Software review: The KNIME workflow environment and its applications in Genetic Programming and machine learning

[1,2]Steve O'Hagan & [1,2,*]Douglas B. Kell

[1]School of Chemistry and [2]The Manchester Institute of Biotechnology, The University of Manchester, 131 Princess St, Manchester M1 7DN, UK.

*corresponding author

Tel: +44 161 306 4492 dbk@manchester.ac.uk http://dbkgroup.org/

Abbreviated title: The KNIME workflow environment

Introduction

Software comes in various forms, from the hair shirt style of the command line to fully blown, GUI-based commercial offerings. The former tends to give its users more control, but disenfranchises many other potential users who cannot themselves program yet who might otherwise benefit from it. A kind of halfway house is represented by software environments that provide both flexibility (power) and ease of use. A particular subset is represented by Workflow environments, in which loosely coupled, individual processing nodes can be 'bolted together' to permit complex computational operations. Taverna (Wolstencroft, Haines, Fellows, Williams, Withers, Owen, Soiland-Reyes et al. 2013) (http://www.taverna.org.uk/) is a very well known scientific workflow system, especially in bioinformatics. It is a fully open environment, freely available, and workflows can be shared via its sister site myExperiment http://www.myexperiment.org/. It has some extensions for cheminformatics (Kuhn, Willighagen, Zielesny, Steinbeck 2010). A particular strength is the means by which it can use Web services to link federated Web-based resources, a particular feature of bioinformatics.

For cheminformatics (see also (Mazanetz, Marmon, Reisser, Morao 2012)), we have been using the KNIME environment (O'Hagan, Kell 2015; O'Hagan, Swainston, Handl, Kell 2015). KNIME stands for the Konstanz Information Miner (Berthold, Cebron, Dill, Gabriel, Kötter, Meinl, Ohl et al. 2008) and is pronounced 'NIME' (with a silent 'K', like knife). It is freely available via www.knime.org for unrestricted use on the desktop (and with versions that operate under MS-Windows, Linux and Mac OSX). As datasets may be large, a reasonably beefy machine is advised. The download itself is just over 1 Gb, and installation is both automated and simple. (There is an otherwise identical commercial offering available at www.knime.com; its chief differences are that the environment may be extended to servers, and to clusters that run the Sun Grid Engine.) Our main experience is with the Windows desktop version. Under the hood, KNIME is built on the ECLIPSE environment, with Java as its main internal language. Many other languages can be used with it, however, as detailed below.

The interface is configurable, but the more-or-less default version is shown in Figure 1. This shows a workflow (written by SO'H) that takes a ChEMBL dataset (https://www.ebi.ac.uk/chembl/target/inspect/CHEMBL4333) of drug binding to a particular receptor, and compares three data analytical methods (GP, random forests and partial least

squares regression). To create the workflow, nodes are dragged from the node repository, dropped into the main workflow window, and linked using the mouse to join their output and input ports (shown as small triangles). Those nodes in the node repository that contain letters as typed into the appropriate window are shown. A vast number of nodes exist, including general ones for data and text mining, statistics and machine learning (at least one of KNIME's originators has a background in fuzzy logic), with other more specialised ones for cheminformatics, mass spectrometry, image processing, time series analysis, and so on. Trusted users can contribute new nodes or entire collections; these are optionally downloadable and/or updated nightly. A right click on each node allows one to configure it. Thus a node for reading in an MS-Excel file would require information on the filename, whether the first row defines column headers, and so on. A second right-click allows a successfully configured node to be executed. If it does so, the 'traffic light' system shown on each node goes green, as in Figure 1. Each node can be annotated with a simple description of its function, again as in Figure 1. Large and complex workflows do not necessarily fit legibly into the main window, and a navigation window appears below under 'outline'. Left clicking on a node provides a description of what it does (and, if the description is well written, how to configure it).

The particular beauty of KNIME for cheminformaticians is that a great many tools have been produced that allow standard procedures to be implemented without additional programming, e.g. converting chemical structures to computer-readable encodings. We regularly use the RDKit (e.g. (Riniker, Landrum 2013)) nodes. Most nodes shown in the figure come with the vanilla-flavoured version of KNIME and/or the many free add-ons. One such is the 'Tree Ensemble Learner', which is from KNIME labs. However, for programmers it is possible to create nodes of arbitrarily complex function by 'wrapping' code in any nodes that 'understand' (parse) one of a number of languages, such as Matlab, R, Perl and Python (native nodes use a freely available SDK and are in Java). Thus the PLS regression node simply wraps a call to a standard R library, while the GP metanode wraps a fairly standard but detailed GP written (by SO'H) in Python. This metanode can easily be configured by its user. The chief disadvantage of this implementation is that one cannot see the GP running, but its progress can be recorded post hoc and exported (here to show fitness vs time for training and validation sets). The final two windows show a list of available workflows (top left) and a list of frequently or recently used nodes.

To give an idea of speed, to write the GP metanode took a few days. Given this, however, to assemble the workflow of Figure 1 took just a couple of hours, and to run it for 1000 GP generations with a population size of 200 and including niching (the slow step) took only 20 minutes on a standard desktop PC.

Where KNIME and related workflow systems come to the fore is in their ability to let 'naïve' users (re)create complex analyses just by reusing existing nodes or whole workflows, and even just by changing file names for instance. Thus some rather sophisticated workflows that compared the structures of 'natural' human metabolites with those of marketed drugs and other chemicals, outputting the analysis in the form of a 2D-biclustered heatmap (O'Hagan, Kell 2015; O'Hagan, Swainston, Handl, Kell 2015), were actually just a single workflow with simple filename changes. Given the base workflow, a novice could learn to do these changes in less than an hour, though of course time spent learning to create new workflows can be almost limitless. There are also API links to commercial software such as the Spotfire visualisation system.

Conclusion

Overall, this is a very sophisticated and professional piece of software. Because of its flexibility, it is nowadays our chief cheminformatics workhorse, and voting with one's feet is surely the best possible endorsement. The KNIME philosophy and business model of mixed commercial and free (but Open) software, allows its continued improvement while making it freely available to desktop users. Some minor gripes relate to the fact that it seems only to read but not write .xlsx files – we are confident that someone will write a node to let it do so soon. There is a substantial community of users, increasing all the time, and many training schools and the like. Because of this, we think it will continue to grow in popularity. It is well worth a look for the GP community.

References

Berthold, M. R., et al. (2008). KNIME: the Konstanz Information Miner. In: Preisach, C., H. Burkhardt, L. Schmidt-Thieme, R. Decker (eds) Data Analysis, Machine Learning and Applications.(pp 319-326). Berlin: Springer. doi: 10.1007/978-3-540-78246-9_38

Kuhn, T., E. L. Willighagen, A. Zielesny, C. Steinbeck (2010). CDK-Taverna: an open workflow environment for cheminformatics. BMC Bioinformatics 11, 159 doi:10.1186/1471-2105-11-159

Mazanetz, M. P., R. J. Marmon, C. B. T. Reisser, I. Morao (2012). Drug discovery applications for KNIME: an open source data mining platform. Curr Top Med Chem 12, 1965-79 doi:10.2174/156802611212180004

O'Hagan, S., D. B. Kell (2015). Understanding the foundations of the structural similarities between marketed drugs and endogenous human metabolites. Front Pharmacol 6, 105 doi: 10.3389/fphar.2015.00105

O'Hagan, S., N. Swainston, J. Handl, D. B. Kell (2015). A 'rule of 0.5′ for the metabolite-likeness of approved pharmaceutical drugs. Metabolomics 11, 323-339 doi: 10.1007/s11306-11014-10733-z

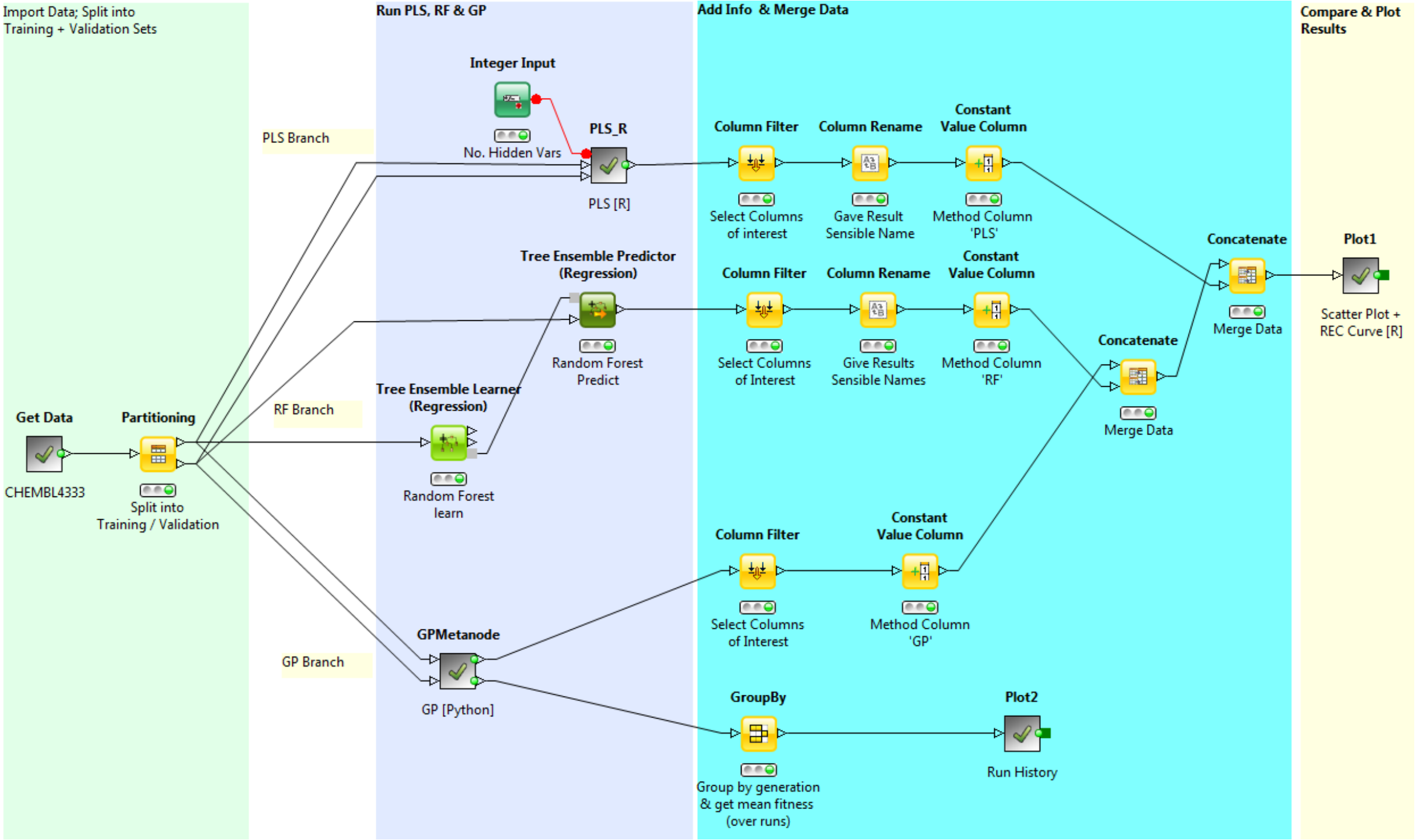
Riniker, S., G. A. Landrum (2013). Open-source platform to benchmark fingerprints for ligand-based virtual screening. J Cheminform 5, 26 doi:10.1186/1758-2946-5-26

Wolstencroft, K., et al. (2013). The Taverna workflow suite: designing and executing workflows of Web Services on the desktop, web or in the cloud. Nucleic Acids Res 41, W557-61 doi:10.1093/nar/gkt328

Funding: We thank the Biotechnology and Biological Sciences Research Council (BBSRC) for financial support under grant BB/M017702/1. This is a contribution from the Centre for Synthetic Biology of Fine and Speciality Chemicals (SYNBIOCHEM).

Legend to figure

Figure 1. Using KNIME to compare GP, Random Forests and Partial Least Squares Regression on a Cheminformatics benchmark dataset (ChEMBL4333).


Import Data; Split into Training + Validation Sets
Run PLS, RF & GP
Add Info & Merge Data
Compare & Plot Results
Integer Input
No. Hidden Vars
PLS_R
PLS [R]
PLS Branch
Column Filter
Column Rename
Constant Value Column
Select Columns of interest
Gave Result Sensible Name
Method Column 'PLS'
Tree Ensemble Predictor (Regression)
Random Forest Predict
Tree Ensemble Learner (Regression)
Random Forest learn
RF Branch
Select Columns of Interest
Give Results Sensible Names
Method Column 'RF'
Concatenate
Merge Data
Plot1
Scatter Plot + REC Curve [R]
Get Data
CHEMBL4333
Partitioning
Split into Training / Validation
GPMetanode
GP [Python]
GP Branch
Select Columns of Interest
Method Column 'GP'
GroupBy
Group by generation & get mean fitness (over runs)
Plot2
Run History