2, TiO2, Al2O3, Fe2O3, FeO, MnO, MgO, CaO, Na2O, FeOt, Fe2O3t, Li2O, mg#, Li, Rb, Cs, Be, Sr, Ba, Sc, Y, Ti, Zr, Hf, V, Nb, Ta, Cr, Mo, W, Re, Ru, Os, Co, Rh, Ir, Ni, Pd, Pt, Cu, Ag, Au, Zn, Cd, B, Ga, In, Tl, C, Ge, Sn, Pb, P, As, Sb, Bi, S, Se, Te, F, Cl, Br, I, At, La, Ce, Pr, Nd, Pm, Sm, Eu, Gd, Tb, Dy, Ho, Er, Tm, Yb, Lu.

Also trailing spaces in the end of the column names are disposed of. However note that, with some exceptions such as importing from online databases, **no duplicated column or sample names are allowed!** The correct syntax is checked upon loading.

After loading, all the completely empty columns are removed first. In the text files, any line starting with the hash symbol (#) is ignored and can be used to introduce comments or to prevent the given analysis from loading temporarily. Any non-numeric items found in a data column with one of the names listed in the above dictionary are replaced by ‘NA’ automatically. At the next stage all fully numeric data columns are stored in a numeric data matrix *WR*. Decimal commas, if present in text file, are converted to decimal points. For any



missing major- and minor-element data (SiO₂, TiO₂, Al₂O₃, Fe₂O₃, FeO, MnO, MgO, CaO, Na₂O, K₂O, H₂O.PLUS, CO₂, P₂O₅, F, S), an empty (NA) column is created automatically.

The remaining, that is all at least partly textual data columns are transferred into a data frame *labels*, in which are also stored codes of plotting symbols, their colours and sizes.

Subsequently, several extra parameters are calculated (if not provided already by the user), namely concentrations of some elements from oxides (P, K, Ti, Ba, Sr) in ppm, total iron expressed as ferrous oxide (FeOt), Shand's indexes in molar % (A/NK, A/CNK):

$$A / NK = \frac{Al_2O_3}{Na_2O + K_2O} [mol. \%]$$

$$A / CNK = \frac{Al_2O_3}{CaO + Na_2O + K_2O} [mol. \%]$$

two variants of Mg numbers:

$$mg \# = 100 \frac{MgO}{FeO + MgO} [mol. \%]$$

$$Mg \# = 100 \frac{MgO}{FeOt + MgO} [mol. \%]$$

and K₂O/Na₂O ratios by weight.

At the same time are the analyses recast to millications (data matrix *milli*) and major-element analyses to anhydrous basis (data matrix *WRanh*).

Data editing and grouping

The *GCDkit* contains basic functions for data input and output, such as pasting, loading and saving or union of two data files (either appending new samples or new data columns for samples already present in the system). If need be, the data set can be modified in a simple spreadsheet. Subsets of the data can be displayed/selected, data columns deleted or new ones appended. In addition, there is a possibility of editing a single value (in the R jargon a *level*) of the given label (a *factor*).

Important feature of data handling in the *GCDkit* is **grouping**. Each of the samples can be assigned to one of the groups that are subsequently utilised by some statistical and plotting functions. Groups can be defined on the basis any of the labels (locality, rock type...), a value of a chosen numerical variable, position in classification diagram (e.g., TAS) or cluster analysis. The information concerning current grouping is stored in a vector *groups*; default grouping after loading a new data file or selection of subset is on plotting symbol.

Searching and subsetting in GCDkit

Great emphasis in the *GCDkit* is on managing the data, in particular **searching and selecting subsets**.

1. **Searching** is useful for temporary selections of samples that are, for instance, to be displayed or plotted in a diagram. In this case, the data stored in memory are fully unaffected. Search dialog appears automatically within each relevant function (plots, norms etc.)

2. Upon **selection of a subset** (via menu items *Select subset by sample name or label*, *Select subset by range*, *Select subset by Boolean*), the data in memory are replaced by their part (subset) fulfilling the given criteria. Still, the changes are temporary as each time a file is loaded, a backup of the data is kept that can be restored anytime using the menu *Select the whole data set* (connected to the function *selectAll*). However, in that case all changes made e.g. to plotting symbols, grouping etc. would be lost.

The searching and subsetting functions employ powerful regular expressions (see R help pages for *regex*). Syntax of search patterns employed in searching and subsetting is dealt with in a considerable detail in the corresponding manual/help entries, here are shown only examples.

a) By sample name or label

These queries of a single textual column employ regular expressions [2]. Put in simple terms, most characters, including all letters and digits, are regular expressions that match themselves. However, metacharacters with a special meaning ('?', '+', '{', '|', '(', and ')') must be preceded by a backslash. Two regular expressions may be joined by the infix operator '|'; the result then matches either of them (logical **or**).

Regular expression	Matches
.	Any character except NULL
^	Beginning of the expression
\$	End of the expression
[]	Any of the characters given in square brackets
[m-n]	Any character in the range given by <i>m</i> and <i>n</i>

A regular expression may be followed by one of several repetition operators.

Repetition operator	The preceding item will be matched
?	At most once (i.e. is optional)
*	Zero or more times
+	One or more times
{n}	Exactly <i>n</i> times
{n,}	At least <i>n</i> times
{n,m}	At least <i>n</i> times, but not more than <i>m</i> times

Example – Subset by label

Let us consider that the searched field corresponds to the following localities:

Mull, Rhum, Skye, Coll, Colonsay, Hoy, Westray, Sanday, Stronsay, Tiree, Islay.

Search pattern	Matches
ol	<u>Coll</u> , <u>Colonsay</u>
n.a	<u>Colonsay</u> , <u>Sanday</u> , <u>Stronsay</u>
^s	<u>Skye</u> , <u>Sanday</u> , <u>Stronsay</u>
e\$	<u>Skye</u> , <u>Tiree</u>
[ds]ay	<u>Colonsay</u> , <u>Sanday</u> , <u>Stronsay</u>
[p-s]ay	<u>Colonsay</u> , <u>Westray</u> , <u>Stronsay</u>
ol oy	<u>Coll</u> , <u>Colonsay</u> , <u>Hoy</u>
l{2}	<u>Mull</u> , <u>Coll</u>

Example – Subset by name

The sample names are: B1-1, B1-3, Koz-1, Koz-2, Koz-5, Koz-11, KozD-1, Ri-1.

Search pattern	Matches
oz-[1-3]	Koz-1, Koz-2, Koz-11
oz- B1-	B1-1, B1-2, B1-3, Koz-1, Koz-2, Koz-5, Koz-11

b) By range

In this case the search pattern is treated as a selection of sample sequence numbers (effectively a list separated by commas that may also contain ranges expressed by colons).

Example – Subset by range

Search pattern	Matches
1:5	First to fifth samples in the data set
1,10	First and tenth
1:5, 10:11, 25	Samples number 1, 2, ... 5, 10, 11, 25

3. By Boolean conditions

The search pattern, representing a Boolean condition, may contain names of both the textual and numeric columns as well as comparison operators '<' (lower than), '>' (greater than), '<=' (lower or equal to), '>=' (greater or equal to), '=' or '==' (equal to), '!=' (not equal to). The character strings should be quoted. Two or more conditions can be combined together by logical **and** (expressed as '.and.', '.AND.', '&'), **or** ('.or.', '.OR.', '|') and brackets.

Example – Subset by Boolean conditions

Search pattern	Matches
Locality="Rhum"	All analyses from Rhum
Locality="Rhum".and.SiO2>65 Locality="Rhum".AND.SiO2>65 Locality="Rhum"&SiO2>65	Analyses from Rhum with SiO ₂ greater than 65 (all three expressions are equivalent)
MgO>10&(Locality="Skye" Locality="Islay")	Analyses from Skye or Islay with MgO greater than ten
MgO>=10&Locality!="Skye"&Locality!="Islay"	Analyses from anywhere except Skye and Islay with MgO at least ten

Using multiple datasets

As a relatively new feature, *GCDkit* has introduced a possibility of keeping several datasets in memory concurrently. Only one among them at the time is designated as the so-called current dataset (that one whose name is indicated in the title bar of the R Console Window) and used for calculations and plotting. The current dataset can be chosen from the menu *GCDkit/Switch/restore a dataset* (attached to the function `selectDataset`) or directly by the function `peekDataset`.

New dataset, as soon as it is loaded from file, clipboard or Internet connection, is stored in the memory under its root filename. If needed, a time stamp is attached to make its name unique.

Modified datasets can be stored/replaced anytime from the menu *GCDkit/Store the current dataset* (function `pokeDataset`).

All the datasets stored in memory, apart from the current one, can be deleted at once (menu *GCDkit/Purge stored datasets*).

Moreover, any of the datasets stored in memory can be added to the current standard (Figaro-compatible) binary/ternary plot or spiderplot, or their plate. This is done either in the foreground or in the background (menus *GCDkit/Overplot another dataset* and *GCDkit/Underplot another dataset*, respectively). These tools allow

effortless comparisons, say with real-world data from a similar pluton or a numeric matrix spanning from petrogenetic modelling.

Choosing numeric variable(s), calculation core routine

Many functions (e.g. binary plots, some statistics) require a single numeric variable to be chosen first. The easiest way is to type in the name of the numerical column (e.g., 'SiO2') or its sequence number (2 for the second column). However, it is not necessary to enter the name in its entirety. Only a substring that appears somewhere in the column name or other forms of regular expressions can be specified.

If the result is ambiguous, the correct variable has to be picked manually from a drop-down list of the multiple matches. Ultimately, empty response invokes list of all available variables.

As an useful alternative in many instances, the system allows to enter a formula that can involve any combination of names of existing numerical columns with constants, brackets, arithmetic operators $+*$ / $^$ and R functions. The most useful of these are 'sqrt' (square root), 'log' (natural logarithm), 'log10' (common logarithm) and 'exp' (exponential function). Potentially useful can be also 'min', 'max' (extremes), 'length' (number of elements/cases), 'sum', 'mean', and 'prod' (product of the elements). In any case, no one prevents the advanced users from writing their own functions that can be invoked here.

When multiple columns are to be selected (e.g. for Harker plots or correlation plots), the easiest way is to type in directly their names separated by commas. Alternatively can be used a comma-delimited list of sequence numbers that may also contain ranges expressed by colons. Also user-defined or built-in lists can be employed, such as 'LILE', 'REE', 'major' and 'HFSE' or their combinations with the column names.

Calculations

All the calculation algorithms produce output temporarily stored in the memory, in the variable (a vector, matrix or list) *results*. Before they are rewritten by the next calculation procedure, they can be saved or appended to the current dataset for further processing (e.g., CIPW normative albite, quartz and orthoclase to be plotted in the Ab–Q–Or ternary).

Most of the normative recalculation algorithms have been adopted from an earlier QuickBasic program NORMAN (Janoušek, 2001). Currently are available R modules for calculation of the CIPW norm, including the modification with biotite and hornblende (Hutchison, 1974; 1975), Niggli Catanorm (Hutchison, 1974 and references therein), Niggli's cationic values (Niggli, 1948), multicationic parameters of the French authors (De La Roche et al., 1980; Debon and Le Fort, 1983; 1988), improved Mesonorm for granitoid rocks (Mielke and Winkler, 1979). As an optional outcome of the latter, implemented is a Q'–ANOR binary chemically approximating the modal QAPF classification for the igneous rocks (Streckeisen and Le Maitre, 1979).

The *GCDkit* (so far only in Windows) provides a menu-driven interface to (a fraction of) statistical functions built in R. These include simple descriptive statistics, histograms and boxplots (for the whole data set or individual groups), correlation plots as well as more sophisticated methods of multivariate statistics (cluster analysis and principal components).

Plotting

The plotting symbols and colours can be allocated, if need be independently of each other, according to current grouping or any of the labels. If two distinct criteria for symbols and colours are chosen, two legends are built (see <http://blog.gcdkit.org/2017/04/legends.html>).

The *GCDkit* produces publication quality plots in PDF or PostScript formats that can be easily imported into other graphical or DTP package for further editing (simply right-click the graphical window and select the desired format). All the plots can be saved simultaneously to PDF (*Save all graphics to PDF*) or PostScript (*Save all graphics to PS*).

The strength of *GCDkit* consists in the wealth of built-in plots most of which (the stand alone, i.e. not multiple, diagrams) are defined as templates compatible with *Figaro*, a set of graphical utilities for R.

Figaro provides means to create figure objects, which contain both the data and methods to make changes to the figure (via the menu *Plot editing*). So, for example, the title can be changed or the histogram fill colour altered and any changes are automatically made visible on interactive devices. In addition one can zoom in and out of the data. *Figaro* objects currently permit the editing of the text, font, size and colour of the main title, sub title and axis labels; colour, size, and symbol of points; colour, line type, and width of lines. Thus *Figaro* provides a degree of interactive editing before committing to hardcopy.

From version 3.0, *GCDkit* employs the concept of the so-called plates, i.e. collections of two or more stand-alone plots in a single plotting window. The examples are Harker plots, or diagrams using the same *Figaro* template plotted for individual data groups (menu item “*Multiple plots by groups*” – function *figMulti*), based e.g. on the petrographic type they come from. Alternatively, plates can be fully user-defined and contain any mixture of *Figaro*-compatible plots, such as binary plots, ternary plots, classification diagrams or spiderplots.

A single plot can be selected by graphical cursor from the plate already plotted (just right click the plot window and invoke the menu option *Select slot – Select another slot*) and subsequently edited in the manner previously available exclusively to stand-alone *Figaro*-compatible plots (submenu of the same menu called *Plot editing*). Moreover, properties of the whole plate or all its diagrams can be changed simultaneously (submenu *Plate editing*). For instance, the whole plate can be set to black and white, the font size of axis labels can be altered, scaling of the common axis on Harker plots set, or the minima of all y axes of binary plots set to 0.

NOTE Use the *Plate redraw*, or dedicated *Export to PS (Colour)* and *Export to PS (BW)* menu items from submenu *Plate editing* to produce “clean” version of your plate before exporting, if further editing is desired in a specialized graphical program (CorelDraw, Adobe Illustrator...).

Where to go next?

Arguably a very useful introduction to using R with, or without, *GCDkit* in interpretation and modelling of whole-rock geochemical data provides our monograph (Janoušek *et al.*, 2016). Courtesy of the publisher, the Appendices explaining the R syntax and the workings of the *GCDkit* system, are freely available online at: <http://blog.gcdkit.org/2015/10/>. But if you want to support our efforts, and learn more, buy a copy!

Future

Our long-term aim is to build a single, coherent and eventually platform-independent system with high-level plotting capabilities for interpretation of whole-rock geochemical data that would be straightforward to use by ordinary users but, at the same time, easily expandable by the more demanding ones. In a nearest future *GCDkit* should be expanded by functions for direct and inverse modelling of the main petrogenetic processes (fractional crystallization, AFC, binary mixing). These are ready but not yet stable enough to be released.

Enjoy! On behalf of the authors

Vojtech Janousek
vojtech.janousek@geology.cz

12 March 2018

<http://www.gcdkit.org>

References

- Carr, M. (1995). *Program IgPet*. Terra Softa, Somerset, New Jersey, U.S.A.
- Clarke, D., Mengel, F., Coish, R. A. & Kosinowski, M. H. F. (1994). *NewPet for DOS, version 94.01.07*. Department of Earth Sciences, Memorial University of Newfoundland, Canada.
- De La Roche, H., Leterrier, J., Grandclaude, P. & Marchal, M. (1980). A classification of volcanic and plutonic rocks using R_1R_2 -diagram and major element analyses – its relationships with current nomenclature. *Chemical Geology* **29**, 183–210.
- Debon, F. & Le Fort, P. (1983). A chemical–mineralogical classification of common plutonic rocks and associations. *Transactions of the Royal Society of Edinburgh, Earth Sciences* **73**, 135–149.
- Debon, F. & Le Fort, P. (1988). A cationic classification of common plutonic rocks and their magmatic associations: principles, method, applications. *Bulletin de Minéralogie* **111**, 493–510.
- Hutchison, C. S. (1974). *Laboratory Handbook of Petrographic Techniques*. New York: John Wiley & Sons.
- Hutchison, C. S. (1975). The norm, its variations, their calculation and relationships. *Schweizerische mineralogische und petrografische Mitteilungen* **55**, 243–256.
- Janoušek, V. (2001). Norman, a QuickBasic programme for petrochemical re-calculation of whole-rock major-element analyses on IBM PC. *Journal of the Czech Geological Society* **46**, 9–13.
- Janoušek, V., Farrow, C. M. & Erban, V. (2006). Interpretation of whole-rock geochemical data in igneous geochemistry: introducing Geochemical Data Toolkit (GCDkit). *Journal of Petrology* **47**, 1255–1259.

- Janoušek, V., Farrow, C. M., Erban, V. & Trubač, J. (2011). Brand new Geochemical Data Toolkit (GCDkit 3.0) – is it worth upgrading and browsing documentation? (Yes!). *Geologické výzkumy na Moravě a ve Slezsku* **18**, 26–30.
- Janoušek, V., Moyer, J. F., Martin, H., Erban, V. & Farrow, C. (2016). *Geochemical Modelling of Igneous Processes – Principles and Recipes in R Language. Bringing the Power of R to a Geochemical Community*. Springer-Verlag, Berlin, Heidelberg, pp 1–346. doi: 10.1007/978-3-662-46792-3
- Mielke, P. & Winkler, H. G. F. (1979). Eine bessere Berechnung der Mesonorm für granitische Gesteine. *Neues Jahrbuch für Mineralogie, Monatshefte* 471–480.
- Niggli, P. (1948). *Gesteine und Minerallagerstätten*. Basel: Birkhäuser.
- Petrelli, M., Poli, G., Perugini, D. & Peccerillo, A. (2005). PetroGraph: A new software to visualize, model, and present geochemical data in igneous petrology. *Geochemistry Geophysics Geosystems* **6**, 1–15.
- Richard L. R. (1995). *MinPet: Mineralogical and Petrological Data Processing System, Version 2.02*. MinPet Geological Software, Québec, Canada.
- Streckeisen, A. & Le Maitre, R. W. (1979). A chemical approximation to the modal QAPF classification of the igneous rocks. *Neues Jahrbuch für Mineralogie, Abhandlungen* **136**, 169–206.