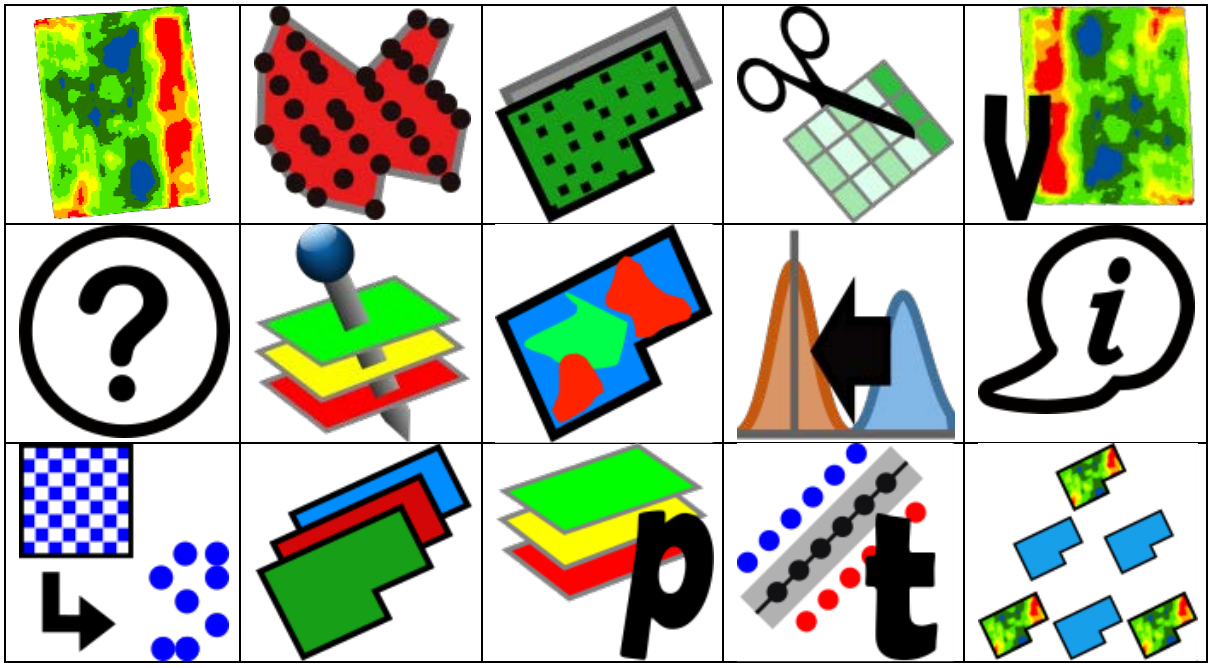
Australian Government
Wine Australia

Australian Government
Department of Agriculture

CSIRO

Simple tools for spatial analysis – key enabling technologies for Precision and Digital Viticulture



FINAL REPORT TO WINE AUSTRALIA

Project Number: **CSA 1603**

Principal Investigator: **Rob Bramley**

Research Organisation: **CSIRO**

Date: **July, 2019**

This project was supported by Wine Australia, through funding from the Australian Government Department of Agriculture as part of its Rural R&D for Profit program, and CSIRO.

CSIRO Agriculture and Food, Waite Campus.

Citation

Bramley RGV, Ratcliff C, Gobbett DL, Bakar S, Jin H, and Henderson B. (2019) Simple tools for spatial analysis – key enabling technologies for Precision and Digital Viticulture. Wine Australia / CSIRO, Australia.

Copyright

© Commonwealth Scientific and Industrial Research Organisation 2019. To the extent permitted by law, all rights are reserved, and no part of this publication covered by copyright may be reproduced or copied in any form or by any means except with the written permission of CSIRO.

Important disclaimer

CSIRO advises that the information contained in this publication comprises general statements based on scientific research. The reader is advised and needs to be aware that such information may be incomplete or unable to be used in any specific situation. No reliance or actions must therefore be made on that information without seeking prior expert professional, scientific and technical advice. To the extent permitted by law, CSIRO (including its employees and consultants) excludes all liability to any person for any consequences, including but not limited to all losses, damages, costs, expenses and any other compensation, arising directly or indirectly from using this publication (in part or in whole) and any information or material contained in it.

CSIRO is committed to providing web accessible content wherever possible. If you are having difficulties with accessing this document, please contact csiroenquiries@csiro.au.

This page intentionally blank

Abstract

Precision Agriculture (PA) is an approach to production in which an understanding of field and vineyard variability is used to enhance the certainty and efficiency of production. Thus, inputs are applied only where they are needed, and importantly for winegrowing, grapes may be selectively harvested and 'streamed' to particular target end uses (wines of different styles or price points). Key enabling technologies include geographical information systems (GIS) and methods of spatial analysis. Drawing on past (mainly viticultural) research, this project produced 'PAT – Precision Agriculture Tools' - a set of accessible 'freeware' tools that promote and facilitate low-cost adoption of PA.

This page intentionally blank

Table of Contents

Abstract 3
Table of Contents 5
Executive summary 7
1 Project rationale and objectives 8
2 Method 9
3 Project results and discussion 11
3.1 Project level achievements 11
3.2 Contribution to programme objectives 14
3.3 Productivity and profitability 14
4 Collaboration 15
5 Extension and adoption activities 15
6 Lessons learnt 16
7 Recommendations 16
7.1 Training in PAT for users 16
7.2 Software maintenance and ensuring legacy value of the PAT software 16
8 Additional project information 17
8.1 Project, media and communications material and intellectual property 17
8.2 Equipment and assets 17
8.3 References 17
8.4 Staff 19
8.5 Evaluation report 19
8.6 Financial report 20
8.7 Acknowledgements 20
Appendix 1: Free GIS for Precision Viticulture 21
Appendix 2: Multivariate spatial prediction and comparison for on-farm experimentation 22
Appendix 3: PAT – Precision Agriculture Tools Plugin for QGIS, User Manual 23

This page intentionally blank

Executive summary

Precision Agriculture (PA) is a management strategy that gathers, processes and analyses spatial, temporal and other data and combines it with other information to support management decisions according to estimated variability for improved resource use efficiency, productivity, quality, profitability and sustainability of agricultural production. Precision Viticulture (PV) is the application of PA in winegrowing.

Over the last 20 years, vineyard variability and the development of PV has been a growing area of research worldwide, to the extent that, in Australia, winegrowing is arguably as much to the fore in the sophistication of its use of digital technologies as is the case in other cropping systems that have explored the use of PA. Thus, it has been demonstrated that vineyards are variable and that managing them as though they are homogenous through the use of uniform management is a sub-optimal strategy. Accordingly, PV approaches to grape and wine production, which employ high resolution spatial data to inform strategies such as selective harvesting, may be highly profitable.

Whilst the sophistication and potential profitability of PV is suggestive of advanced application of digital technology, recently surveys in both the wine and grains sectors suggest that adoption of PV/PA, including new methods of on-farm experimentation which may support the targeting of management, will be greatly facilitated by factors which simplify the adoption process. Access to assistance, low cost and process simplification were identified as particularly critical. This project sought to address an opportunity in terms of both the cost and simplification of the spatial data analysis which is critical to PV/PA, by producing a suite of open source tools, accessible through a freeware platform, which may enable a non-expert to implement the core data analysis tasks which PV/PA require. This project was supported by Wine Australia, through funding from the Australian Government Department of Agriculture as part of its Rural R&D for Profit program, and CSIRO.

Drawing on CSIRO-led research in PV/PA conducted over the last 20 years, the project produced a suite of tools for spatial analysis that are freely available and accessible. *pyprecag* is a Python library containing these tools which is accessed using 'PAT' (Precision Agriculture Tools), a Plugin for QGIS, a freeware geographical information system. This is supplemented by tools for the analysis of so-called 'whole of block' experiments, coded in R, a freeware statistical programming language, which is also accessible via PAT. Users do not need to have expertise in spatial analysis and geostatistics to make use of PAT and it is therefore hoped that the availability of these tools will greatly facilitate adoption of PV/PA.

Instructions for setting up QGIS for PAT are accessible from https://github.com/CSIRO-Precision-Agriculture/PAT_QGIS_Plugin#precision-agriculture-tools-pat-plugin-for-qgis. Users are encouraged to install PAT from the QGIS Plugin repository having downloaded QGIS. Enquiries regarding PAT can be directed to PAT@csiro.au.

It is recommended that as a follow-up to this work, additional investments be made in user training, and also in software maintenance and upgrading to maintain compatibility with QGIS and Python as these systems continue to evolve.

1 Project rationale and objectives

Over the last 20 years, vineyard variability and the development of Precision Viticulture (PV) has been a growing area of research worldwide, to the extent that, in Australia, winegrowing is arguably as much to the fore in the sophistication of its use of digital technologies as is the case in other cropping systems that have explored the use of Precision Agriculture (PA; Bramley and Trengove 2013). Thus, it has been demonstrated that vineyards are variable and that managing them as though they are homogenous through the use of uniform management is a sub-optimal strategy. Accordingly, PV approaches to grape and wine production, which employ high resolution spatial data to inform strategies such as selective harvesting, may be highly profitable; Bramley (2019) provides a review.

Whilst the sophistication and potential profitability of PV is suggestive of advanced application of digital technology, it's level of adoption appears to be somewhat less than in broadacre systems, especially grains. Whereas a small survey of wine sector attitudes to the adoption of PV (Bramley 2013) highlighted that 66% of respondents believed that PV was either already delivering or will most likely deliver a benefit to their business, the survey also identified that the perceived high cost of PV was a significant constraint to adoption, that a lack of technical support / advice was limiting and that perceptions that PV was too difficult and/or time consuming were common. A larger, more recent survey of Australian grain growers (Bramley and Ouzman 2019) strongly suggested that adoption of digital technologies was heavily dependent on the prior adoption of yield mapping – which in broadacre systems is generally the entry point to more sophisticated PA adoption. This survey also showed that adoption of PA was greatest amongst grain growers who used a paid agronomic consultant. Collectively, these surveys, along with much anecdotal evidence, suggest that adoption of PV/PA will be facilitated by factors which simplify the adoption process, whether in terms of access to assistance, low cost or process simplification. Likewise, 'spatially distributed experimentation' has been shown to be a potentially powerful tool in support of identifying targeted management strategies (Bramley et al. 2013) and is an approach which vineyard managers have found highly attractive (Panten et al. 2010; Panten and Bramley 2011; Bramley et al. 2011a). However, in the absence of simple software for the analysis of such trials, implementation is constrained by the requirement for skills in the complex geostatistics necessary for analysis of results and an associated need for computer power. This is especially so when experimental treatment response variables are measured using digital technologies, such as yield monitors or remotely sensed imagery, which deliver data at a high spatial resolution and density. Access to variable rate equipment to assist with laying down a trial is also a potential impediment, albeit one that was not a constraint to the work of Panten and Bramley (2011) or Bramley et al. (2011a).

The bulk of the Australian PV research effort, much of which was funded through Wine Australia and/or conducted through the former Cooperative Research Centre for Viticulture, was conducted using methods of analysis which, at least at that time, were reliant on access to expensive software, including geographical information systems (GIS). Some of the sensing technologies involved in PA/PV were also perceived as expensive. However, novel, low-cost sensing technologies, some now available on smartphones, is a major means by which data (never before available due to the high cost of data capture) can now be captured. The associated analytics, the ability to make sense of massive amounts of highly variable types of data, is a key source of the growing power of 'big data' (see Leonard et al. 2017 and the underpinning reports referenced therein; Sonka 2014). Recently, a number of GIS programs have become available in a low cost or freeware format. Accordingly, this project had three simple aims:

a. To identify a suitable freeware GIS platform to support spatial analysis, map display and the various map production/analysis tools used in our previous research;
b. Refine methods for geostatistical analysis of field experiments in order for analysis to be performed on standard desktop computers; and
c. Develop a code to be engineered into a user friendly, open source software tool, which outputs to the GIS freeware.

These aims were couched as outputs 8a-c (*Enabling Technology*) in the broader Department of Agriculture and Water Resources-funded project (15-02-018) *Digital Technologies for Dynamic Management of Disease, Stress and Yield.* Outputs (a) and especially (b) involved key research tasks whilst output (c) was essentially development task dependent on (a) and (b) and the large CSIRO-led research effort in PV/PA conducted over the last 20 years or so. As is described below, these outputs were collectively delivered through the production of 'PAT' (Precision Agriculture Tools; Ratcliff et al., 2019a).

2 Method

Consistent with the three aims, the project was delivered in three phases.

a. Identification of a suitable freeware GIS platform to support spatial analysis, map display and the various map production/analysis tools used in our previous research.

The work undertaken in this phase is described in some detail in Appendix 1. In brief, the approach taken was to summarise the features required of GIS for use in a PV/PA context, based on our previous research experience, and to then identify a suitable GIS platform from the range of freeware options available based on some key criteria. Further detail is presented in Appendix 1. This work was undertaken by the CSIRO Agriculture and Food team based in Adelaide (see section 8.4). An important element of this work was to catalogue our existing PA analysis tools and their provenance and to obtain 'freedom to operate' approvals from the funders of the projects in which they were generated. This approval, without exception, was readily offered.

b. Refine methods for geostatistical analysis of field experiments in order for analysis to be performed on standard desktop computers

This work formed the key 'new' research undertaken in the project. Its primary focus was to address the shortcoming of the geostatistical method of analysis used for 'landscape scale' experiments (Bishop and Lark 2006), as described by Bramley et al. (2013). In brief, this limitation derives from the large matrix inversion involved in interpolating using kriging (e.g. Webster and Oliver 2007) which means that very large datasets, such as those generated by sensors such as yield monitors, cannot be analysed on a standard desktop computer using the Bishop and Lark (2006) method. It is also a 'global' method which therefore makes it inconsistent with the kind of local analysis used for interpolating yield maps from yield monitor data. Our aim in this work was to address these limitations through identification of modifications to the Bishop and Lark (2006) method. Ultimately our aim was to implement a solution so that such experiments could be conducted and analysed by non-experts. This work on method refinement, which was undertaken by CSIRO Data 61 colleagues based in Canberra, is described in some detail in Appendix 2 with some further contextual background and description of the approach taken provided as follows.

Context

Whole-of-block experimentation, a form of spatially distributed experimentation (Bramley et al. 2013), involves on-farm experiments where treatments are applied in a highly replicated design over an entire block, land management unit, or large portion thereof. These experiments are typically carried out by farmers using standard farming equipment and are spatially explicit in the treatments applied and responses (yields) measured. Whole-of-block experimentation requires a different analytical approach than traditional design-based field experimentation (e.g. plot trials with results analysed using linear mixed models) given the explicit spatial nature

and scale, the heterogeneity within the block, and the resolution at which data is collected. These features present new technical and computational challenges. At its core, the challenge in whole-of-block experimentation is one of multivariate spatial prediction and comparison for different 'treatments' applied at block or farm-scale. These 'treatments' may reflect both different varieties and management options. Understanding where differences that are statistically significant and practically important exist within the block is key to supporting farm decision making (e.g. partitioning the block into several zones) and optimal targeting of management.

Approach

A naïve approach to the spatial prediction and comparison problem is to consider each variable (i.e. treatment) on its own and make a spatial prediction with respect to that variable across the farm. These spatial predictions could be compared and used to assess treatment differences across the block. Whilst simple and effective (e.g. Lanyon and Bramley 2005), this approach is univariate and can result in considerable information loss because it ignores the spatial dependence between variables and does not make the comparison between variables (or treatments) directly. Improved prediction and comparison of treatments requires consideration of the covariance or dependence between different variables (treatments) in the joint spatial modelling of variables across the spatial domain (i.e. block, field or farm).

For the present project, a review of methods for the multivariate spatial prediction and comparison was undertaken. This review covers work in research and approaches in different communities, such as geostatistics, statistics, machine learning, as well as application domains like soil science, precision agriculture, natural resources, and climate. The review recognised particular challenges in: (i) modelling the dependence between variables (cross-covariances), (ii) handling the data density and resolution afforded by digital technologies, (iii) dealing with spatial heterogeneity and scale, and (iv) ensuring computational feasibility / efficiency. As described in Appendix 2, a range of techniques were considered, given these constraints and the data characteristics of whole-of-block experimentation. Cokriging based on Linear Model of Coregionalisation (LMCR), where the spatial dependence structure is modelled by a linear combination of simpler functions, was identified as the most appropriate in this particular context.

The existing approach of Bishop and Lark (2006) as used in our previous viticultural work (e.g. Panten and Bramley 2011; Bramley et al. 2011a) uses an implementation of cokriging based on a linear model of co-regionalization (LMCR) to make the multivariate spatial predictions and comparison of variables (treatments). It is a non-standard implementation of cokriging given there are no co-located observations. Executable software is available (Panten et al. 2010) but does not necessarily scale to high resolution, or highly dense, spatial data sets (e.g. from on-the-go sensors) and may either not work or be prohibitively time-consuming when implemented on a standard desktop computer (Bramley et al. 2013). The existing approach also requires substantial user input and tuning (e.g. around variogram fitting). Our literature review found that standardised ordinary cokriging is more suitable for spatially distributed experimentation, in order to obtain consistent results. The project team consequently investigated and then developed local (ordinary) cokriging techniques for analysing treatment comparisons. The focus on local versions of cokriging was important for two reasons: (i) it allowed for spatial heterogeneity to be better represented by local (cross-)covariance functions; (ii) it ensured that the cokriging was computationally feasible because for large high dimensional data sets, global cokriging is often not even possible; and (iii) it allowed the analytical methodology used to for experimental analysis to 'match' that used for yield mapping, for which local, rather than global kriging is used.

The project focussed heavily on ensuring the cokriging system developed was robust, computationally reasonable based on a standard desktop computer and required no or minimal user input.

The identification of an appropriate and computationally efficient local neighbourhood (or window) was an important challenge to overcome, and a neighbourhood radius was used rather than a set number of points (as in yield map interpolation, for example – see Appendix 3). Adjustments to that radius were necessary to deal with edge effects where there may be a limited number of neighbouring data points. A spatially varying neighbourhood is implemented in our cokriging system.

Global cokriging, local cokriging and local cokriging with subsampling were all included in the cokriging system developed. For small data sets global cokriging may be possible and appropriate, while for very large datasets the system supports subsampling prior to local cokriging to make it computationally feasible. For the great majority of cases, local cokriging is the appropriate choice based on accuracy, computational feasibility and efficiency, and responding to local heterogeneity.

c. Develop a code to be engineered into a user friendly, open source software tool, which outputs to the GIS freeware.

Further to the work undertaken in phases a and b, this third phase of the work was conducted in two parts. The first, drawing on the freeware GIS review (Appendix 1) and our previous research in PV/PA, involved converting our previous PA analytical tools, which were originally constructed for use in conjunction with the ArcGIS software suite (ESRI, 2016), into a Python library (Python Software Foundation, 2010) of PA analysis tools containing the various core processing and analysis steps. This Python library was then made accessible through a 'front end' to QGIS (the selected freeware GIS – see below). This 'front end' was given the name 'PAT' (Precision Agriculture Tools). Python 2.7 and QGIS 2 (QGIS Development Team, 2016) were selected as the platforms for this work. The second part of the work involved coding the new co-kriging methods for 'whole of block' experimental analysis (see above and also Appendix 2) using R (R Core team 2013) and also making this accessible through PAT. Finally, a user manual was written to support the PAT toolset. During this third phase of the work, the project advisory team was engaged to assess useability and identify additional functionality that prospective end-users were likely to desire.

3 Project results and discussion

3.1 Project level achievements

a. Identification of a suitable freeware GIS platform to support spatial analysis, map display and the various map production/analysis tools used in our previous research.

QGIS 2.18 (QGIS Development Team, 2016) was selected as the freeware GIS package that would be used to underpin the project outputs. Appendix 1 provides further detail underpinning this decision. Consistent with this decision, it was decided that programming of the analysis tools would be carried using Python 2.7 (Python Software Foundation 2010).

b. Refine methods for geostatistical analysis of field experiments in order for analysis to be performed on standard desktop computers.

The techniques implemented within the cokriging system were implemented, refined, and verified extensively on five data sets collected from previous whole-of-block experiments and through dedicated simulation studies. The

spatially-varying local cokriging techniques were shown to perform indistinguishably from global cokriging on small data sets such as those derived from hand sampling (e.g. Panten and Bramley 2011). For large data sets, for which global cokriging is not possible, they were found to perform statistically significantly better than spatial splines or sampling-based cokriging. The computation time on a standard desktop computer is less than half-hour even for tens of thousands of sensed observations from a 6.8 ha field. The spatially-varying local cokriging technique is recommended for large whole-of- block experiment datasets (see Appendix 3).

Given these findings, this phase of the project delivered the following outputs:

- R package: The package "oft" (On farm Tools) was developed. This is a combination of wrapper functions that includes novel algorithmic implementations of multivariate geostatistical methods to understand dissimilarities in the production of a farm by providing their corresponding statistical uncertainty and significances. The models implemented in the package are: (1) Global cokriging; (2) local cokriging based on a radius expanding algorithm; (3) local cokriging based on cross-validations with either (i) an adaptive neighbourhood radius, or (ii) adaptive neighbourhood points; and (4) Subsampling methods using: (i) bootstrapping and (ii) *k*-means clustering. Currently, the package supports the cokriging system using 2 or 3 treatments.
- Implementation of the R package via the PAT library 'front end' – see below.

Supporting publications describing this work are under development, copies of which will be provided to Wine Australia once accepted:

- *To Enable farmers for efficient geostatistical analysis of whole-of-block experiments*
  The paper will focus on the technical details of the proposed spatially varying local cokriging techniques, and its performance in terms of prediction accuracy and computation time. It will also illustrate some visualisation to help farmers to make an evidence-based, quick decision, such as zoning for differential management within a farm.

- *Better local cokriging techniques for whole-of-block experiment data analysis*
  The paper will focus on performance comparison among various local cokriging techniques implemented in the R package using large data sets.

- *An R toolbox to identify uncertainty of farm productivity based on treatment dissimilarities*
  A standalone paper focusing on the R package.

c. Develop a code to be engineered into a user friendly, open source software tool, which outputs to the GIS freeware.

PAT (Ratcliff et al. 2019a,e) was produced as a plugin for QGIS and is the front end to *pyprecag* (Ratcliff et al. 2019b) – an open source Python library of PA analysis tools which contains the processing steps. Collectively, PAT and *pyprecag* are the substantive outputs from this project. PAT/*pyprecag* contains all the tools for spatial data manipulation and analysis that, based on our own data manipulation and analysis undertaken during many years of PV/PA research, we believe will be needed by adopters of PV/PA. They also contain tools for the analysis of spatially distributed experiments whether these use a 'whole-of-block' (e.g. Cook and Bramley 1998; Panten and Bramley 2011; Bramley et al. 2011a, 2013) or simple strip (Lawes and Bramley 2011) design. Note that a key enabler to PAT, essential for the map interpolation steps, is the VESPER software (Minasny et al. 2005) which users also need to install; this is also freeware.

Both '*pyprecag*' and the PAT 'front end' were released in January 2019 as per the agreed project schedule; a second release was made in late June 2019 with enhancements to PAT which enable greater automation of some processes steps. Following the initial release (v0.1.0) there were two minor updates and two new feature releases such that the current version is now v0.3.0. As at 30 June, 2019, there had been 1086 downloads of v0.1.2 via QGIS.

The user manual is provided at Appendix 3 as the most efficient way to describe the toolset available, what each tool does and its mode of operation. However, in the first release, PAT contained tools for:

- generating a block boundary;
- deriving a block grid from this for mapping onto;
- projecting data to the correct datum;
- cleaning and trimming data prior to map interpolation;
- formatting data for interpolation in VESPER;
- importing the VESPER output;
- normalising a raster map layer;
- randomly sampling points from a raster map layer;
- extracting the values in each pixel in a raster maps layer;
- clustering of map layers to identify zones;
- analysis of remotely sensed imagery and calculation of vegetation indices such as NDVI, PCD, etc…
- co-kriging tool for the analysis of spatially distributed (whole-of-block) experiments; and
- the analysis of strip trials (albeit with the subsequent moving window t-test conducted using MS-Excel).

Given the rate of progress to the end of 2018 and also the late appointment of some project staff, a no-cost extension to the project was granted which enabled additions and enhancements made available in the second release. These were:

- 'Persistor' – an alternative to clustering, this tool generates a probability surface based on several years of yield or imagery data so as to indicate the consistency of performance at any given location (Diker et al. 2003; Bramley and Hamilton 2004);
- An enhancement to the strip trial analysis tool such that the analysis is now carried out within 'PAT'.
- A tool which enables interpolation of maps from 'low density' data sets derived from hand sampling; in the initial release, the mapping protocol followed was designed for mapping of high density data derived from an on-the-go sensor such as a yield monitor.
- Generation of automatic symbology/legends for map layers.

In addition, documentation/tutorials for programmers has been available. It is a convention / licencing requirement in the freeware software community for appropriate documentation to be provided to other developers who may wish to incorporate existing tools into new products.

Instructions for setting up QGIS for PAT are accessible from https://github.com/CSIRO-Precision-Agriculture/PAT_QGIS_Plugin#precision-agriculture-tools-pat-plugin-for-qgis. Users are encouraged to install PAT from the QGIS Plugin repository having downloaded QGIS. Enquiries regarding PAT can be directed to PAT@csiro.au.

3.2 Contribution to programme objectives

The objective of the programme was specified as being to realise significant productivity and profitability improvements for primary producers, through:

- generating knowledge, technologies, products or processes that benefit primary producers;
- strengthening pathways to extend the results of rural R&D, including understanding the barriers to adoption; and
- establishing and fostering industry and research collaborations that form the basis for ongoing innovation and growth of Australian agriculture.

As indicated above, the primary driver for the production of PAT was the realisation that the perceived difficulty of PA/PV data analysis and the prohibitive cost (at commercial rates) of the GIS system which has underpinned much of our prior research. With PAT now available, users can, with limited knowledge, and with no cost other than in terms of their time (including time needed for familiarisation with GIS), perform the various spatial analysis steps which we regard as being core to the adoption of PA/PV, irrespective of the cropping system that they work in. We therefore believe that PAT makes a substantial contribution to delivery against the first two of the above dot points.

Since *pyprecag* has been provided as open source, and given also that a consultant could use a single download of PAT to service data analysis undertaken for a number of clients, we would suggest that PAT also makes a contribution to delivery against the third dot point.

3.3 Productivity and profitability

PAT makes no direct contribution to the productivity of primary production. However, since it is a key tool for the facilitation of adoption of PA, it potentially makes an important contribution to enterprise profitability through enhanced resource use efficiency and in the case of the wine sector, selective harvesting (Bramley and Trengove 2013). For example, Bramley et al. (2018) provide an example in which the targeted application of gypsum to a sugarcane field variably affected by soil salinity and sodicity led to a saving in the costs of applying gypsum of $330/ha. All of the spatial data analysis which underpinned this example may be performed in PAT. The 'PA in Practice' manuals produced by SPAA (Society for Precision Agriculture, Australia – accessible at https://spaa.com.au/resources/) provide several examples of the profitable use of PA by graingrowers, whilst examples of the potential profit enhancement available to grapegrowers and winemakers through the adoption of selective harvesting are provided by Bramley et al. (2005, 2011b); again the spatial analysis underpinning these examples can be completed in PAT. More broadly, Leonard et al. (2017) suggest that 'unconstrained decision agriculture' could lead to increases in the gross value of production in Australian agriculture in the order of 51%. In making this claim, Leonard et al. (2017) defined 'decision agriculture' as a "conclusion or action resulting from the application of knowledge and/or information that may be derived from digital agriculture." PA is a major part of digital agriculture and PAT has the potential to therefore make a significant contribution.

As at 30 June, 2019, there had been 1086 downloads of PAT v0.1.2 via QGIS.

4 Collaboration

The nature of this project – development of software – was such that it was neither critically dependent on collaboration, nor likely to be enhanced by it. On the other hand, the project team put together to deliver this project involved the establishment of a collaboration between CSIRO teams in Agriculture and Food (Adelaide) and Data 61 (Canberra). A number of collaborations of note are worth mentioning:

The project made use of a project advisory panel to guide the utility and useability of the software being developed. This panel comprised Mr Colin Hinze (Bird in Hand Winery, Adelaide Hills – now with Rural Directions consultancy), Mr Hans Loder (Vineyard Manager, Penley Estate, Coonawarra) and Dr Kathy Evans (University of Tasmania/Tasmanian Institute of Agriculture). Their freely offered input on the PAT tools, which we trained them in, was much appreciated and enabled some tool finessing to make them easier for users to understand. Mr Hinze is a long time collaborator with the CSIRO team, especially through his previous role at Taylors Wines where much PV research has been carried out and where he was an enthusiastic adopter of PV approaches to vineyard management. Mr Loder has also been an enthusiastic adopter of PV, especially through his previous role at Katnook Estate; he also has a background in GIS and further to his participation in the project advisory panel, uses PAT on an on-going basis. In contrast, Dr Evans had no prior exposure to GIS and her only exposure to PV was through conference attendance and also as an early adopter of the spatially distributed experimentation approach (Bramley et al., 2011). Accordingly, the panel had a mixed skill based with respect to their approach to PAT and their feedback was valuable.

One benefit to accrue through the link to Dr Evans has been that a PhD student co-supervised by her and the Project Supervisor has been an early adopter of PAT and is using it in her research (Song et al., 2019).

Finally, the project benefitted from an additional collaboration internal to CSIRO; through the 'eResearch' initiative operated by CSIRO's Information Management and Technology business unit, we had access to specialist assistance with aspects of Python coding and software packaging and the mechanisms by PAT was released; the input of Daniel Collins, Andrew Spiers and Adrian D'Allesandro (CSIRO IMT) in this regard was much appreciated.

5 Extension and adoption activities

The project has taken a number of opportunities to raise awareness of PAT, both through conferences (Ratcliff et al., 2019c,d) and industry fora (Ratcliff et al., 2018). PAT has also featured in an article in the SPAA newsletter (Knights, 2019) and items produced by Wine Australia – see:

- https://www.wineaustralia.com/news/articles/making-precision-viticulture-a-practical-reality
- https://www.wineaustralia.com/news/articles/embrace-vineyard-variability
- https://www.wineaustralia.com/news/articles/to-be-precise-its-all-viticulture

Finally, in addition to Ratcliff et al. (2019c), PAT will gain substantial exposure at the forthcoming 17th Australian Wine Industry Technical Conference and Trade Exhibition (21-24th July, 2019) through a presentation to a workshop on vineyard technology and through the Wine Australia display booth. As described below (section 7 – Recommendations), it is recommended that a major extension and adoption effort be resourced so as to enable development and delivery of training assistance for users of PAT.

6 Lessons learnt

We were somewhat naive in commencing this project in thinking (a) that release of software is easy, and (b) that once released, it would be enduring. The second of these issues is dealt with below (section 7 – Recommendations). In regard to the first, we now understand that preparing a piece of software like PAT with the intention of releasing it via an open source freeware community, such as those which drive both QGIS and Python, does not mean de facto, that these communities will accept it. It seems clear that we would not have been able to release PAT according to the timelines set by the project without the input of the CSIRO eResearch team (see section 4 above). In effect, the project was somewhat under-resourced in terms of both time and staff.

7 Recommendations

7.1 Training in PAT for users

In both the lead up to the first release of PAT, via feedback provided to the PAT@csiro.au email address, and at various fora (e.g. the annual SPAA PA Symposium held in September, 2018), the project team have received numerous enquiries regarding access to PAT and requests for training; several of these have come from within the wine sector. The need for training to be made available was also clearly identified as a priority by both the Management and Steering Committee of the RnD4Profit-15-02-018 project and also by our project advisory team. Accordingly, it is recommended that a follow-on activity be supported aimed at providing training to end-users. This could be done through face-to-face presentation to core user groups, via a series of 'how to' YouTube videos, a recorded webinar presentation, and a PAT Twitter feed.

7.2 Software maintenance and ensuring legacy value of the PAT software

Closely associated with the need to provide support to PAT users through training, is the need to support them through on-going maintenance of the software; such maintenance is also critical to delivery of a sound ROI for the initial development of PAT since we clearly need to maximise the timeline over which it delivers benefit to stakeholders.

At the outset of PAT development in 2017, the analysis of suitable GIS products and programming languages (Appendix 1) led to the selection of QGIS 2 as the GIS package which PAT would be delivered through. Since QGIS relies on, and provides programming interfaces for Python version 2.7, this was the obvious choice as the development language for PAT and also for the underpinning *pyprecag* package. While we were aware of a slated release of QGIS 3 at that time, the date for official release was unknown, so we were concerned that delays in QGIS 3 release could interfere with the planned timeline for release of PAT. As it was, the first stable release of QGIS 3 occurred in Feb 2019, in the same month that we first released PAT.

Software developers need to keep abreast of ongoing changes to the platforms they develop on, whether these are commercially based, or free and open source. Both QGIS and Python are driven by separate communities of open source developers. QGIS 3 is now the default version available to users, and further development of QGIS 2 has effectively been discontinued as of March 2019 having been first released in October 2013. Coincidentally, the Python community has decided to discontinue Python 2.7 which has been in existence for around 10 years, and Python 3 is now the current version. Because QGIS 3 integrates with Python 3, earlier Python 2.7 plugins (such as PAT) will not work in QGIS 3, and need to be upgraded to Python 3.

While QGIS 2 will still be available for users to download for some time to come, and PAT will continue to work in QGIS 2, there are several reasons why an upgrade of PAT to QGIS 3 and Python 3 is highly desirable:

1. PAT will not work in QGIS 3 and will not be visible in the QGIS Plugin Manager to existing users of QGIS 3. This means PAT would only be available to users who are willing to use an older version of QGIS, and who do not have any other reasons for using QGIS 3, such as other plugins or newer features that are only available in QGIS 3. This is likely to limit the adoption of PAT, especially among users who are already QGIS 3 users.
2. Ability to respond to and fix bugs will become more difficult. In some cases bugs appear that are caused in some of the additional 3rd party Python modules that PAT relies on. The developers of these modules already have, or are planning to, cease supporting Python 2.7, so new versions of such modules, whether released to resolve bug fixes or provide new functionality are likely to be unavailable for PAT to use.
3. GDA2020 is a new national Australian Coordinate system (https://www.icsm.gov.au/gda2020), which is expected to be in use nationally by 30 June 2020, replacing the older GDA94. This coordinate system has been developed to support the far greater accuracy of modern GPS/GNSS receivers, and so will provide important benefits to precision agriculture practitioners. GDA2020 support in QGIS 2 will be limited, but it will be fully supported and maintained in QGIS 3.
4. VESPER, which is currently used by PAT for interpolating yield data into maps via block kriging is also undergoing similar changes to bring it in-line with the newer programming languages and to provide better processing efficiencies. PAT will need to be upgraded to best make use this modernisation.

In summary, while the current version of PAT will continue to work as it is, the potential uptake and value to industry will be limited by not upgrading PAT to work with QGIS 3 and Python 3. Accordingly, it is recommended that an initiative be supported to ensure that the initial investment in the development of PAT, in the training outlined above, and indeed, in the Precision Agriculture/Viticulture RDE on which PAT draws has an enduring legacy. This legacy will be ensured by upgrading PAT to work with both QGIS 3 and Python 3.

8 Additional project information

8.1 Project, media and communications material and intellectual property

Please refer to sections 5 and 8.3.

8.2 Equipment and assets

Nil

8.3 References

Bishop TFA, Lark RM. 2006. The geostatistical analysis of experiments at the landscape-scale. Geoderma 133, 87–106.

Bramley R. 2013. Wine sector attitudes to the adoption of Precision Viticulture. Wine and Viticulture Journal 28 (5), 69-73.

Bramley, RGV. 2019. Precision Viticulture: Managing vineyard variability for improved quality outcomes. Chapter 12 in: Reynolds, A.G. (Ed) Managing Wine Quality. Volume 1. Viticulture and wine quality. 2nd Ed. In press.

Bramley RGV, Trengove, S. 2013. Precision Agriculture in Australia: present status and recent developments. Engenharia Agricola 33, 575-588.

Bramley RGV, Evans KJ, Dunne KJ, Gobbett DL. 2011a. Spatial variation in response to 'reduced input' spray programs for powdery mildew and botrytis identified through whole-of-block experimentation. Australian Journal of Grape and Wine Research 17, 341-350.

Bramley RGV, Jensen TA, Webster AJ, Robson AJ. 2018. Precision Agriculture and sugarcane production – A case study from the Burdekin region of Australia. Chapter 9 in Rott PC (Ed) Achieving sustainable cultivation of sugarcane. Volume 1: Cultivation Techniques, quality and sustainability. Burleigh Dodds Science Publishing, Cambridge, England. pp.185-202.

Bramley RGV, Lawes RA, Cook SE. 2013. Spatially distributed experimentation: tools for the optimization of targeted management. Chapter 12 in: Oliver MA, Bishop TFA, Marchant BM. (Eds). Precision Agriculture for Sustainability and Environmental Protection. Earthscan, Food and Agriculture Series. Routledge, Abingdon, UK. pp. 205-218.

Bramley RGV, Ouzman J, Thornton C. 2011b. Selective harvesting is a feasible and profitable strategy even when grape and wine production is geared towards large fermentation volumes. Australian Journal of Grape and Wine Research 17, 298-305.

Bramley RGV, Proffitt APB, Hinze CJ, Pearse B, Hamilton RP. 2005. Generating benefits from Precision Viticulture through selective harvesting. In: Stafford JV. (Ed) Proceedings of the 5th European Conference on Precision Agriculture. Wageningen Academic Publishers, The Netherlands. 891-898.

Cook SE, Bramley RGV. 1998. Precision agriculture: opportunities, benefits and pitfalls of site-specific crop management in Australia. Australian Journal of Experimental Agriculture 38, 753–63.

Diker K, Buchleiter GW, Farahani HJ, Heerman DF, Brodahl MK. 2003. Frequency analysis of yield for delineating management zones. In P.C. Robert, R.H. Rust & W.E. Larson (Eds.) Proceedings of the 6th International Conference on Precision Agriculture and Other Precision Resources Management, Madison, WI, USA: ASA-CSSA-SSSA. CDROM.

ESRI. 2016. ArcGIS 10.4. Environmental Systems Research Institute, Inc. (Esri), Redlands, CA, USA.

Knights S. 2018. Free GIS 'toolkit' released. Precision Ag News 15 (3), 12-13.

Lanyon DM, Bramley RGV. 2005. Aiding soil management decisions through the use of precision viticulture technologies. In: Blair RJ, Williams PJ, Pretorius IS. (eds). Proceedings of the 12th Australian Wine Industry Technical Conference. Winetitles, Adelaide. pp. 249.

Lawes RA, Bramley RGV. 2011. A simple method for the analysis of on-farm strip trials. Agronomy Journal 104, 371-377.

Leonard, E., Rainbow, R., Trindall, J., Baker, I., Barry, S., Darragh, L., et al. 2017. Overview - Accelerating precision agriculture to decision agriculture: Enabling digital agriculture in Australia. Cotton Research and Development Corporation. http://farminstitute.org.au/LiteratureRetrieve.aspx?ID=161055. Accessed June 2019.

Minasny B, McBratney AB, Whelan BM. 2005. VESPER version 1.62. (Australian Centre for Precision Agriculture, University of Sydney: Sydney, NSW, Australia). www.sydney.edu.au/agriculture/pal/software/vesper.shtml (accessed June 2019).

Panten K, Bramley RGV. 2011. Viticultural experimentation using whole blocks: Evaluation of three floor management options. Australian Journal of Grape and Wine Research 17, 136-146.

Panten K, Bramley RGV, Lark RM, Bishop TFA. 2010. Enhancing the value of field experimentation through whole-of-block designs. Precision Agriculture. 11: 198-213.

Python Software Foundation. 2010. Python Programming Language, version 2.7. http://www.python.org (Accessed June 2019).

QGIS Development Team. 2016. QGIS Geographic Information System. Version 2.18. Open Source Geospatial Foundation Project. https://qgis.org (Accessed June 2019).

R Core Team. 2013. R: A Language and Environment for Statistical Computing; R Foundation for Statistical Computing. R v3.2.2. R Core Team, Vienna, Austria. http://www.R-project.org (accessed June 2019).

Ratcliff, C., Gobbett, D. and Bramley, R. 2018. PAT and QGIS – your new best mates for PA data analysis. Proceedings of the 21st Symposium of Precision Agriculture in Australasia. 10-11 September, Adelaide Oval, Adelaide. SPAA / University of Sydney. pp. 47-49.

Ratcliff C, Gobbett D, Bramley R. 2019a. PAT - Precision Agriculture Tools. v1. CSIRO. Software Collection. https://doi.org/10.25919/5c731a813b91a.

Ratcliff C, Gobbett D, Bramley R. 2019b. pyprecag - a Python package for the analysis of precision agriculture data. v1. CSIRO. Software Collection. https://doi.org/10.25919/5c731a41954ce.

Ratcliff C, Gobbett D, Bramley R. 2019c. 'PAT' – Making Precision Viticulture data analysis tools accessible. Proceedings of the 17th Australian Wine Industry Technical Conference, Adelaide, July 2019. In press.

Ratcliff C, Gobbett D, Bramley R. 2019d. PAT': Accessible tools for precision agriculture data analysis. Proceedings of the 19th Australian Agronomy Conference, Wagga Wagga, September 2019. In press.

Ratcliff C, Gobbett D, Bramley R. 2019e. PAT - Precision Agriculture Tools. v2. CSIRO. Software Collection. https://doi.org/10.25919/5d23f9fb58722.

Song X, Bramley R, Evans K. 2019. Exploring a simple strip approach to vineyard experiments for understanding spatial variation in treatment responses. Proceedings of the 17th Australian Wine Industry Technical Conference, Adelaide, July 2019. In press.

Sonka, S. 2014. Big data and the Ag sector: More than lots of numbers. International Food and Agribusiness Management Review 17, 1–20.

Webster R, Oliver MA. 2007. Geostatistics for Environmental Scientists. 2nd Ed. John Wiley & Sons, Chichester, UK.

8.4 Staff

CSIRO Agriculture and Food – Waite Campus, Adelaide

Dr Rob Bramley, Senior Principal Research Scientist – Project Supervisor.
Ms Christina Ratcliff, Principal Research Technician – Principal Investigator.
Mr David Gobbett, Senior Experimental Scientist – Technical oversight.

CSIRO Data 61 – Black Mountain, Canberra

Dr Brent Henderson, Group Leader – Project supervision and Technical oversight.
Dr Huidong Jin, Senior Research Scientist – Investigator.
Dr Shuvo Bakar, Research Scientist – Investigator.

As described in section 4, Daniel Collins and Adrian D'Allesandro (CSIRO IMT) also provided input to the project through a CSIRO 'eResearch' project. However, these staff were resourced internally by CSIRO and quite separately to the present project.

8.5 Evaluation report

As described in this final report, all outputs (8a-c) identified in in the monitoring and evaluation plan for the 'Digital technologies for dynamic management of disease, stress and yield' project (RnD4Profit-15-02-018) have been completed and delivered. The associated communication activities are described in section 5.

8.6 Financial report

To be provided by separate cover.

8.7 Acknowledgements

This project was funded by CSIRO and the Federal Department of Agriculture as a part of its Rural R+D for Profit Program (project 15-02-018 *Digital Technologies for Dynamic Management of Disease, Stress and Yield*) with sponsorship from Wine Australia (project CSA 1603). However, it drew heavily on work conducted over many years in the wine, sugar and grains sectors which was variously co-funded by CSIRO, the former Cooperative Research Centre for Viticulture (CRCV), Wine Australia (WA), the Grains Research and Development Corporation (GRDC), the Birchip Cropping Group (BCG), Sugar Research Australia (SRA) and The University of Southern Queensland (USQ). We are most grateful to these organisations and to the Australian Wine Research Institute (AWRI - on behalf of CRCV) for their permission to further develop the various tools created during this previous work into what, collectively, is now PAT.

Appendix 1: Free GIS for Precision Viticulture

AGRICULTURE AND FOOD
www.csiro.au

Free GIS for Precision Viticulture

Christina Ratcliff, David Gobbett

Internal Report for Wine Australia Project No. CSA1603, Simple Tools for Spatial Analysis – key enabling technologies for Precision and Digital Viticulture
8 May 2017



Wine Australia

Citation

Ratcliff, C and Gobbett, D (2017) Free GIS for Precision Viticulture. Internal Report for Wine Australia Project No. CSA1603, Simple Tools for Spatial Analysis – key enabling technologies for Precision and Digital Viticulture. CSIRO, Australia.

Copyright

© Commonwealth Scientific and Industrial Research Organisation 2017. To the extent permitted by law, all rights are reserved and no part of this publication covered by copyright may be reproduced or copied in any form or by any means except with the written permission of CSIRO.

Important disclaimer

CSIRO advises that the information contained in this publication comprises general statements based on scientific research. The reader is advised and needs to be aware that such information may be incomplete or unable to be used in any specific situation. No reliance or actions must therefore be made on that information without seeking prior expert professional, scientific and technical advice. To the extent permitted by law, CSIRO (including its employees and consultants) excludes all liability to any person for any consequences, including but not limited to all losses, damages, costs, expenses and any other compensation, arising directly or indirectly from using this publication (in part or in whole) and any information or material contained in it.

CSIRO is committed to providing web accessible content wherever possible. If you are having difficulties with accessing this document please contact csiroenquiries@csiro.au.

Contents

Acknowledgments iii
1 Introduction 4
2 Selecting GIS for Precision Viticulture 6
2.1 Key Functionality Required 6
3 Identifying initial freeware GIS candidates 7
4 Detailed selection criteria 8
4.1 Selecting a shortlist 9
4.2 Detailed assessment of the shortlisted candidates 9
5 Results 10
5.1 Initial candidates 10
5.2 Shortlisting the candidates 10
6 Functionality of shortlisted candidates 13
6.1 GRASS – Geographic Resources Analysis Support System 13
6.2 SAGA - System for Automated Geoscientific Analyses 14
6.3 QGIS - Quantum Geographic Information System 15
7 Recommendation 17
References 18

Figures

Figure 1. Common data sources, formats, operations and output visualisations used in precision viticulture.....6

Figure 2. Size of the online support community for the 12 selected 12

Figure 3. An example yield map illustrating key features such as classified raster symbology and associated legend, multiple scale maps used in the same layout.....13

Figure 4. A 3D Map Stack produced in GRASS showing yields over 3 years 14

Tables

Table 1. List of selected desktop free GIS candidates. The software applications from here on will be referred to by their acronyms as shown in bold. 10

Table 2. 12 Free GIS identified as potential precision viticulture software solutions. The three short listed software solutions are highlighted in bold 11

Acknowledgments

This work is funded by Wine Australia and CSIRO as Wine Australia Project No. CSA1603. The Wine Australia component of the funding is supported by the Australian Department of Agriculture and Water Resources through 'Rural R+D for Profit' Project No. 15-02-018 (Digital Technologies for dynamic management of disease, stress and yield. The support of these funders is gratefully acknowledged.

1 Introduction

The 'Simple Tools for Spatial Analysis – key enabling technologies for Precision and Digital Viticulture' project recognises that a major constraint to the adoption of precision viticulture (PV) and precision agriculture (PA) research outputs has been the fact that most of the research has been conducted using 'high end' GIS software that is not accessible to most industry practitioners, due to cost and/or complexity. The current suite of tools developed and used within CSIRO PV/PA research are built upon ESRI's ArcGIS Desktop, a proprietary desktop GIS application commonly used for GIS analysis. The tools are written in the python programming language, a high level general purpose programming language. In the past, proprietary GIS software provided users with far superior functionality compared to available alternatives, but a shift has occurred where free and open source GIS software has matured to become a suitable platform for PV/PA for both scientists and vineyard managers. The 'Simple tools for spatial analysis – key enabling technologies for Precision and Digital Viticulture' project will focus on making the current suite of PV/PA tools available within a free GIS software platform, with this document summarising the process undertaken to select an appropriate GIS software for this purpose.

The term *free* in *'free and open source software'* does not represent free of cost, but freedom of use. Essentially it gives a user or community the right to run, distribute, copy, study, enhance, change and improve the software. *Open source* refers to the fact that the source code is accessible and released to the public (Free Software Foundation, 2017). The two terms are often used together as a user requires the source code to have the ability to change, enhance or improve the software. Free and open source software is also often available free of cost. Another term, '*freeware',* is used to indicate software that is available free of cost, and where the developer retains ownership. Freeware often comprises small free utilities, or is provided as part of a service offering (e.g. Skype) or alongside commercial software (e.g. Adobe PDF Reader which relies on paid-for Adobe Acrobat to create PDF files), so is more likely to be developed and released for underlying commercial purposes.

There are several foundations and organisations supporting the development of geospatial open source software and standards. Two of the most recognised are

a) the Open Geospatial Consortium (OGC) which provides open standards and supporting documents for geospatial content, services, data processes and data sharing. These standards underpin the open interfaces and encoding used in free GIS applications and services.

b) the Open Source Geospatial Foundation (OSGeo) which was formed to support collaborative development of open source geospatial projects including desktop and web mapping applications, geospatial libraries and metadata catalogues.

Both these organisations are responsible for providing ways to access, manipulate data and services in GIS applications. As a result, it is common to find applications implementing OSGeo's Geospatial Data Abstraction Library (GDAL) for its ability to translate and process raster (GDAL)

and vector (OGR) data formats. These libraries are regularly updated. The use of these popular open source libraries and services is vital to the success of developing PV/PA GIS tools.

In this document, we summarise the features required of GIS for use in a PV/PA context, then outline a methodology for selecting a suitable GIS platform. The results of applying this methodology to the large range of freely available GIS software are presented, and a recommendation of the most suitable free GIS software for the development and deployment of PV/PA tools is made based on our assessment.

2 Selecting GIS for Precision Viticulture

2.1 Key Functionality Required

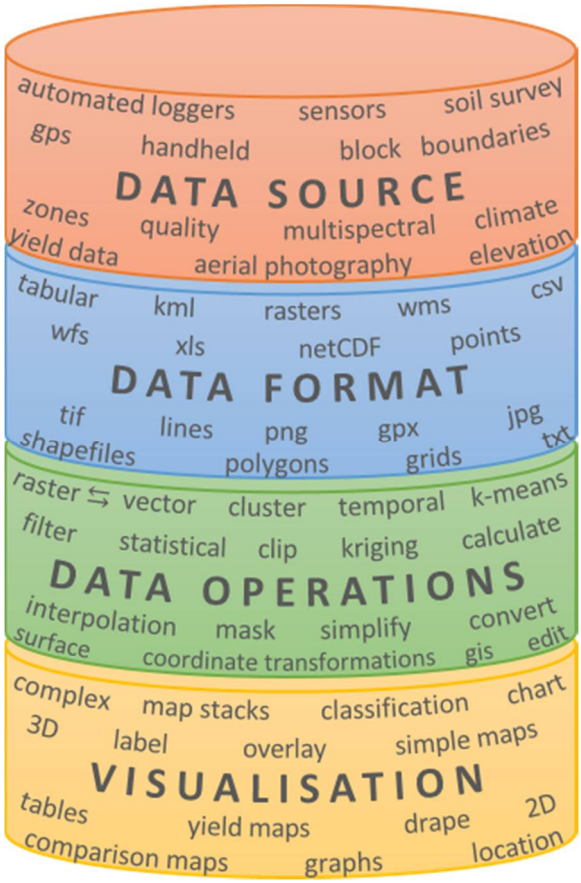


Figure 1. Common data sources, formats, operations and output visualisations used in precision viticulture

An assessment of the typical work was undertaken in the current GIS environment to determine the common types of data sources, formats, GIS tasks and visualisation used in PV/PA (Figure 1). Data is received from various sources including automated loggers (e.g. yield monitors), hand-held sensors, GPS, remotely sensed imagery of vine vigour and tabular data, all of which is analysed and presented in many ways to make the information more meaningful to the end user.

A key source of data for PV/PA is the use of automated loggers to record yield at harvest time. The yield monitor data is generated in a variety of different formats and generally requires cleaning and filtering as a precursor to interpolating a surface to produce yield map rasters. These rasters can be compared over time, and combined with soil surveys and climate records, to produce both simple and complex maps to assist vineyard managers to make informed management decisions; the delineation of management zones is often a part of this process.

The selected free GIS must cater for the majority of these data sources, formats and functionalities directly or through extensions, add-ins, plugins or toolboxes and be able to create output deliverables in the form of derived data, maps and tables.

3 Identifying initial freeware GIS candidates

An internet search was undertaken to identify the range of free GIS available. This identified hundreds of software applications including cloud, server, mobile, desktop and web mapping solutions, basic viewers, geospatial and statistical libraries, development tools, and open GIS data formats both for general and/or specialised areas of interest. To narrow the list of candidates it was determined that free GIS must:

a. be a general purpose desktop GIS software application
b. support both raster and vector data for editing and analysis
c. enable customisation through scripts and/or plugins
d. support multiple operating systems (e.g. Windows, Mac)

4 Detailed selection criteria

To restrict the larger number of candidates identified in the first selection stage, the following seven categories were considered as requirements of free GIS for PV.

Currency: As the free GIS market is quickly evolving, it is important that the selected software application has reached mature development, but is also able to keep-up with the rate of change and provide regular releases to users.

Application Focus: Some free GIS are started as research projects by universities or other organisations and as a result are highly specialised, focusing on a specific area of interest. An example of this is DIVA-GIS which specialises in species distribution modelling but also fits the space of a general purpose GIS application. Such applications, while providing specialised tools and functionality, may lack the basic requirements of a general purpose GIS.

Data Focus: Data focus gives an indication of the strengths and weaknesses of each software application with respect to supported raster and/or vector data formats and GIS operations. This was determined subjectively by reviewing the software user manuals, the number and types of tools available, and the range of supported data formats for both raster and vector data while excluding any extra functionality made available through extensions, add-ins, plugins or toolboxes.

Development Platform: While the development platform of free GIS isn't critical to the final software selection, it influences how readily weaknesses can be addressed by developers, and can affect compatibility with other software which may be used to fill a functionality gap. If the software is developed using a popular language, technical developers will also be more able to actively participate in the software development to improve or add functionality to suit their needs, while also sharing it back with the wider community.

Extensible Functionality: Where the data and/or application focus of a GIS software application is lacking functionality, gaps may be filled by providing access to extensions, add-ins, plugins or toolboxes from other software suites or libraries. If GIS software provides a framework to facilitate and simplify the development of tools and extensions, a larger user community is likely to develop, and functionality missing in the base software might be obtained through such extensions and add-ins. It is important to remember here that the further maintenance and development of these tools is reliant on the source packages whose developers are experts in their field. It is also beneficial if the GIS software can incorporate tools/functions from other platforms: for example, QGIS, which has strengthened its statistical and raster functionality through incorporating functionality from the R statistical package and toolboxes from SAGA and GRASS GIS.

Scripting Functionality: As the current set of CSIRO PV/PA tools are written in Python, it is preferable that the selected software supports Python as a means of customization. Python is a popular, open, high-level general-purpose programming language with numerous modules to provide specialised functionality.

Support Community & Resources: The support community must be mature enough to support a range of users from the GIS novice to professionals and developers. User manuals, tutorials and user forums are important resources and must cover this broad range of skill level and is

important to the successful acceptance of the chosen software. Where possible, forum activity was assessed to determine how reactive the community is to assisting users to resolve issues. To quantify the support community, an online search was undertaken for each free GIS software to determine the breadth and currency of the support communities. Each free GIS was entered into a) Google search engine, b) Stack Exchange, a network of Q&A websites, and c) GIS Stack Exchange, a Q&A website specifically for GIS users, and the number of hits was recorded. Special attention was paid when selecting the search terms to reduce the number of returned miss-hits. For example, the search term *"GRASS GIS"* was used to minimise the influence of *lawn* or *turf* based sites being recorded. Inflated numbers may also occur due to cross-hits where toolboxes have been incorporated to improve functionality in a core piece of software. For example, QGIS contains SAGA toolboxes, so searches for *"SAGA GIS"* will also include results for the QGIS SAGA Toolbox.

4.1 Selecting a shortlist

The freeware GIS which met the above criteria were investigated in greater depth to assess each software application's purpose and functionality. Further online searches were undertaken to determine the size of the online support community. Through this process, a shortlist of three free GIS was identified by eliminating software which:

a. has not had a new stable release in the last 12 months
b. does not support customisation via Python
c. has a small online support community

4.2 Detailed assessment of the shortlisted candidates

To assess the three shortlisted candidate software applications, each was downloaded and installed on a Windows 7 desktop computer. A GIS professional with no experience of each of the shortlisted applications then proceeded in replicating a simple yield map as shown in Figure 3. This process allowed for a *hands-on* experience of each software application covering simple tasks like:

a. loading different data types,
b. projecting data either on-the-fly, or by creating a new output
c. stylising raster and vector data
d. basic cartography like labelling data, adding legends and scale bars
e. create multiple side-by-side map views on one page (i.e. main map and locality map or two or more years of yield data)

Limitations in each application which impacted the above tasks were noted.

5 Results

5.1 Initial candidates

Fourteen Desktop GIS software applications were selected in the initial selection process, (Table 1). After preliminary investigations, Capaware and Marble Desktop were eliminated as they are primarily desktop globe viewers.

Table 1. List of selected desktop free GIS candidates. The software applications from here on will be referred to by their acronyms as shown in bold.

NAME	VERSION	RELEASE DATE*	HOMEPAGE
QGIS (Quantum Geographic Information System)	2.18.5	24/03/2017	http://www.qgis.org/en/site/
SAGA (System for Automated Geoscientific Analyses)	4.0.1	20/03/2017	http://www.saga-gis.org/en/index.html
GeoDa (Geographic Data Analysis)	1.8.16.4	07/03/2017	http://geodacenter.github.io/
OpenJUMP (Open Java Unified Mapping Platform)	1.10	27/01/2017	http://openjump.org/
Whitebox GAT (Geospatial Analysis Tools)	3.4	26/01/2017	http://www.uoguelph.ca/~hydrogeo/Whitebox/index.html
GRASS (Geographic Resources Analysis Support System)	7.2.0	28/12/2016	https://grass.osgeo.org/
gvSIG (Generalitat Valenciana, Sistema d'Informació Geogràfica)	2.3.1	26/10/2016	http://www.gvsig.com/en/products/gvsig-desktop
uDIG (User-Friendly Desktop Internet GIS)	2.0.0.rc1	12/08/2016	http://udig.refractions.net/
MapWindow	5.1.1.0	15/06/2016	http://www.mapwindow.org/
Marble Desktop	1.14.1	12/06/2016	https://marble.kde.org/index.php
ILWIS (Integrated Land and Water Information System)	3.8.5	03/09/2015	http://www.ilwis.org/open_source_gis_ilwis_download.htm
DIVA-GIS	7.5	01/05/2012	http://www.diva-gis.org/
TerraView	4.2	3/10/2011	http://www.dpi.inpe.br/terraview_eng/index.php
Capaware	rc2	7/04/2011	http://www.capaware.org/index.php?Itemid=58

**** current stable release available on 30/03/2017***

5.2 Shortlisting the candidates

The links shown in Table 1 were used as a starting point to research each software application's purpose and functionality. Online searches were also undertaken to determine the size of the online support community. Table 1 shows the date and version number for the current software

release, Table 2 shows the summary of the listed categories and Figure 2 shows the combined size of the online support community. While gvSIG is a strong contender in several areas, it does not appear in the short list as it has a more limited sized online community and is currently heavily oriented towards Spanish language end-users.

Table 2. 12 Free GIS identified as potential precision viticulture software solutions. The three short listed software solutions are highlighted in bold

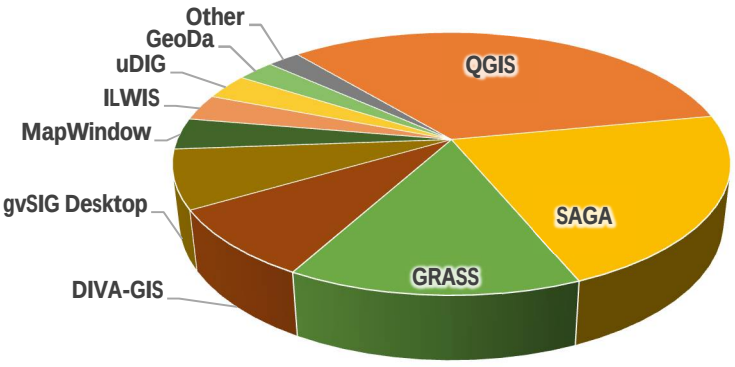
NAME	APPLICATION FOCUS	DATA FOCUS	EXTENSIBLE FUNCTIONALITY	DEVELOPMENT PLATFORM	SCRIPTING FUNCTIONALITY
QGIS	Core GIS Functions, Editing, Analysis and Cartography.	More Vector	GRASS, SAGA, ILWIS, OTB[1], R[2]	Python, C++, QT4	Python, C++, QT4
SAGA	Analysis, 2D & 3D visualisation, spatial modelling, Imagery processing	More Raster	-	C++	API, SAGA Command Line, Python.
GRASS	Analysis, 2D & 3D visualisation, spatial modelling, Imagery processing	More Raster	R[2], MatLab[3]	Python, C	API,Python, C
gvSIG Desktop	Core GIS Functions, Vector Editing,	More Vector	Sextante[4]	Java	API,Jython, JavaScript, Groovy
OpenJUMP	Vector Editing, Analysis	Vector	Sextante[4]	Java	API, Beanshell, Jython
GeoDa	Specialised (Spatial Autocorrelation & Spatial regression Modelling)	More Vector	-	C++	-
Whitebox GAT	Core GIS Functions, Analysis, spatial modelling,	More Vector	-	Java	API,Jython, JavaScript, Groovy
uDIG	Core GIS Functions, Vector Editing,	More Vector	-	Java	Groovy
MapWindow	Core GIS Functions, Vector Editing,	Vector	-	C#	API, C#, .NET
ILWIS	Analysis, 2D & 3D visualisation, spatial modelling, Imagery Processing	More Raster	-	C++, QT	C++, QT, Python 3.3x
DIVA-GIS	Specialises in species, distribution modelling, biological Diversity and climate analysis	Vector &Raster	-	Java	Command Line
TerraView	Core GIS Functions, Analysis, Spatial Statistics	More Vector	-	C++	API, C++

[1] Orfeo ToolBox (OTB) contains tools used in remote sensing for high resolution imagery classification and processing.

[2] R is a language and environment for statistical computation and visualisation

[3] Matlab is a matrix-based computing language for scientific and engineering numerical analysis research

[4] Sextante geoprocessing toolbox contains raster and vector data analysis and processing tools.



internet searches undertaken on 9/03/2017

Figure 2. Size of the online support community for the 12 selected

6 Functionality of shortlisted candidates

The top three shortlisted free GIS applications, GRASS, SAGA and QGIS, underwent further testing to evaluate their capabilities and user-friendliness in greater detail.

6.1 GRASS – Geographic Resources Analysis Support System

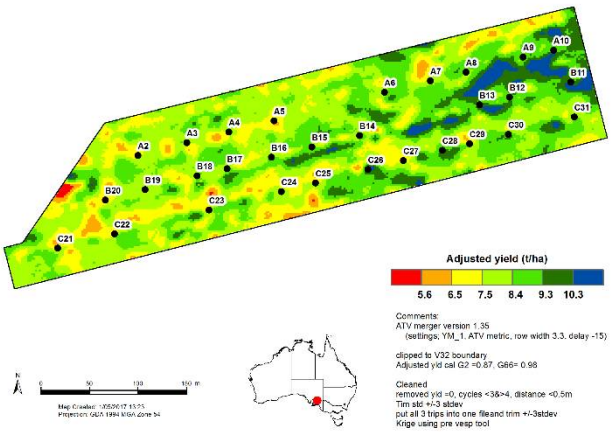


Figure 3. An example yield map illustrating key features such as classified raster symbology and associated legend, multiple scale maps used in the same layout.

GRASS is a software suite used by researchers, scientists and GIS professionals by providing over 350 raster and vector modules for data processing, analysis and 2D and 3D visualisation. Originally developed for the US Army Construction Engineering Research Laboratories in the 1980's and 90's, GRASS has matured into an impressive GIS tool used all over the world (Grass Development Team, 2015).

GRASS uses the concept of locations and mapsets to store and present data. A location is a container for datasets of the same geographic extent in the same coordinate system, and mapset is used to organise data into similar themes or projects. GRASS uses its own internal data format which allows for efficient data processing and analysis. All data must be registered with a GRASS database via either a linked dataset or imported to the native GRASS format prior to display or analysis. Projection on-the-fly is not permitted within GRASS to ensure the accuracy of the datasets during analysis is maintained. The provided import tools allows for re-projection of the data to the GRASS database in a single tool.

GRASS does not support floating number classified symbology of a raster like other GIS applications. To achieve this, the raster must be converted to an integer based grid, classified to a discrete set of codes, then a colour applied to each code. Each step produces a new raster, so if the source changes, these steps must be reapplied. This process impacts the presentation of a live legend. During the colour application, the user may add a label for each code which can be displayed in the legend while hiding the code. A horizontal legend similar to that showed in Figure 3 was difficult to achieve while maintaining live legend labelling.

While GRASS is powerful for analysing raster datasets, limitations were found during the cartographic assessment. GRASS supports a cartographic composer, but it appears that vector symbology isn't inherited from the display viewer. Vector labelling is possible in the display view via layer properties, while labelling within the composer requires a saved labels file to be applied as an overlay. The task of labelling within the composer was not achieved. Unlike other software, the map configuration within the composer is saved as a separate instructions file not with the

workspace. This enables multi-pages documents to be created. There are also limitations in the output map format from the composer, currently it supports only encapsulated postscript (EPS) and portable document format (PDF).

The GRASS user interface supports the concept of multiple display views which can be useful undertaking side-by-side comparisons, yet it has proven difficult to match this in the cartographic composer. At present, it seems that only one display view is allowed per page and therefore adding the Australia location map as shown in Figure 3 has not been achieved.

While the print composer has limitations for map production, GRASS also enables data to be presented in 3D within the display view. Although this also has limitations, it contains functionality that other GIS software (free and proprietary) do not. For example, it is possible to offset a surface on the X and Y plane in addition to the Z, also while in 3D display view, it is possible to place north arrows, scale bars and legends. The available output formats from the display view are restricted to Portable Pixmap (PPM) and Tag Image File (TIFF). An example of a 3D map produced in GRASS is shown in Figure 4.

6.2 SAGA - System for Automated Geoscientific Analyses

SAGA was first released in 2004 and ongoing development has enabled it to become a specialised tool for raster based scientific analysis and modelling. As its name suggests, it allows geoscientific analysis to be undertaken through more than 600 spatial tools and algorithms and is used by scientists and GIS professionals all over the world (Conrad et al., 2015).

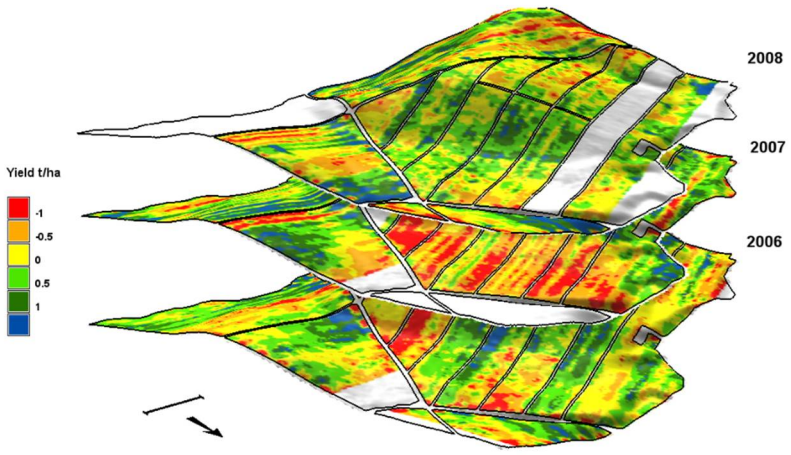


Figure 4. A 3D Map Stack produced in GRASS showing yields over 3 years

Like GRASS, SAGA operates best when raster data is stored in its own native data format and provides easy to use import and export tools to convert for most GDAL/ORG formats. It uses Shapefiles as its default vector format, but also has conversion tools for other GDAL/ORG formats. SAGA will register the import operations with the saved SAGA project file and next time the project is opened, the registered datasets will be re-imported.

While projection of both raster and vector dataset is supported, on-the-fly projection is not. Datasets can be projected and stored in memory until the project is closed, when the user will be prompted to save any new or altered files.

Stylising raster and vector data is easy, with the ability to apply colouring via a lookup table, RGB composite or a single, discrete or graduated colours. A lookup table option can be used for classified rendering as shown in Figure 3 where a single colour is applied to a range of values along with unique values for both vector (numeric and textual fields) and raster. It should be noted that unlike other GIS software, there is no 'load unique values' option. The user manually enters in all codes or attributes and assigns a colour which can be time consuming for complex vector textual attributes. The resulting lookup table can however be saved to a file for future use.

Compared to GRASS, the cartographic display is even more limited. SAGA does not allow labelling for more than one field and label placement is extremely restrictive with only vertical or horizontal alignment options available with no label offset options. Similarly, there is only one style of north point, and two for scale bars, with very simple property settings for both including size and position. The text size of the scale bar is set by the height of the graphic element, and there are no properties to control the size or number of divisions. Scale text can also be shown, but no properties are available to adjust text content, size or position on the page. There are no legend properties to set fonts, and patch sizes etc.

The SAGA interface does support the concept of multiple display views for analysis of different datasets, but having two display views on the one map is not possible. SAGA's layout capabilities are extremely limited; the user is unable to alter the location of the legend, include extra text like a map title, insert logo graphics, etc. It was therefore was not possible to reproduce the yield map shown in Figure 3. Producing 3D views is also possible in SAGA, but multiple offset layers like GRASS allows are not possible. Adding the scale bar and north point are possible, but the readability is reduced and the positioning is difficult when the view is rotated.

6.3 QGIS - Quantum Geographic Information System

QGIS development started in 2002 when GIS software applications for the Linux operating system was limited and the QGIS project was initiated to enable an interactive 2D viewing platform for spatial data in that environment. QGIS is now a mature cross-platform GIS software application supporting the viewing, editing, analysis and presentation of spatial data. QGIS is a modular application, where additional functionality can be added through plugins or toolboxes (Hugentobler, 2008).

Unlike GRASS and SAGA, QGIS facilitates most OGR vector formats, and GDAL raster formats, including Shapefiles, TIFFs and comma separated values (csv). Other formats like Excel are supported through the use of 3rd party plugins. QGIS also supports projection on-the-fly for all datasets and also has the ability to permanently change the projection.

Out of the three shortlisted applications, QGIS has the most sophisticated options for symbolising raster and vector datasets. Raster renderers include, single or multiband options, on-the-fly hill-shade representation for elevation datasets; interpolation methods for display include linear, discrete or exact values with no pre-processing required. Vector renderers are just as extensive and allow data to be presented as single symbol, categorized or graduated and even rule-based among others. Unlike SAGA, when displaying a layer based on single or multiple fields, the user can automatically load all attributes, or manually add them one by one.

QGIS vector labelling options are also extensive, ranging from a simple label by one or more attributes through to complex rule-based labelling with the ability to add shadows or buffering and even multi-line labels. Label placement options allow for offset and around a point, data defined priority and placement along with the ability to control labelling over other features types. The addition of map elements like scale bars, north points and legends to a map is also possible, with many different styles and properties available. Recreating a live legend as shown in Figure 3 is complex and involves several steps, but is possible.

While QGIS supports only one display view, layers can be grouped and visibility settings applied. Using the print composer, each of these grouped layers can be applied to a different map frame allowing for the presentation of a locality map or side-by-side map views to compare different datasets. In addition, these layouts can be saved as a map template and applied to other QGIS projects for quick map production. Maps can be saved as a range of common formats including PNG, JPG, TIFF and PDF.

Viewing maps in 3D can be achieved through the use of 3rd party plugins. Using the Qgis2three plugin, a single 3D layer map was produced as html format for viewing in an internet browser, however the addition of titles, scale bars, north points and legends was not achieved. GRASS provides tools to QGIS, which should achieve the same result as shown in Figure 4, but due to the existence of a software bug, the reproduction of the 3D map stack was not achieved from with QGIS.

7 Recommendation

QGIS provides the best well-rounded 'out-of-the-box' solution for a free GIS application due to its strengths in cartographic presentation. The online community is extensive and up-to-date books, tutorials, and user manuals exists to assist all level of users.

QGIS is released with a GNU General Public License version2 (GPL) which a common license type for Free and Open Source Software. This license a) gives access to the software and source code, b) gives the legal right to copy, distribute and/or modify the software, and c) protects the software and source code from being incorporated into proprietary software. All copies issued are exempt from warranty. When copies are distributed, either in their original or altered form, they must be distributed with a copy of the GNU GPL and the source code be made available (Free Software Foundation, 2016).

The QGIS installation files are made available for use in a variety of scenarios which is important for the adoption of PV/PA in rural areas with poor internet connectivity. Standalone installers, allows the download of one file for installation and is advised for new users. Additionally, portable versions may be created on a device (ie USB or cloud drive) for use anytime, anywhere, on any PC and enables distribution of QGIS with predefined configurations, customizations, and templates. The software can be launched directly from the device, or copied to a PC with no extra installation required. A compatible mobile application QField for QGIS is currently being developed for use on Android platforms, and is targeted as a field data capture and management app compatible with QGIS.

While QGIS's primary focus is for vector analysis, the main package comes with some basic raster functionality. One shortcoming of the tested version of QGIS is it lacks the ability to create a 3D map stack of raster layers (Figure 4). This functionality, if needed can be achieved using GRASS GIS. The strength of QGIS compared to other free GIS applications is the ability to use plugins and processing toolboxes to further enhance functionality. By incorporating toolboxes, GRASS adds 314 tools, SAGA 248 tools, and GDAL/OGR 48 tools to QGIS's 116 existing tools to creating a well-rounded application for analysis and visualisation of vector and raster datasets. Users also have the ability of accessing the official plugin repository containing over 600 user-developed plugins covering tasks like loading popular web map services like Google maps or Open Street Map into QGIS, to providing extra editing tools. Users can create their own plugins, tools or scripts for specific tasks which can be shared with the wider user community. QGIS provides several mechanisms to distribute plugins including, a) an OSGeo hosted central plugins repository, b) a self-hosted plugin repository, and c) via a file. When connected to an online repository, the currency of a plugin can be checked, and updated at the user's discretion.

References

Conrad, O., Bechtel, B., Bock, M., Dietrich, H., Fischer, E., Gerlitz, L., Wehberg, J., Wichmann, V., and Böhner, J. (2015) System for Automated Geoscientific Analyses (SAGA) v. 2.1.4, Geosci. Model Dev., 8, 1991-2007, doi:10.5194/gmd-8-1991-2015.

Free Software Foundation (2017), What is Free Software? https://www.gnu.org/philosophy/free-sw.html Accessed 13 Apr 2017.

Free Software Foundation (2016), GNU General Public License, version 2. https://www.gnu.org/licenses/old-licenses/gpl-2.0.html, Accessed 13 Apr 2017

Grass Development Team (2015), GRASS GIS: Bringing advanced geospatial technologies to the world. https://grass.osgeo.org/documentation/general-overview/. Accessed 11 Apr 2017.

Hugentobler, M. (2008). Quantum GIS. Encyclopedia of GIS. S. Shekhar and H. Xiong. Boston, MA, Springer US: 935-939. http://link.springer.com/referenceworkentry/10.1007/978-0-387-35973-1_1064

CONTACT US

t 1300 363 400
+61 3 9545 2176
e csiroenquiries@csiro.au
w www.csiro.au

AT CSIRO, WE DO THE EXTRAORDINARY EVERY DAY

We innovate for tomorrow and help improve today – for our customers, all Australians and the world.

Our innovations contribute billions of dollars to the Australian economy every year. As the largest patent holder in the nation, our vast wealth of intellectual property has led to more than 150 spin-off companies.

With more than 5,000 experts and a burning desire to get things done, we are Australia's catalyst for innovation.

CSIRO. WE IMAGINE. WE COLLABORATE. WE INNOVATE.

FOR FURTHER INFORMATION

CSIRO Agriculture and Food
Dr Rob Bramley
t +61 8 8303 8594
e rob.bramley@csiro.au
w www.csiro.au/en/Research/AF

Appendix 2: Multivariate spatial prediction and comparison for on-farm experimentation

Data61
www.csiro.au

Multivariate spatial prediction and comparison for on-farm experimentation

Huidong Jin, Shuvo Bakar and Brent Henderson

Report Number: EP194220
3 April 2018

Wine Australia

Commercial In Confidence

Data61
Building 801 Synergy, North Science Road
Acton ACT 2601, Australia
GPO Box 1700, Canberra ACT 2601 Australia
Telephone: +61 2 62167055
Fax: +61 2 62167000

Copyright and disclaimer

© 2018 CSIRO To the extent permitted by law, all rights are reserved and no part of this publication covered by copyright may be reproduced or copied in any form or by any means except with the written permission of CSIRO.

Important disclaimer

CSIRO advises that the information contained in this publication comprises general statements based on scientific research. The reader is advised and needs to be aware that such information may be incomplete or unable to be used in any specific situation. No reliance or actions must therefore be made on that information without seeking prior expert professional, scientific and technical advice. To the extent permitted by law, CSIRO (including its employees and consultants) excludes all liability to any person for any consequences, including but not limited to all losses, damages, costs, expenses and any other compensation, arising directly or indirectly from using this publication (in part or in whole) and any information or material contained in it.

Contents

Executive summary . . . 1

1 Introduction and motivation . . . 2

2 Brief review of methods . . . 4

2.1 Classical multivariate geostatistics . . . 4

2.1.1 Empirical cross-covariance with LMCR . . . 5

2.2 High resolution data and computational challenges . . . 5

2.2.1 Compactly supported covariance functions . . . 5

2.2.2 Reduced rank kriging . . . 6

2.2.3 Subsampling and localised methods . . . 6

2.3 Summary . . . 6

3 Implementations . . . 7

3.1 Advantages and challenges of the approaches . . . 7

3.1.1 Global co-kriging method . . . 7

3.1.2 Local co-kriging method . . . 8

3.1.3 Mixed co-kriging method . . . 8

3.2 Example 1: Clare J3 . . . 9

3.2.1 Results for the global and mixed co-kriging methods . . . 9

3.3 Example 2: Taylors V32 . . . 13

3.3.1 Results for the local co-kriging method . . . 13

3.4 Summary . . . 14

4 Discussion and next steps . . . 18

5 Supplementary . . . 19

5.1 Default neighbourhood size . . . 19

5.1.1 Mixed co-kriging method . . . 19

5.1.2 Local co-kriging method . . . 19

5.2 R code implementation . . . 19

Executive summary

On-farm experimentation, also known as spatial experimentation, requires a different analytical approach than traditional design-based field experimentation. The large spatial scale, heterogeneity and the data resolution present new technical and computational challenges.

At its core the challenge in on-farm experimentation is one of multivariate spatial prediction and comparison for different 'treatments' applied at farm-scale or landscape-scale. This scale differs from traditional small block experiments. These 'treatments' may reflect both different varieties or genotypes and different management actions. Improved prediction and comparison of treatments requires consideration of the covariance or dependence between different variables (treatments) and the joint spatial modelling of variables across the spatial domain (i.e. farm). The univariate spatial prediction and comparison, which considers each variable (treatment) on their own and makes a spatial prediction of the response of interest (e.g. yield) across the farm, could result in considerable information loss because it ignores the spatial dependence between variables and does not make the comparison between variables (or treatments) directly.

This report focuses on one of the approaches to classical multivariate spatial geostatistics that may take advantage of cross-variable and spatial dependencies, as Part 2 of the 'Simple Tools for Spatial Analysis – key enabling technologies for Precision and Digital Viticulture' project. The report considers a popular classical multivariate approach – 'co-kriging', with it's extensions to localised analysis of on-farm experimentation. This report discusses advantages and challenges of the extensions and develops user-friendly computing platform using `R` programming language. It also provides critical information on further development of the multivariate methods that can potentially benefit the output of the on-farm experimentation.

The report implements classical localised co-kriging (ordinary co-kriging) for high resolution spatial data, and also highlights several speed-up strategies, including reduced-rank approaches, covariance tapering, and aggregation co-kriging. Some of these strategies are good for efficient likelihood estimation at a large number of locations. The classical local approaches (e.g. local co-kriging), which is similar to covariance tapering in principle, can handle some of the spatial heterogeneity and manage the high resolution data (Section 2.2) quite well.

Going forward the intention is to further improve the local ordinary co-kriging based on linear model of coregionalisation (LMCR), and extend it to be adaptable with different data size and co-kriging resolution. Some working directions are further validating against the existing methods, including regression covariates, and comparing with efficient spatial prediction techniques like splines for large data sets.



Figure 1.1: Photo for three different vineyard floor treatments in an organically managed block of Merlot (4.8 ha) – Clare Valley, South Australia [Bramley et al., 2008, Panten et al., 2010]

1 Introduction and motivation

The 'Simple Tools for Spatial Analysis – key enabling technologies for Precision and Digital Viticulture' project, funded by Wine Australia, recognises that a major constraint to the adoption of precision viticulture (PV) and precision agriculture (PA) has been the fact that most of this research has been conducted using 'high end' GIS software that is not accessible to most industry practitioners, due to cost and/or complexity. As Part 2 of the project, this report focuses on methods for the analysis of on-farm experimentation or spatial experimentation.

On-farm experimentation seeks to drive improvement to on-farm productivity and efficiency, by adopting an experimental paradigm and utilising the increasing availability of sensing and observation technologies and continuously indexed data sets and techniques. Continuously indexed data sets with multiple variables have become ubiquitous in the precision agriculture, digital agriculture, geophysical, ecological, environmental and climate sciences [Genton and Kleiber, 2015]. Availability of modern sensor technologies means that farmers are now able to routinely collect geo-referenced data with a large number of repeated-measurement spatial points within plots or fields at an affordable cost. Mainly because of the farm equipment used, farmers prefer large plots (such as split field or half-field as illustrated in Figure 1.1), certainly much larger than in classical on-station trials. Different from pot experiments or on-station experiments, on-farm experiments are integrated into regular farming operations using a participatory approached with the aim to understand the variable, possible long-range, agriculture production, caused by, e.g., soil, topography, management [Bramley and Trengove, 2013]. One example is illustrated in Figures 1.1 and 1.2, in which three different mid-row management options (treatments) for enhancing vine vigour in an organically managed vineyard, Clare Valley, South Australia. This Clare experiment was analyzed by geostatistical techniques to illustrate the spatially varying treatment responses and the significance of treatment contrasts [Panten et al., 2010].

As can be observed in Figure 1.2, there are several features of data from on-farm experiments.

1. The response observations for different treatments are normally not co-located.

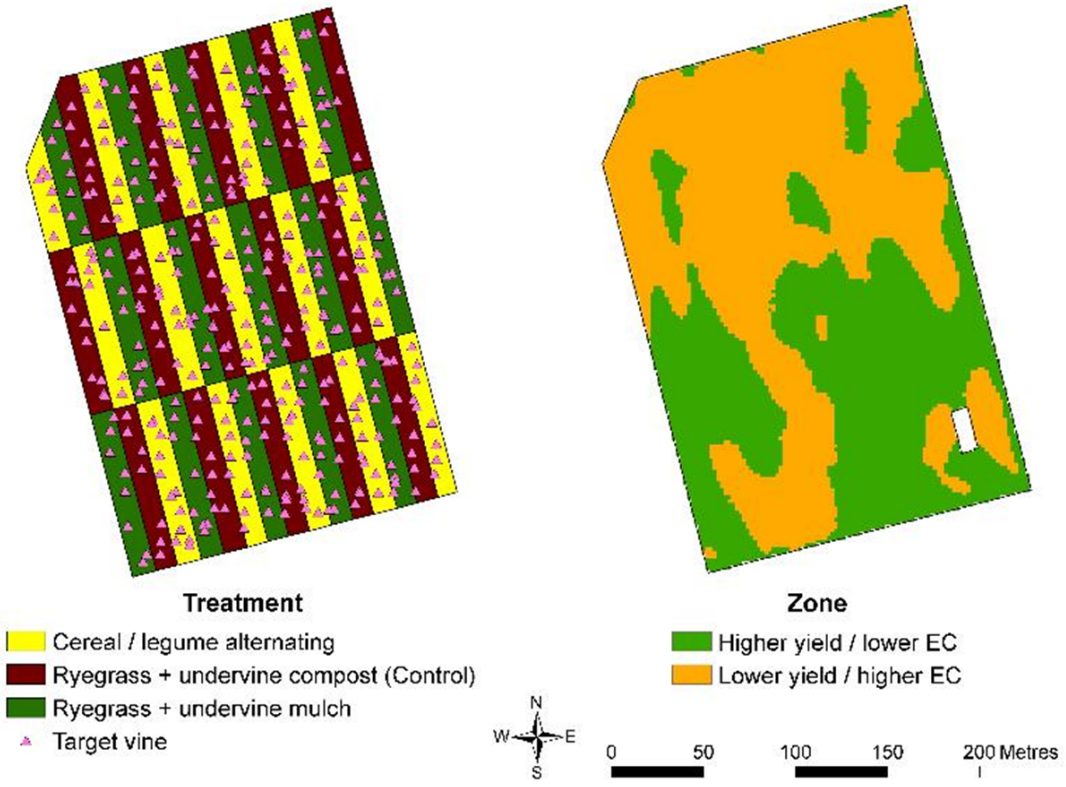


Figure 1.2: Manually sampled sites for three vineyard floor treatments in an organically managed block of Merlot (4.8 ha) – Clare Valley, South Australia [Panten et al., 2010]

2. There are typically a quite small number, often only 2 or 3, of different types of treatments.
3. The number of response observations may range from small, such as the manual samples in the Clare experiment, to large, such as high resolution spatial yield maps from harvesters.
4. The response observations could randomly distributed with some measurement errors.
5. The required resolution for predictions may be high (and the process for generating them potentially time-consuming).

On-farm experimentation like one illustrated in Figure 1.2 requires an analysis that makes (i) spatial predictions of yield or other response characteristics under different treatments; (ii) spatial predictions of the differences in those yields between the treatments; and (iii) standard errors (or equivalent) for the predictions and the predicted treatment contrasts.

Important attributes of the problem to address include the multivariate nature of the problem and the spatial dependence between different treatments, non-stationarity of the spatial random fields, potentially large data sets and heavy or prohibitive computational constraints, and the need for localised predictions analogous to local kriging.

In the remaining of the report, Section 2 briefly reviews the classical approaches for multivariate geostatistics, with focus on cross-covariances for co-kriging. It also provides the computation challenges from a big volume of observations or high-resolution prediction, and discusses possible solutions. Extensions of the co-kriging methods are discussed, implemented and tested in Section 3. Section 4 gives concluding comments and a possible future project plan.

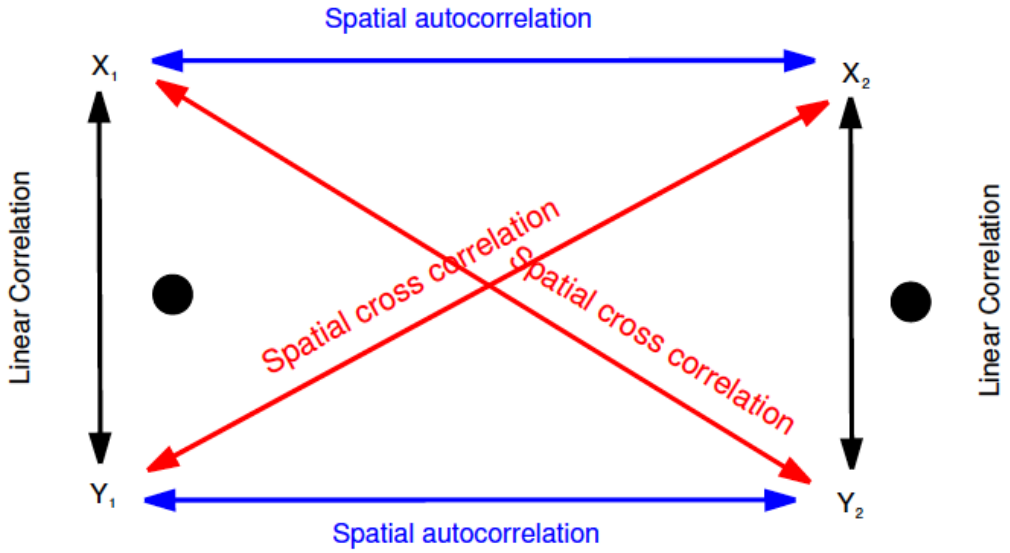


Figure 2.1: Cross-correlation may help improve spatial prediction and comparison

2 Brief review of methods

2.1 Classical multivariate geostatistics

The primary analytical focus in on-farm experimentation is spatial prediction and comparison of different treatments at farm or landscape scale. The spatial scale means that spatial variation may be significant and should be considered in any analysis. This section presents and discusses classical multivariate geostatistical methods within the broad area of multivariate spatial statistics for addressing these challenges. One area that is particularly important for on-farm experimentation is that of co-kriging. As illustrated in Figure 2.1, considering the cross-variable dependency may improve the spatial predictions and comparisons. Wackernagel [2013] notes that in principle co-kriging yields smaller prediction errors than kriging when there are multiple processes that are spatially dependent.

The key challenge in multivariate spatial prediction is to specify a valid cross-variogram or cross-covariance function, which is responsible for the relationships between distinct variables. Cross-covariance functions requires the resulting covarinace matrix for associated data to be positive definite [Gelfand and Banerjee, 2010, Genton and Kleiber, 2015]. This challenge becomes rapidly harder as the dimension increases as there are much more cross-covariance functions and marginal covariance functions to consider.

Identifying the cross-covariance function is one of the challenging tasks in multivariate modelling. Ad-hoc plug-in methods are often popular for constructing an empirical cross-covariance function. Additionally, linear combination of independent processes can lead to a richer modelling of cross-covariance, which is also known as the linear models of coregionalization (LMCR) [Wackernagel, 2013]. Other methods of constructing the cross-covariance functions include: kernel convolution [Ver Hoef and Barry, 1998], locally stationary models [Fuentes, 2002], conditional methods [Cressie and Zammit-Mangion, 2016] etc. Once the covariance structure is trained and optimised, co-kriging can be used to infer other locations of interest. Several possible methods like ordinary cokriging (OCK), standardized ordinary cokriging (SOCK) as well as the PCK developed by Papritz and Flühler [1994] are available. All of them formulate weights to minimise the cokriging variance and an unbiased estimate. PCK is good to estimate the difference between two variables, and SOCK often has the best performance according to Bishop and Lark [2006].

Below a brief description of the LMCR method is presented to estimate the empirical cross-covariance function for the classical co-kriging.

2.1.1 Empirical cross-covariance with LMCR

The linear Model of coregionalisation (LMCR) is a popular simplification, which ensures cross-covariance assumptions required for co-kriging prediction are met. In LMCR, the multivariate random field is represented as a linear combination of several or one independent univariate random fields (i.e., univariate spatial covariance function). These random fields are known as basic structures [Desassis and Renard, 2013].

In LMCR the empirical cross-semivariograms are estimated from data, often based on some prespecified bins to characterise the spatial dependence. It is necessary for each bin to contain observations and often requires to adjust the bins based on the spatial distribution of the data. The pseudo cross-variograms would be recommended to used if these observations for different treatments are not co-located [Papritz et al., 1993, Ver Hoef and Cressie, 1993]. As the number of model parameters increases quadratically with the number of variables (or treatment), LMCR is commonly used for two or three different variables.

Once the sample (or experimental or pseudo) cross-variograms are known it is necessary to identify a set of basic variogram models and positive semidefinite coregionalisation matrices that yield the best fit. Because this problem has possibly many solutions (different variogram models may take on the same values at the lag vectors under consideration), it seems difficult, if not impossible, to design a completely automated fitting algorithm.

2.2 High resolution data and computational challenges

Due to the advancement of technologies, such as on the ground sensors, machinery sensors, aerial images, and remote sensing, massive amounts of data may be observed at a large number of spatial locations or even across the whole farms (e.g. yielding). These massive volumes of data, with a wide variety, had better be captured, analysed and then used for evidence-based decision-making [Sonka, 2015]. This section illustrates the computational issues caused by big data, that is,

(1) how to make the multivariate spatial analysis of large data sets feasible, and

(2) how to improve computational efficiency?

The co-kriging techniques for large data require the solution of a large cross-covariance matrix of all the involved variables. For large amounts of data, it is difficult, if not impossible, to solve the model with direct methods as the large dimensional cross-covariance matrices must be inverted for co-kriging or prediction [Sun et al., 2012].

It becomes a challenging task to create an effective and efficient statistical method to handling large spatial data sets, especially where the number of spatial locations is large. There are several ways to solve this problem: likelihood approximation or covariance function approximation techniques like covariance tapering, reduced ranked approximations (Gaussian predictive process or fixed rank kriging), and Markov random fields or their mixtures etc. We briefly review few of these techniques below.

2.2.1 Compactly supported covariance functions

The underlying mechanism of this kind of approaches is to approximate a covariance or cross-covariance matrix with some sparse matrix, which is able to be inverted computationally much more efficiently. There are several different versions for this kind of approximation techniques, such as covariance tapering, local kriging, and nearest neighbor [Fuentes and Raftery, 2005, Furrer et al., 2012].

It is mathematically non-trivial to build up appropriate tapering functions for multivariate covariance. For example, tapers had better be smooth at the origin; possibly different compact supports; and a good balance between the number of parameters to be fixed and the multi-taper flexibility. The sparse matrices from tapering may still lead to prohibitively high computation, e.g., the sparse matrices from a large tapering radius may still require cubic complexity for inverse. A possible better solution is to go

with block diagonal structure, as used in [Sang et al., 2011].

2.2.2 Reduced rank kriging

The underlying mechanism of reduced rank kriging is to use some reduced rank matrices to approximate the original full-rank cross-covariance matrices [Cressie and Kang, 2010]. In univariate setting, Gaussian Predictive Process (GPP) creates a process onto a subspace generated by the realisations of the original process at a specified set of locations (or knots) [Guhaniyogi et al., 2011]. Similarly, for multivariate spatial processes, the multivariate predictive process extends the preceding concepts to multivariate Gaussian processes [Banerjee et al., 2008]. Basically it uses the multivariate realisations on previously chosen knot locations to approximate the high-dimensional Gaussian processes.

2.2.3 Subsampling and localised methods

Besides reduced rank and sparse matrix approximation, there are other approximation techniques in literature, which is often popular in order to speed up (co-)kriging. Subsampling techniques are also used to reduce computation burden, such as picking up a random, or representative, subset of data for prediction [Quiñonero-Candela and Rasmussen, 2005]. Furthermore, local kriging [Cressie, 1993] also alleviates the computational burden, however mostly depends on the definition of the term 'local'. Furrer and Genton [2011] proposed aggregation-cokriging for highly-multivariate spatial data sets to reduce the computational burden. This method is based on a linear aggregation of the covariables with carefully chosen weights [Sun et al., 2012].

2.3 Summary

The classical multivariate geostatistcs method developed by Bishop and Lark [2006] uses an implementation of co-kriging to make the multivariate spatial predictions and comparison of variables. It is in non-standard implementation of cokriging given there are no co-located observations. Software code is available but does not necessarily scale to high resolution, or highly dense, spatial data sets, and require substantial user input/tuning [Panten et al., 2010]. We will handle these issues in Section 3.

The approximation techniques for high resolution spatial data are quite efficient, especially for model inference and out-of-site prediction. They often require that the observations for different treatments are co-located. We will take an extension of the local kriging approach into local co-kriging to speed-up co-kriging based on cross-covariance estimated using LMCR.

The review for multivariate geostatistics above is not exhaustive. There are multivariate covariance functions with specific features, such as nonstationarity, asymmetry, compact support (tapering), spatio-temporal, or Bayesian multivariate spatial modelling. Genton and Kleiber [2015] gives a comprehensive overview of the multivariate covariance function with special features. Gelfand and Banerjee [2017] gives a good account of Bayesian geostatistical analysis, efficient and user-friendly computation. Under spatial regression setting, where the expectations are driven by some co-located covariates, Bayesian hierarchical models are often the choice. Examples include multivariate Gaussian predictive process [Finley et al., 2008, Banerjee et al., 2015, 2008, Bakar et al., 2015], full scale approximation [Sang et al., 2011].

3 Implementations

This section will focus on the implementation of co-kriging methods (ordinary co-kriging) for both small and large spatial data. In particular, Clare J3 [Panten et al., 2010] and Taylors V32 data sets are used, with data points 377 and 9,065 respectively.

The computing code is written in R [RDevelopmentCoreTeam, 2006] (with a possible extension to the development of an open-sourced software package). Three approaches are constructed for the co-kriging implementations. For a relatively small data set a global cross-covariance function is used, which ensures the positive definite (semi-definite) property and lower computational cost for the matrix. The local co-kriging method, an extension of the local kriging discussed in Section 2 is developed, which is mainly aimed for large dimensional multivariate data. Here, local cross-covariance functions are constructed to obtain localised predictions to minimise the high computational cost for the cross-covariance matrix.

The local co-kriging is sometimes challenging due to the number of multivariate data points that can contribute to identify the cross-covariance function in a pre-defined local area. For example, consider a high-dimensional data set with 3 treatments, and we often observe the presence of all 3 treatmetns are not equal in a defined localised neighbourhood. Furthermore, we might see a particular variable has more observations compared to the others. In practice, these situations influence the cross-covariance matrix to become unstable and non-invertible. Hence, the localised method develops an algorithm that can iteratively identify the best range of a local neighbourhood that is effective for the local co-kriging. Explaining it further, for a given neighbourhood size if a cross-covariance matrix becomes non-invertible, then the algorithm will increase the neighbourhood size – until it produces a stable cross-covariance matrix.

Furthermore, the mixed method is developed that combines both global and local cross-covariance functions, and aims to identify the localised effects of a small number of spatial data (not a large dimensional data). For example, often we are interested to see any local variations from a sparse and smaller set of data, however this initiates the problem of very small number of data points for all treatments, even often no data points for a particular treatment in the pre-defined local area. Hence, the mixed method initiates to construct an algorithm that will use a global cross-covariance to tackle these situations. However, for large dimensional multivariate data the mixed method will introduce an increased computation cost for computing the global corss-covariance matrix.

Note that the local co-kriging with the adjustment algorithm can also be applicable for relatively small data sets, however it is computationally costly to identify the range as often the small data sets are distributed sparsely.

The co-kriging methods and related code described in this report can take maximum three treatments. The methods can be implemented using (1) exponential, (2) Gaussian and (3) spherical covariance functions.

3.1 Advantages and challenges of the approaches

In this section, we briefly discuss the pros and cons of three different co-kriging methods.

3.1.1 Global co-kriging method

This approach uses a fixed cross-covariance function for co-kriging, and as discussed earlier it can be implemented to a smaller set of multivariate spatial data with maximum 3 treatments.

Advantages

1. Can predict a smooth spatial field with a reduction of an overall bias for the prediction.
2. Computationally easy for small number of spatial points, and useful to construct positive-definite cross-covariance matrix.

3. Useful for a stationary spatial field.

Challenges

1. Often hard to identify localised effects and corresponding biases.
2. Non-stationarity may be missed.
3. Computationally expensive, and sometimes not feasible, to obtain the predictions when observation data size is large (e.g., high resolution observations).

3.1.2 Local co-kriging method

The local approach uses a local cross-covariance function for co-kriging. Similar to the global approach it can take up to 3 treatments and can support 3 different types of covariance models.

The local approach requires an input to define the local neighbourhood, there is a default value for this, however users can pre-select the size of the neighbourhood.

This method is suitable for large volumes of data. Users can implement the “local” version to a smaller data dimension, however as discussed earlier it may be computationally expensive to identify the range of the local neighbourhood.

Advantages

1. Can identify localised effects and non-stationarity.
2. Applicable for the large dimensional data, as it reduces cross-covariance calculation to a local neighbour with small number of points.
3. Often comes with the problem of non-positive definite cross-covariance matrix. Hence the algorithm is developed to identify the proper invertible cross-covariance matrix using a looping method.

Challenges

1. The spatial surface is not smooth and often very irregular based on the selection of localised neighbourhood.
2. The recurrence looping algorithm is often computationally expensive given the number of loops it requires to fix the issue of the inversion of the cross-covariance matrix.
3. Co-kriging output varies with the choice of the neighbourhood size.

3.1.3 Mixed co-kriging method

The mixed co-kriging approach is an extension of the local approach for small dimensional data. If the number of samples are large, then local co-kriging with a robust recurrent algorithm of selecting the neighbourhood size is more appropriate. For small number of samples, the mixed approach performs a localised co-kriging with user defined neighbourhood size (or default neighbourhood size). However, if the cross-covariance function is not stable for a particular kriging location then it uses the global cross-covariance. Note that for small number of samples the “mixed” approach does not perform the robust method for selecting the neighbourhood size.

Advantages

1. Can identify both global and localised effects and corresponding biases.
2. Computationally efficient, given the choice of the neighbourhood size.

Challenges

1. Model output is constrained on the initial neighbourhood size.
2. Often can exhibit a very non-smooth surface, which is completely different than a localised ap-

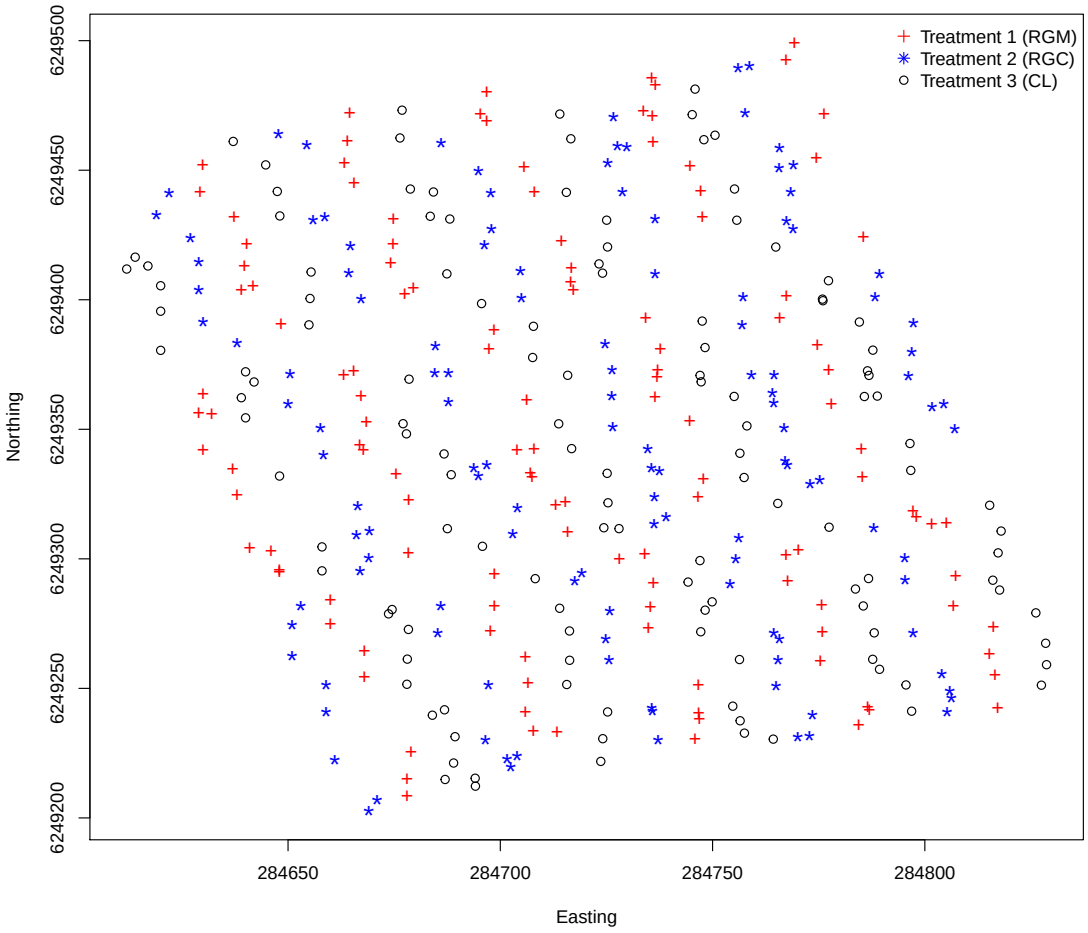


Figure 3.1: Three different treatments in the Clare J3 data set.

proach. Similar situation may occur for comparing it with the global approach.

3. Co-kriging output varies with the choice of the neighbourhood size.

3.2 Example 1: Clare J3

The Clare J3 data set is a good example for co-kriging with small number of spatial data. Figure 3.1 illustrates all the three treatments and corresponding sampling sites. The co-kriging methods are performed to predict in the 2 metre grids (a total of 12,040 grid points). Here,

- Treatment 1 = Ryegrass + Undervine Mulch (RGM),
- Treatment 2 = Ryegrass + Undervine Compost (RGC, control) and
- Treatment 3 = Cereal/Legume Alternating (CL).

The (1) global and (2) mixed methods for co-kriging are implemented to the Clare J3 dataset.

3.2.1 Results for the global and mixed co-kriging methods

The cross-semivariograms are presented in Figure 3.2 for the global method. Figures 3.3(a), (c) and (e) show the estimated treatment effects for RGM, RGC and CL. Furthermore, we also report the treatment differences for RGM vs RGC and CL vs RGC in Figures 3.4(a) and (c). Note that in the experiment RGC is treated as a control.

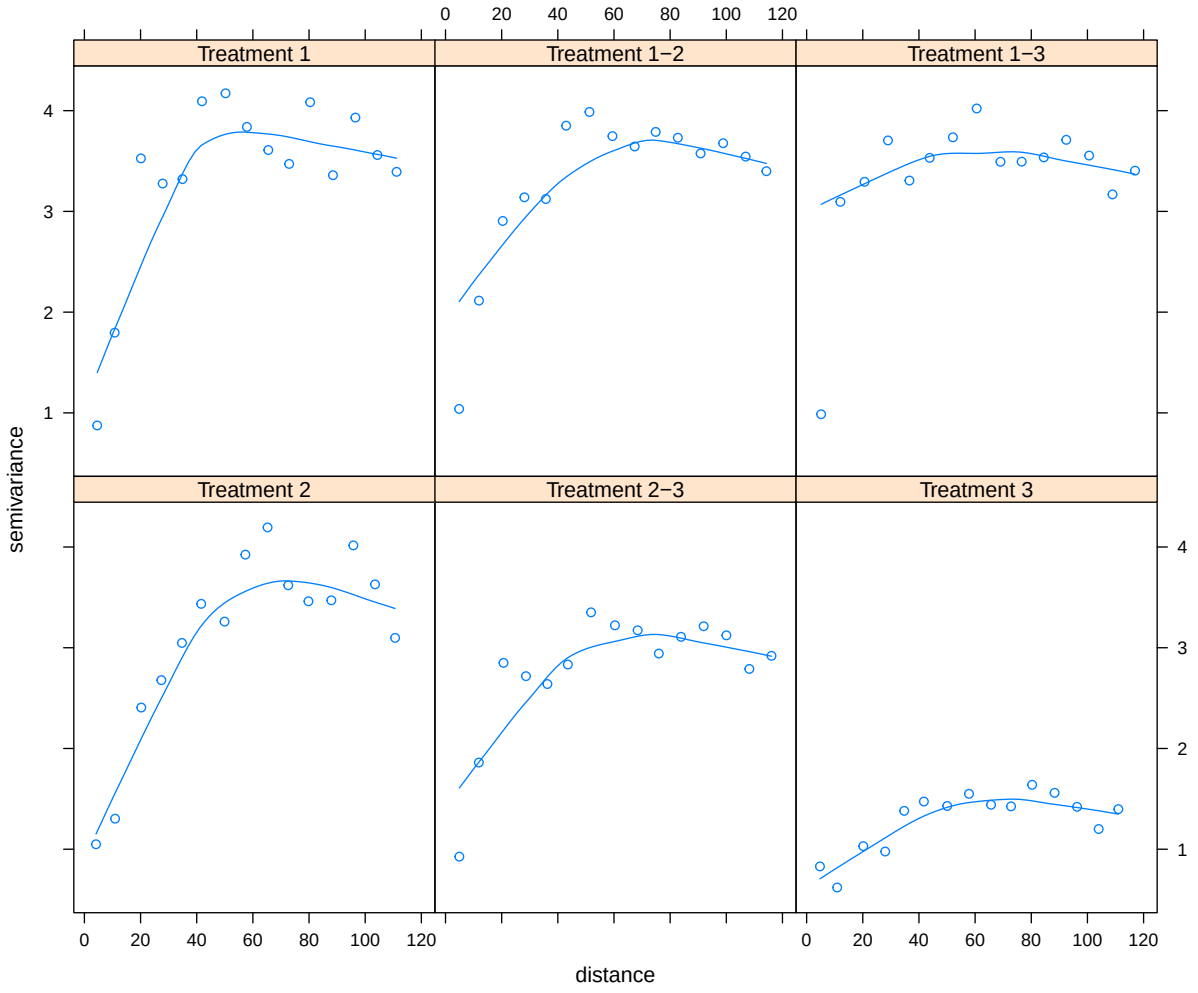


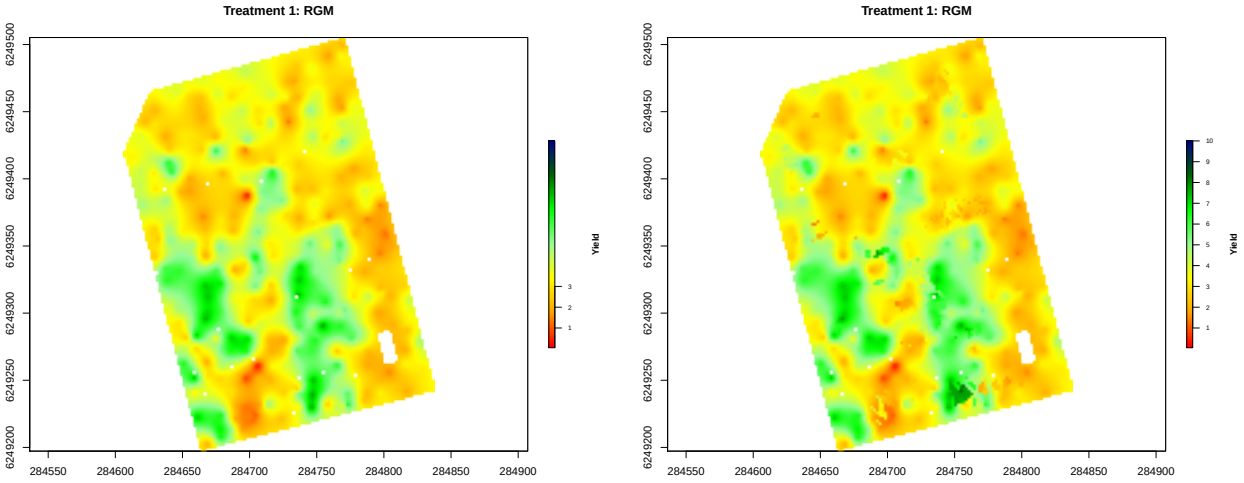
Figure 3.2: Estimated semivariograms for the global co-kriging method.

The mixed method is also used in this example to understand the local estimated effects obtained from the Clare J3 data. As discussed earlier, the neighbourhood size is often important for the localised approach. It represents the area to construct the local cross-covariance. User can define the neighbourhood size. For example, Figures 3.3(b), (d), and (f) represent the estimated treatment effects based on an arbitrary value – 20 units (in this example units are in metres) radius. The software also provides percentages for the number of times the local and global cross-covariance functions were used in the model, in this example it was 93.96% and 6.04% respectively. Explaining it further, about 94% of the kriging points (i.e., about 11,318) used local cross-covariance and only 6% points (i.e., 722) used the global cross-covariance for estimating the treatment effects.

Figures 3.4(a), (c) represent the treatment differences of RGM and CL with the control treatment–RGC respectively. Comparing with Figures 3.4(d)–(e), we see that the differences are prominent in some localised areas.

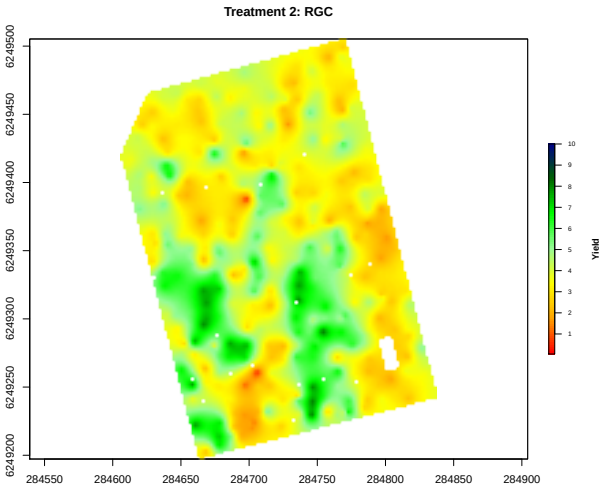
We also checked the mixed method using a neighbourhood size of five metres, which yielded a 100% use of global cross-covariance function for co-kriging at the 12,040 data points.

The software also can consider a default value for the neighbourhood size, when no value is defined by the user. The default value is calculated based on a function of maximum spatial distance between the data points, range and the total number of spatial locations (see details in Supplementary Section). The default neighbourhood size is calculated 10.54 metres, where for all data points global cross-covariance is used.

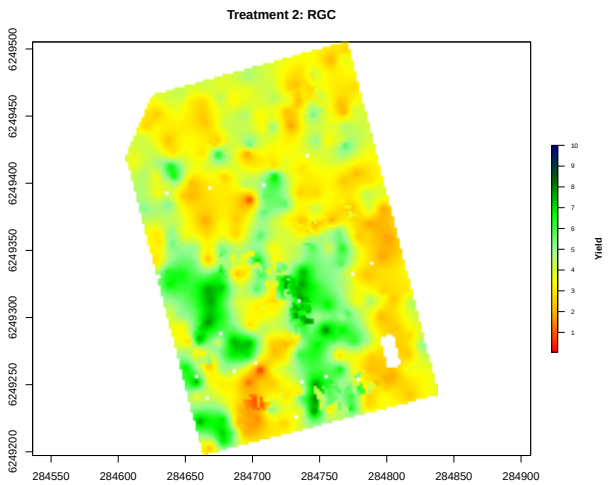


(a) Global

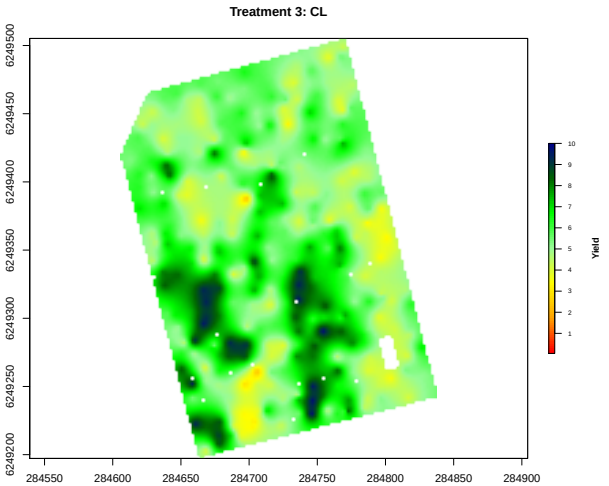
(b) Mixed



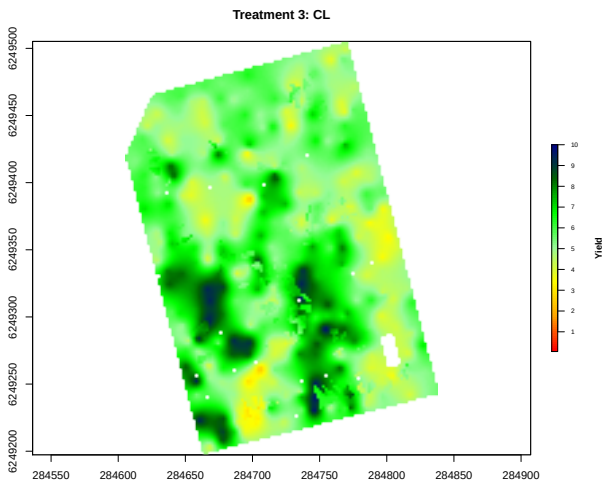
(c) Global



(d) Mixed

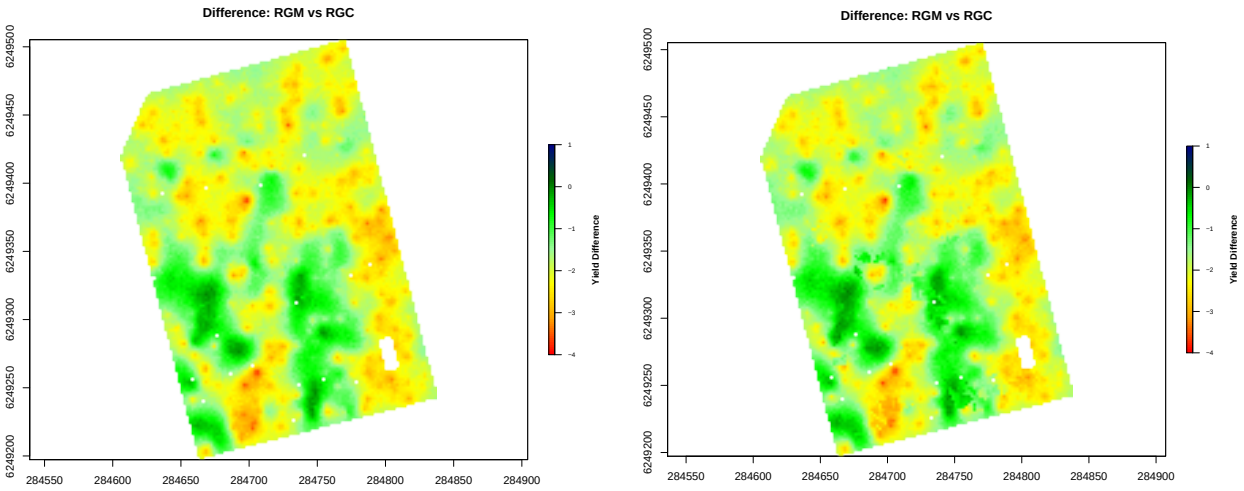


(e) Global



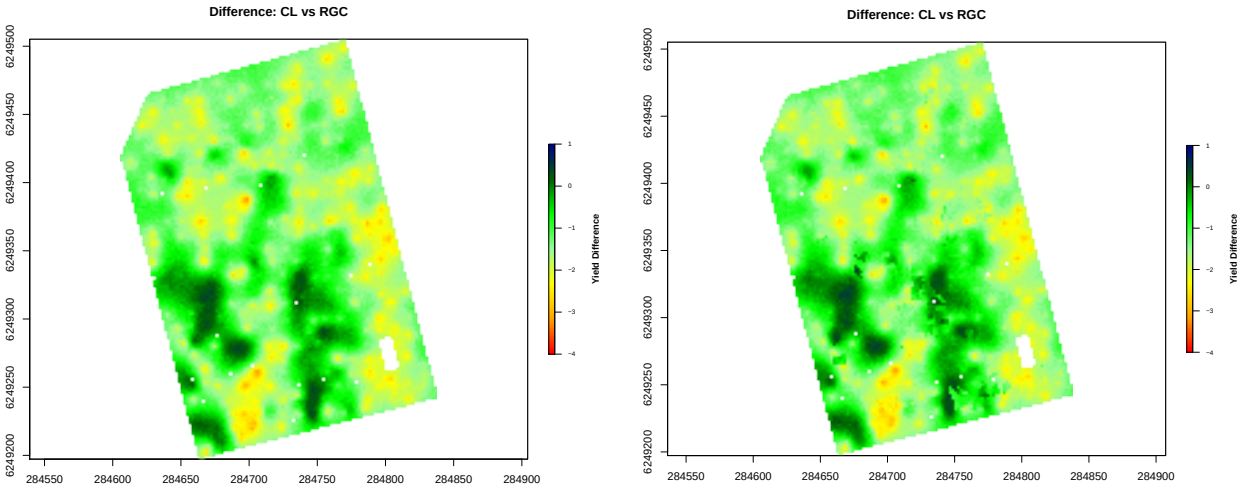
(f) Mixed

Figure 3.3: Treatment effects estimated for the RGM, RGC and CL using global (a)–(c) and mixed (d)–(f) methods (considering neighbourhood size 20 metres).



(a) Global

(b) Mixed



(c) Global

(d) Mixed

Figure 3.4: The estimated treatment differences are shown in (a) RGM vs RGC and (c) CL vs RGC using the global method. The estimates differences using mixed method are presented in (b) RGM vs RGC and (d) CL vs RGC

3.3 Example 2: Taylors V32

Taylors V32 is an ideal data set to explore the co-kriging method for the large dimensional data problem. Here two harvesters G2 and G66 were used to measure the yields, which harvested alternate rows in the same vineyard; the G2 had a slipping belt, which initiated two different treatment measurements for G2 and G66.

In this section for illustration of the local co-kriging method, the G2 (Treatment 1) and G66 (Treatment 2) are used as two treatments, see Figure 3.5. Here 9,065 data points are used to obtain co-kriging estimates in 2-metre grids, a total 20,424 grid points.

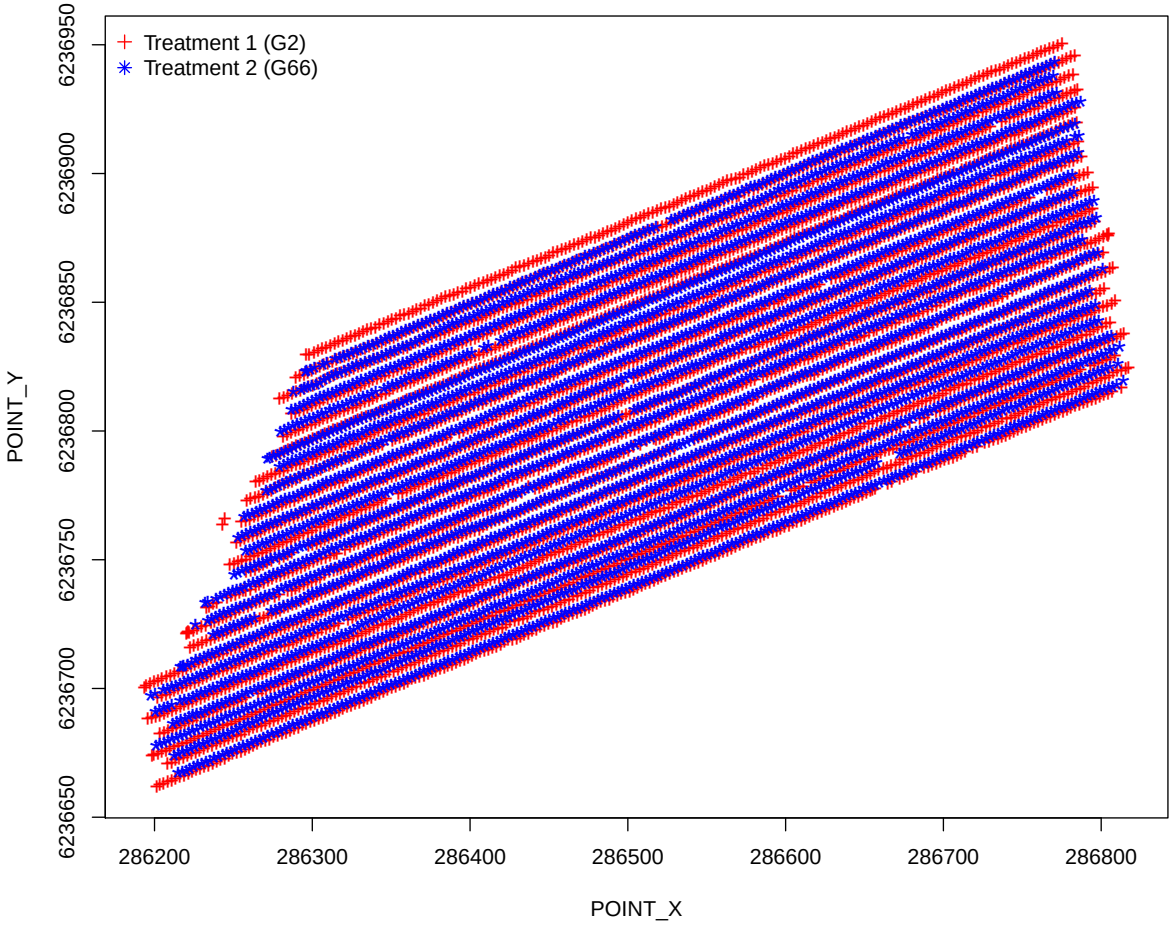


Figure 3.5: Treatment plots for Taylor V32 data set.

3.3.1 Results for the local co-kriging method

Similar to the mixed method, user can define the initial neighbourhood size for the local co-kriging method. However, unlike the mixed method, the local method updates the neighbourhood size dynamically for co-kriging estimates in each grid points. This updating uses an algorithm that provides reasonable size of the neighbourhood to obtain an invertible localised cross-covariance function.

Figures 3.6(a) and (b) show the values of the neighbourhood sizes for each prediction points. Here, the initial neighbourhood sizes are considered 5 and 10 metres respectively, and as expected the algorithm identifies the optimum size, which varies across the prediction points. The maximum neighbourhood size is estimated 20 metres, however 55% and 99% of the prediction locations used 5-metre and 10-metre as their neighbourhood sizes. The local co-kriging method also can take a default neighbourhood size. For the Taylors V32 data set the default neighbourhood size is calculate 71 metres

(see details in the Supplementary Section).

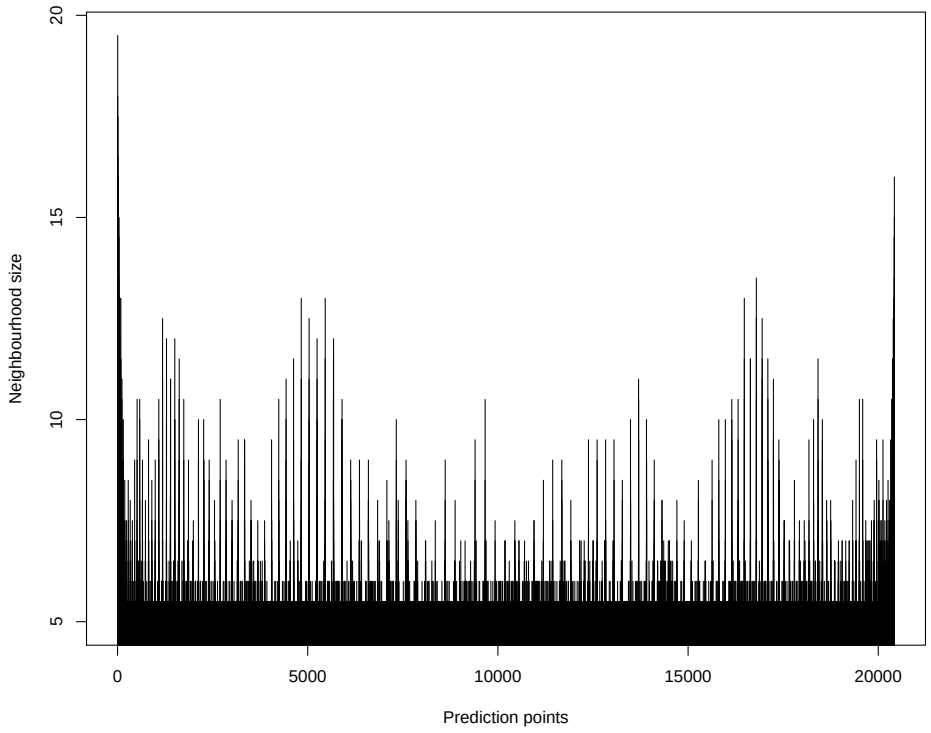
Figures 3.7(a), (c) and (d) show the estimated treatment effects using harvester G2 for robust neighbourhood with initial value 5, 10 metres and default value 71 metres respectively. We see that smaller neighbourhood yields a localised nonstationary estimates compared to a larger neibourhood. Similar results are seen for the harvester G66, see Figures 3.7(b), (d) and (e).

Figures 3.8(a), (b) and (c) show the estimated treatment effect differences for harvester G2 and G66. As expected, the treatment differences vary based on the neighbourhood size.

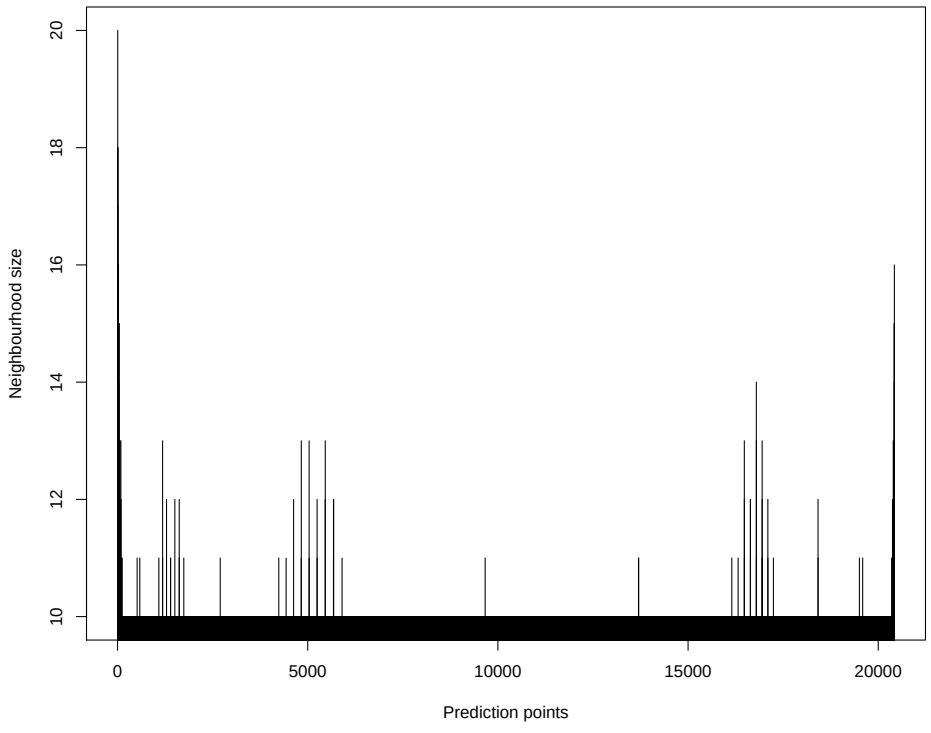
3.4 Summary

The co-kriging methods explained above are to address the challenges in on-farm experimentation. We illustrate the co-kriging methods for both small and large data sets. A global co-kriging method is easily applicable to smaller data set, however to identify the localised effects the mixed co-kriging method is used, where both global and local cross-covariance functions are utilised. To address the problems related to large dimensional data – we developed a localised co-kriging method.

To identify the parameters, e.g., neighbourhood size for the local co-kriging, a robust recurrence algorithm is developed. This approach perfectly achieves the need, however often requires more computation time. For example, for the Taylors V32 data set it took about 3 hours to obtain estimates in 20,424 grid points. Note that the computation time also depends on the size of the neighbourhood. This initiates us to explore further to identify the optimum value for the neighbourhood to minimise the computation time.

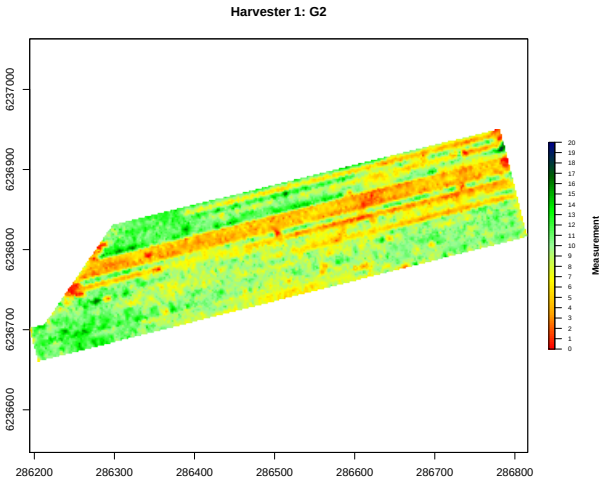


(a) Initial neighbourhood size: 5-metre

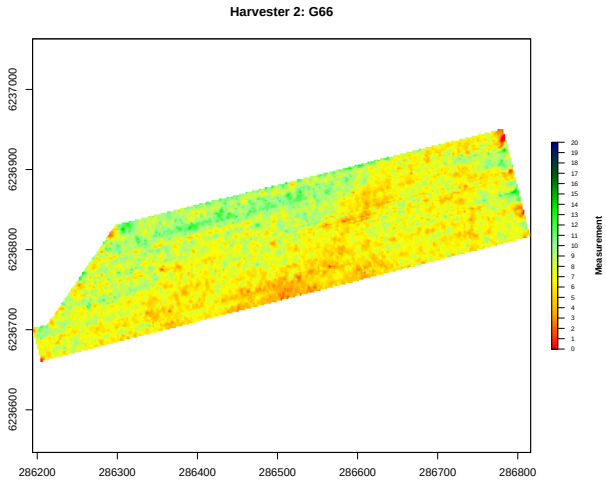


(b) Initial neighbourhood size: 10-metre

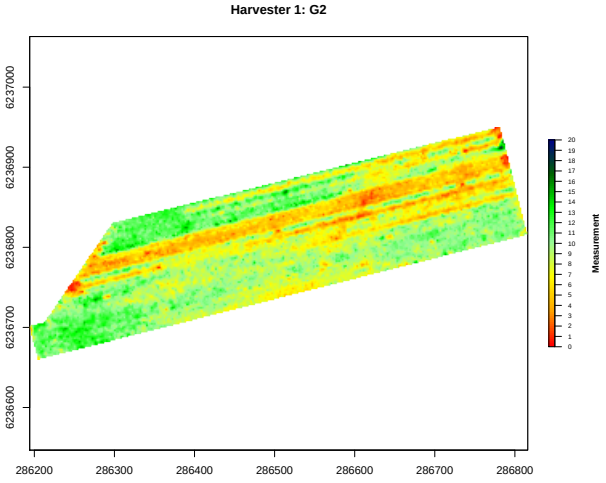
Figure 3.6: Distribution of the neighbouhood sizes for local co-kriging method considering initial neighbourhood size (a) five and (b) 10 metres.



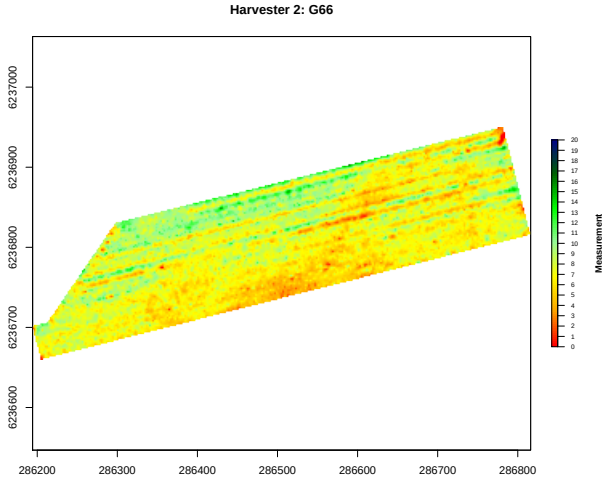
(a) Robust Neighbourhood (5 M)



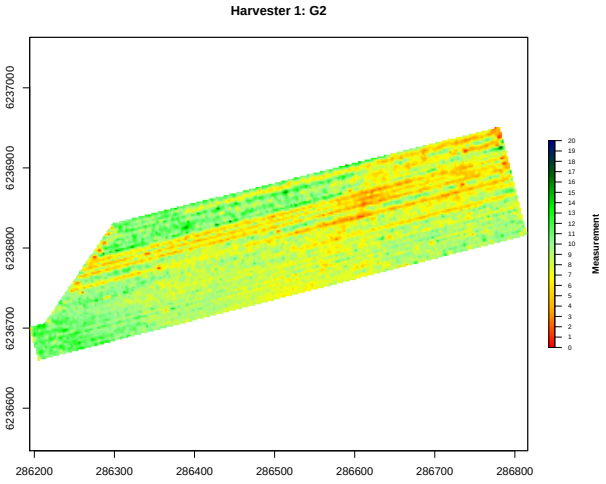
(b) Robust Neighbourhood (5 M)



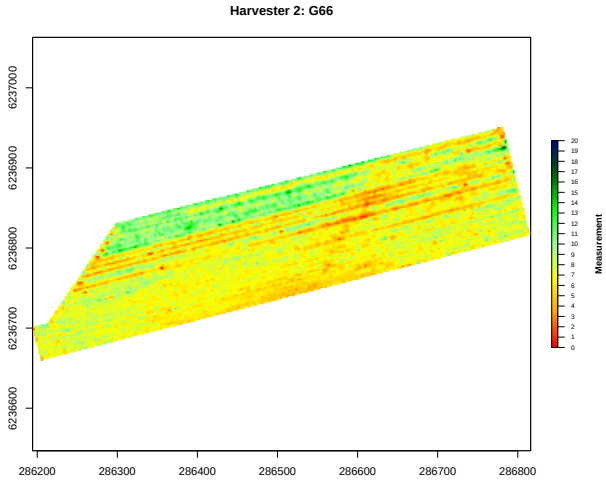
(c) Robust Neighbourhood (10 M)



(d) Robust Neighbourhood (10 M)

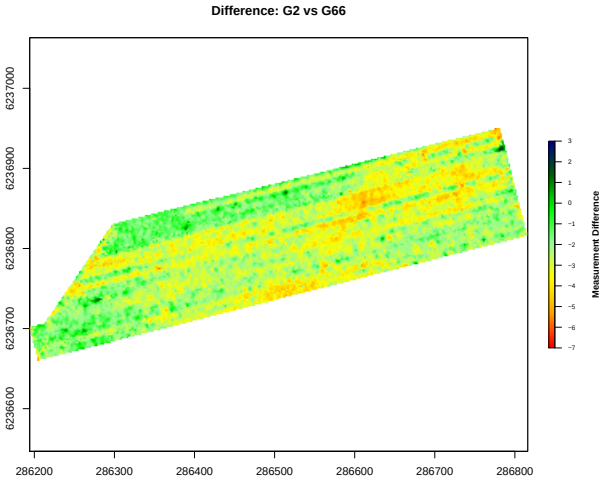


(e) Default Neighbourhood

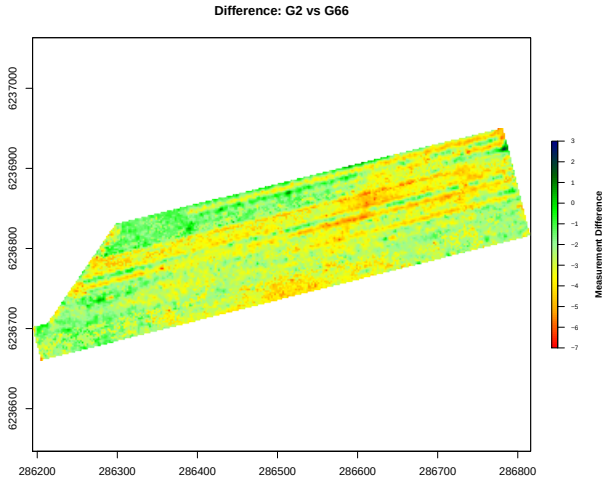


(f) Default Neighbourhood

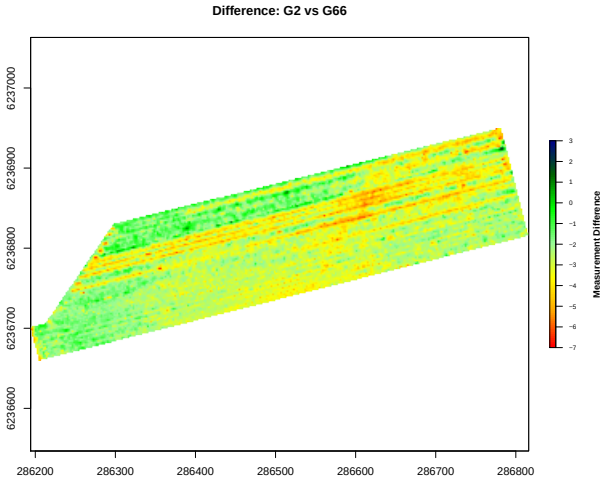
Figure 3.7: Treatment effects estimated using local co-kriging method for G2 and G66 considering (a)–(d) robust neighbourhood with starting sizes 5 and 10 metres respectively and (e)–(f) with default size 71 metres.



(a) Robust Neighbourhood (5 M)



(b) Robust Neighbourhood (10 M)



(c) Default Neighbourhood

Figure 3.8: The treatment differences are shown in (c) G2 vs G66.

4 Discussion and next steps

This report has reviewed various multivariate spatial prediction including classical multivariate geostatistics and computation techniques that may be appropriate for on-farm experiments. Implementation has subsequently focused on co-kriging under LMCR. Preliminary results demonstrate that it is working and fit for purpose.

There are however several improvements to consider and more detailed testing and comparisons to undertake in the near future. In particular, the immediate next steps include:

1. Refining the default parameter setting in the implementation such that the system can perform smoothly without manual intervention for various data sets.
2. Addressing the edge effects in the local co-kriging method.
3. Identifying the minimum number of data points that can be used to construct a workable cross-covariance matrix for the local co-kriging method.
4. Introducing a sub-sampling technique that will work as a pilot approach to understand the large-dimensional data (an extension of the local co-krging) and also to determine the corresponding neighbourhood size in the edge points of the farm.
5. Further testing the system on automatic sensed data, such as yield maps, and determining where computational thresholds may exist.
6. Extending the local co-kriging using other methods like standardised ordinary co-kriging and PCK.
7. Potentially including regression covariates, such as topography, in the co-kriging based on LMCR to improve local predictions and address non-stationarity.
8. Comparing performance against efficient spatial prediction techniques like spline, tensor products, regularisation, and multiresolution LASSO for large data sets where the intensity of observation may reduce the need to account for the spatial dependence.

While the underlying focus is on yield prediction and comparison, e.g. comparing different genotypic varieties and management options, there is also demanding for understanding the underlying ‘soil fertility’ or ‘latent yield’ as leads to the identification of different zones across the farm and drives how those zones are managed operationally. This review has highlighted alternative approaches to achieving this end that may be considered in the future if time permits.

This project builds up a user-friendly interface between the R implementation described here with an open-source GIS platform, QGIS. Based on several selection criteria, including application focus, extensible functionality, development platform, scripting function, popularity and user-friendliness, QGIS is chosen by the project team as the open-source GIS system to host the tools the project is developing/re-implementing.

5 Supplementary

5.1 Default neighbourhood size

5.1.1 Mixed co-kriging method

The default neighbourhood size is calculated as:

$$NS = f(r, d_{max}, n) = \begin{cases} d_{max} \times r/n, & \text{if } d_{max}/n \leq 1 \\ r, & \text{Otherwise} \end{cases}$$

where, NS is the neighbourhood size, r is the range parameter that can be estimated from the global cross-variogram (often in spatial statistics the range is defined as a reciprocal of the spatial decay parameter ϕ), d_{max} is the maximum distance between two spatial locations, and n is the total number of spatial locations.

5.1.2 Local co-kriging method

The neighbourhood size is chosen based on a robust approach, where in each loop it increases 10% to accommodate workable data into the model. The initial default size of the neighbourhood is calculated from a function of d_{max},T and n, where d_{max} is the maximum distance, T is the tolerance site (this is to define the threshold for the big-n problem), and n is the number of spatial locations in the different set. Hence we write:

$$NS = f(d_{max}, T, n) = d_{max}(T/n)$$

Our programming code defaults the threshold for the default big-n as 1000 spatial points, while users can specify it separately.

5.2 R code implementation

We develop R code for implementing the Global, Local and Mixed approaches. Figure 5.1 shows a snapshot of the front end R code, which has a defined class – “cokriging”.

```
# main function
coKrig <-
  function(data = NULL,
           type = "global",
           grid.size = NULL,
           grid.coords = NULL,
           shapefile = NULL,
           model = "Exp",
           neighbourhood.size = NULL,
           plot.global = FALSE,
           plot.local = FALSE,
           tol.site = 1000,
           ...) {
    #
```

Figure 5.1: R code snapshot.

We can list the input arguments of the function as follows:

`data:` This argument is used to input data. The data set should be pre-processed into 4 columns. First two columns will represent x and y coordinates of the data points. Third column will represent the values, and the fourth column will represent the treatment index or name. For example:

```
> head(y)
   Easting Northing    yield treatment
1 284769.3  6249498 2.097618         1
2 284776.3  6249471 2.345208         1
3 284767.3  6249492 3.472111         1
4 284785.5  6249424 3.162278         1
5 284774.5  6249454 3.651484         1
6 284805.1  6249313 1.977142         1
> tail(y)
    Easting Northing    yield treatment
372  284617  6249413 5.691567         3
373  284620  6249405 6.444756         3
374  284614  6249416 5.277987         3
375  284620  6249396 6.368870         3
376  284612  6249412 5.413658         3
377  284620  6249380 7.011152         3
```

`type:` The cokriging approach that developed here is based on three different methods, hence this argument can take "global", "local" and "mixed".

`grid.size:` This is an option for user to define the grid size. The format is a vector input with number of rows and columns. For example, $c(10, 10)$ implies that it will be a 10×10 grid.

`grid.coords:` Users can also define the grid coordinates instead of the grid size. Hence, it will be a $2 \times m$ matrix or data frame of (x, y) grid points. Here m is the number of kriging points.

`shapefile:` Users can also give a shape file input using this argument.

`model:` This argument defines the covariance model. Currently it can take "Exp", "Sph" and "Gau", where Exp=exponential, Sph=Spherical and Gau=Gaussian model.

`neighbourhood.size:` This argument is used for local cokriging only. This will define the boundary of the local co-kriging and requires one vector input, e.g. $c(25)$ implies the boundary size is 25 units (e.g. metres/km). Currently it is implemented as a radius.

`plot.global:` This is a logical function (TRUE or FALSE). Only applicable for global approach. If TRUE then it will draw a plot of the semi-variogram.

`plot.local:` This is a logical function and only applicable for the mixed approach. If TRUE then draw plots of the semi-variogram for all kriging points.

`...:` Other arguments, see details in "par" function of R.

The programming code also provides cross-semivariogram plots on the fly for each localised points. Note that to minimise the computational costs, the cross-semivariogram plot option is not available for the "local" approach. To obtain the co-kriging plots and corresponding standard errors, we define a S3 class plot function. Thus `> plot(output)` will convert the co-kriging output into a shapefile and then plot the predictions and corresponding SEs.

References

Rob Bramley, Kerstin Panten, and David Gobbett. Making sense of vineyard variability – completion of crv99/5n – 1.1.1. Report CSL 06/03, Grape and Wine Research and Development Corporation, CSIRO Sustainable Ecosystems, Adelaide, Jan 2008.

K Panten, RGV Bramley, RM Lark, and TFA Bishop. Enhancing the value of field experimentation through whole-of-block designs. *Precision Agriculture*, 11(2):198–213, 2010.

M. G. Genton and W. Kleiber. Cross-covariance functions for multivariate geostatistics (with discussion). *Statistical Science*, 30:147–163, 2015.

Robert Bramley and Sam Trengove. Precision agriculture in Australia: present status and recent developments. *Engenharia Agrícola*, 33(3):575–588, 2013.

Hans Wackernagel. *Multivariate geostatistics: an introduction with applications*. Springer Science & Business Media, 2013.

Alan E Gelfand and Sudipto Banerjee. Multivariate spatial process models. *Handbook of Spatial Statistics*, pages 495–515, 2010.

Jay M Ver Hoef and Ronald Paul Barry. Constructing and fitting models for cokriging and multivariable spatial prediction. *Journal of Statistical Planning and Inference*, 69(2):275–294, 1998.

Montserrat Fuentes. Spectral methods for nonstationary spatial processes. *Biometrika*, 89(1):197–210, 2002.

Noel Cressie and Andrew Zammit-Mangion. Multivariate spatial covariance models: A conditional approach. *Biometrika*, 103(4):915–935, December 2016. DOI: https://doi.org/10.1093/biomet/asw045.

A Papritz and H Flühler. Temporal change of spatially autocorrelated soil properties: optimal estimation by cokriging. *Geoderma*, 62(1-3):29–43, 1994.

T.F.A. Bishop and R.M. Lark. The geostatistical analysis of experiments at the landscape-scale. *Geoderma*, 133(1):87–106, 2006. https://doi.org/10.1016/j.geoderma.2006.03.039.

Nicolas Desassis and Didier Renard. Automatic variogram modeling by iterative least squares: univariate and multivariate cases. *Mathematical Geosciences*, 45(4):453–470, 2013.

A Papritz, HR Künsch, and R Webster. On the pseudo cross-variogram. *Mathematical Geology*, 25 (8):1015–1026, 1993.

Jay M Ver Hoef and Noel Cressie. Multivariable spatial prediction. *Mathematical Geology*, 25(2): 219–240, 1993.

S Sonka. Big data: from hype to agricultural tool. *Farm Policy Journal*, 12(1):1–9, 2015.

Ying Sun, Bo Li, and Marc G Genton. Geostatistics for large datasets. In *Advances and challenges in space-time modelling of natural events*, pages 55–77. Springer, 2012.

M Fuentes and A E Raftery. Model evaluation and spatial interpolation by Bayesian combination of observations with outputs from numerical models. *Biometrics*, 61:36–45, 2005.

Reinhard Furrer, Marc G Genton, and Douglas Nychka. Covariance tapering for interpolation of large spatial datasets. *Journal of Computational and Graphical Statistics*, 15:502–523, 2012.

Huiyan Sang, Mikyoung Jun, and Jianhua Z Huang. Covariance approximation for large multivariate spatial data sets with an application to multiple climate model errors. *The Annals of Applied Statistics*, pages 2519–2548, 2011.

Noel Cressie and Emily L Kang. High-resolution digital soil mapping: Kriging for very large datasets. In *Proximal Soil Sensing*, pages 49–63. Springer, 2010.

Rajarshi Guhaniyogi, Andrew O Finley, Sudipto Banerjee, and Alan E Gelfand. Adaptive Gaussian predictive process models for large spatial datasets. *Environmetrics*, 22(8):997–1007, 2011.

Sudipto Banerjee, Alan E Gelfand, Andrew O Finley, and H Sang. Gaussian predictive process models for large spatial data sets. *Journal of Royal Statistical Society B*, 70:825–848, 2008.

Joaquin Quiñonero-Candela and Carl Edward Rasmussen. A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research*, 6(Dec):1939–1959, 2005.

N A C Cressie. *Statistics for Spatial Data*. John Wiley & Sons, New York, 1993.

Reinhard Furrer and Marc G. Genton. Aggregation-cokriging for highly multivariate spatial data. *Biometrika*, 98(3):615–631, 2011. ISSN 00063444.

Alan E Gelfand and Sudipto Banerjee. Bayesian modeling and analysis of geostatistical data. *Annual Review of Statistics and Its Application*, 4:245–266, 2017. DOI: 10.1146/annurev-statistics-060116-054155.

A. O. Finley, S. Banerjee, A. R. Ek, and R. E. McRoberts. Bayesian multivariate process modeling for prediction of forest attributes. *Journal of Agricultural, Biological, and Environmental Statistics*, 13: 60–83, 2008.

S. Banerjee, B. P. Carlin, and A. E. Gelfand. *Hierarchical Modeling and Analysis for Spatial Data,* 2nd edn. Chapman and Hall/CRC, Boca Raton, FL, 2015.

K Shuvo Bakar, Philip Kokic, and Huidong Jin. A spatiodynamic model for assessing frost risk in south-eastern Australia. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 64 (5):755–778, 2015.

RDevelopmentCoreTeam. R*: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2006. URL http://www.R-project.org. ISBN 3-900051-07-0.

CONTACT US

t 1300 363 400
+61 3 9545 2176
e csiroenquiries@csiro.au
w www.csiro.au

WE DO THE EXTRAORDINARY EVERY DAY

We innovate for tomorrow and help improve today Ű for our customers, all Australians and the world.

Our innovations contribute billions of dollars to the Australian economy every year. As the largest patent holder in the nation, our vast wealth of intellectual property has led to more than 150 spin-off companies.

With more than 5,000 experts and a burning desire to get things done, we are AustraliaŠs catalyst for innovation.

WE IMAGINE. WE COLLABORATE. WE INNOVATE.

FOR FURTHER INFORMATION

Data61
Warren JIN
t +61 2 6216 7055
e Warren.JIN@csiro.au
w www.data61.csiro.au

Data61
Shuvo Bakar
t +61 2 6218 3624
e Shuvo.Bakar@data61.csiro.au
w www.data61.csiro.au

Data61
Brent Henderson
t +61 2 6216 7095
e Brent.Henderson@csiro.au
w www.data61.csiro.au

Appendix 3: PAT – Precision Agriculture Tools Plugin for QGIS, User Manual

AGRICULTURE AND FOOD
www.csiro.au

PAT - Precision Agriculture Tools Plugin for QGIS

User Manual

Christina Ratcliff, David Gobbett, Rob Bramley
June 2019 PAT v0.3.0



Wine Australia

Citation

Ratcliff, Christina; Gobbett, David; Bramley, Rob (2019): PAT - Precision Agriculture Tools. v1. CSIRO. Software Collection. https://doi.org/10.25919/5c731a813b91a

Copyright

© Commonwealth Scientific and Industrial Research Organisation 2018. To the extent permitted by law, all rights are reserved and no part of this publication covered by copyright may be reproduced or copied in any form or by any means except with the written permission of CSIRO.

Important disclaimer

The notes and software described here are based on a number of technical, circumstantial or otherwise specified assumptions and parameters. The user must make their own assessment of the suitability for use of the information or material contained in or generated from these notes. To the extent permitted by law, CSIRO excludes all liability to any party for expenses, losses, damages and costs arising directly or indirectly from using these notes and software tools.

Enquiries

Contact PAT@csiro.au for queries about the tools described in this manual.

Contents

PAT - Precision Agriculture Tools 4
1 Installing, Upgrading and Uninstalling 7
1.1 License 7
1.2 Requirements 7
1.3 Installing PAT 7
1.4 Update PAT 12
1.5 Uninstall PAT 13
2 Individual Tools 14
2.1 Create Block Grid 14
2.2 Clean, Trim, Normalise Point Data 16
2.3 Run Kriging Using VESPER 20
2.4 Import VESPER Results 23
2.5 Create Polygon from On-The-Go GPS Point Trail Data 24
2.6 Rescale or Normalise a Raster 28
2.7 Generate Random Pixel Selection 29
2.8 Extract Pixel Statistics for Points 30
2.9 Calculate Image Indices for Blocks 33
2.10 Resample Image Band for Blocks 36
2.11 Create Zones with *k*-means Clusters 39
2.12 Create Strip Trial Points 41
2.13 Run Strip Trial *t*-test Analysis 43
2.14 Whole-of-block Analysis 47
2.15 Persistor 51
2.16 Apply Raster Symbology 54
2.17 Settings 55
3 Technical Notes 56
4 How-To's 57
4.1 Create a block boundary polygon from a CSV of GPS collected points 57
4.2 How to reproject a shapefile and why you need to do this 59
4.3 Loading PAT Symbols into QGIS 60
4.4 Kriging 'low density' data in PAT 61
5 Contributors 65
6 Acknowledgments 65
7 References 66

PAT - Precision Agriculture Tools

The Precision Agriculture Tools (PAT) plugin is a suite of open source tools developed by CSIRO for Precision Agriculture (PA) data analysis. The tools run within Quantum Geographic Information System (QGIS) version 2.18 (QGIS Development Team 2017), a free and open-source desktop geographic information system that supports viewing, editing, and analysis of geospatial data. PAT aims to provide an easy-to-use interface for processing PA data through an established workflow developed for constructing maps using on-the-go data (e.g. from yield monitors or EM38 soil surveys) as shown in Figure 1 and Table 1 (Bramley and Williams 2001; Taylor et al. 2007; Bramley et al. 2008; Bramley and Jensen 2014). It also includes tools for analysis of remotely sensed imagery and some on farm experiments and for clustering map layers to create potential management zones. Over time more tools and 'how-to' instructions will be added to expand the functionality and usefulness for both practical and research purposes.

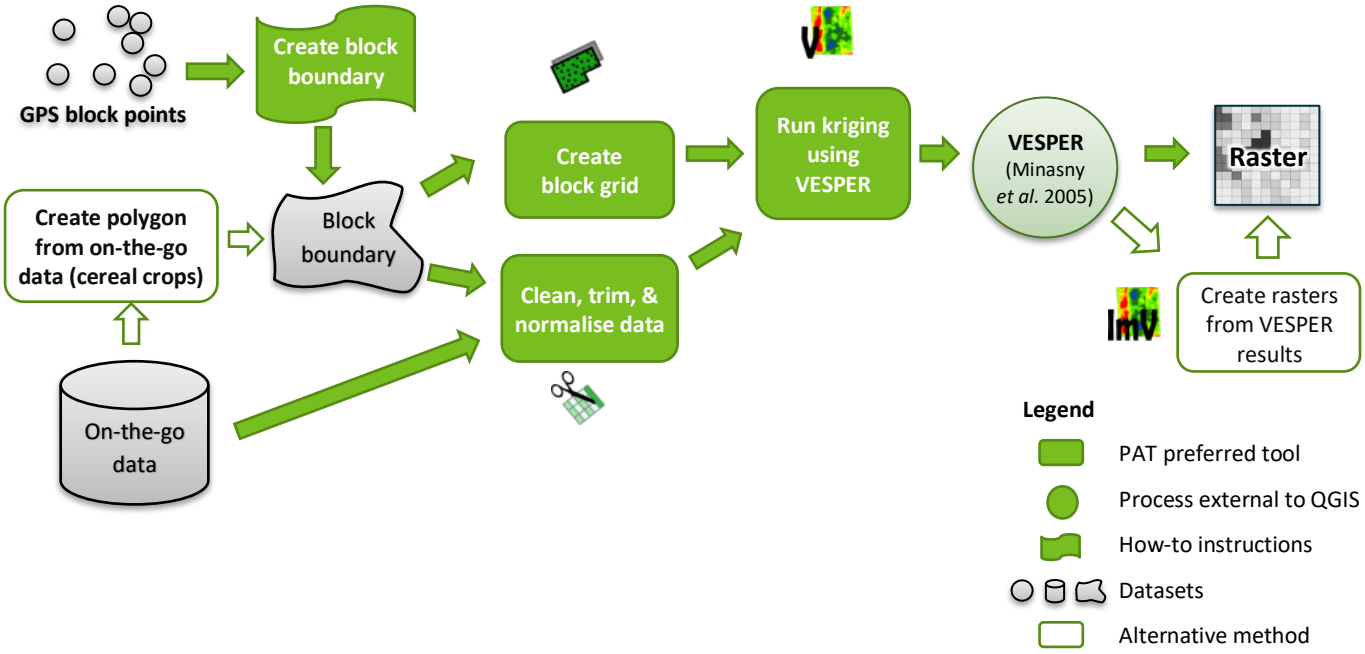


Figure 1 The basic map production workflow used in PAT and some of the existing tools available for processing high density on-the-go data.

Table 1 Brief description of tools available in PAT

	PAT TOOLS	DESCRIPTION
	**Create Block Grid**	Convert polygon features (e.g. a block boundary – see page 57) to • a TIFF raster; and • a VESPER text grid file of X,Y point values used by VESPER.
	**Clean, Trim, Normalise Point Data**	Process an on-the-go data file (e.g. from yield monitors) by applying clipping, cleaning and filtering rules and output as a CSV file.
	**Run Kriging Using VESPER**	Create a VESPER control file and data files and run VESPER kriging.
	**Import VESPER Results**	Convert VESPER outputs to raster TIFF format.
	**Create Polygon from On-The-Go GPS Point Trail Data**	Generate a polygon block boundary from on-the-go data (e.g. from yield monitors) containing GPS points. The GPS points must cover the entire block (not just be a set of points around the boundary - see page 57 for instructions on creating a boundary from a set of surveyed data points).
	**Rescale or Normalise a Raster**	Create rasters by • rescaling (standardising) values between a fixed range (i.e. 0-1, or 0-255); or • normalising values to a mean of 0 and standard deviation of 1.
	**Generate Random Pixel Selection**	Generate a selection of random pixels from a raster and save to a points Shapefile.
	**Extract Pixel Statistics for Points**	Extract pixel statistics using a square neighbourhood footprint from multiple rasters at set locations.
	**Calculate Image Indices for Blocks**	Resample and smooth imagery to a larger pixel size, as well as calculate indices such as PCD and NDVI.
	**Resample Image Band for Blocks**	Clip imagery to one or more blocks and resample to a different pixel size.
	**Create Zones with k-means Cluster**	Create zones with *k*-means clustering.
	**Create Strip Trial Points**	Tool for analysing simple strip-based trials. It generates pairs of points from a central line within a strip and with 'a pair' offset at a distance outset the strip. These points can be used for the comparison of performance along and adjacent to a strip trial (such as an N-rich strip) using the Lawes and Bramley (2012) moving window t-test tool.
	**Run Strip Trial *t*-test Analysis**	The comparison of performance of a treatment, such as different fertiliser rates, along and adjacent to a strip trial (such as an N-rich strip) using the moving window t test as described in Lawes and Bramley (2012).
	**Whole-of-block Analysis**	Analyse whole-of-block experiments where 2 or 3 treatments are applied across a block or paddock.
	**Persistor**	Summarise the patterns of yield across a block over a number of years relative to a target, as described by Bramley and Hamilton (2005).
	**Apply Raster Symbology**	Apply predefined symbology to a displayed raster.
	**Settings**	User settings for PAT. Provides the ability to set data directories and the location of VESPER (if installed).
	**About**	About Precision Agriculture Tools (PAT). This provides information on the currently installed release, and relevant open source licences.
	**Help**	Display the PAT user manual.

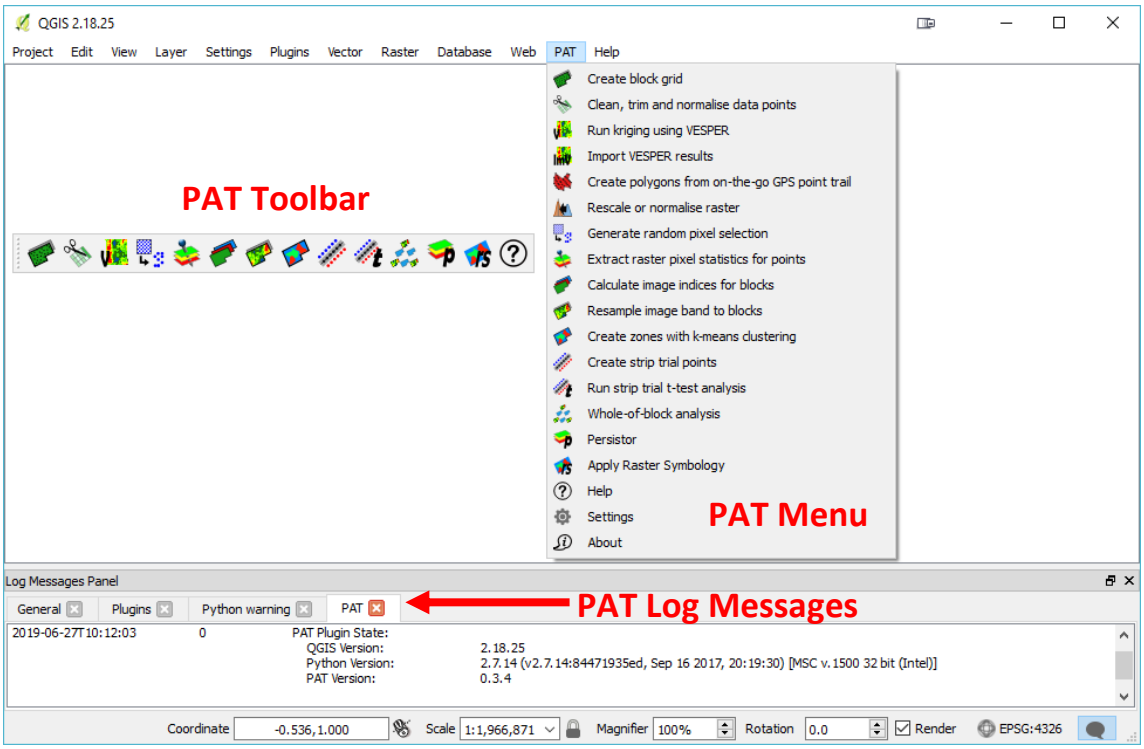


Figure 2 PAT toolbar, menu and log messages panel within QGIS.

1 Installing, Upgrading and Uninstalling

1.1 License

The PAT plugin for QGIS is free to download and use under a GPL V3 Open Source license. PAT license details can also be found in the ***About*** tool under PAT menu. The companion Python library 'pyprecag' (Ratcliff *et al.* 2019) is also freely available and is released under the CSIRO Open Source Software License Agreement (variation of the BSD / MIT License). All source code and license files are available from https://github.com/CSIRO-Precision-Agriculture.

1.2 Requirements

VESPER: To use the ***Run Kriging Using VESPER*** tool requires installation of VESPER (note: only available for Microsoft Windows operating systems). VESPER is a kriging (map interpolation) program (Minasny *et al.* 2005). It needs to be installed independently if you would like to undertake kriging of maps using VESPER. VESPER is **not** distributed with this plugin but is the recommended map interpolation tool. To download or view more information on VESPER visit https://sydney.edu.au/agriculture/pal/software/vesper.shtml.

R 3.5.1+: The R statistical language (R Core Team 2018) is required to carry out the analytical processing for the *Whole-of-block Analysis* (Section 2.14). R is **not** distributed with this plugin. If ***Whole-of-block analysis*** is required, then you must download and install R in addition to QGIS. The minimum R version suitable for this is v3.5.1 Configuring QGIS to use R section on page 10.

QGIS LTR 2.18.21-26: Download and install the QGIS standalone long-term release (versions 2.18.21 to 2.18.26). **Note that the PAT tools do not currently work with QGIS version 3 or later.**

QGIS 2.18.26 can be downloaded directly using these links.

32bit : http://download.osgeo.org/qgis/win64/QGIS-OSGeo4W-2.18.26-1-Setup-x86.exe
64bit: http://download.osgeo.org/qgis/win64/QGIS-OSGeo4W-2.18.26-1-Setup-x86_64.exe

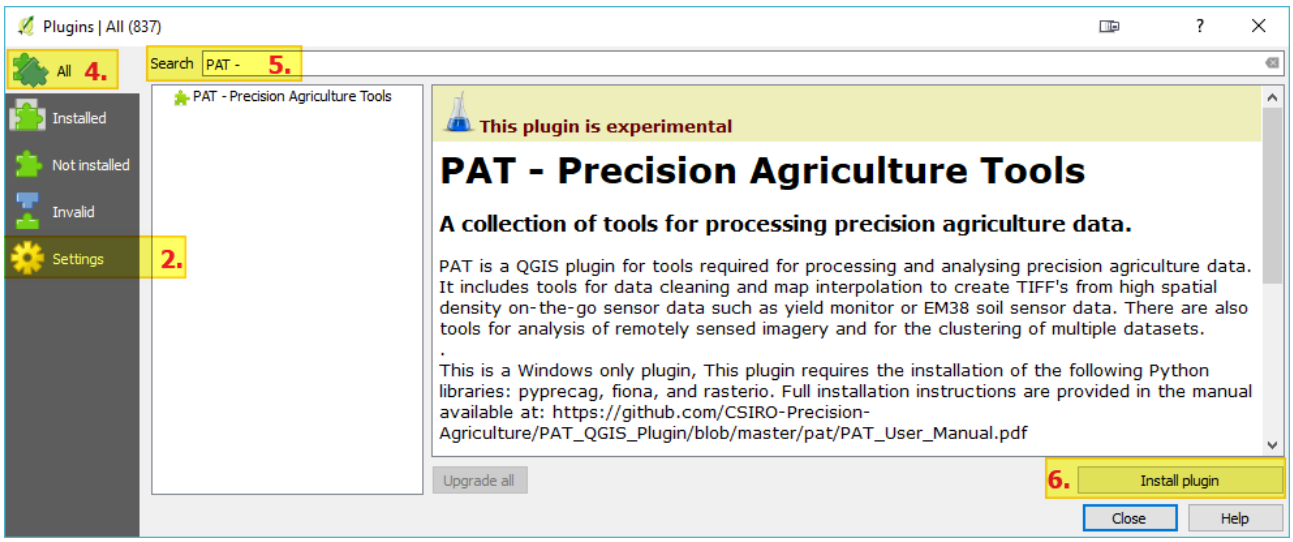
Additional Python Packages: QGIS includes Python and numerous Python packages. However, PAT requires the following additional packages: ***pyprecag*** (Ratcliff *et al*. 2019), ***Fiona*** (Gilles *et al*. 2011) ***and Rasterio*** (Gilles *et al.* 2013). The ***Fiona*** and ***Rasterio*** packages have been bundled with the PAT plugin and instructions on how to install them are included.

1.3 Installing PAT

1.3.1 Install the PAT - Precision Agriculture Tools Plugin.

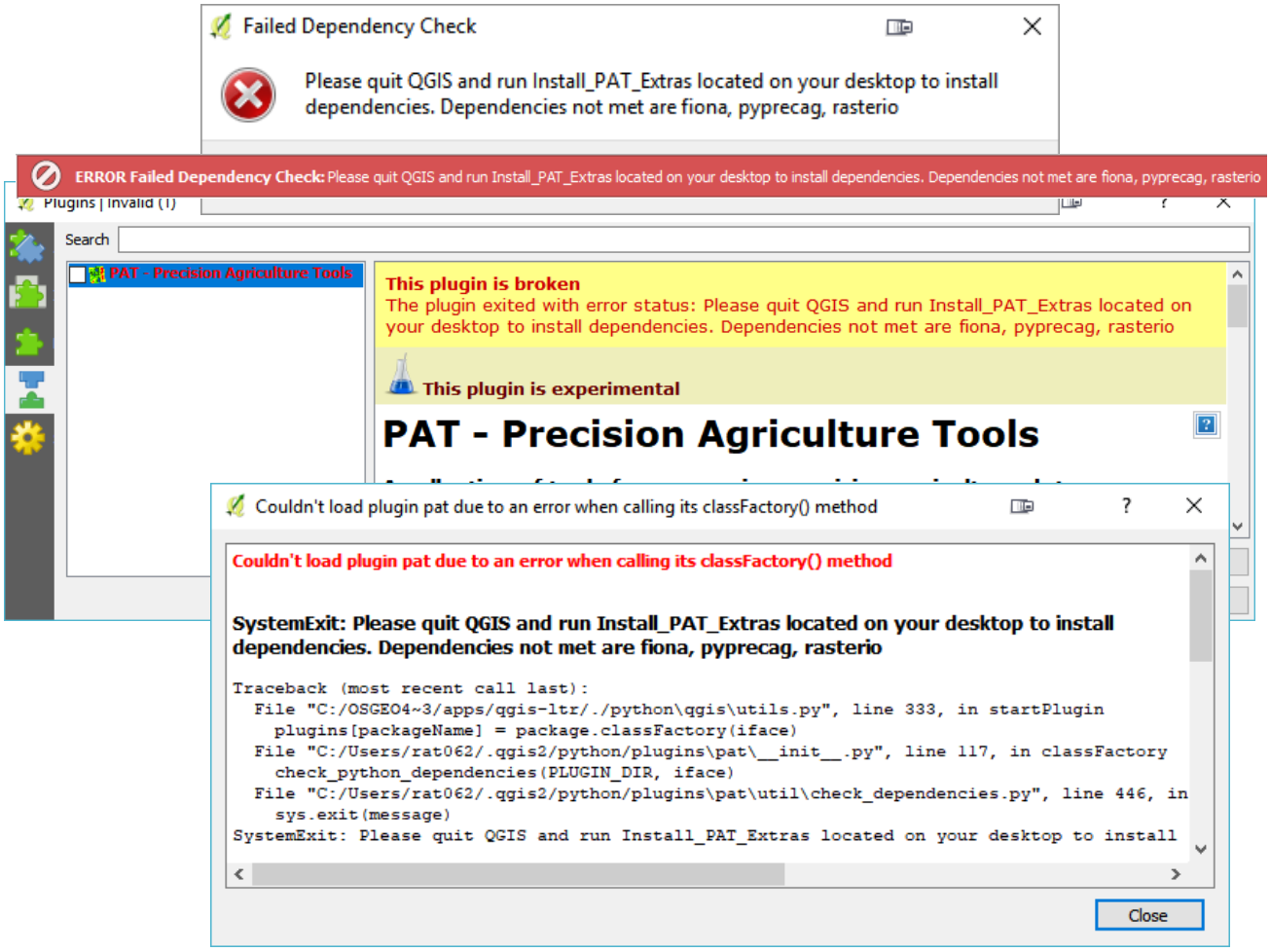
1. In QGIS open the plugin manager.
   Plugins Menu → Manage and Install Plugins
2. Select the ***Settings*** side tab.
3. Tick ***Check for Updates*** and ***Show Experimental Plugins*** options.
4. Select the ***All*** or ***Not installed*** side tab.
5. Search for and select **PAT - Precision Agriculture Tools.**
6. Click ***Install plugin***.

You will receive an ***Failed Dependency Check*** message box and ***Error loading plugin*** messages. This is expected and is resolved by completing the next section titled **Installing or Upgrading PAT Python Dependencies.**

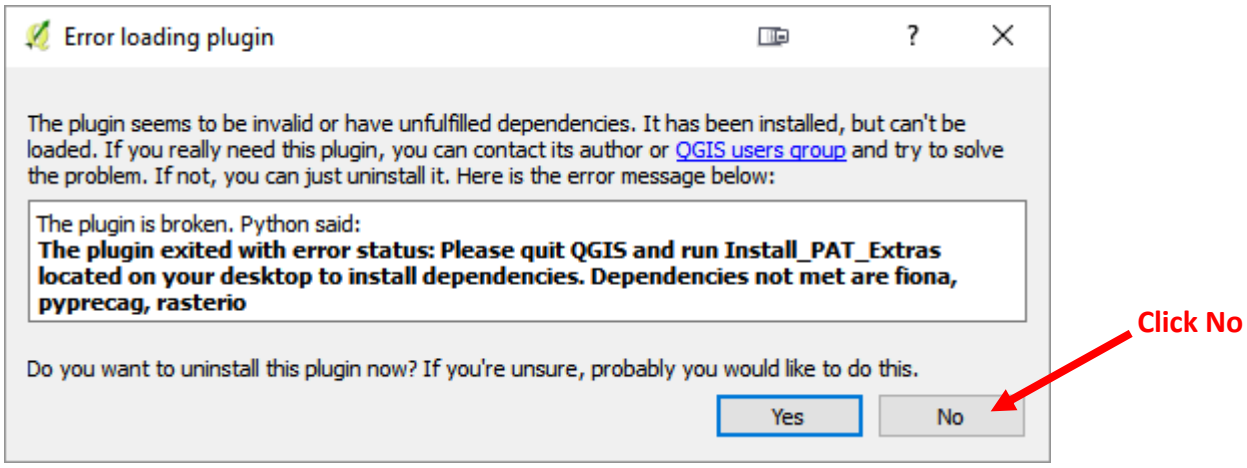


1.3.2 Installing or Upgrading PAT Python Dependencies

While installing or upgrading PAT, a check will be completed to ensure the required Python packages are installed on your system. If any of the images below appear, then this check has failed. This is normal.



If the following message appears click **No.**



Steps to install dependencies.

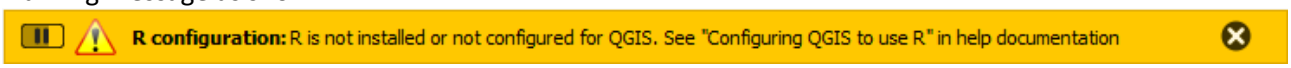


1. Close QGIS and go to your windows desktop.
2. A shortcut has been created on your desktop named ***Install_PAT_Extras***. Double click the shortcut to run it.
3. Choose **YES** for the ***Windows Account Control*** and any other messages which may appear.
4. Restart QGIS. The check will run again to ensure the installation occurred correctly.

If the PAT menu and toolbar are not present, then reinstall and/or check/activate PAT using the plugin manager. QGIS should now contain the PAT menu, toolbar and log panel similar to Figure 2 on page 6. PAT has now been successfully installed.

1.3.3 Configuring QGIS to use R

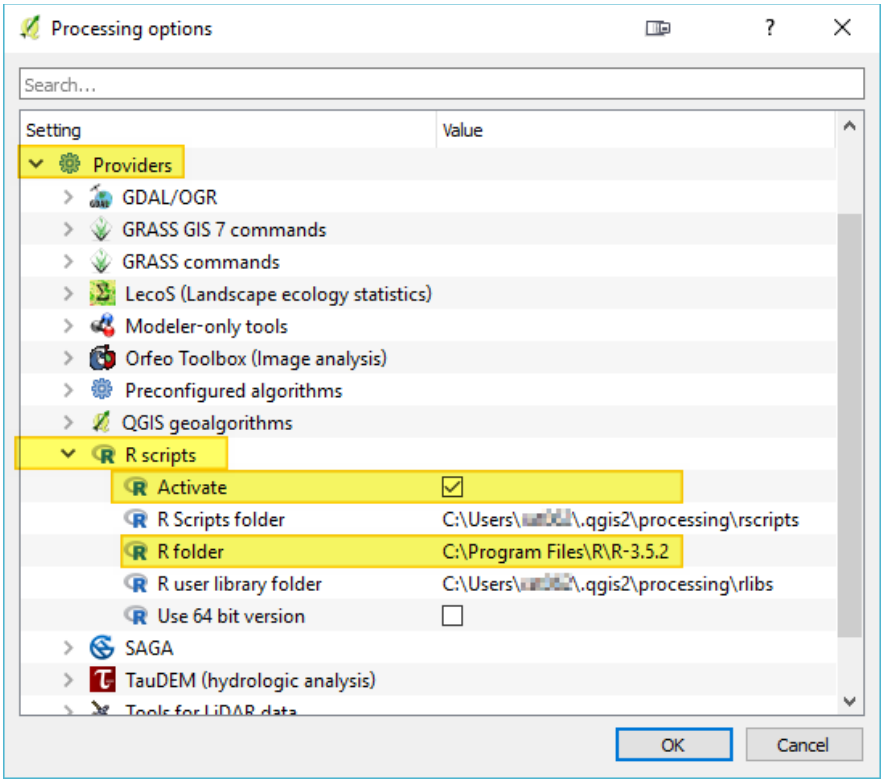
If R is not installed and you launch the ***Whole-of-block analysis tools*** via the menu or toolbar you will a warning message as shown.



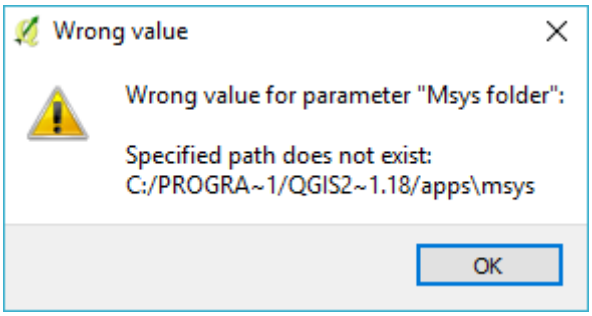
To resolve this, complete the following steps to configure R and QGIS

1. Download R from Windows version can be downloaded from http://r-project.org/ . The minimum R version suitable for this is v3.5.1.
2. Install using all the default options
3. If required, activate the **Processing** plugin in QGIS plugin manager.
    a. In QGIS open the plugin manager.
    Plugins Menu → Manage and Install Plugins
    b. Select the ***Installed*** *section*
    c. Find **Processing** and ensure the adjacent checkbox is ticked.
    d. Close the Plugins dialog.
    This will give you a processing menu and the processing side panel.

4. From the ***Processing*** menu select ***Options***.
    a. Expand ***Providers***
    b. Expand ***R scripts***
    c. Tick on ***Activate***
    d. Ensure that the R installation folder is shown against ***R folder***
    e. Click OK.

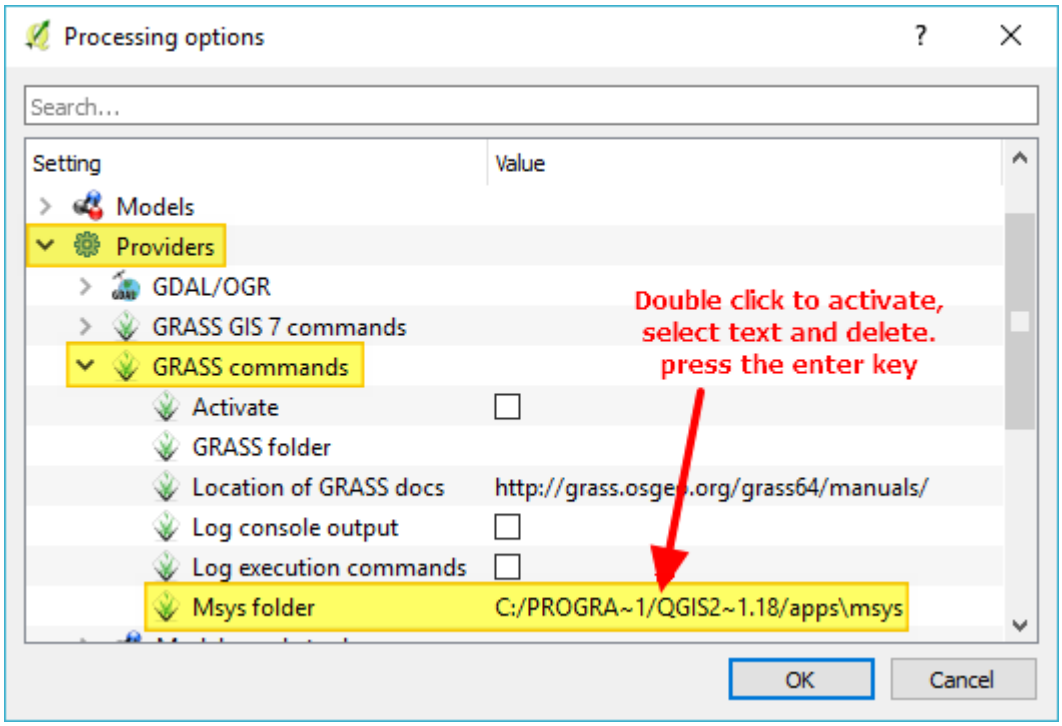


When clicking OK you may get a ***Wrong value*** error message as shown below, and your mouse cursor will remain a circle.



To resolve treat the mouse circle as an arrow and

1. Under ***Providers***, expand ***GRASS commands***
2. Double click the text to the right of Msys folder, select and delete it and press the enter key.



3. Click OK
4. Close and re-open QGIS to reset your mouse cursor.

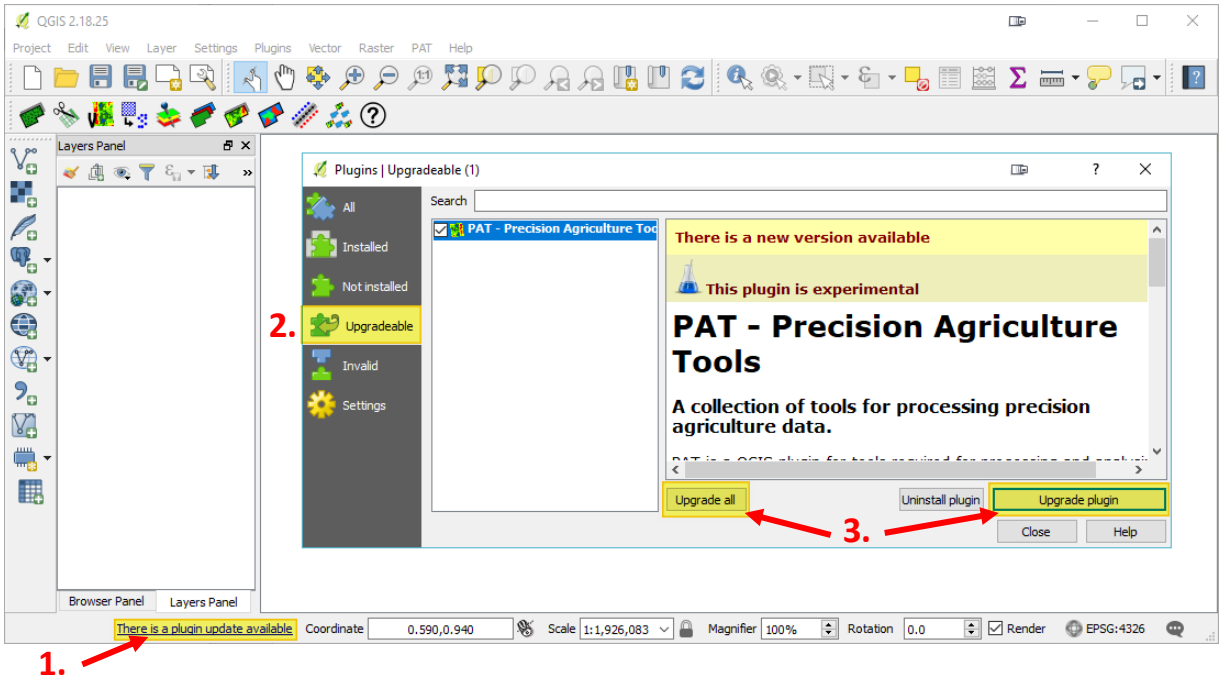
1.4 Update PAT

If the ***Check for Updates*** option is checked on in the plugin manager, QGIS will advise the user if new plugins are available, or if installed plugins have been updated. This notification is displayed in the QGIS interface's status bar as shown below. It is recommended that ***Check for Updates*** is checked.

To Update

1. In QGIS open the Plugin Manager by clicking on the link in the status bar or via ***Plugins. Menu → Manage and Install Plugins.***
2. Either select the ***Upgradeable*** left side tab or search for your plugin.
3. Upgrade by selecting the PAT plugin and click ***Upgrade plugin.***

A check will be run to ensure the Python packages required by PAT are installed and are of the correct version. If this check fails, follow the instructions in *Installing or Upgrading PAT Python Dependencies* to upgrade the dependencies.

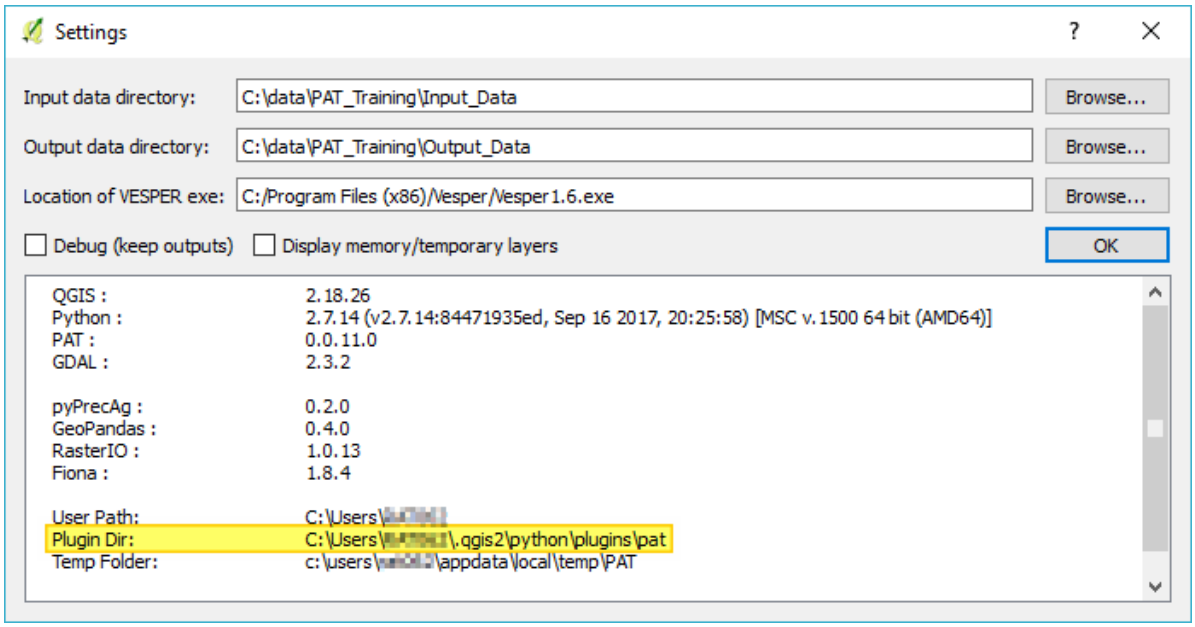


1.5 Uninstall PAT

If you wish to uninstall PAT and its Python dependencies you must first uninstall the Python dependencies, followed by the plugin as instructed below.

1.5.1 Uninstall PAT Python dependencies for all users.

1. Navigate to the current user's PAT installation directory. This folder is listed in as **Plugin Dir** in the ***settings dialog*** as shown highlighted in yellow below.



2. Find the ***UnInstall_PAT_Extras.bat*** file in the python_packages folder.
3. **Right click** the file and select ***Run as Administrator*** and choose ***Yes*** for any messages which appear.

1.5.2 Uninstall PAT Plugin.

PAT Python dependencies should be uninstalled PRIOR to uninstalling the PAT plugin. The plugin can be uninstalled in one of two ways.

1. **Via the QGIS Plugin Manager**
    a. In QGIS, open the plugin manager, and find the PAT plugin.
    b. Click Uninstall plugin.

2. **Via Windows Explorer.**
    a. Navigate to the current users PAT installation directory as show in 1.5.1.1 above.
    b. Delete the entire pat folder.

2 Individual Tools



2.1 Create Block Grid

Summary

This tool converts polygon features such as a block boundary to a raster grid using a set pixel size. This is a critical part of setting up the analysis environment for creating maps from on-the-go data because it generates the base grid onto which maps will be interpolated, by using the outer most extent of the block boundary as the 'frame' for the grid. Refer to instructions on page 57 on how to generate a block boundary.

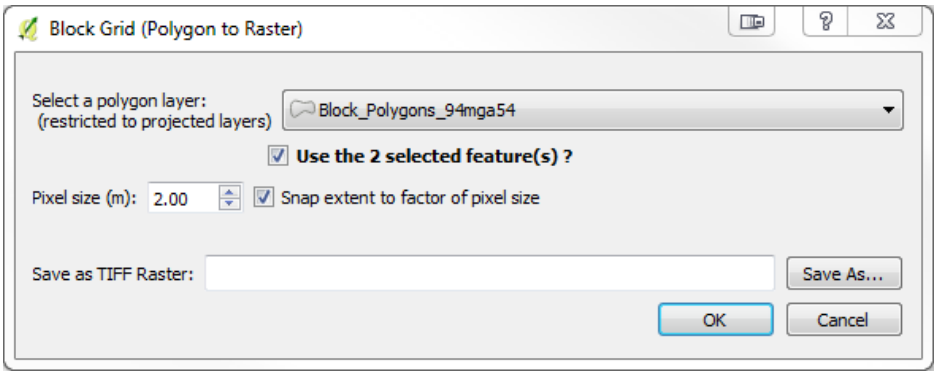
The raster outputs created by the Block Grid tool are:

- a TIFF raster; and
- a grid file of X, Y point values used by VESPER for kriging.

Areas inside the polygon boundary will be assigned a value of 1 while areas outside will be assigned a no-data value of -9999.

If your shapefile layer does not appear in the layer list, then it is probably in a geographic coordinate system. Refer to page 59 for information on why using a projected coordinate system is important, and how to reproject it. A future version of the ***Create Block Grid*** tool will include this conversion when required.

Dialog



Block Grid (Polygon to Raster)		
Select a polygon layer		A layer containing the block boundary polygon features in a projected coordinate system to be converted to the raster outputs.
Use selected features	Default is unchecked	If checked, will only use the selected features, if unchecked, all the features will be used. Useful in creating a grid for a single field from a boundary layer comprising more than one field.
Pixel size (m)	0.00m to 6 km Default is 2 m	The pixel size to assign to the raster outputs. This is expressed in metres. Recommended Values: Viticulture: 2 m Sugar: 2 m Broadacre grains: < 200 ha 2m > 200 ha 10 m
Snap extent to factor of pixel Size	Default is checked	Snap the output raster extent to a factor of the pixel size. This will ensure adjacent rasters use a common origin which is important for future analysis.
Save as TIFF raster	Default derived from input layer	The output raster TIFF file to be created. The VESPER grid file will have a suffix of _v.txt .

2.2 Clean, Trim, Normalise Point Data

Summary

This tool processes on-the-go data files (e.g. from a yield monitor or EM38 soil survey) containing GPS coordinates recorded as latitude and longitude in decimal degrees, to output point values in a projected coordinate system. The tool also applies cleaning and filtering rules.

This tool:

- retains all columns (except coordinate columns) from the original file.
- converts coordinate columns to a projected coordinate system and renames them to Easting and Northing. An additional column (EN_EPSG) will be created and assigned the EPSG number for the projected coordinate system used to reproject data. An EPSG number is an international numbering system for coordinate systems (see http://www.epsg.org).
- optionally saves a Shapefile version matching the output CSV file. A second Shapefile will also be saved containing the string '*_removed*' in its filename. This Shapefile will contain all the points the filter discards and will be attributed by filter type. The description of the values can be found in Table 2.

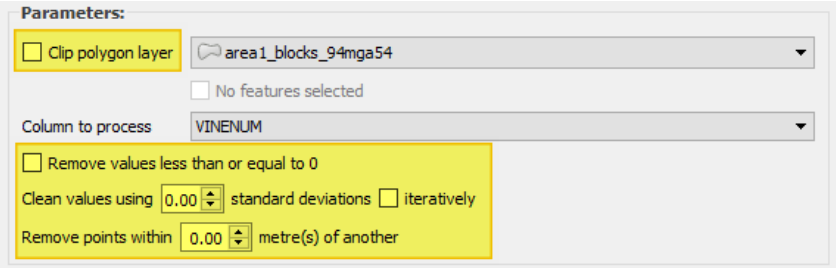
This tool filters data by:

- optionally clipping the data to a Shapefile boundary (i.e. removes points that lie outside the boundary).
- optionally removing values from a given column which are null (missing) or are less than or equal to zero.
- creating a normalised column using a prefix of nrm_ and calculating normalised values for the data such that their mean is zero and standard deviation is 1. The normalised value of column Z is calculated as (Z - mean(col Z))/(s.d.(col Z)) .
- optionally trimming normalised outliers based on a set number of standard deviations. This trim can optionally be performed iteratively with the normalised value re-calculated for each iteration. It is recommended that, by default, normalised data are trimmed to +/- 3 standard deviations.
- optionally thinning the data by removing points that are closer together than a set distance. For data that have been georeferenced using a differential GPS, we recommend removing points that are closer than 1 m apart.

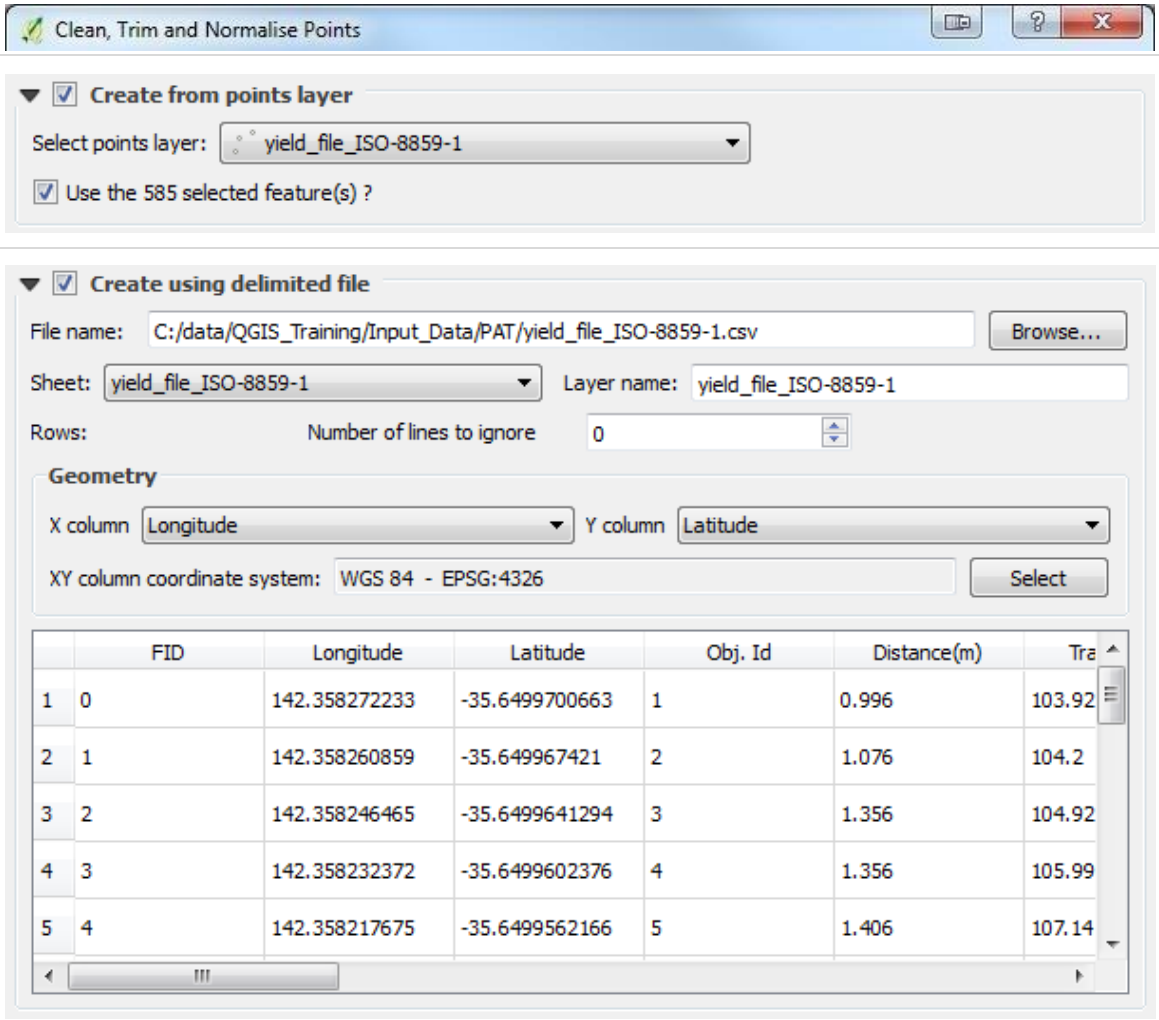
As part of the filtering process, the tool may rename some data columns to adhere to Shapefile column name limitations. The new names are displayed as a PAT log message and written to the log file. For more information on the location of the saved log file and temporary folder refer to the Technical Notes section on page 56.

Once processing is complete, the results of filtering are shown as a PAT log message. Only filters which remove points are listed. When iteratively filtering, results of all iterations are shown. An explanation of these is shown in Table 2.

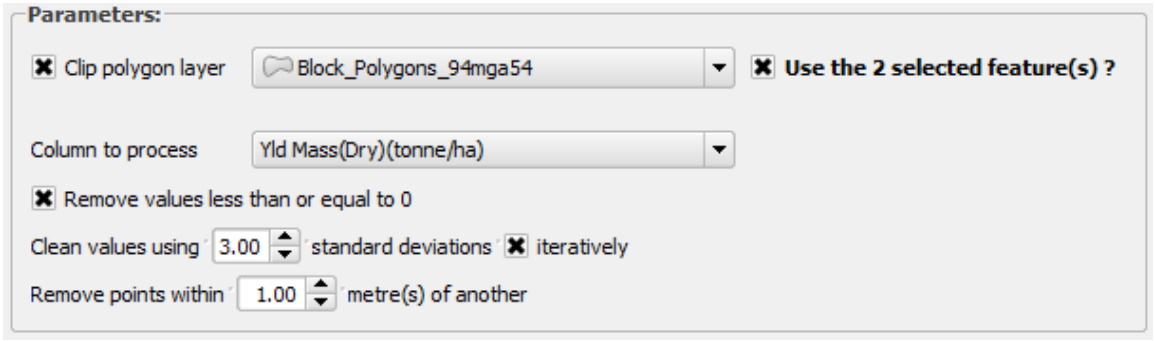
The ***Clean, Trim, Normalise Point Data Tool*** can also be used to reproject csv files containing coordinate columns without the cleaning, trim and normalise functionality by changing the parameters to those highlighted below.



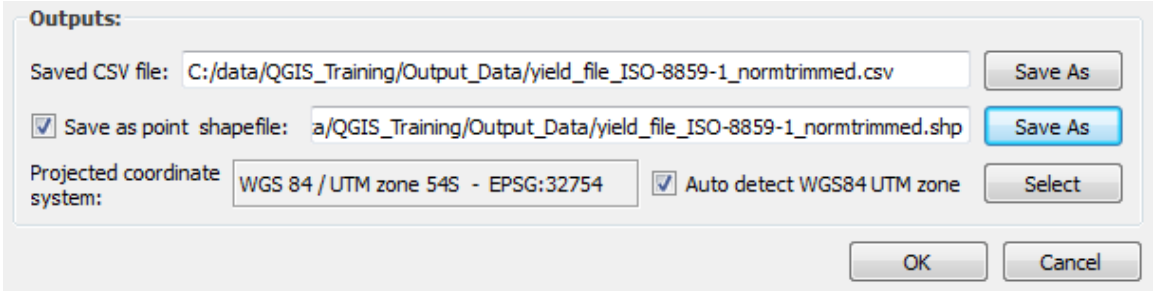
Dialog



Choose whether to use GPS data currently loaded in QGIS as a Points Layer		
Select points layer:		A point layer currently loaded in QGIS.
Use selected features	Default is unchecked	If checked, use only the selected features to generate a polygon. If unchecked, all the features will be used.
Or alternatively using a delimited text file		
Filename	Options include XLS, TXT, CSV	The filename of the table file.
Sheet	The sheet containing data. If the file type doesn't support sheets, then the filename will be used.	
Layer name	Default is the sheet name.	The name used when displaying QGIS layers, and as the default value for the output point and/or polygon files.
Number of lines to ignore	Default is 0	The number of lines at top of file to ignore. Can be used for skipping header text.
X column **Y column**	The columns representing the X and Y coordinates within the input file. By **default** will attempt to match to commonly used column names like X,Y, longitude, latitude; lon, lat etc.	
XY columns coordinate system	The coordinate system of the XY Columns. Note: GPS normally records data in WGS 1984 (EPSG: 4326)	
Clip polygon layer	Default is unchecked	A layer containing polygon features used to clip the points data. Any points that fall outside the polygon features will be removed.



Use selected features	Default is unchecked.	If checked, use the selected features to clip the input points. If unchecked, all the features will be used.
Column to process	The column to normalise, trim and clean.	
Remove values less than or equal to 0	Default is checked.	If checked, rows from the chosen column will be removed where the value is null or less than or equal to zero.
Clean values using X standard deviations	0.00 to 5.00 Default is 3. Use 0 to omit.	Points where the normalised value is outside a set number of standard deviations will be removed.
Iteratively	Default is checked	Clean the points iteratively using the set number of standard deviations until all values are within +/- the set standard deviation.
Remove points within X metres of another	0.00 to 20.00 Default is 1. Use 0 to omit.	Any points spaced at less than this distance apart will be removed.



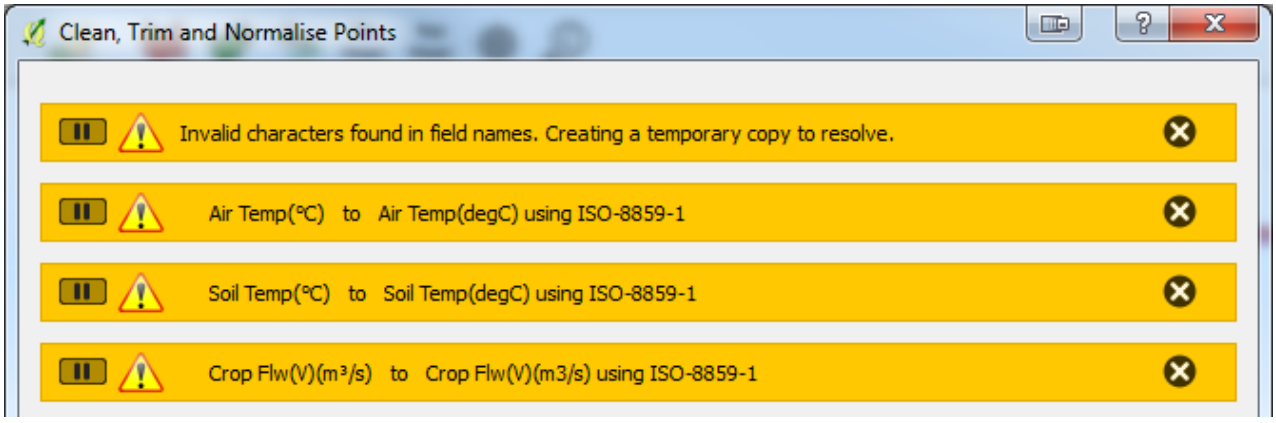
Save CSV file	Derived from layer name.	The name and path of the output CSV file containing the final data points.
Save as point Shapefile	Default is unchecked. Derived from layer name.	If checked, save a Shapefile version of the output CSV file. A second point Shapefile will be saved with a suffix of ***_removed*** containing the points discarded and attributed by a filter column.
Projected coordinate system	The output projected coordinate system of the Shapefile.	
Auto detect WGS84 UTM zone	If Auto detect is checked the following rules will apply: - attempt to match the input if it is a projected coordinate system. - If the inputs are in Geographic's (lat/long) it will calculate the relevant WGS84 UTM Zone. Note: If in Australia then MGA GDA 1994 zones, will be used – see Technical Notes on page 56.	

Notes

Table 2 A description of the filter type used when reporting filtering results.

ORDER	FILTER TYPE	DESCRIPTION
01	clip	Points removed when clipping by a polygon.
02	zero	Nulls and/or Zeros removed.
03	3.0 std iter 1	Points removed during the standard deviation (std) iteration (iter) number 1 and uses 3 standard deviations.
04	3.0 std iter 2	Points removed during the standard deviation (std) iteration (iter) number 2 and uses 3 standard deviations.
05	3.0 std iter 3	Points removed during the standard deviation (std) iteration (iter) number 3 and uses 3 standard deviations.
06	pointXY (1.0 m)	Points removed which are spaced at less than the specified distance apart, in this case **1m.**
07	pointX (1.0 m)	Points removed after sorting by the X coordinate, which are spaced at less than the specified distance apart, in this case **1 m.**
08	pointY (1.0 m)	Points removed after sorting by the Y coordinate, which are spaced at less than the specified distance apart, in this case **1 m.**
	Pts remaining	The number of points left after filtering occurs.
	Total	The total number of points in the dataset.

- The following notification occurs when a delimited text file contains invalid characters in a column name. A copy of the file with corrected column names will be made in the PAT temporary folder (location shown in PAT log message). For more information on the location of the saved log file and temporary folder see the Technical Notes section on page 56.



2.3 Run Kriging Using VESPER

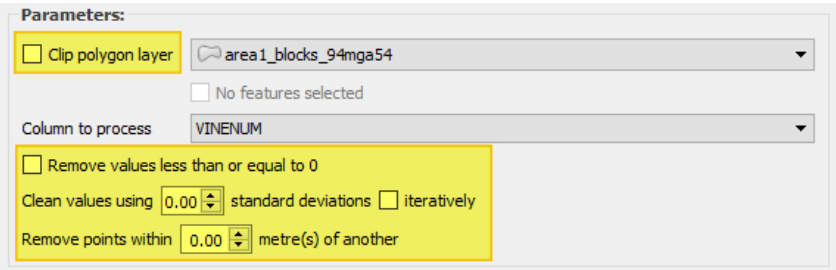
Summary

This tool is used for mapping on-the-go data at high spatial density, such as are collected by a yield monitor or an EM38 soil survey, following established protocols (Bramley and Williams 2001; Taylor et al. 2007; Bramley et al. 2008; Bramley and Jensen 2014) using local block kriging. It will create a VESPER control file and collate the files required for kriging. The following files will be created in a VESPER sub-folder located in the specified output folder.

- the VESPER control file. The control filename will be used as a base to derive other VESPER output files like the kriged map result.
- a subset of data to krige. All non-required columns are deleted.
- a Windows batch file (*Do_VESPER.bat)* which can be used to launch VESPER processing for all control files in the VESPER sub-folder. This process can be run outside of QGIS and the Python/pyPrecAg environment.

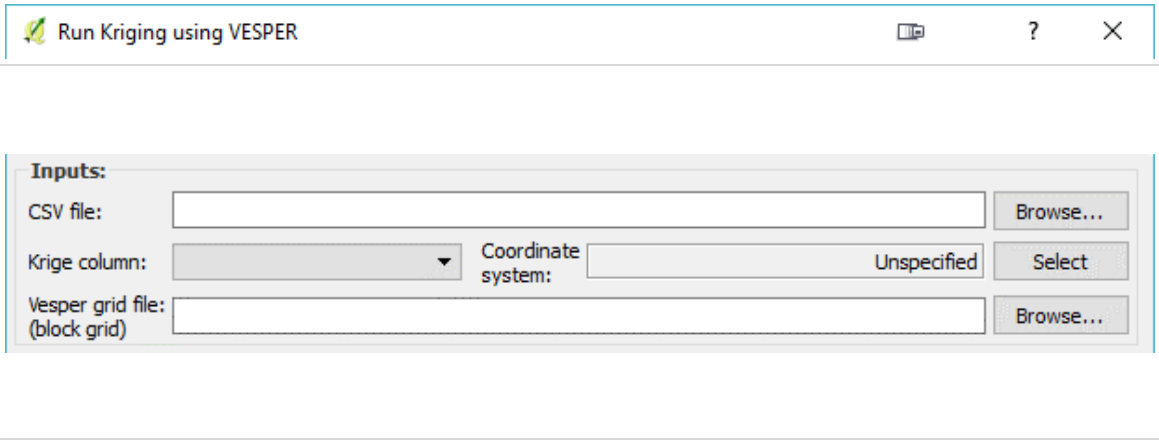
To run VESPER control files from within QGIS, VESPER must already be installed on the PC and configured in the PAT settings **(2.17 Settings** on page 55).

Data validation checks will be completed before running the tool. This aims at ensuring that the input csv file which contains eastings and northings, overlaps with provided VESPER grid text file. If you receive a warning, please check that these two files use the same coordinate system and indeed overlap. If your csv coordinates are in a different coordinate system to the VESPER grid file, the ***Clean, Trim, Normalise Point Data Tool*** can be used to project coordinates without using the cleaning, trim and normalise functionality, by setting the options as shown highlighted below.



Kriging of data collected at much lower spatial density, such as through hand sampling (e.g. grape bunch number, mid-season cuts of wheat biomass) is also possible in PAT. However, it should be noted that we DO NOT recommend interpolation of maps from 'low density' data when fewer than 100 data points are available (Webster and Oliver, 2007). Experience also suggests that, in vineyards and sugarcane fields, a sample intensity of around 26 points per ha is required, given the typical dimensions of such fields. A lower sample density may suffice for broadacre situations but the 100 point minimum always applies. Note also that this analysis should only be undertaken by users with an understanding of map interpolation using kriging and the generation and role of the variogram in this process (Webster and Oliver, 2007). Detailed instructions on this process can be found in section ***4.4 Kriging 'low density' data in PAT***.

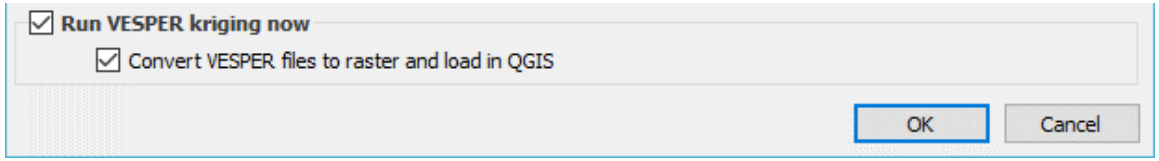
Dialog



Cleaned and trimmed CSV file		A CSV file containing the data to krige. Use the ***Clean, Trim, Normalise Point*** tool to generate this input or to convert the coordinate system.
Coordinate system for CSV file:		This is the coordinate system of the coordinate columns in the CSV file. By default, the value of the EN_EPSG or ENEPSG column in the input CSV file will be used.
Krige column		The column containing values to krige.
VESPER grid (block grid) file		The VESPER grid file representing the area to be kriged. Use the ***Create Block Grid Tool*** to generate this input.
HIGH DENSITY KRIGING		
Block kriging size	0 to 1000 Default is 10	The value in metre to set as the block size when using block kriging in VESPER. This value should be approximately 5 times the pixel value. Recommended Values: Viticulture: 10 m Sugar: 10 m Broadacre grains: < 200 ha 30 m > 200 ha 60 m
LOW DENSITY KRIGING – FOR ADVANCED USERS ONLY		
Variogram Text File:		The saved VESPER variogram text file. See ***4.4 Kriging 'low density' data in PAT*** for creation instructions.
Minimum number of points to use	Default is the number of data rows minus 2	Minimum number of neighbourhood points used for kriging interpolation.
Show VESPER graph and map graphics while kriging	Default is unchecked	Checking this option will display the map and graph while VESPER is running but will increase the processing time.



Folder	The output folder for the VESPER files. A sub-folder called VESPER will be created if required.
Control file	The name to use as the VESPER control file. This will be appended to the output folder.
Auto create control file name	If checked, the name will be generated from the first 20 characters of the input CSV file, the string '*control*' and the Krige column. Uncheck this option to prevent manual edits from being overwritten when changing source CSV file or Krige column.



Run VESPER kriging now	Default is checked	Add the control file to the VESPER Queue for processing.
Convert VESPER Files to Raster and Load in QGIS	Default is checked	On completion of VESPER processing, convert the outputs to raster TIFF formats.

File Naming Conventions

- <> denotes an existing element or input
- non-alphanumeric characters are removed with the exception of hyphens (-) and underscores (_).

DESCRIPTION	FILENAME	EXTENSION	EXAMPLE
*Files **required** by VESPER and copied to the VESPER output folder.*			
VESPER control file	< *20 characters of CSV file*>_<*low or high density*>_<*krige column*>_control	.txt	swblock_HighDensity_YldMassDry_control.txt
VESPER grid file	<*control file*> where control is replaced by vespergrid.	.txt	swblock_HighDensity_YldMassDry_vespergrid.txt
VESPER data file	<*control file*> where control is replaced by vesperdata.	.csv	swblock_HighDensity_YldMassDry_vesperdata.csv
*Files **created** by VESPER and saved to the VESPER output folder.*			
VESPER Krige file (contains Prediction and SE)	<*control file*> where control is replaced by kriged.	.txt	swblock_HighDensity_YldMassDry_kriged.txt
VESPER parameter file	<*control file*> where control is replaced by parameter.	.txt	swblock_HighDensity_YldMassDry_parameter.txt
VESPER report file	<*control file* > where control is replaced by report..	.txt	swblock_HighDensity_YldMassDry_report.txt
*If **converting VESPER files to raster** is checked or the **Create Rasters from VESPER Results** tool is used.*			
Kriged Prediction TIFF source: VESPER Krige result above	<*control file*> where kriged is replaced by PRED.	.tif	swblock_HighDensity_YldMassDry_PRED.tif
Standard Error TIFF source: VESPER Krige result above	<*control file*> where kriged is replaced by SE.	.tif	swblock_HighDensity_YldMassDry_SE.tif
Confidence Interval (CI) metadata text file	<*control file*> where control is replaced by CI.	.txt	swblock_HighDensity_YldMassDry_CI.txt



2.4 Import VESPER Results

Summary

This tool converts the output files generated by VESPER into raster TIFFs.

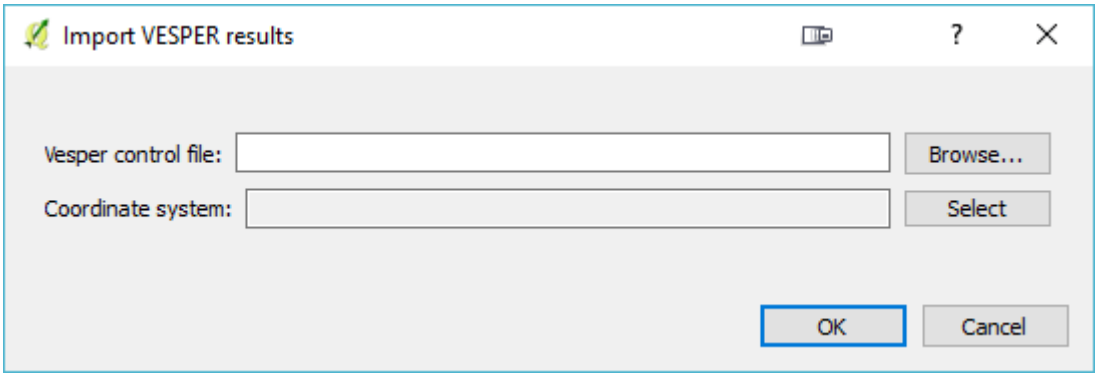
The output filenames will be created using the VESPER control file name as a base and demonstrated in the ***Run Kriging Using VESPER Tool - File Naming Conventions*** section on page 22.

The files created include:

- a TIFF file representing the predicted (kriged) value; these are the values used to generate the resulting map.
- a TIFF file representing the standard error (SE) of the kriging prediction.
- a text file containing the calculated median kriging prediction SE and the 95% confidence interval for the map. This is important information for testing the significance of the difference between different zones when different map layers are clustered to generate potential management zones.

The coordinate system to be assigned to the rasters will be extracted from the value stored within the control file. If the coordinate system cannot be found or is incorrect, it can be selected manually.

Dialog



Import VESPER Results		
VESPER Control File:		A VESPER control file used to identify VESPER results files and convert to rasters.
Coordinate System for CSV File:	By default it is extracted from the input control file.	This is the coordinate system of the VESPER results.



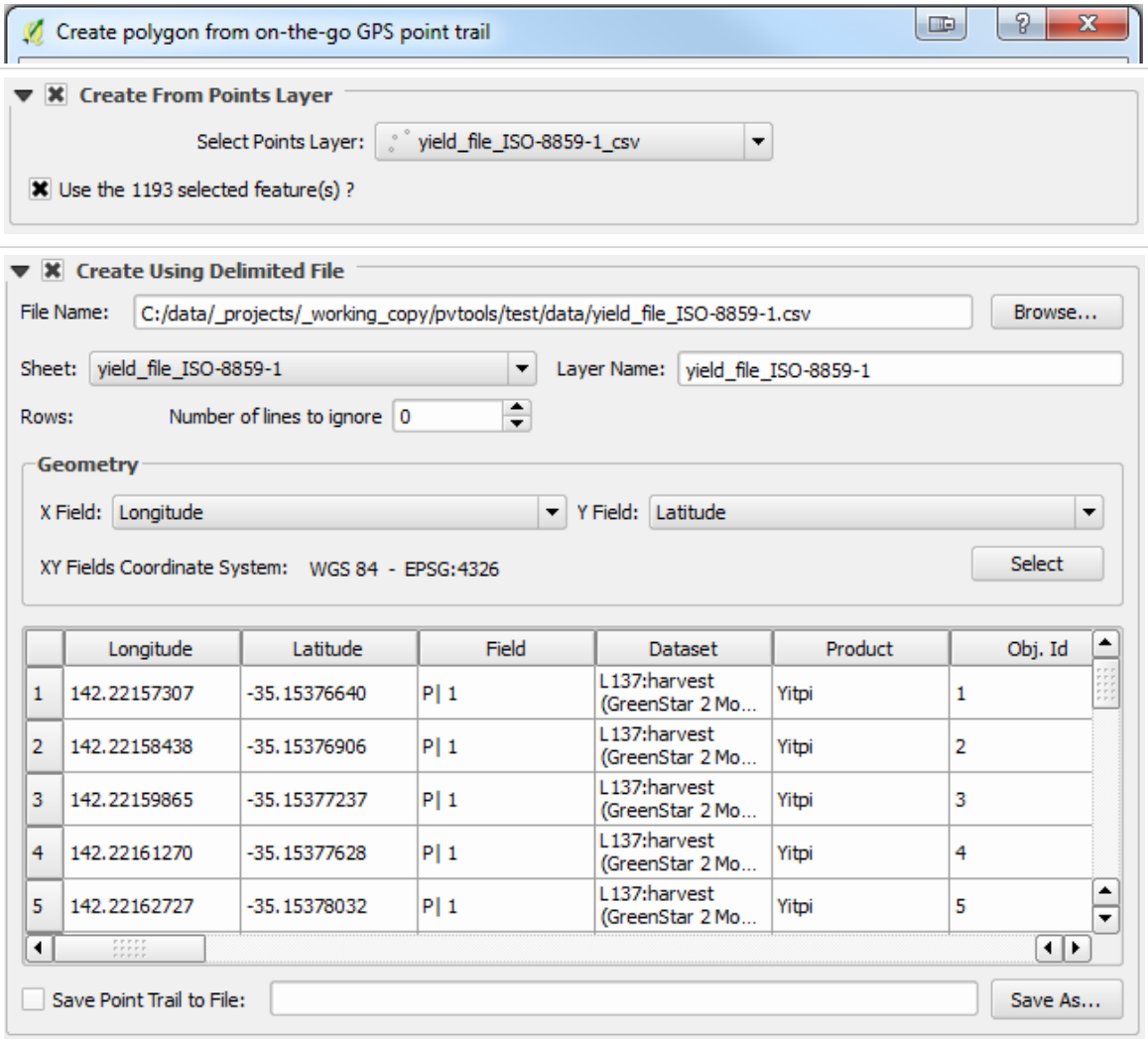
2.5 Create Polygon from On-The-Go GPS Point Trail Data

Summary

Having a block boundary polygon is central to many of the PAT processing steps. Boundary polygons are used to constrain data to a fixed extent. It is preferably that a boundary polygon is created by collecting accurate GPS points around the block and editing them in QGIS to create polygons (refer to ***How-To - Create a block boundary polygon from a CSV of GPS collected points*** on page 57 for instructions on this method). However, if accurate GPS data for the boundary is not available, then a less accurate block boundary polygon can be created using this tool, based upon a file of on-the-go GPS points (i.e. from a yield monitor or EM38 survey data). Note that this method is not preferred, but the tool is provided as a 'quick fix' for those wishing to generate maps in the absence of surveyed block or field boundaries.

As the process involves a dot-to-dot approach, it is critical that the input file of points are in order (i.e. sorted by an increasing time sequence - as would be normal for data collected by an on-the-go sensor such as yield monitor). For efficiency, points can be thinned by removing points closer than a set distance apart as justified by the accuracy of the GPS. Resulting points will be connected to form lines and then converted to polygons.

Dialog

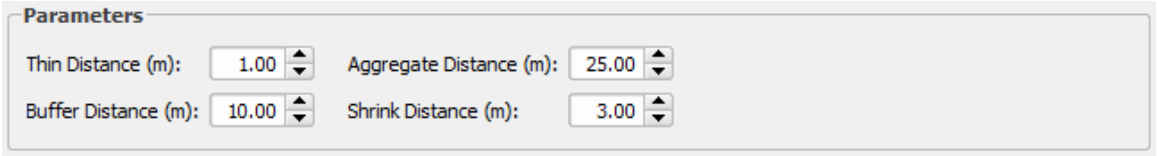


Choose whether to use GPS data currently loaded as a QGIS Points Layer

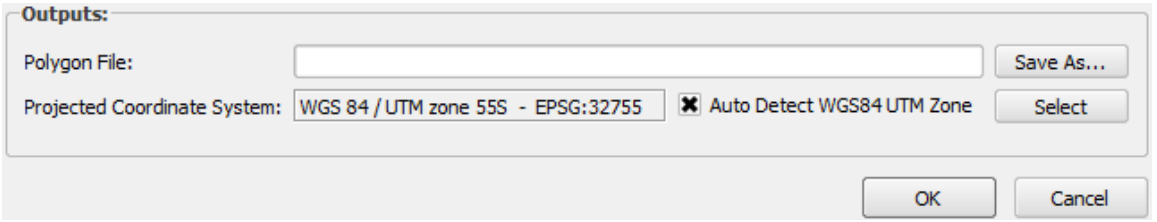
Use selected features	Default is unchecked	If checked, only uses the selected features to generate a polygon. If unchecked, all the features will be used.

Or alternatively using a delimited text file

Filename	Options include XLS, TXT, CSV	The filename of the table file.
Sheet	The sheet containing data. If the file type doesn't support sheets, then the filename will be used.	
Layer name	Default is the sheet name.	The name used when displaying QGIS layers, and as the default value for the output point and/or polygon files.
Number of lines to ignore	Default is 0	The number of lines at top of file to ignore. Can be used for skipping header text.
X column **Y column**	The columns representing the X and Y coordinates within the input file. By **default** will attempt to match to commonly used column names like X,Y, longitude, latitude; lon, lat etc.	
XY columns coordinate system	The coordinate system of the XY Columns. Note: GPS normally records data in WGS 1984 (EPSG: 4326).	
Save point trail to file	Default is unchecked	If checked, save the input delimited file as a points Shapefile in the output coordinate system.
Thin distance (m)	0.00 to 100.00 Default is 1 Use 0 to omit	Any points spaced at less than the minimum distance apart will be removed.



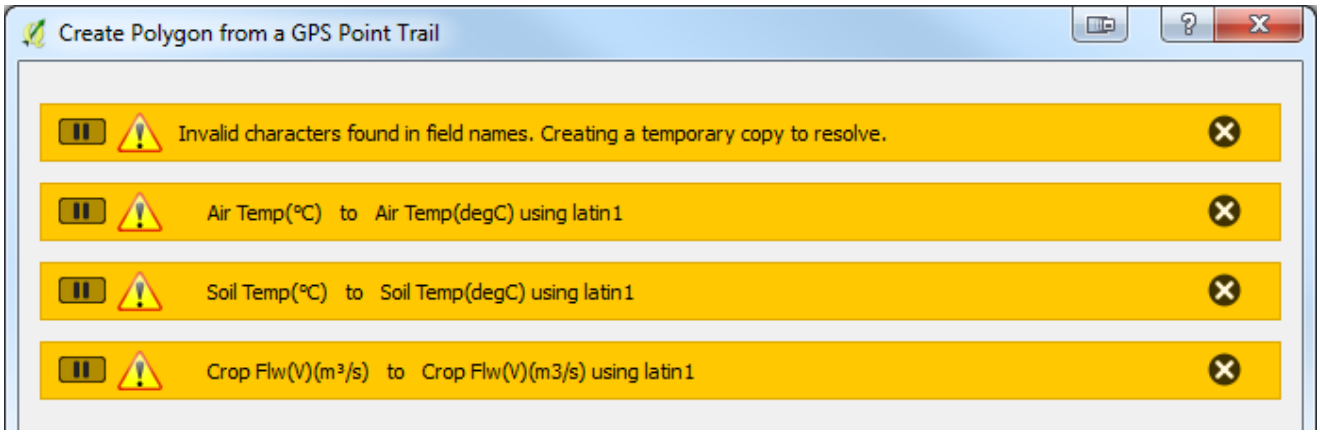
Aggregate distance (m)	0.00 to 100.00 Default is 25	The maximum distance between points used to separate strings of connected points into individual lines. Typically, this is slightly larger than the row/swath width.
Buffer distance (m)	0.00 to 100.00 Default is 10	A distance in metre used to buffer the lines to create overlapping polygons. Typically, this is half the row/swath width.
Shrink distance (m)	0.00 to 100.00 Default is 3	A distance in metre used to shrink the overlapping polygons after dissolving. Typically, this is slightly less than the buffer distance.



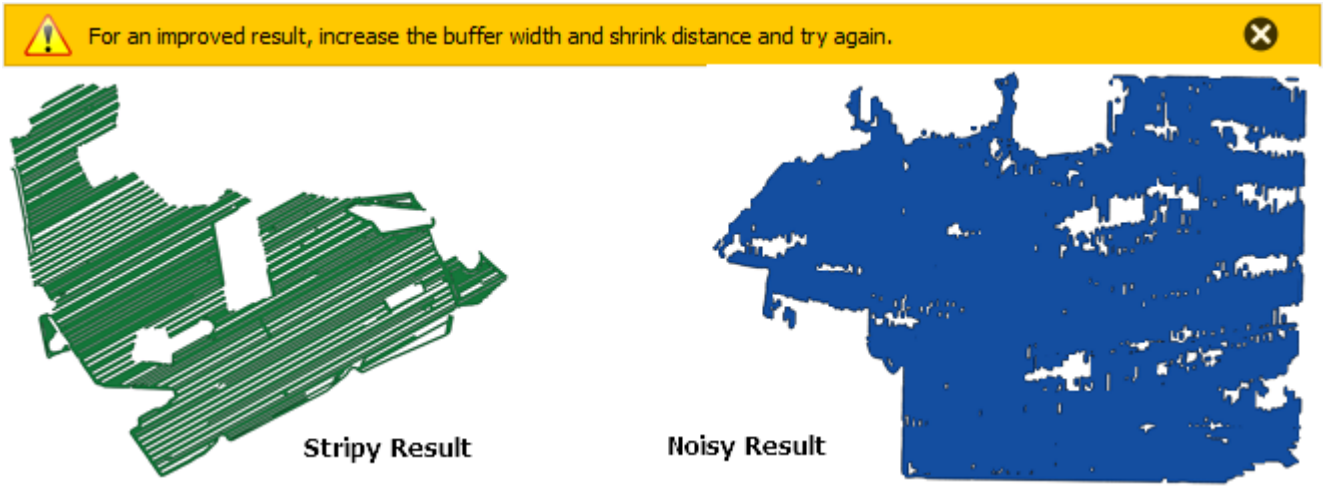
Polygon file	Derived from layer name The output filename for the polygon Shapefile.
Projected coordinate system	The output projected coordinate system of the Shapefile.
Auto detect WGS84 UTM zone	If Auto detect is checked the following rules will apply: - attempt to match the input if it is a projected coordinate system. - If the inputs are in Geographic's (lat/long) it will calculate the relevant WGS84 UTM Zone. Note: If in Australia then MGA GDA 1994 zones, will be used – see Technical Notes on page 56.

Notes

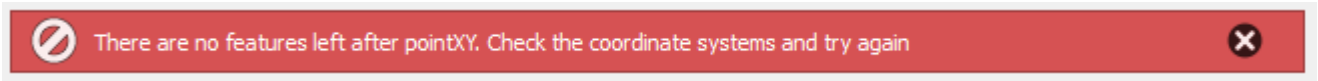
- The following notification occurs when a delimited text file contains invalid characters in the column name. A copy of the file will be made in the PAT temporary folder containing corrected column names as specified in the PAT log message and to file. For more information on the location of the saved log file and temporary folder see the Technical Notes section.



- A warning is triggered when the resulting polygon is considered stripy or noisy. This can usually be corrected by increasing the buffer and shrink distances.



- The following error is caused when the wrong coordinate system is applied to the input source file or layer. Thinning and/or clipping will result all points being removed leaving no points for further processing.



2.6 Rescale or Normalise a Raster

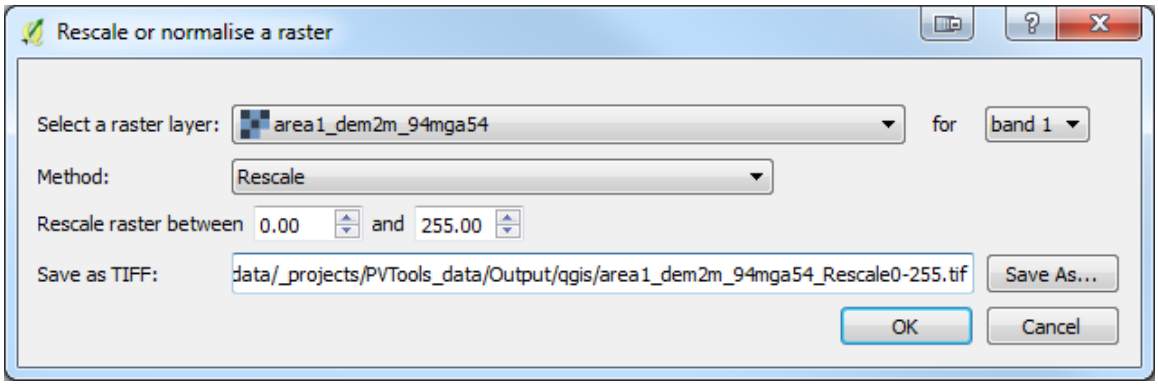
Summary

This tool rescales or normalises a raster and outputs to a new TIFF file. Existing no-data values will be ignored in any calculation.

- Rescale will adjust the raster between the specified values (e.g. output values to a range between 0 and 1)
- Normalise will adjust the raster to a mean of zero and standard deviation of one

Such re-scaling is useful when you wish to compare several map layers, for example, yield maps for several years.

Dialog



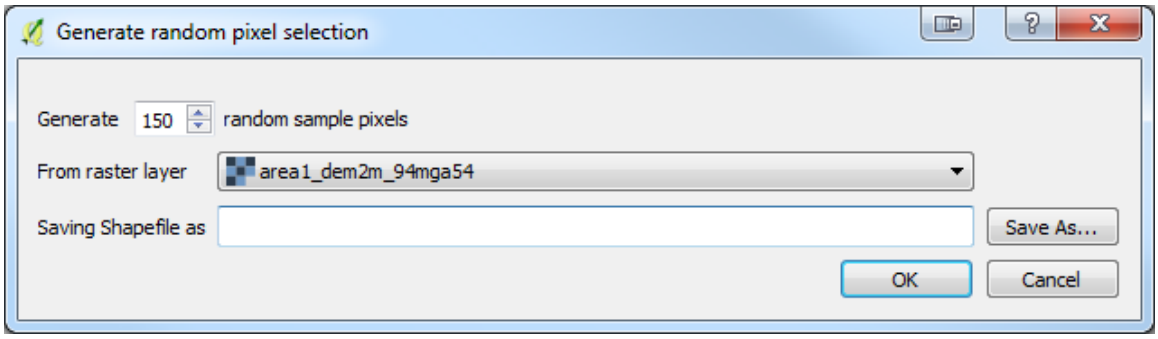
Rescale or normalise a raster		
Select a raster layer for band X	Default is band 1	The raster layer and band to rescale or normalise.
Method	Default is rescale	Options: • **Rescale** – adjust values to a fixed range; or • **Normalise** – adjust values to a mean of zero and a standard deviation of one.
Rescale between (when selected)	Default 0 to 255	The range of values used with rescaling.
Save as Shapefile	Default will be derived from the input layer name	The output Shapefile to be created.

2.7 Generate Random Pixel Selection

Summary

This tool is used to select randomly distributed pixel locations from an existing raster and save them to a point Shapefile. Pixels will be selected from areas inside the raster extent which contain valid data. The resulting points will be located in the centre of the chosen pixel and the output shapefile will contain columns representing the X and Y coordinates. This tool provides the key location input for the ***Extract Pixel Statistics for Points Tool*** described on the following page.

Dialog



Generate random pixel selection		
Generate random sample pixels	Default is band 1	The number of pixels to randomly select.
Select a raster layer		The raster to use as the base for selecting pixels. No-data pixels will be exempt from selection.
Save as Shapefile	Default will be derived from the input layer name	The output Shapefile to be created.

2.8 Extract Pixel Statistics for Points

Summary

Extract pixel statistics for points is used to extract pixel statistics from multiple rasters using a pre-defined set of points. These points could be identified using the ***Generate Random Pixel Selection Tool*** described on the previous page. Statistics are calculated on pixel values within a square neighbourhood and extracted to a CSV file. This tool is useful when relationships are being sought between different maps layers – for example, a regression or correlation analysis between yield and soil map data.

Applying a neighbourhood filter to rasters is useful for removing (smoothing) small anomalies introduced from instrument inaccuracies or on-the-go movement. The neighbourhood consists of a centre pixel and a number of pixels forming a square around the central pixel, as shown in Figure 3.

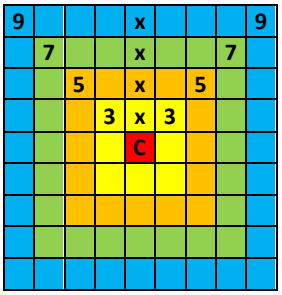


Figure 3. A representation of the 3x3, 5x5, 7x7, 9x9 neighbourhood size around a central pixel (C) as is used when calculating filtered statistics.

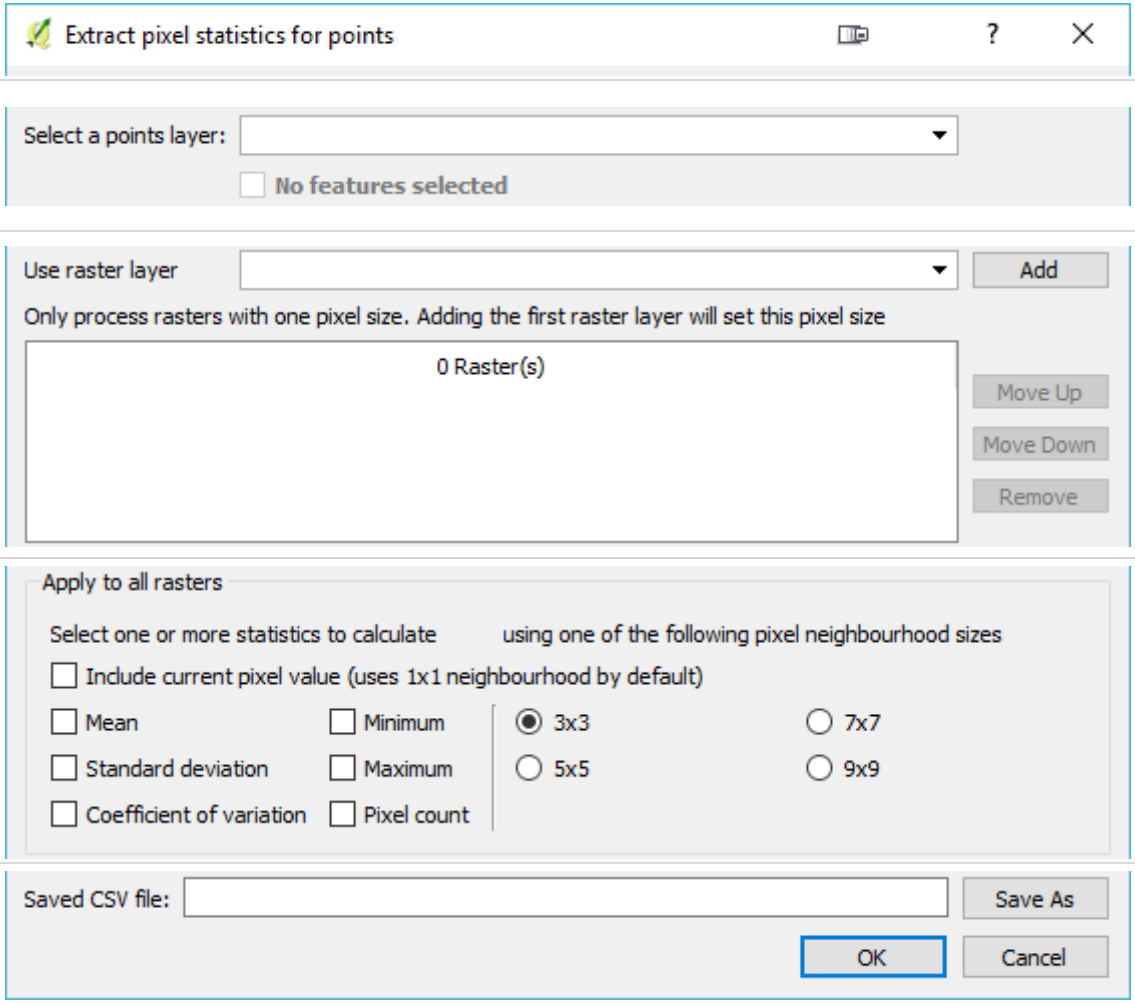
For example, a mean statistic for a 3x3 neighbourhood on 2 m pixels, will calculate the mean of the central pixel (red) and the surrounding 3x3 area of 8 pixels (yellow) equating to 36 m2 on ground.

Pixels designated no-data will be excluded from the statistical calculation. However, a central no-data pixel may be assigned a value if at least one pixel in the neighbourhood has a valid value.

Currently, the statistical methods supported by this tool are: mean, standard deviation, co-efficient of variation (CV), minimum, maximum and a count of pixels contributing to the statistical calculation.

The output values are saved to a CSV data file. Column names for each raster and statistic combination are explained below.

Dialog



Select point layer		A point layer in QGIS used to extract pixel statistics values.
Use selected features	Default is unchecked	If **checked**, extract pixel values only for the selected points. If **unchecked**, all the points will be used.
Use raster layer		The raster layer to extract values from. Only raster layers of the same pixel size can be analysed at a time and this is set by the first raster layer added. The *Add, Delete, Move Up, Move Down,* buttons can be used to manage the list of rasters. The order of rasters is retained in the output CSV file with separate columns for each raster and statistic combination.
Calculate pixel neighbourhood statistics		Select the required statistical methods for each raster in the list. **Current pixel** just extracts the pixel value, and/or **Statistics** to be calculated using the set neighbourhood size. The output column naming format is explained below.
Neighbourhood size (pixels)	Default is 3x3	The size of the neighbourhood footprint to use while calculating statistics.
Save CSV file	Derived from layer name	The name and path of the output CSV file containing the final data points.

File and Column Naming Conventions

- <> denotes an existing element or input
- non-alphanumeric characters are removed with the exception of hyphens (-) and underscores (_).

Output Filenames:

CSV filename	<*QGIS points layer name*>_pixelvals.csv

Column Names:

NAMING RULE	EXAMPLE	EXPLANATION
<***statistic***><***size***>_<***raster file***>	mean3x3_swblock_YldMassDry_PRED	The column containing values from the mean 3x3 neighbourhood filter for the raster swblock_YldMassDry_PRED.tif.
	pixel_swblock_2009YldMassDry_PRED	The column containing the pixel values for the raster swblock_2009YldMassDry_PRED.
	cv5x5_seblock_2014YldMassDry_PRED	The column containing values for coefficient of variation (CV) 5x5 neighbourhood filter for the raster seblock_2014YldMassDry_PRED.tif.
	pixelcount7x7_swblock_2024YldMassDry_PRED	Column containing values representing the count of pixels used for statistical calculations with a 7x7 neighbourhood filter for the raster seblock_2014YldMassDry_PRED.tif.



2.9 Calculate Image Indices for Blocks

Summary

Many providers of remotely sensed imagery deliver data to clients as multi band files. The Calculate Image Indices for Blocks tool is used to calculate indices for a multi band image and aligns the output to the extent and pixel size of a block grid. Indices currently supported are Normalised Difference Vegetation Index (NDVI); the 'simple ratio' or Plant Cell Density index (PCD); Green Normalised Difference Vegetation Index (GNDVI); Chlorophyll Red-Edge index (CHLRE); and Normalised Difference Red-Edge index (NDRE). Table 3 describes how these indices are calculated.

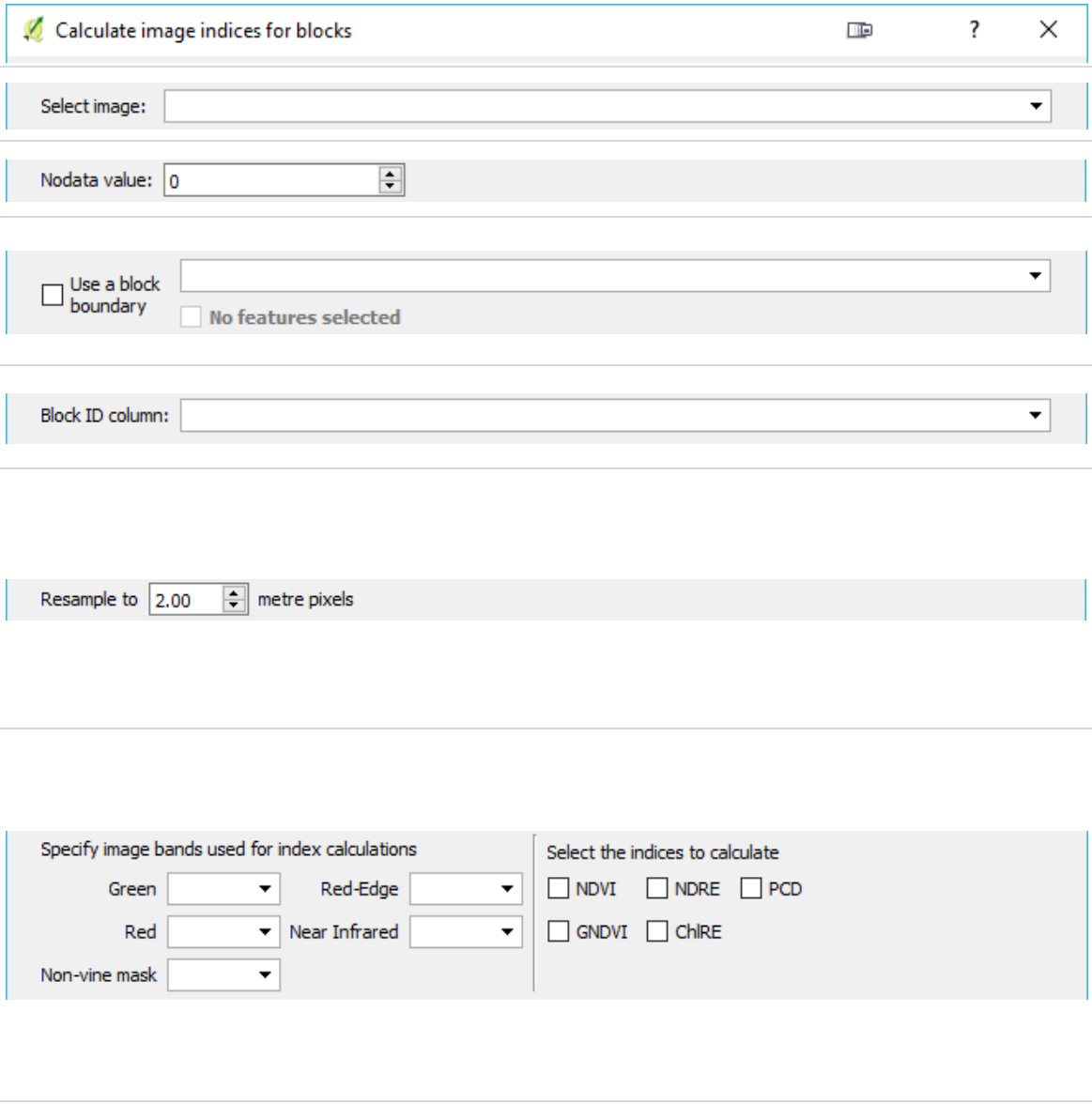
Each band in the image file is mapped to a spectral band e.g. Red, Red-edge, Near Infrared and is then used to calculate relevant image indices. In the case of vineyard data, if a non-vine mask is present in an existing band of the image, it can be used to remove non-vine signals prior to the resampling and alignment to the block grid. This will ensure that spectral signatures relating to ground cover are excluded from the resulting image outputs. This non-vine masking is not relevant for broadacre crops.

An optional block boundary polygon layer, and a column containing the block name or ID, can be used to separate the resulting images into individual blocks. By default, if no column is specified, then all polygons are assumed to be from the one block and will be processed accordingly. If no block boundary layer is specified, then a single polygon outlining the image (excluding no-data) will be used. The output from this tool will be aligned to a block grid that will be identical to the one created separately using the ***Create Block Grid Tool*** on page 14.

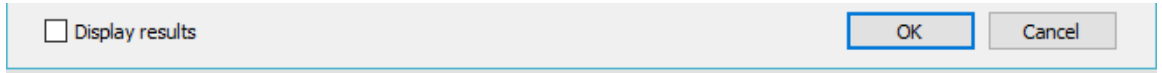
Table 3 The list of supported indices

Normalized Difference Vegetation Index (**NDVI**)	$\mathbf{NDVI} = \frac{(\boldsymbol{NIR} - \boldsymbol{Red})}{(\boldsymbol{NIR} + \boldsymbol{Red})}$
Green Normalized Difference Vegetation Index (**GNDVI**)	$\mathbf{GNDVI} = \frac{(NIR - Green)}{(NIR + Green)}$
Normalised Difference Red-Edge Index (**NDRE**)	$\mathbf{NDRE} = \frac{(NIR - Red\ Edge)}{(NIR + Red\ Edge)}$
Plant Cell Density Index (**PCD**)	$\mathbf{PCD} = \frac{NIR}{Red}$
Chlorophyll Red Edge Index (**ChlRE**) (Gitelson 2004; Gitelson *et al.* 2005)	$\mathbf{Chl}_{red\ edge} = \left(\frac{NIR}{Red\ Edge}\right) - 1$

Dialog



Select Image		The image layer containing the appropriate bands required for calculate indices.
No-data value	Default from image.	Can be used to specify a different no-data value.
Use a block boundary	Default is unchecked	A layer in QGIS containing polygon(s) representing blocks.
Use selected features	Default is unchecked	If **checked**, only the selected polygons will be used.
Block ID column		A column containing the block id or name. This will be used treat multiple polygons with the same block id or name as one.
Resample pixel size (m)	0.00m to 6 km Default is 2 m	The pixel size to apply to the raster outputs. This is expressed in metre. Ideally this should match your **Block Grid TIFF** file. Recommended Values: Viticulture: 2 m Sugarcane: 2 m Broadacre grains: < 200 ha 5 m > 200 ha 10 m
Specify image bands used for index calculations		Mapping of band numbers to band types. The mapped bands will enable/disable indices based on individual index requirements. For most images, Green is band 2, Red is band 3, Near Infrared is band 4. Non-vine mask is a band where pixels not containing vine signals are set to no-data. Not required for broadacre data.
Select the indices to calculate		The indices to calculate. If an index is disabled (greyed out) the band it requires hasn't been mapped. For index acronym and equation, see table above.

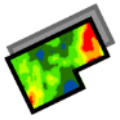
	**Projected coordinate system**		The projected coordinate system to apply to the output files. By **default**, it will calculate the relevant coordinate system from the input image or block boundary's coordinate system. Note: If in Australia then MGA GDA 1994 zones, will be used – see Figure 8 on page 59.
	**Output folder**		The folder to save output TIFF files. A folder based on the input image name will be created and all TIFF files will be saved here. See table below for *File Naming Conventions* used for output files.
	**Display results**	Default is unchecked	If checked all resulting TIFF files will be loaded into QGIS.

File Naming Conventions

- <> denotes an existing element or input
- non-alphanumeric characters are removed from strings with the exception of hyphens (-) and underscores (_).

Filenames:

NAMING RULE	EXAMPLE	EXPLANATION
Output Folder:		
<***output_folder***>\<***image_name***>	C:\data\vineyard**rgbi_jan_50cm_84sutm54_tif**	A new folder based on the image name is created in the output folder and all created images are saved here. In this example the image **rgbi_jan_50cm_84sutm54.tif** is used to create a new folder called **rgbi_jan_50cm_84sutm54_tif**.
Image Names:		
<***block_id***>_<***index***>_<***pixel_size***>.*tif*	B1_NDVI_2m.tif	The TIFF file resampled to **2m** pixels for the Normalised Difference Vegetation Index (**NDVI**) created for block id/name of **B1**.
	PCD_250cm.tif	The TIFF file resampled **250cm** pixels for the Plant Cell Density (**PCD**) Index created **without** specifying a block id column.



2.10 Resample Image Band for Blocks

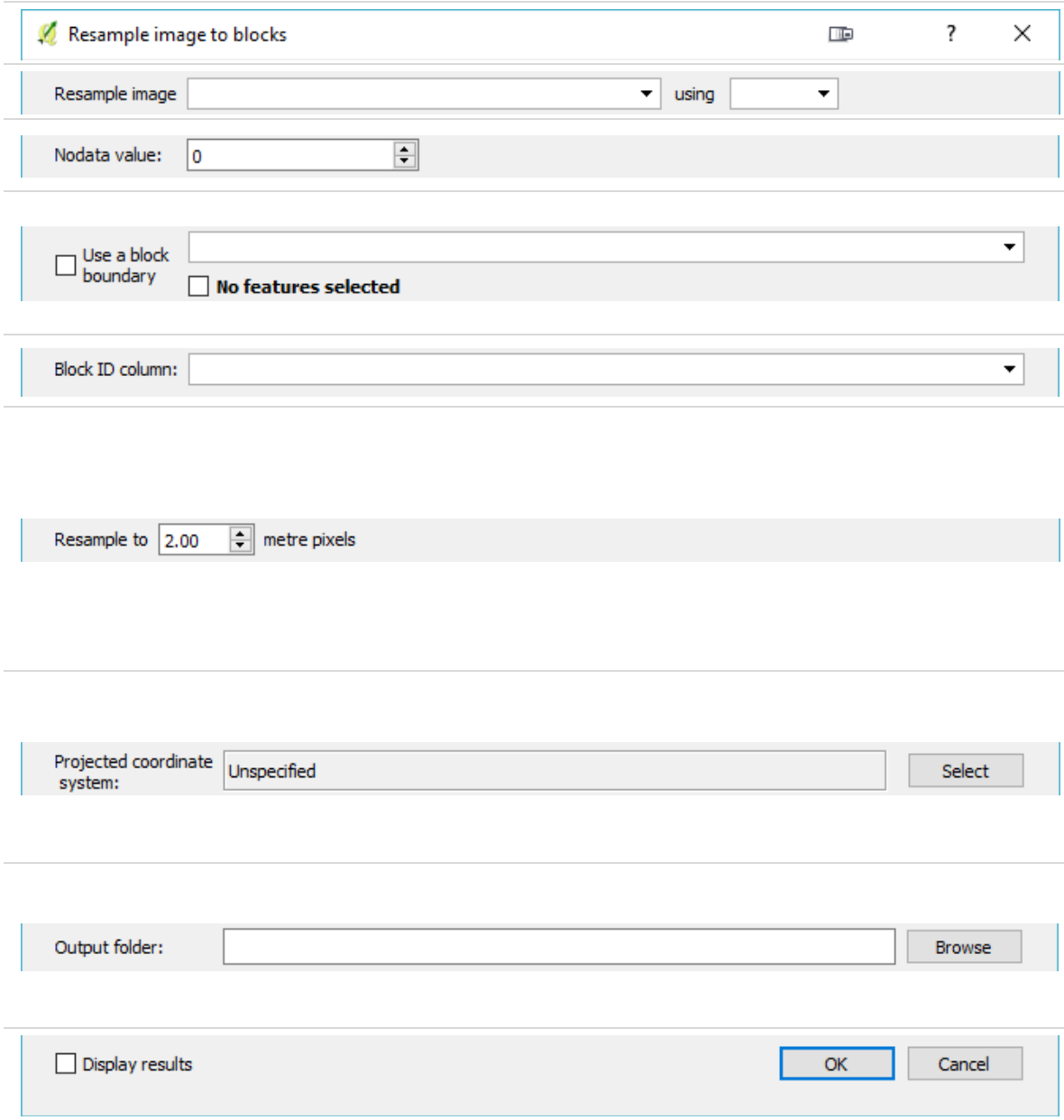
Summary

Resample Image Band for Blocks is used to resample, align and smooth an existing band of an image to match a block grid.

An optional block boundary polygon layer and a column containing the block name or ID, can be used to separate the resulting images into individual blocks. By default, if no column is specified, then all polygons are assumed to be from the one block and will be processed accordingly. If no block boundary layer is specified, then a single polygon outlining the input image will be used.

The output from this tool will be aligned to a block grid that will be identical to the one created separately using the ***Create Block Grid Tool*** on page 14.

Dialog



Select Image		The image layer containing the band to resample.
No-data value	Default from image	Can be used to specify a different no data value.
Use a block boundary	Default is unchecked	A layer in QGIS containing polygon(s) representing blocks.
Use selected features	Default is unchecked	If **checked**, only the selected polygons will be used.
Block ID column		A column containing the block id or name. This will be used treat multiple polygons with the same block id or name as one.
Resample pixel size (m)	0m to 6 km Default is 2 m	The pixel size to apply to the raster outputs. This is expressed in metre. Ideally this should match your **Block Grid TIFF** file. Recommended Values: Viticulture: 2 m Sugar: 2 m Broadacre grains: < 200 ha 2 m > 200 ha 10 m
Projected coordinate system		The projected coordinate system to apply to the output files. By default it will calculate the relevant coordinate system from the input image or block boundary's coordinate system and extent coordinate system. Note: If in Australia then MGA GDA 1994 zones, will be used – see Figure 8 on page 59.
Output folder		The folder to save output TIFF files. A folder based on the input image name will be created and all TIFF files will be saved here. See table below for *File Naming Conventions* used for output files.
Display results	Default is unchecked	If checked all resulting TIFF files will be loaded into QGIS.

File Naming Conventions

- <> denotes an existing element or input
- non-alphanumeric characters are removed from strings with the exception of hyphens (-) and underscores (_).

Filenames:

NAMING RULE	EXAMPLE	EXPLANATION
Output Folder:		
<***output_folder***>\<***image_name***>	C:\data\vineyard\area1_rgbi_jan_50cm_84sutm54_tif	A new folder based on the image name is created in the output folder and all created images are saved here. In this example the image **area1_rgbi_jan_50cm_84sutm54.tif** is used to create a new folder called **area1_rgbi_jan_50cm_84sutm54_tif**.
Image Names:		
<***block_id***>_<***Band***>_<***pixel_size***>.*tif*	B1_**Band6**_2m.tif	The TIFF file where **Band 6** is resampled to **2m** for block id/name of **B1**.
	Band7_250cm.tif	The TIFF file where **Band 7** is resampled **250cm** pixels **without** specifying a block id column.



2.11 Create Zones with *k*-means Clusters

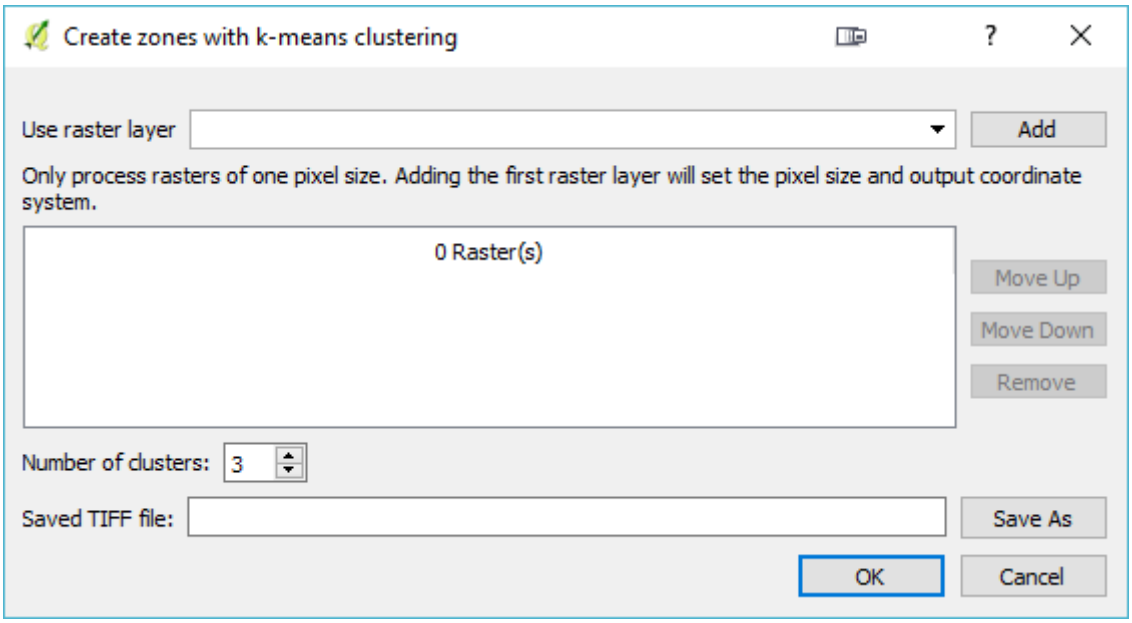
Summary

This tool allows zones to be created by performing k-means clustering on multiple rasters (i.e. map layers) to create clusters of similarity by minimising variability within clusters while maximising variability between clusters. If significant differences between clusters are observed, then the clustered results can be used as potential management zones.

Raster files with the same single pixel size, and coordinate systems are used as inputs and the common area of overlap will be used to generate an output TIFF containing the clustered result.

On completion of k-means clustering, the mean and standard deviation for each zone/cluster and source raster combination will be calculated and written to a CSV File alongside the output TIFF file as well as being displayed in PAT's log messages panel.

Dialog



Create zones with *k*-means clustering		
Use raster layer:	The raster layer to extract values from. Only raster layers of the same pixel size can be analysed at a time. The first raster layer added sets both the pixel size and the output coordinate system. The *Add, Delete, Move Up, Move Down,* buttons can be used to manage and order the list of rasters. The order of rasters is retained in the output CSV file with separate columns for each raster and statistic combination.	
Number of clusters:	Default is 3	The number of clusters/zones to create. Aside from trying 3, it is recommended that users also try 2 clusters. 4, 5 or an even larger number of clusters may also be appropriate.
Save TIFF file	The name and path of the output TIFF file representing the zones. In addition, a statistics CSV file will be written to disk along with the TIFF, and results printed to PAT's log messages panel.	

File and Column Naming Conventions

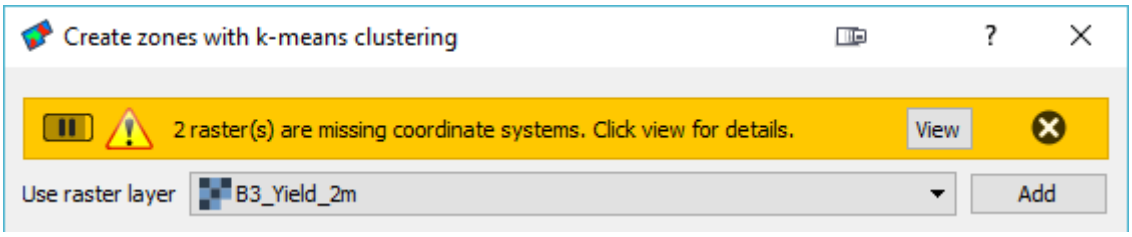
- <> denotes an existing element or input
- non-alphanumeric characters are removed from strings with the exception of hyphens (-) and underscores (_).

FILENAMES:	
Output TIFF	
Naming Rule:	k-means_<*n_clusters*>*clusters*_<***n_rasters***>*rasters*_<***pixel_size***>*.tif*
Example:	k-means_3clusters_5rasters_2m.tif
Description:	A 2m pixel TIFF file for 3 k-means clusters using 5 input rasters.
Output Statistics CSV	**<** output TIFF name**>** where **.tif** is replaced by **_statistics.csv**

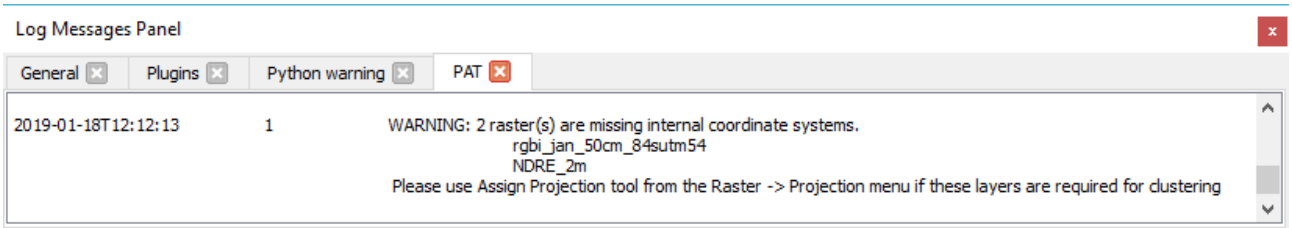
CSV COLUMN NAMES:	
Naming Rule:	**<*raster_file*> _<*statistic*>**
Example:	swblock_2024YldMassDry_PRED_**std**
Description:	The **standard deviation** value for swblock_2024YldMassDry_PRED.tif for the corresponding cluster/zone.

Notes

- To successfully use this tool, all input raster files must contain the coordinate system internal to the file. When a file without a coordinate system is loaded into QGIS, an external coordinate system will be applied based on your QGIS settings. These files cannot be used by this tool. The user will be notified of this when launching the tool as follows.



Clicking on ***view*** will open the PAT Log Messages Panel and provide a list of those images. Users can then use the *Assign Projection* Tool from the *Raster -> Projection* menu to assign the appropriate internal coordinate system to the files. If the *Assign Projection* tool is unavailable activate the GDALTools plugin.



2.12 Create Strip Trial Points

Summary

This tool is used to generate the data needed to conduct the moving window *t*-test method of Lawes and Bramley (2012) and is available in PAT's ***Run Strip Trial t-test Analysis Tool***. It may be useful to users who, for example, have implemented a treatment like an N-rich strip in their cereal paddock, or who want to compare a row of vines that have been pruned differently to the rest of a vineyard.

The tool uses a line representing the centre of a strip trial (or treatment) and creates points along the line spaced using the nominated distance. In addition, two sets of parallel '*control'* points are created at an offset to the strip trial line as shown in Figure 4. This allows the treatment strip data, to be compared to data from the neighbouring (control) area.

The tool creates a number of identifiers to facilitate the analysis: each point has values assigned to new columns headers TrialID, Strip_Name, PointID and DistOnLine. The treatment strip is assigned a unique identifier in the column TrialID and the value of '*Strip'* in the Strip_Name column. Points are created along the line and assigned a unique point identifier in the column PointID. In addition, two sets of control points are placed at the offset distance either side of the strip. The PointID of the control points are paired with the corresponding treatment point. The Strip_Name values of the control points are assigned based on the direction of offset from the original treatment line as shown in Figure 4. Each point is also assigned a value in the column DistOnLine representing the distance in metres of the point from the start of respective line.

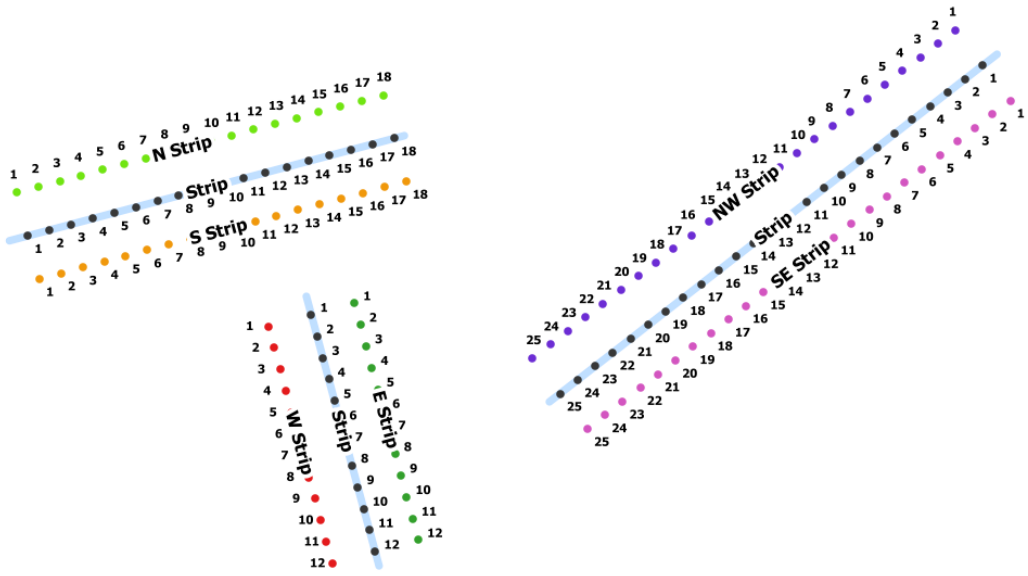
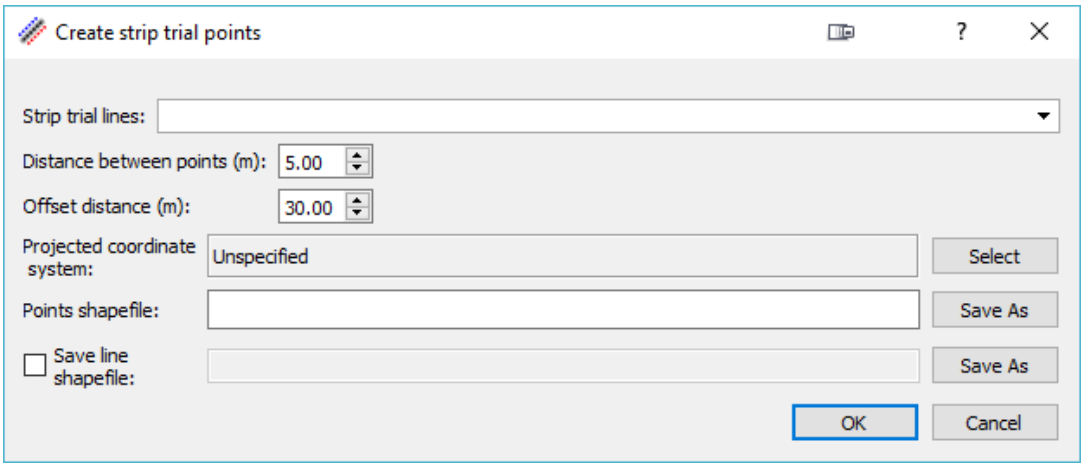


Figure 4 Output from the *Create Strip Trial Points tool*. The blue line represents the treatment strip and the dots indicate the equally spaced treatment and control (i.e. offset) points.

Dialog



Create strip trial points		
Strip trial lines:	The layer containing the strip trial lines.	
Distance between points (m):	1 – 500 m Default is 5m	The distance used to space points along the centre line. In this instance, users are advised to select a value appropriate to their trial rather than accepting the default. If used in conjunction with a strip trial t-test then this distance should be the pixel size of the input rasters used for the t-test. This will result in every pixel in the strip being used for the analysis.
Offset distance (m)	1 – 500m Default is 30m	The distance used to offset the points from the line. If used in conjunction with a strip trial t-test then this should be far enough away from the strip trial to remove the trial's treatment influence. In the case of an N rich strip trial laid down using a boom spray, the offset distance ought to be at least equivalent to the length of the boom.
Projected coordinate system:	The projected coordinate system to apply to the output files. By default it will calculate the relevant coordinate system from input layer's coordinate system. Note: If in Australia then MGA GDA 1994 zones, will be used – see Figure 8 on page 59.	
Points shapefile:	Derived from layer name	The name and path of the output points shapefile.
Save line shapefile:	Default is unchecked. Derived from layer name.	The name and path of the output points shapefile.

Column Naming Conventions

CSV COLUMN NAMES:	
FID	Unique feature identifier for each point.
TrialID	The unique strip trial identifier.
Strip_Name	A name assigned to each line. The treatment will be assigned a value of *'Strip'* while the offset strips will be assigned the cardinal direction of the offset to the treatment as shown in Figure 4.
PointID	The identifier of each point along the centreline, and its paired control points.
DistOnLine	The distance in metres of the point from the start of each line.

2.13 Run Strip Trial *t*-test Analysis

Summary

This tool enables a user to analyse strip trials to determine the effectiveness of treatments. For example, it could be used analyse the difference between an N-Rich strip treatment and the adjacent normally fertilized paddock, or to compare different pruning or spray treatments in a vineyard.

Using points created from the ***Create Strip Trial Points Tool,*** values will be extracted from input rasters and a pairwise moving window *t*-test conducted based on a calculated Response Index (RI) and treatment differences. These are then graphed as described in Lawes and Bramley (2012).

The input rasters include:

- **Strip values raster:** A map produced through following the high density on-the-go sensor mapping workflow (Figure 1) which includes data sourced from the trial strip.
- **Control values raster** (optional): A map produced through following the high density on-the-go sensor mapping workflow after data sourced from the trial strip have been removed. Often this is desirable because, when interpolating a map containing all the data for a 'nitrogen rich' strip, the higher values in the strip may impact the interpolation of adjacent areas. In general, our recommendation would be to make use of a control values raster by re-kriging the original data excluding the data points under the trial strip.
- **Zone raster** (optional): A zone raster can be created using the ***Create Zones with k-means Clusters*** tool. You may want to use this to identify which zone different parts of the experimental strips are located in. Desirably in a field with defined zones the strip would run across these zones.

For each input strip a map image will be produced along with a graph and CSV file, for each strip combination a) Strip vs Control 1, b) Strip vs Control 2, c) Strip vs Average of Control 1 & 2 (Figure 5).

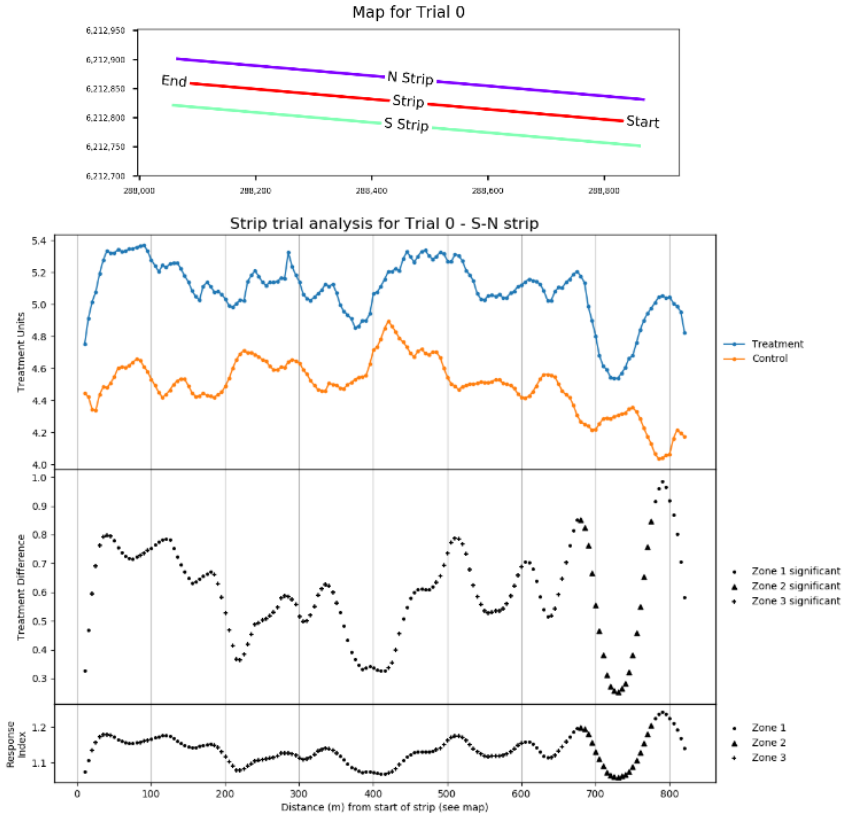
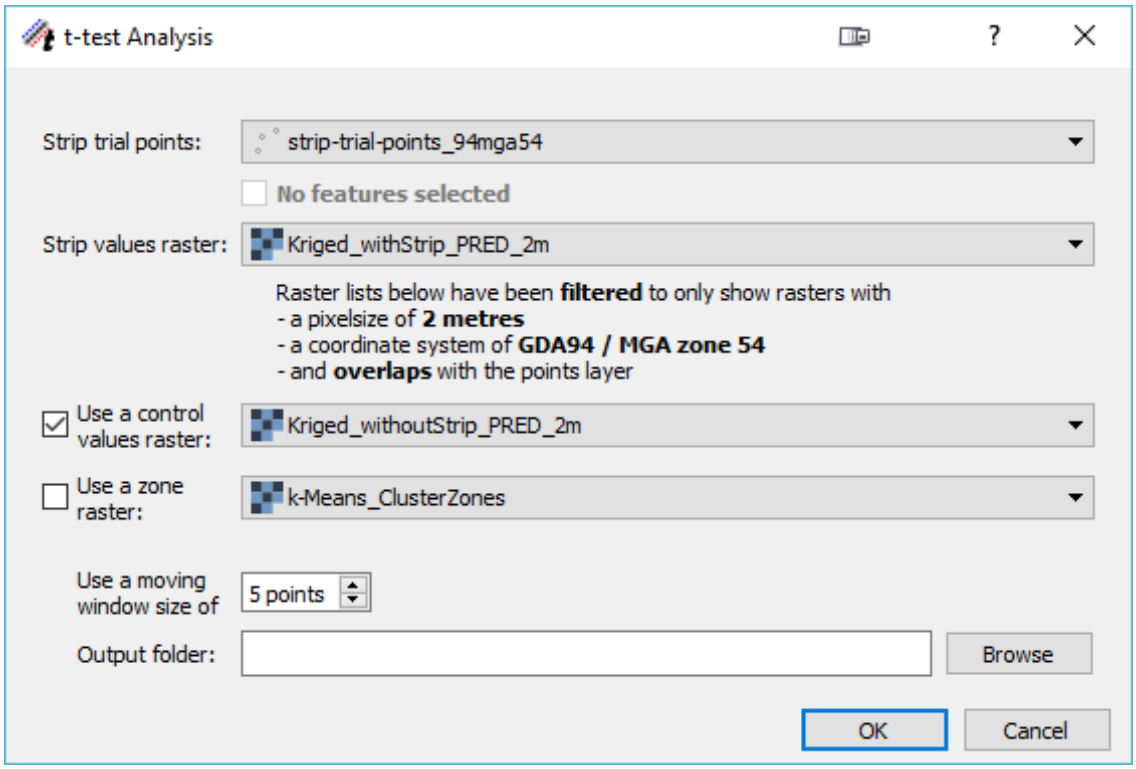


Figure 5 Output graph and map from the *Run Strip Trial t-test analysis tool*.

Dialog



Run Strip Trial *t*-test Analysis		
Strip trial points:	The layer containing the strip trial points. This must be the output from the ***Create Strip Trial Points Tool***.	
Strip values raster:	The raster containing values to extract for analysis.	
Use a Control values raster:	Default is unchecked	The raster layer created after treatment points have been removed.
Use a Zone raster:	Default is unchecked	The raster containing the zone values. If not provided, a zone of 1 will be applied to the entire strip.
Use a moving window size	Derived from layer name	The name and path of the output points shapefile.
Output folder:	The folder to save output map, graph and CSV files. See table below for used for output files.	

File and Column Naming Conventions

- <> denotes an existing element or input
- non-alphanumeric characters are removed from strings with the exception of hyphens (-) and underscores (_).

TrialID represents the unique strip trial identifier from input points ***TrialID*** attribute.

strip represents the value(s) from input points ***Strip_Name*** attribute. When the analysis uses both control strip points their values will be hyphenated e.g. N-S

FILENAMES:	
Map file:	A map showing the Trial points.
Naming Rule:	*Trial-<**TrialID**>_map.png*
Example:	Trial-**3**_map.png
Description:	The map for the strip with a TrialID of **3.**
Graph files:	The results displayed as graphs.
Naming Rule:	Trial-<***TrialID***>_<***strip***>_graph.png
Example:	**Trial-3_N-strip**_graph.tif
Description:	The graph result for Trial **3** which compares the **N** control strip to the treatment strip.
CSV files:	The csv file used to create the graphs.
Naming Rule:	Trial-<***TrialID***>_<***strip***>.csv
Example:	Trial-**3_N-S-strip**.csv
Description:	The csv file containing the Trial **3** results for the comparison of **N-S** control strip to the treatment strip.

CSV COLUMN NAMES:	
TrialID	The unique strip trial identifier from the input *Strip Trial Points*.
PointID	The identifier of each strip trial point from the input *Strip Trial Points*, and the paired control points.
DistOnLine	The distance in metres of the point from the start of the centreline of the treatment strip.
Strip Value	The number extracted from the **Strip value raster**. The units of value will be inherited from the source data (eg NDVI, Yield t/ha).
<strip> Control (if provided)	The number extracted from the **Control values raster.** Applies only when a **Control values raster** is used. The units of value will be inherited from the source data (eg NDVI, Yield t/ha).
Strip Zone (if provided)	The zone identifier extracted from the **Zone raster**. Creates a Zone value of 1 if not specified.
<***strip***>_mean	The mean of the input control strips for each point if both control strip are used.
treat_diff	The difference between the treatment and controls points columns.
av_treat_diff	The average of the column treat_diff using a moving window.
p_value	The t-test p-value calculated using a moving window.
RI	The RI calculated using a moving window: RI = mean of treatment / mean of control

The column names for the output file are dependent on the input scenarios. In the following example, a subset of the column names for the East-West strip trial shown in Figure 4 where a **strip values raster**, **control raster** and **zone raster** were provided as inputs.

CSV COLUMN NAMES:	
Strip Value	The pixel value from the strip value raster for the strip trial points.
N Strip Control **S Strip Control** **- or -** **N Strip Value** **S Strip Value**	The pixel value from the control raster for control line to the north or south of the treatment strip. If a control raster is not provided, then pixel value will come from the strip value raster and the column name will be ***N Strip Value*** or ***S Strip Value***.
Strip Zone	The pixel value from the zone raster for the strip trial. If a zones raster is not provided, then a value of 1 will be assigned.
N Strip Zone **S Strip Zone**	The pixel value from the zone raster for control points to the north or south of the strip trial. If a zones raster is not provided, then a value of 1 will be assigned.
N-S_mean	The average of the control strip points when both of sets of control strips are used in the analysis.



2.14 Whole-of-block Analysis

Summary

Whole-of-block experimentation, also known as spatially distributed experimentation, involves on-farm experiments where treatments are applied in a highly replicated design over an entire block or land management unit, or a large section of it. Given their scale, they are typically laid down using the farmer's normal equipment. These experiments can be thought of as a three-dimensional version of the strip trial, discussed in the previous section ***Create Strip Trial Points.*** Users interested in this experimental approach may wish to read the book chapter by Bramley et al. (2013) and perhaps the references listed in that chapter. In brief, the objective of whole-of-block experimentation is to recognise the possible effect of underlying spatial variability in the field in which the experiment is located, on the crop response to experimental treatments, and so use this variability as an experimental tool to guide decisions about variable rate or targeted management. Whole-of-block experimentation therefore moves away from the classical approach to agronomic experimentation which seeks to identify whether treatment A is better than treatment B and typically uses small plots. Instead, it is recognised that treatments A and B may both deliver benefit, albeit in different parts of the block or field; the whole-of-block approach identifies these different areas.

An example is given in Figure 6. Here, some different vineyard floor management treatments have been evaluated in an experiment aimed at improving the availability of water to the vines. The cereal/legume treatment delivers a significant benefit over the other treatments, but this is expensive and so the vineyard manager would only want to use this in those areas where it is beneficial. The whole-of-block analysis tool enables users to generate maps such as those shown in Figure 6 – maps showing the response to the different treatments over the entire block, and maps indicating the significance of differences between these treatments.

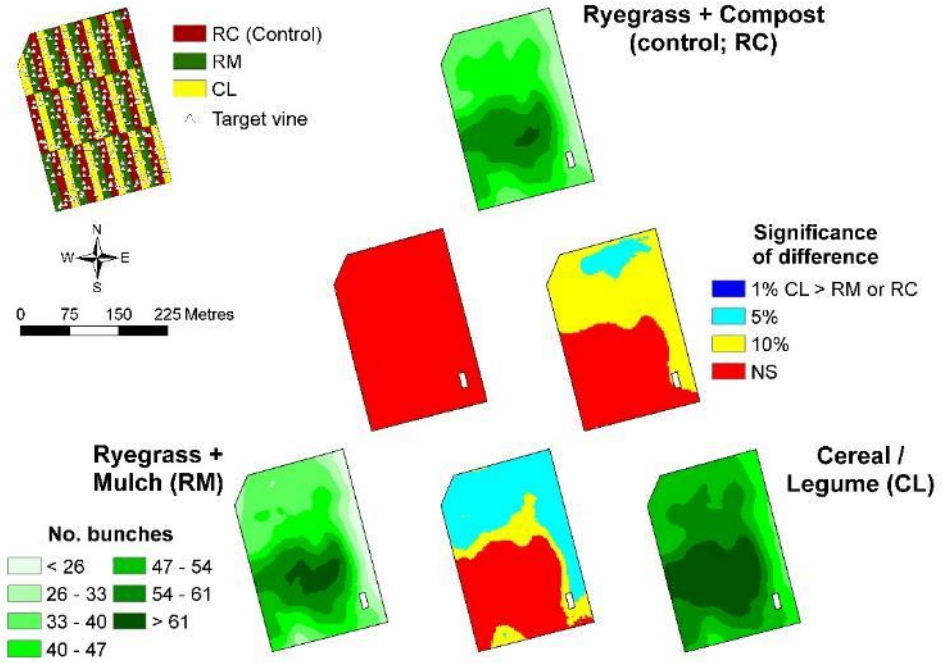


Figure 6 An example of a whole-of-block experiment conducted in a 4.8 ha Clare Valley vineyard. Panten and Bramley (2011) provide further information

The purpose of the ***Whole-of-Block analysis tool*** is to predict treatment responses and their differences over the entire experimentation area as illustrated by Figure 6. Similar to the use of VESPER for kriging, this tool interpolates point data (low or high density) to the area of the entire block. Low density data would typically include up to a few hundred points, perhaps collected through hand sampling, dispersed across a block, whereas high density data includes a few thousand or more points, typically collected using a sensor such as a yield monitor. The tool will handle either two or three treatments, and will identify areas of the block where the treatments are significantly different from each other. To address the technical and computational challenges in this analysis, the tool uses fast and efficient global and local cokriging techniques.

Use of R

This tool is somewhat different from the other PAT tools in that it uses the R statistical language (http://r-project.org/) to carry out the analytical processing. This means that R must be installed on your PC for this to work – see ***Configuring QGIS to use R*** on page 10. The minimum R version suitable for this is v3.5.1.

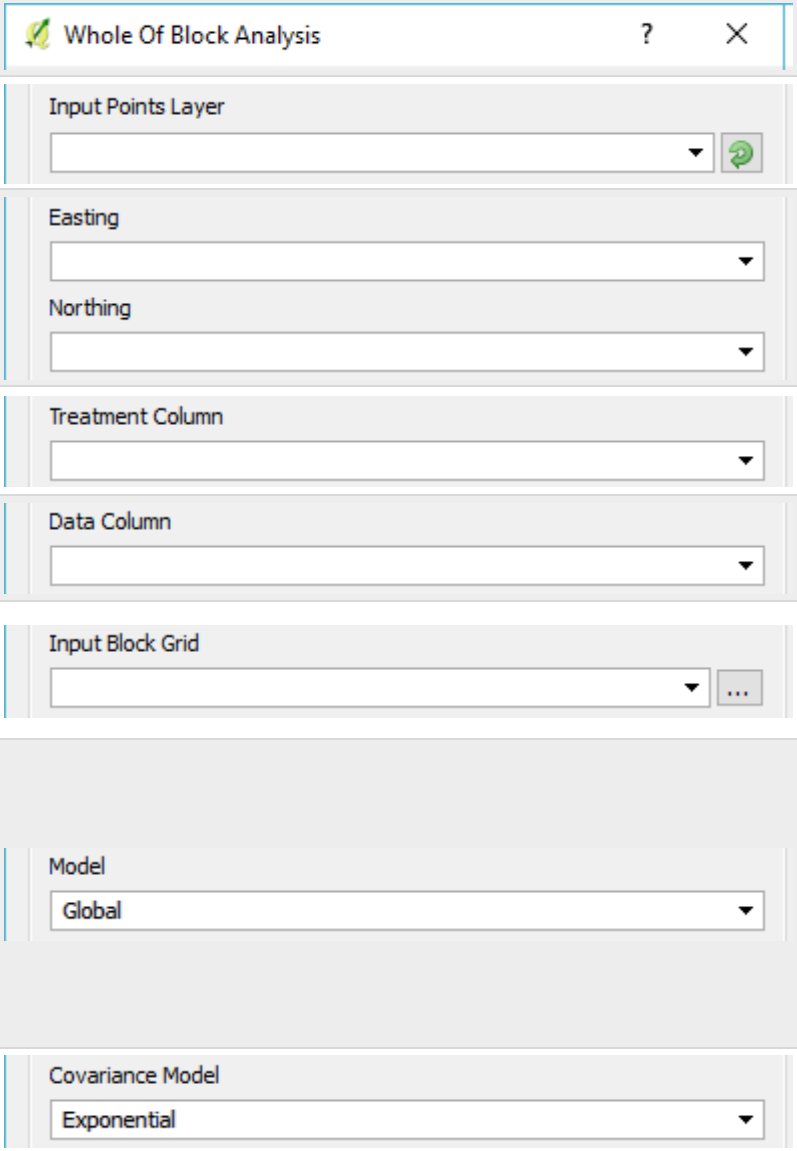
Algorithm for the Local Cokriging System – a brief commentary on the data radius used

A spatially-varying local co-kriging method is used to balance the trade-off between the computation efficiency and prediction accuracy. We use a neighbourhood radius r(s) to identify neighbours for location s.

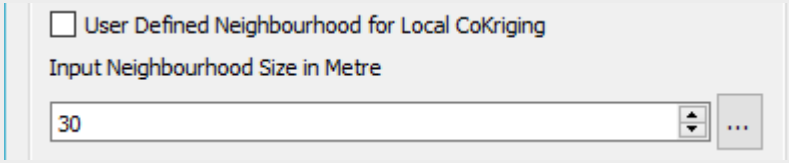
For subregions with lower density observations, such as at the corners, or edges of a farm field, the neighbourhood radius required by local cokriging will be relatively bigger to ensure the stability of the estimated cross-variograms and balance the number of observations from different treatments within the neighbourhood.

For high density datasets we will use a relatively small neighbourhood radius, and ensure it will change relatively smoothly across the field. This adaption is achieved by two specific algorithm choices. Firstly, an initial neighbourhood radius is selected automatically based on the observation density and achieving a reasonable balance in the number of observations from different treatments. This neighbourhood radius increases iteratively to include more observations if the current radius is not large enough to meet prediction quality criteria. Secondly, a spatial-varying neighbourhood radius surface is estimated based on neighbourhood radius values from a subset of locations. This exploits the spatial-smoothness of the neighbourhood radius and provides a better default neighbourhood radius from which to optimise.

Dialog



Input points layer		A text format data file (CSV format) containing the experimental data that has been loaded into QGIS. The file should be a minimum of four columns containing, the following:
Easting and Northing		The names of columns in the CSV data file containing the projected Easting and Northing point coordinates
Treatment column		The name of column in the CSV data file containing the treatment applied at each point. The treatments can be represented by numbers (1,2,3), short characters (“AM”,”BL”,”CS”) or text 10 characters or less **without** spaces like (“Control“,“Mulch”,”Cover_crop”).
Data column		The name of column in the CSV data file containing the measured treatment response (e.g. Yield) at each point.
Input Block Grid or Prediction Grid		The grid onto which treatment predictions, differences and standard errors are to be interpolated by the tool. The block grid created by PAT Create Block Grid tool is ideal for this, or it can be a raster (TIF) file with the same coordinate system as the Input Points Layer.
Model	Default is Global	There are two model options. • Global cokriging (default) is useful for small datasets (a few hundred points) • Local cokriging is useful for moderate or larger datasets (thousands of points), particularly with local heterogeneity. This option initiates a starting radius for localised cokriging and uses an iterative algorithm to optimise the local cross-covariance structure. The cross-variogram is fitted using a Linear Model of Coregionalisation (LMC). The current version of the software defines 20 lag classes, giving 20 points to the cross-variogram. The maximum distance used in this process (i.e. largest lag class) is set as 1/3 of the maximum distance between points in the input dataset.
Covariance Model	Default is Exponential	Exponential (default); users may also select Spherical and Gaussian models

	**User Defined Neighbourhood radius for Local Cokriging**	Default is unchecked Default is 30m or 15 times the pixel size	Only applicable to the Local Cokriging method, and allows a user to define an initial neighbourhood radius. If this option is ticked then user should define the "Input Neighbourhood Size in Metres". The default is 30 m. A suggested starting value can be calculated as 15 times the pixel size of the block grid. However, if "User Defined Neighbourhood for Local Cokriging" is not ticked then the initial neighbourhood radius will be chosen as a function of total number of data points and corresponding number of treatments. This is the recommended starting point for using this tool.
	**Save output**	Folder in which to save the output files. See File Naming Conventions (below) for an explanation of the different output files that will be created in this folder.	

File Naming Conventions

- <> denotes an existing element or input. For example <Data Col> refers to the name of the Data Column selected in the input parameters (above).
- non-alphanumeric characters are removed from strings with the exception of hyphens (-) and underscores (_).

Filenames:

NAMING RULE	EXAMPLE	EXPLANATION
Output file names:		
<Data Col>_dump_file_<Model><Covariance Model>.tif	Bunch_wt_dump_file_GlobalExp_.log	Logged information mainly useful for debugging purposes
<Data Col>_model_parameters_<Model><Covariance Model>.txt	Bunch_wt_model_parameters_GlobalExp_.txt	Summary of parameters used during processing
<Data Col>_list_of_tif_files_<Model><Covariance Model>.txt	Bunch_wt_list_of_tif_files_GlobalExp_.txt	List of TIF files generated as output
<Data Col>_tr_<Tmt1>_<Model><Covariance Model>_.tif	Bunch_wt_tr_KC_GlobalExp_.tif	Treatment inference in "Input Block Grid or Prediction Grid".
<Data Col>_tr_<Tmt1>_var_<Model><Covariance Model>_.tif	Bunch_wt_tr_KC_GlobalExp_.tif	Corresponding variances of the treatments.
<Data Col>_ tr_diff_<Tmt1>_<Tmt2>_<Model><Covariance Model>_.tif	Bunch_wt_p_val_KC_SP_GlobalExp_.tif	Predictions of treatment response differences between treatments
<Data Col>_tr_<Tmt1>_var_<Model><Covariance Model>_.tif	Bunch_wt_tr_KC_var_GlobalExp_.tif	Corresponding covariances of the treatment differences.
<Data Col>_p_val_<Tmt1>_<Tmt2>_<Model><Covariance Model>_.tif	Bunch_wt_p_val_KC_SP_GlobalExp_.tif	p-values of the treatment differences between pairs of treatments.
<Data Col>_z_<Tmt1>_<Tmt2>_<Model><Covariance Model>_.tif	Bunch_wt_z_KC_SP_GlobalExp_.tif	Z-statistics of the treatment differences.

2.15 Persistor

Summary

Persistor is used to summarise the similarities in patterns of variation across a block over a number of years (Bramley and Hamilton 2005). It might be used as an alternative to *k*-means clustering for identifying clusters based on yield, but also enables zones to be assessed in terms of a target level of performance (e.g. mean yield + 10%) and so provides information about how frequently such a target is met.

The "***Target over all years***" method assigns a value to each pixel to indicate the number of instances (in the raster list) in which that pixel was either less than or greater than the mean (+/- a nominated percentage) of that raster.

The "***Target probability***" method builds on the *target over all years* method, in that it includes an upper range (i.e. cells with a given frequency of values that are above the mean +/- a given percentage) and a lower range (i.e. cells with a given frequency of values that are below the mean +/- a given percentage).

The tool assigns a value to each pixel which indicates whether the performance in that pixel over a given proportion of years is:

a) Greater than the mean plus or minus the nominated percentage (value = 1)
b) Less than the mean plus or minus the nominated percentage (value = -1)

The remaining pixels which do not fall into category a) or b) are given a value of 0.

All input grids must overlap and have the same coordinate system and pixel size.

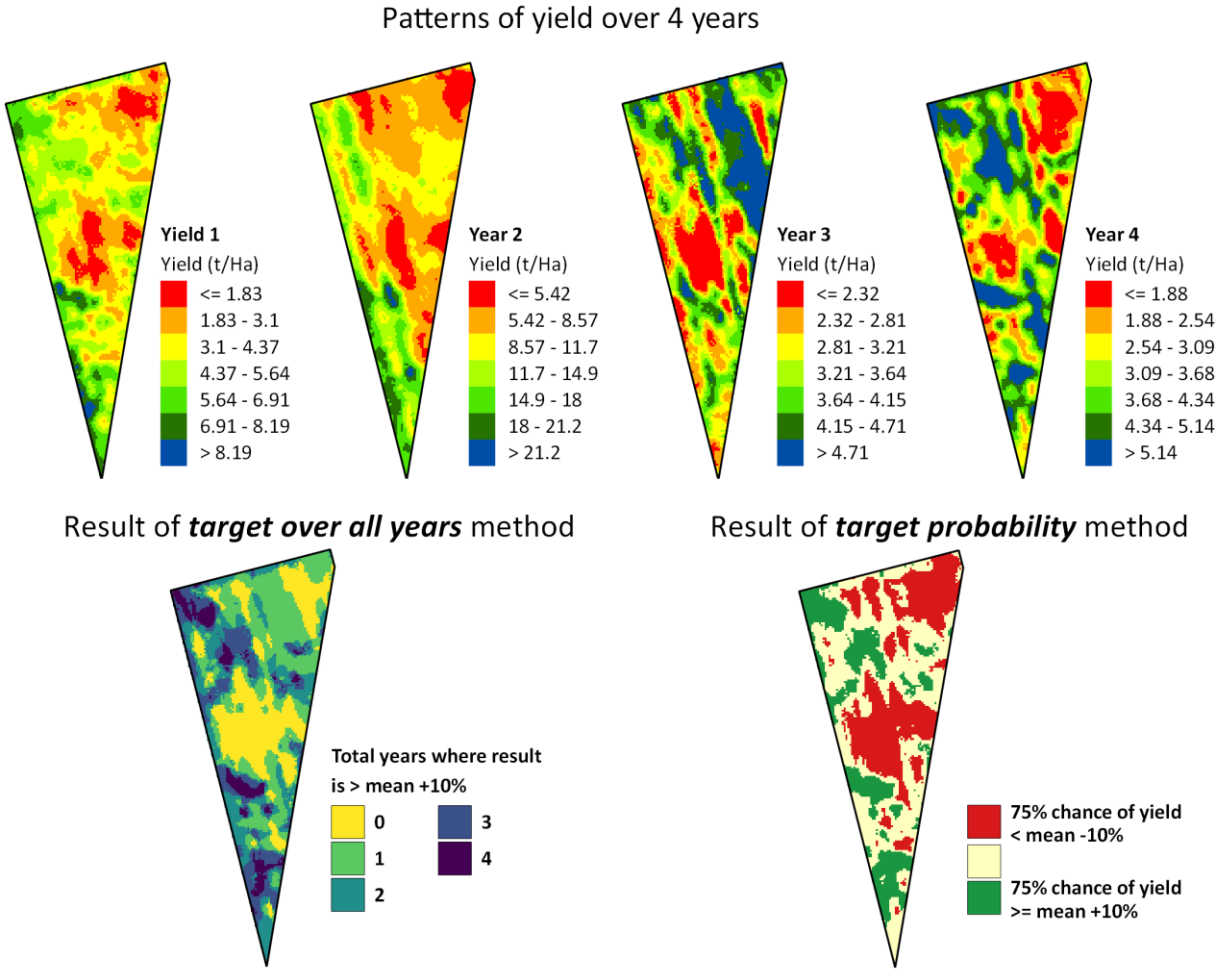


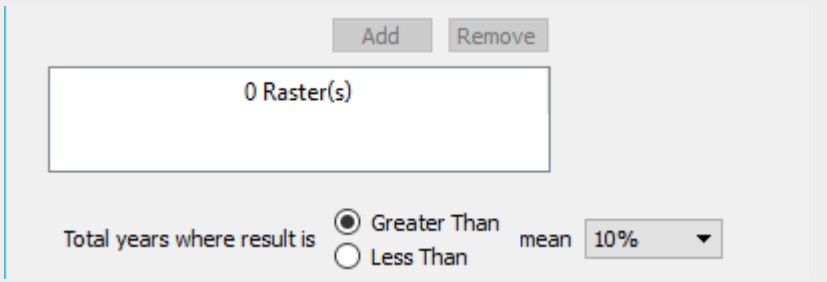
Figure 7 An illustration of the *Persistor Tool* results.

Dialog



Persistor method: Target Probability	**Persistor Method:**	Default is Target Probability	Select the analysis method from: **Target Over All Years** –the performance persistence of yield across multiple years. **Target Probability** – indicates which pixels meet an upper and a lower limit.
Select raster layer:	**Select a raster layer:**		Use *Select Raster Layer* with the *Add* button to add rasters to the list(s) below. Raster layers will be filtered ensure they have the same pixel size as the first raster layer added to either the upper or lower raster lists. The values for these filters applied will be shown on the dialog.

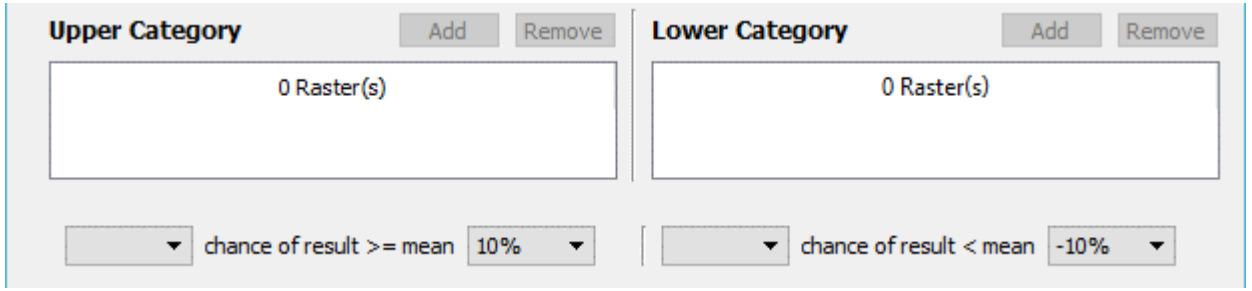
TARGET OVER ALL YEARS *The performance persistence of yield across multiple years.*



Raster List		The list of rasters used for the ***Target Over All Years Method.*** Use *Select Raster Layer* with the *Add* button to add rasters to the list. The *Remove* button removes a selected raster from the list.
Total years where result is	Default is Greater Than and +10%	Count the years (from the list of rasters) where the pixel value is either **greater than** or **less than** the mean plus or minus the given **percentage** of the raster mean.

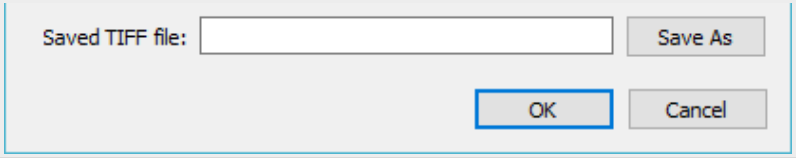
continued on next page……

TARGET PROBABILITY	*Indicate which pixels meet an upper and a lower limit.*



Raster Lists	Use *Select Raster Layer* with the *Add* button and the *Remove* button to manage the rasters in the upper and lower category lists.

UPPER LIMIT			**LOWER LIMIT**		
Upper Category Raster List	The list of rasters to apply upper category rules too**.**		**Lower Category Raster List**	The list of rasters to apply upper category rules too.	
Upper Category Settings	Default mean percentage is **+10%**	The **frequency** (as percentage) with which a pixel must exceed the raster mean (plus or minus the given **percentage)** to be included in the upper category. Pixels in this category will be assigned a value of +1.	**Lower Category Settings**	Default mean percentage is **10%**	The **frequency** (as percentage) with which a pixel must fall below the raster mean (plus or minus the given **percentage)** to be included in the lower category. Pixels in this category will be assigned a value of -1.



Saved TIFF file	The name and path of the output TIFF file.

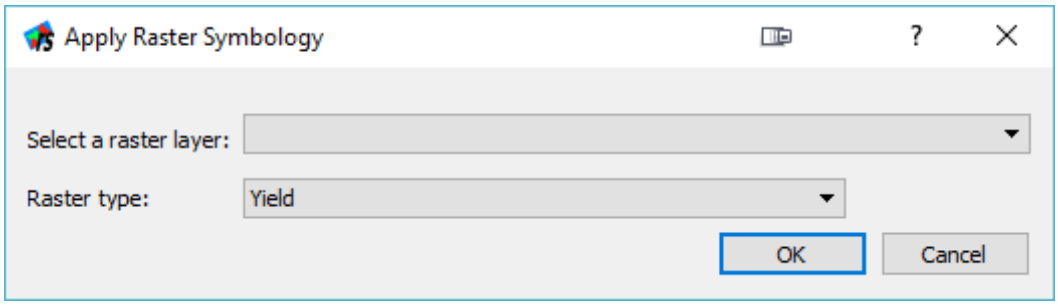
2.16 Apply Raster Symbology

Summary

The ***Apply Raster Symbology Tool*** allows for the quick application of the symbology for the different PAT tool output rasters.

Instructions for installing the PAT symbology can be found in ***4.3 Loading PAT Symbols into QGIS.***

Dialog



Supported raster types	Used by	Colour Ramp
Yield	Run Kriging Using VESPER, Import VESPER Results.	Yield 7 Colours
Image Indices (ie PCD, NDVI)	Calculate Image Indices for Blocks.	Imagery 5 Colours
Zones	Create Zones with k-means Clusters.	Random Colours
Block Grid	Create Block Grid.	Random Colours
Persistor - All Years	Persistor – Target All Years Method.	Viridis
Persistor - Target Probability	Persistor - Target Probability Method.	RdYlGn

2.17 Settings

Summary

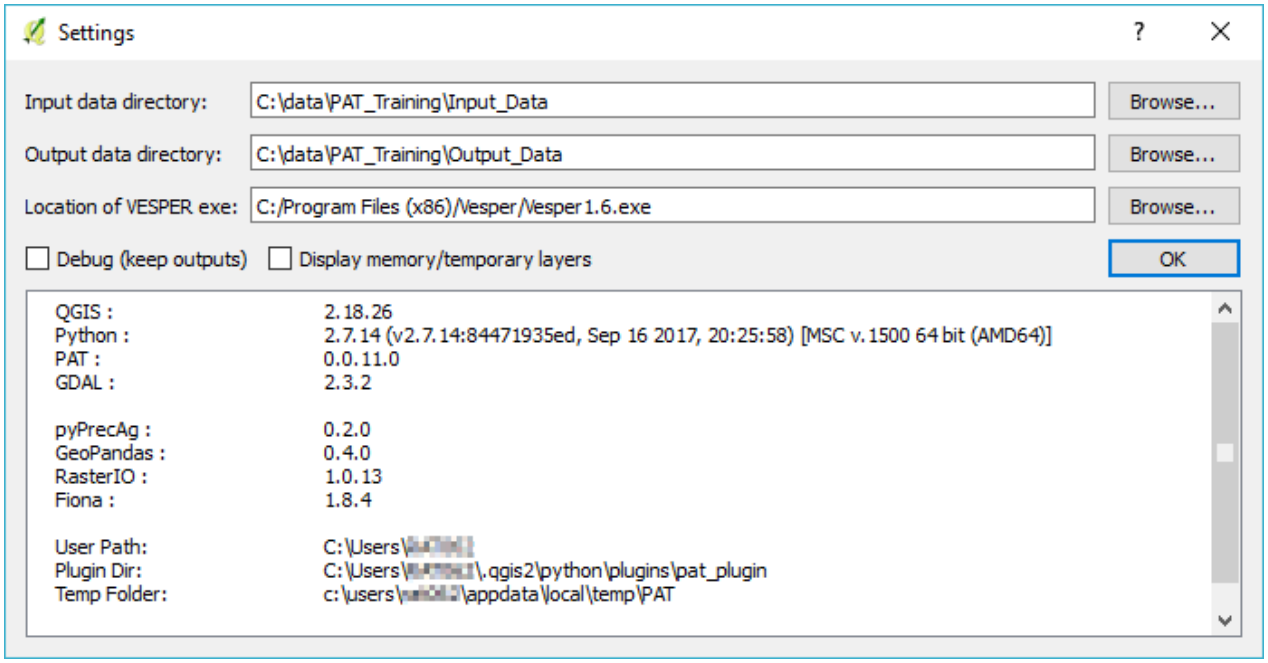
This tool is used to display and edit PAT settings.

The input and output data directories set here will be used to set the default paths by the browse for file/folder functionality for all tools. Each tool will store and access its own values after first time use.

Checking the Debug (keep outputs) box will save intermediate files created while processing data to file. It should be noted that this will slow down the time taken to run tools, but can be a useful diagnostic tool.

The **Display Memory/Temporary Layers** checkbox can be used to add the intermediate files to QGIS along with any in-memory or virtual layers which are used but not saved to disk.

Dialog



Input Data Directory	A folder containing input data.*
Output Data Directory	This is the location of the output file.
Location of VESPER Exe	Set the location of the VESPER executable. If this is not found or specified, VESPER cannot be run, but relevant VESPER input files can still be created.
Debug (Keep Outputs)	When checked, various intermediate files will be written to disk to assist with error/debugging analysis. However, this will slow down processing.
Display Memory/Temporary Layers	If checked, any files written to disk, or created in memory will be loaded into QGIS into a grouped layer labelled DEBUG and can be used for error/debugging analysis.

* By default this will be the PAT sub-folder in the user's home directory. To quickly navigate here type *%homepath%/PAT* in the address bar of Windows Explorer.

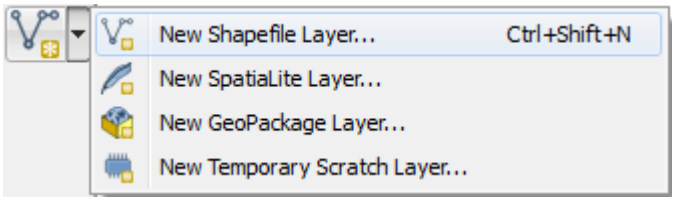
3 Technical Notes

- PAT makes use of the CSIRO-developed pyPrecAg Python module which is an open source Python package containing a range of specialised analysis functions.
- All intermediate files created while processing is located in the PrecisionAg folder of the user's temporary folder. To quickly navigate to the temporary folder, type %temp%/PrecisionAg in the address bar of windows explorer. This folder is deleted when QGIS exits.
- All progress, messages and errors are displayed in the PAT tab of the Log Panel as shown in Figure 2 and are saved to a log file located in the PrecisionAg folder of the user's temporary folder. A list of important paths including the location of temp and the user's plugin folder.
- A Users QGIS Plugin folder can be found by typing %homepath%/.qgis2/python/plugins into the address bar of Windows Explorer.

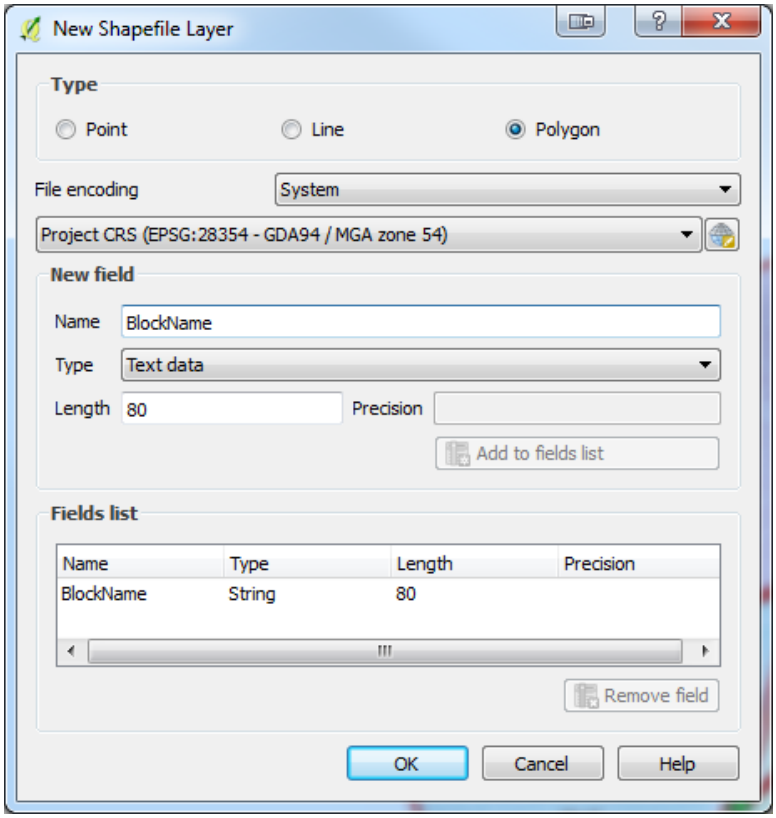
4 How-To's

4.1 Create a block boundary polygon from a CSV of GPS collected points

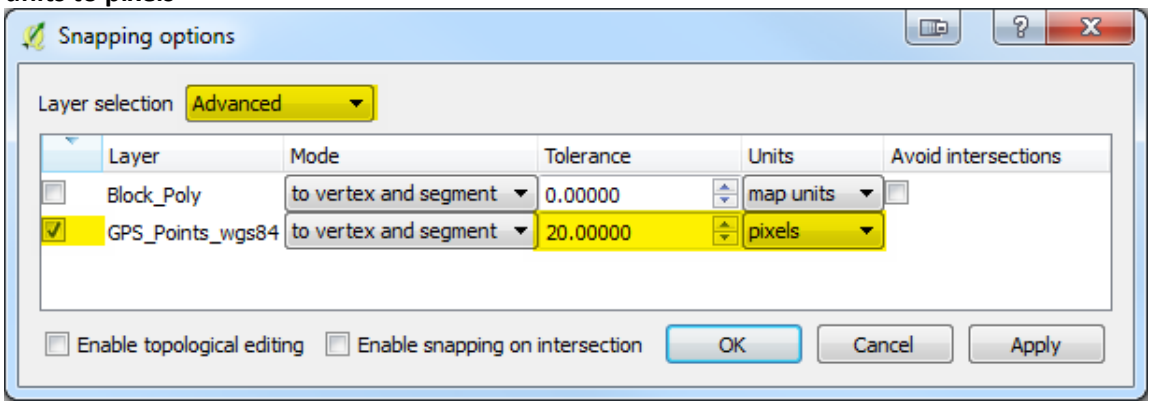
1. Using the ***New Shapefile Layer*** tool from the ***manage layers toolbar*** *(or Layer menu -> Create Layer -> New Shapefile Layer*) create a new polygon shapefile adding the relevant coordinate system* and attribute fields you require. Clicking OK will prompt you for the location to save the shapefile.



More information on coordinate systems can be found at



2. Set a style and labelling to the polygon layer. A hatching polygon fill works well for editing.
3. Launch the ***Add Delimited Text Layer*** tool from the ***manage layers toolbar*** *(or Layer menu-> Add Layer -> Add Delimited Text Layer)* and load your GPS CSV file as a layer into QGIS.
   Hint: Your coordinate system is probably WGS 84.
4. If required, load other vector or raster data, like imagery, which can be used as reference.
5. **Setup your snapping environment**.
   a. Open ***snapping*** options (*Settings menu -> Snapping Options*)
   b. Change ***layer selection*** to ***advanced***.
   c. **Tick** the layer containing the **point layer** loaded in step 3.
   d. Change the ***tolerance*** to **20** and set **units to pixels**



*More information on selecting a coordinate system can be found on page 59.

6. Select/activate the **polygon** layer in the layers panel and click the ***toggle editing*** icon  from the ***digitizing toolbar***.

7. **Add new features** by using the ***add features*** tool  from the ***digitizing toolbar***.
8. As you move the mouse close to a point the point will change to show a magenta cross hairs (+), this means the mouse has snapped to this point. Clicking the mouse will add this point as a vertex in the polygon. Continue following around the points to form a polygon. Right-mouse-click to finish a polygon.
9. When you finish a polygon a dialog will open to allow you to enter attributes. Click **OK** to add attributes and finalise polygon.

   Save your edits using the ***save layer edits*** icon  on ***digitizing toolbar*** and ***toggle editing*** off  when complete.

To add, move or delete a vertex, toggle to node mode using the ***node tool*** . Click the polygon to edit. Nodes/Vertex will appear as red squares.

- **Double click** to **add** new vertex.
- **Single click** to select existing vertex. The square will turn blue. Use the **DEL** key to delete
- **Click** and **drag** a vertex to **move**.

To add a hole (donut) to a polygon use the ***add ring*** tool  from the ***advanced digitizing toolbar*** and sketch your polygon as described in step 8.

To delete a hole (donut) in a polygon use the ***delete ring*** tool  from the ***advanced digitizing toolbar*** and click in the hole.

To split a polygon, use the ***split features*** tool  from the ***advanced digitizing toolbar*** and sketch the path to split. Multiple polygons will be created having the same attribution.

Useful editing shortcut keys.

Add new feature	Ctrl+.	**Zoom in**	Scroll wheel or	Ctrl ++
Delete last vertex	Del	**Zoom out**	Scroll wheel or	Ctrl +-
Undo	Ctrl+Z	**Zoom full**	Ctrl+Shift+F	
Cancel edit	Escape	**Pan**	Middle mouse or Spacebar (while adding a feature only). Note: spacebar also turns active layer visibility on/off.	

4.2 How to reproject a shapefile and why you need to do this

A projected coordinate system is used to depict the curved earth surface as a flattened surface – as it is when printed on a piece of paper. Currently the block grid tool requires the polygon shapefile to be in a projected coordinate system. However, raw GPS data such as you might have in your yield monitor data file uses a geographic coordinate system most likely with position expressed as decimal degrees or degrees minutes second (such as in Google Maps). These will typically be labelled as latitudes and longitudes.
If your shapefile is in a geographic coordinate system then you will need to reproject it to a projected coordinate system.

Geographic coordinate systems include:
Global: EPSG: 4326, WGS 84
Australian: EPSG: 4283, GDA94

In projected coordinate systems, the Globe/Australia is divided up into zones (see UTM zone map). Map grid of Australia (MGA) zones align closely with the UTM zones, but use the 1994 Geocentric Datum of Australia. Figure 8 shows the position of the MGA zones.

Projected coordinate systems include:
Global: WGS84 / UTM Zone divided in to North and South of the equator
Australian: GDA94 / MGA zones

You need to know the projected coordinate system for your data and which zone it lies in. This projected coordinate system will get used throughout PAT. This is made simpler through the use of EPSG numbers. An EPSG number is an international numbering system for coordinate systems (see http://www.epsg.org)

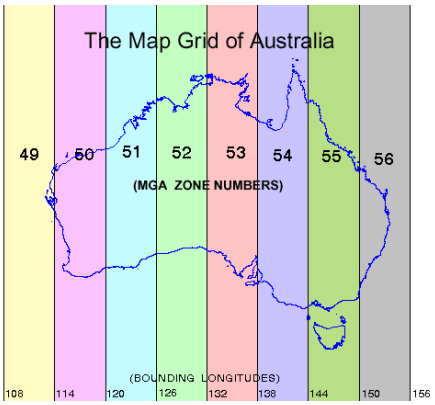
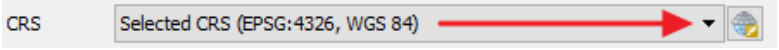


Figure 8 The Map Grid of Australia. (www.environment.gov.au/erin/tools/mga2geo-gda.html).

1. **Determine your projected coordinate system.**
   - Within Australia use GDA 1994 Map grid of Australia (MGA)
     EPSG: 283xx
   - Outside Australia use the WGS84 UTM Zone system.
     North of the Equator - EPSG: 326xx
     South of the Equator - EPSG: 327xx

   Using your EPSG prefix listed replace the xx with your zone. For example Adelaide is an Australian city in zone **54** so projected coordinate system then becomes EPSG: 283**54** GDA94 / MGA zone **54**.

2. **Reproject your shapefile**
   a. In QGIS, select your shapefile and select ***Save as*** from the ***Layer menu.***
   b. In the dialog, browse for a new shapefile and assign a name.
   c. Click the icon to the far right of the CRS options.



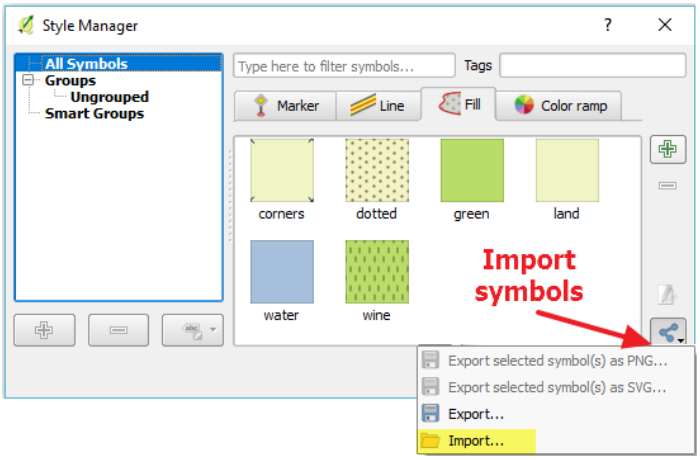
   d. Search for your projected coordinate system. The easiest way is to use the EPSG number from step 1.
   e. Leave all other options as is and click OK

The shapefile will be reprojected and loaded into QGIS and is now ready for use.

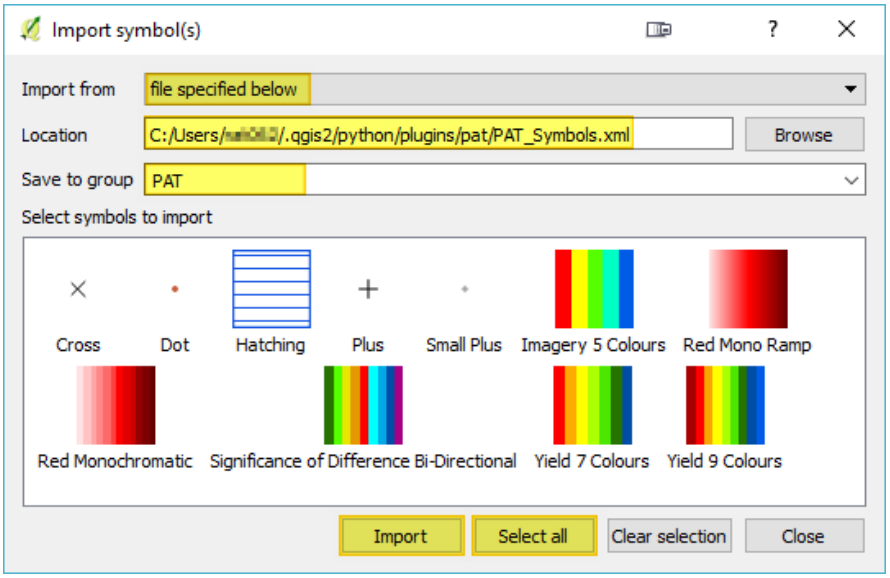
For further information on coordinate systems see
https://docs.qgis.org/2.18/en/docs/gentle_gis_introduction/coordinate_reference_systems.html

4.3 Loading PAT Symbols into QGIS

The PAT plugin includes a pre-defined set of symbols and colour ramps for use with datasets derived while using the plugin.



1. In QGIS, launch the **Style Manager** from the settings menu.
2. From the lower right corner of the dialog, select ***Import*** dialog.
3. Set ***Import from*** to ***file specified below.*** Browse to the users QGIS plugin folder and find the **PAT_symbols.xml** file in the pat folder (see 1.5 Uninstall PAT for help finding this folder).
4. Enter PAT as the ***Save to group.***
5. Select symbols to import or click ***Select all***.
6. Click ***Import***. If symbols with the same name are already loaded, you will be notified and given the option to overwrite.



4.4 Kriging 'low density' data in PAT

The mapping protocol which forms the 'backbone' of the PAT workflow is designed to handle data collected at a high spatial density using an on-the-go sensor – such as a yield monitor, proximal canopy sensor or electromagnetic soil sensor. Sometimes, there will be a need to collect data using hand sampling – for example using biomass cuts in a wheat field, counts of bunch numbers in a vineyard or assessments of disease – and for mapping these. Whereas the interpolation of high density, on-the-go data uses local block kriging, for hand sampled data collected at much lower density, global point kriging is more appropriate. This can be done in VESPER, but requires more user input, not least because the 'variogram' which underpins interpolation using kriging, needs to be generated manually and carefully. Additional information on this can be found in the VESPER manual available here with additional information on geostatistics available in books such as Webster and Oliver (2007). These notes are intended to guide the generation of the variogram and other input that PAT needs to run VESPER for low density data sets. Note for this current release of PAT, the procedure requires some use of VESPER independently of PAT.

Important notes

- It is not the intention in these notes to provide training in geostatistics. Users wishing to gain a better understanding of this topic should consult appropriate texts such as Webster and Oliver (2007). However, in general, we do not recommend that PAT users engage in low data density map interpolation unless they have acquired some prior understanding of map interpolation using kriging.
- It should also be noted that we DO NOT recommend interpolation of maps from 'low density' data when fewer than 100 data points are available (Webster and Oliver, 2007).
- Experience with work in vineyards (3-10 ha) suggests that a sample density of 26 samples or data points per hectare leads to satisfactory maps but a general rule of thumb is that higher data densities give better maps.

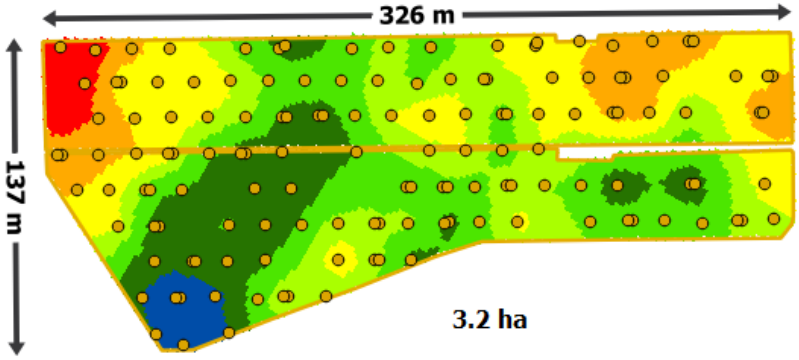


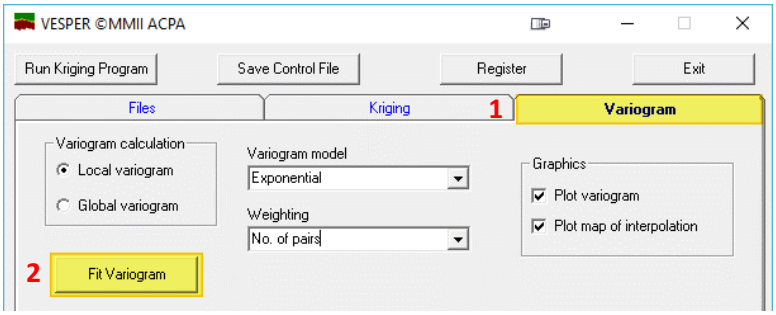
Figure 9 A map showing the data points used for the low data density kriging of 140 measures of grape berry weight in a 3.2 ha vineyard along with the final kriged result. This is the example used for the variogram fitting below.

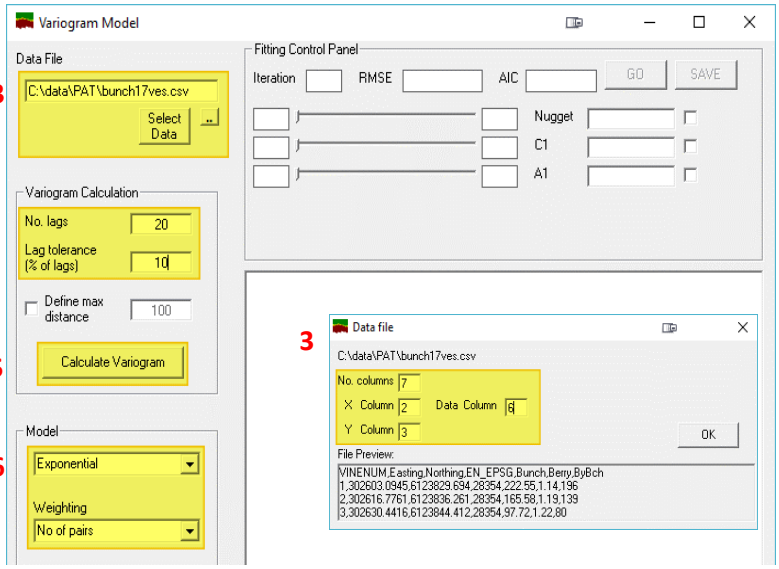
Requirements

- A **CSV file** containing separate columns for each of X coordinates (e.g. Easting), Y coordinates (e.g. Northing) and the data value(s) to be mapped.
- A **VESPER grid file** created using the ***Create Block Grid Tool***
- The coordinate system for both the **CSV file** and **VESPER grid file**. Both these files should have the same coordinate. The ***Clean, Trim, Normalise Point Data Tool*** can be used change the coordinate system of the CSV file to match the VESPER grid file described in 2.2.
- **VESPER:** ***Run Kriging Using VESPER*** tool requires installation of VESPER (see section 1.3.2)

Setup the Variogram

1. Open **VESPER** independently of PAT and QGIS (i.e. as a stand-alone application).
2. In the Variogram tab, click *Fit Variogram*.
3. Browse to your csv file and in the **Data files** dialog that will appear, select the columns representing your X, Y and data value. For this example the X (Easting) is in column 2, Y (Northing) column 3 and data value (Berry) column 6.
4. Change the Variogram computation to
   a. ***Number of Lags*** to **20**
   b. ***Lag tolerance (%)*** to **10**
5. Ensure the ***Variogram Model*** is **Exponential** and ***Weighting*** is **No. of pairs.**
6. Click **Calculate Variogram** to calculate and display the variogram.

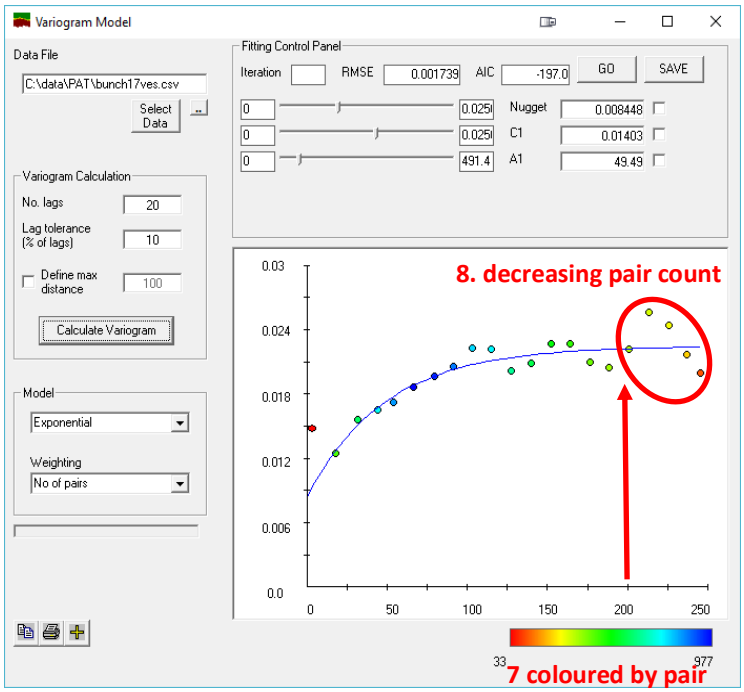




The Variogram

A variogram is a function which describes the variation in a dataset as a function of the distance between the points which comprise that dataset. It is key to the kriging process because the weights used to interpolate a map surface from sample data is a function of the distance between nearby data points and the location of the points for which map values are being interpolated. These weights derive from the variogram. Both Webster and Oliver (2007) and the VESPER Manual (Minasny et al., 2005) provide further information

7. The colour coding in the variogram plot relates to the number of pairs of samples that have been used (or are available) to calculate the 'semivariance' of samples separated by each lag (i.e. distance) class.

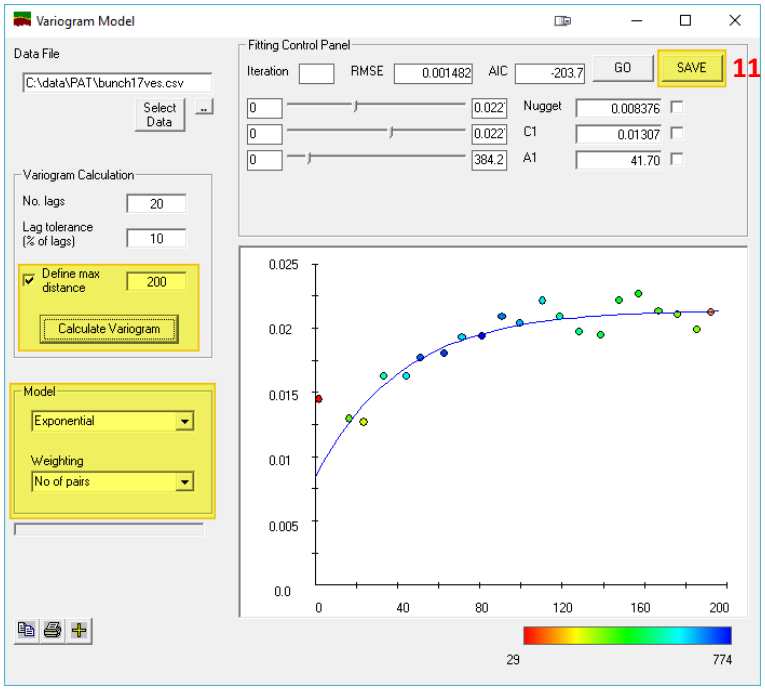


8. In this example, it can be seen that beyond a distance of about 200 m, the number of pairs of points drops off. Because these points are at large lags, and the important part of the variogram is the sloping part at shorter lags, it can often be useful to set the maximum distance. The distance will vary between datasets.

   When a maximum distance is set, it should reflect the drop-in pairs of points and should also be larger than the apparent range of spatial dependence (see Webster and Oliver, 2007 for further discussion of the form of the variogram), and also needs to be sensible in relation to the size of the field being considered. For example, if your vineyard being mapped is a 1 ha square of 100 m side, one would generally expect the maximum distance to be less than 100 m.

9. For this dataset, tick the Define max distance and enter 200 m, then click Calculate Variogram to update the variogram.

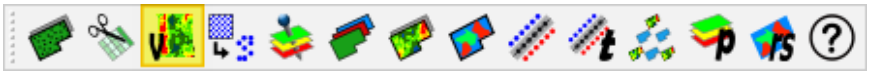
10. To this point, the default settings of an Exponential variogram model have been used with a weighting for the fitting of the curve dependent on the No of pairs. Often, a curve which describes the data a little better might be obtained by selecting a different model and/or using a different weighting method. In general, and with the exception of advanced users, we would recommend that the variogram model selection is confined to either Exponential (the default) or Spherical and that the Weighting used is either No of pairs or No_pairs/st_dev. Further information on settings is available in the VESPER Manual (Minasny et al., 2005).



11. Once satisfied with the fitted variogram, click ***Save*** to save variogram parameters to file. It is useful if the output has the same prefix as the CSV file with the name of the data column you used in step 3 and variogram is added to the end (e.g bunch17ves_berry_variogram.txt). This is the input variogram file for the low density kriging option in ***Run Kriging Using VESPER Tool.***

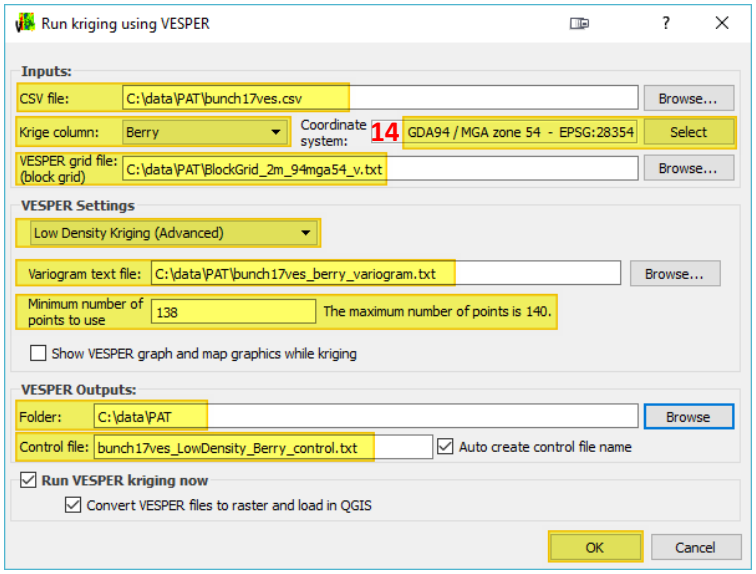
Run Kriging using PAT

12. Close VESPER, launch QGIS and the open then dialog for ***Run Kriging Using VESPER Tool*** from the PAT toolbar or menu. Further details on this dialog can be found section 2.3. The Low Density Kriging implementation is as described below.



13. Select the same ***CSV file*** and ***Krige*** (data) ***column*** used in step 3
14. The ***coordinate system*** will be automatically determined if a column (*EPSG* or *ENEPSG*) exists in the CSV file. If this does not exist then you will be required to set it prior to continuing.

    If your CSV file needs reprojecting, the you can use the ***Clean, Trim, Normalise Point Data Tool*** using the settings described in 2.2.

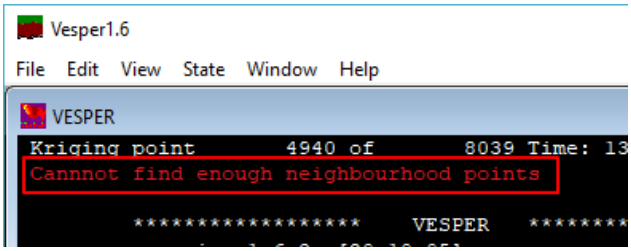


15. Browse to the ***VESPER grid file***.

    At this point a dataset overlap check will be completed if you have provided the CSV and VESPER grid file where it is assumed that these two files have the same coordinates system. If this fails you will be warned as shown. No overlap will cause VESPER kriging to fail.

16. Choose the ***Low Density Kriging (Advanced)*** method***.***
17. Browse for the ***variogram text*** file saved in step 11.
18. Specify the ***output folder*** to store the VESPER files and results in. A sub-folder called VESPER will be automatically created if it does not already exist.
19. If required change the default ***control file name***. This is saved into the **folder** from step 18.
20. Use the default (total point count minus 2) ***minimum number of points to use*** during VESPER kriging.

    In some cases, this will cause a *Cannot find enough neighbourhood points* error in VESPER and create an empty output. If this occurs re-run the kriging by repeating steps 13-21 and decrease the ***minimum number of points to use*** by one and try again . You may have to re-try multiple times to find the right value.



21. Click OK and the control file will be created and added to the VESPER queue. When the VESPER processing is completed, the resulting kriged text file will be converted to a TIF and loaded into QGIS.

5 Contributors

The following individuals have contributed the production of PAT and the underpinning pyprecag Python package, through coding, configuration and expertise in PA, and in the case of the Whole-of-block analysis tool, the development of statistical methodology.

Core Project Team: **Christina Ratcliff** (lead developer), **Rob Bramley** (project leader) and **David Gobbett** (technical lead)

Whole-of-block analysis tool: **Shuvo Bakar**, **Warren Jin** and **Brent Henderson**

pyprecag packaging and publishing: **Andrew Spiers**, **Adrian D'Alessandro, David Benn** and **Daniel Collins**

6 Acknowledgments

CSIRO's development of PAT draws on many years of Precision Agriculture RDE across a range of cropping sectors and variously supported by CSIRO, Wine Australia, the former Cooperative Research Centre for Viticulture, Sugar Research Australia and the Grains Research and Development Corporation. However, the production of the PAT toolset has been sponsored and supported by Wine Australia (Project No. CSA1603) through funding from the Australian Federal Department of Agriculture and Water Resources as part of its Rural R&D for Profit program (Project No. 15-02-018 - Digital Technologies for dynamic management of disease, stress and yield). The input of Andrew Spiers, Adrian D'Alessandro and Daniel Collins was funded through a CSIRO IMT eResearch Collaboration Project. The support of these funders is gratefully acknowledged, as is the assistance of the project advisory panel [Dr Kathy Evans (TIA/University of Tasmania), Colin Hinze (Bird in Hand Winery) and Hans Loder (Penley Estate)].

7 References

Bramley RGV, Hamilton RP. 2005. Understanding variability in winegrape production systems 1. Within vineyard variation in yield over several vintages. *Australian Journal Of Grape And Wine Research* **10**, 32–45. doi:10.1111/j.1755-0238.2004.tb00006.x.

Bramley RGV, Jensen TA. 2014. Sugarcane yield monitoring: A protocol for yield map interpolation and key considerations in the collection of yield data. International Sugar Journal 116, 1–12. doi:http://www.scopus.com/inward/record.url?eid=2-s2.0-84916603197&partnerID=40&md5=f5f44e347b0ac4a1e0d35d95e64ddb27.

Bramley RGV, Kleinlagel B, Ouzman J. 2008. A Protocol for the Construction of Yield Maps From Data Collected Using Commercially Available Grape Yield Monitors - Supplement No. 2. Precision viticulture Cooperative Research Centre for Viticulture, Adelaide 1–4.

Bramley RGV, Lawes RA, Cook SE. 2013. Spatially distributed experimentation: tools for the optimization of targeted management. Chapter 12 in: Oliver MA, Bishop TFA, Marchant BM. (Eds). Precision Agriculture for Sustainability and Environmental Protection. Earthscan, Food and Agriculture Series. Routledge, Abingdon, UK. pp. 205-218.

Bramley RGV, Williams S. 2001. A Protocol for the Construction of Yield Maps From Data Collected Using Commercially Available Grape Yield Monitors. Precision viticulture Cooperative Research Centre for Viticulture, Adelaide 1–4.

Gillies S and others. 2011. Fiona is OGR's neat, nimble, no-nonsense API. Toblerity. https://github.com/Toblerity/Fiona

Gillies S and others. 2013. Rasterio: geospatial raster I/O for Python programmers. Mapbox. https://github.com/mapbox/rasterio

Gitelson AA. 2004. Wide Dynamic Range Vegetation Index for Remote Quantification of Biophysical Characteristics of Vegetation. Journal of Plant Physiology 161, 165–173. doi:10.1078/0176-1617-01176.

Gitelson AA, Viña A, Ciganda V, Rundquist DC, Arkebauer TJ. 2005. Remote estimation of canopy chlorophyll content in crops. Geophysical Research Letters 32, L08403. doi:10.1029/2005GL022688.

Lawes RA, Bramley RGV. 2012. A Simple Method for the Analysis of On-Farm Strip Trials. Agronomy Journal 104, 371-377.

Minasny B, McBratney AB, Whelan BM. 2005. VESPER version 1.62. Aust. Cent. Precis. Agric. McMillan Build. A 5. https://sydney.edu.au/agriculture/pal/software/vesper.shtml.

Panten K, Bramley RGV. 2011. Viticultural experimentation using whole blocks: Evaluation of three floor management options. Australian Journal of Grape and Wine Research 17 136-146.

QGIS Development Team 2016. QGIS Geographic Information System version 2.18. Open Source Geospatial Foundation Project. https://qgis.org

R Core Team 2018. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. http://www.R-project.org/.

Ratcliff C, Gobbett D, Bramley R. 2019. pyprecag - a Python package for the analysis of precision agriculture data. v1. CSIRO. Software Collection. https://doi.org/10.25919/5c731a41954ce

Taylor JA, McBratney AB, Whelan BM. 2007. Establishing management classes for broadacre agricultural production. Agronomy Journal 99, 1366–1376. doi:10.2134/agronj2007.0070.

Webster, R, and Oliver, MA. 2007. Geostatistics for environmental scientists. Wiley & Sons, Chichester, England.

CONTACT US

t 1300 363 400
+61 3 9545 2176
e csiroenquiries@csiro.au
w www.csiro.au
w https://www.csiro.au/en/Research/AF

AT CSIRO, WE DO THE EXTRAORDINARY EVERY DAY

We innovate for tomorrow and help improve today – for our customers, all Australians and the world.
Our innovations contribute billions of dollars to the Australian economy every year. As the largest patent holder in the nation, our vast wealth of intellectual property has led to more than 150 spin-off companies.
With more than 5,000 experts and a burning desire to get things done, we are Australia's catalyst for innovation.
CSIRO. WE IMAGINE. WE COLLABORATE. WE INNOVATE.

FOR FURTHER INFORMATION

CSIRO Agriculture and Food
Christina Ratcliff
t +61 8 8273 8157
e PAT@csiro.au
e christina.ratcliff@csiro.au
w https://people-my.csiro.au/r/c/christina-ratcliff

CSIRO Agriculture and Food
David Gobbett
t +61 8 8303 8741
e david.gobbett@csiro.au
w https://people.csiro.au/g/d/david-gobbett

CSIRO Agriculture and Food
Dr Rob Bramley
t +61 8 8303 8594
e rob.bramley@csiro.au
w https://people.csiro.au/B/R/Rob-Bramley