SCIENCE NOTES

QScout: A QGIS plugin tool suite for georeferencing and analysis of field-scouted and remote sensing data

Joshua R. Evans1,2 | Fernando E. Romero Galvan1 | Kathleen Kanaley1 | Lance Cadle-Davidson1,2 | Kaitlin M. Gold1

1 College of Agriculture and Life Sciences, Cornell AgriTech School of Integrative Plant Sciences, Cornell Univ., Geneva, NY 14456, USA
2 USDA-ARS, Grape Genetics Research Unit, Geneva, NY 14456, USA

Correspondence
Kaitlin M. Gold, College of Agriculture and Life Sciences, Cornell AgriTech, Cornell Univ., Geneva, NY 14853, USA.
Email: kg557@cornell.edu

Funding Information
USDA-ARS Grape Genetics Research Unit CRIS Project 8060-21220-007-00D
NASA JPL Strategic University Research Program: NASA project #80NSSC21K1605

Abstract
Field scouting is an important part of many research methodologies in plant pathology and plant phenomics. However, linking scouting data to field imagery is often hampered by the time-consuming task of georeferencing with a GIS. Here, we present the QScout tool suite for integrating remote sensing imagery and raster data with field-scouting data in QGIS, an open-source GIS program. The central features of QScout are the Drop Pins and Locate Pins plugins, allowing the user to easily link scouted data to remote sensing imagery. QScout also includes the Value Grabber and Grid Aggregator plugins, which transfer raster data into pins and aggregate the data from the pins into a grid, respectively. The final tools, Drop, Grab, and Aggregate and Locate, Grab, and Aggregate, are plugins that combine subsets of the four core plugins. The interface allows GIS users to effectively make use of field-scouted observations with remote imagery and can improve data organization, analysis, and identification of locations of interest for further scouting or targeted management. QScout is publicly available as a GitHub repository: (https://github.com/GoldLabGrapeSPEC/QScout).

1 | INTRODUCTION

Field scouting has a long history of importance in crop production and as a foundational research methodology in many agricultural science disciplines. Field scouting is the act of physically visiting a plot of land and inspecting the plants systematically to assess their health, growth status, or other condition, such as disease (Nutter et al., 1991). Traditionally, field scouting has been the primary method by which growers and scientists collect data on the individual plant level. Field-scouted observations serve as important points of reference and validation in many phenomic investigations (Martinelli et al., 2015; Polgar & Primack, 2011). In contrast to the long history of field scouting, remote sensing has emerged only recently (Martinelli et al., 2015; Zeng et al., 2020). Whereas standard photography techniques sample light at only a few wavelengths, typically red, blue, and green, multispectral imaging, a commonly used type of remote sensing in plant pathology and plant phenomics, consists of sampling light from additional near-infrared wavelengths to yield a more complete absorbance spectrum and understanding of plant condition (Zeng et al., 2020).

The availability and quality of satellite and other types of remote sensing imagery have increased exponentially in recent years. Although kilometer resolutions were once standard, recent developments in constellation architecture (Nagel et al., 2020) have brought satellite imagery resolution to the sub-meter level, with commercial providers offering...
resolutions as fine as 30 to 50 cm, such as the data available from WorldView3 or Planet Labs SkySat (Digital Globe Team, https://gbdxdocs.digitalglobe.com/docs/worldview-3; Planet Team, 2017). For the first time, this allows researchers to analyze satellite datasets at an individual plant level. Much of this analysis is conducted in a GIS program such as the free open-source tool QGIS (QGIS Development Team, 2021). However, the very scale and resolution that make these data so promising can also hamper analysis. Although some software tools exist to easily process and analyze large quantities of multispectral satellite data, such as the built-in QGIS processing toolbox (QGIS Development Team, 2021), GRASS (GRASS Development Team, 2020), and GDAL (GDAL/OGR contributors, 2021), we found that the lack of specific open-source tools for processing and validating imagery with field-scouting data was significantly slowing our workflow. Most significantly, we lacked a tool to easily convert field-scouted observations in spreadsheet form into QGIS points then combine these points into a grid of rectangular polygon geometries.

New tools are needed to aid users in efficiently processing validation data and high volumes of imagery in a standardized format to meet the growing interest in the use of satellite observations in addition to observations from ground-level equipment and unoccupied aerial vehicles in plant phenomics. The integration of ground data with multispectral imagery with a high spatial resolution allows researchers to find correlations between imagery-derived plant health metrics and in vivo plant health. Further, this advancement may aid researchers and extension specialists in helping stakeholders to diagnose and treat problems in their crops more accurately. To more effectively manage, convert, and analyze

Core Ideas

- Linking remote sensing imagery to ground validation data is an essential first step in nearly all use-cases.
- Validation data are frequently not geo-referenced, which makes connecting it to remote sensing data difficult and time consuming.
- We have developed an open source tool that can help users easily link non-georeferenced field data to raster and remote sensing imagery.
- Our tool can improve data organization, analysis, and identification of locations of interest for investigation.

FIGURE 1 Cartoon demonstrating the mechanism of the Drop Pins and Locate Pins algorithms. In both algorithms, the user provides the algorithm with a row vector, a bounding polygon, a starting corner, and values for within and between row spacing. (a) The program calculates the angle between the x axis and the row vector and uses trigonometry to calculate the x and y components of the within- and between-row spacing. (b) The program uses those values to place points above, below, to the right, and to the left of the origin point of the row vector. (c) The program then continues to place points adjacent to existing points, avoiding redundant points and not placing points outside the bounding polygon. (d) The program places points in empty spaces adjacent to existing points. (e) On the 15th iteration, the program will place no additional points because no empty locations adjacent to the existing points remain within the bounding polygon. So, with Δx and Δy referring to coordinate positions in the field ("horizontal" and "vertical"), the trig functions take the angle between the row vector and the line of latitude and use that angle to decompose the row height and point interval (perpendicular and parallel to the row vector respectively) into HΔx, HΔy, VΔx, and VΔy. These vectors are used to place adjacent points. In this example, the row height is 6 ft and the point spacing is 5 ft. Therefore, HΔx = 5 × cos(29) = 4.37 ft; HΔy = 5 × sin(29) = 2.42 ft; VΔx = 6 × cos(29 – 90) = 2.91 ft; and VΔy = 6 × sin(29 – 90) = 5.25 ft
TABLE 1 Example data used as input into Drop Pins: A .csv or .xlsx file containing data such as this is loaded into Drop Pins, and the data in the columns are assigned to the points

| Row | Panel | Vine | DM_incd | DM_severity | Other |
|-----|-------|------|---------|-------------|-------|
| 8   | 1     | 1    | 20      | 1.1         | 0     |
| 8   | 1     | 3    | 0       | 0           | 0     |
| 8   | 2     | 1    | 10      | 0.6         | 0     |
| 8   | 2     | 2    | 15      | 1.5         | 0     |
| 8   | 2     | 2    | 35      | 1           | 5     |
| 8   | 3     | 3    | 20      | 0.8         | 0     |
| 8   | 5     | 1    | 0       | 0           | 0     |
| 8   | 6     | 1    | 15      | 0.6         | 0     |
| 8   | 6     | 2    | 0       | 0           | 0     |
| 8   | 7     | 1    | 0       | 0           | 0     |
| 8   | 7     | 3    | 30      | 1.3         | 0     |

*The Row Panel, and Vine columns identify the location of the plant described in the data row. This example is from a trellised vineyard, in which a panel indicates a linear set of vines (here, three vines) between posts. QScout can accommodate any sort of plant positioning indicator. Unless indicated otherwise (such as here, where we have used “Panel”), QScout will treat data in a column labeled “Column” or “Number” as the absolute position within the row.

**For the plant described in each data row, examples of field scouting data have been provided. The percentage of incidence of downy mildew (DM_incd), the severity of the downy mildew (DM_severity), and the incidence of another plant health issue, such as physical damage (as a percentage). The program can accept any number of attribute columns.

our remote sensing and field-scouted data, we developed the QScout suite of plugins for QGIS. The tool suite applications include routine processes such as locating observations of plants or environmental traits for statistical or spatial analysis in relation to image data, mapping plant locations for repeated observations, and spatially interpolating field data for correlating them with remote sensing-derived plant health indices.

2 | METHODS

2.1 | Sample data

Field scouting data were collected from *Vitis vinifera* L. ‘Chardonnay’ plantings in the Cornell Pathology Vineyards at Crittenden Research Farm in Geneva, NY, on 3 Aug. 2021. The plant attributes collected were disease incidence and severity of grapevine powdery mildew (*Erysyphe necator*), grapevine downy mildew (*Plasmopara viticola*), and miscellaneous plant damage. Incidence was recorded as the percentage of leaves with any disease, and severity was recorded as the percentage of leaf area with visible disease. The row number, panel number, and vine number within the panel were also recorded for each data point.

The satellite terrain raster used in the figures showing the QGIS workflow was captured by SkySat Generation C Planet Labs satellite at 42°52′43″ N, 77°0′57″ W on 2 Aug. 2021. SkySat is a commercial platform of 21 constellation spectroscopic satellites. SkySat imagery captures four regions of the electromagnetic spectrum, and thus the multispectral imagery captures the blue, green, red, and near-infrared regions and a panchromatic image. The spatial resolution of these products is 75 cm for the multispectral measurements and 50 cm for the panchromatic.

2.2 | QScout’s use

The QScout plugin suite can be downloaded from the QGIS plugins dialog. The source code and tutorial are publicly available at https://github.com/GoldLabGrapeSPEC/QScout. For the first use, the tutorial dataset can verify that implementation of the method achieves a standard outcome. QScout was developed on the QGIS (qgis.org) version 3.14.16-Pi platform, running on an Ubuntu 20.04 LTS (Canonical). Python code was written with Pycharm Community 20.3 (JetBrains). Figures were created in GNU Image Manipulation Program (GIMP) version 2.10.22 (gimp.org).

Conceptually, QScout was developed to replace the time-consuming manual processes outlined in Figure 1. QScout plugins can be accessed from the processing toolbox (Ctrl + Alt + T) of QGIS. The *Drop Pins* and *Locate Pins* plugins require the user first to draw a bounding polygon and a row vector. The bounding polygon is a single-feature geometry layer, where the polygon’s sides form the boundary area of the region under analysis. The row vector provides a starting location from which the program will generate points at regular intervals and provides row point orientation. In the case of *Drop Pins*, the user can also provide a spreadsheet file to limit where the pins are dropped and provide data to attach to the dropped pins, in .csv or .xlsx format, as shown in Table 1.

Neither the *Value Grabber* algorithm nor *Grid Aggregator* require a bounding polygon and a row vector. *Value Grabber* requires a point layer and a raster file to run. The *Grid Aggregator* algorithm requires only a layer of point geometries. The grid boundaries produced by the algorithm matches the bounds of the points layer, unless the user specifies bounds in the algorithm’s optional parameters.

2.3 | QScout’s algorithms

The *Drop Pins* and *Locate Pins* algorithms function very similarly. The mechanism by which they identify locations for pins is identical (see Figure 1) but differs in how the pins’ information is used. After the steps shown in Figure 1, the program numbers the row and position values of the points, originating at the starting corner specified by the user. From
FIGURE 2 Cartoon demonstrating the functionality of the Drop Pins algorithm. The user draws a bounding polygon to show the extent of the area and a row vector to provide a direction and anchor point, then uses the Drop Pins plugin to generate a pin geometry layer representing the plants in a plot of land. (a) A view of the QGIS viewport before running Drop Pins. (b) A view of the QGIS viewport after running the Drop Pins algorithm without a data file provided. If the user has provided a data file, pins are dropped only on locations specified in the data file; otherwise, pins are dropped throughout the entire designated area. (c) A view of the QGIS viewport after running Drop Pins with a data file provided. Pins have been dropped on all locations where a plant is predicted to be present.

Here, the behaviors of Drop Pins and Locate Pins diverge. In both cases, the points placed by this stage of the algorithm are used as inputs for further computations; not all of the points are necessarily created as QGIS point geometries.

Drop Pins: After the pins’ locations have been identified, the Drop Pins algorithm numbers the row and position of each pin from the starting corner specified by the user. If the user has not provided a data file or has checked the “Drop Data-Less Points” box, the program will create QGIS features for all the points calculated by the method shown in Figure 1; otherwise, the program will only create QGIS features for points with data.

Locate Pins: After the points’ locations have been identified, the program will take each point in the input layer and calculate which point location is nearest to that input point. The program will then create a copy of the input point in a new geometry layer and assign it attributes for the row number and location within the row.

Grid Aggregator: The program first creates a grid using the parameters of grid cell width, grid cell height, and (optionally) grid extent provided by the user (if the grid extent is not provided, the extent of the points layer will be used). The program then creates a list of the points contained within each grid cell. The program then goes through each attribute of the
point layer and applies the aggregation function to the list of
points created for each grid cell, applies the aggregation func-
tion to the attribute for each point, and assigns the aggregation
function output to the grid cells’ feature.

3 | RESULTS AND DISCUSSION

The QScout algorithms described here for implementation in
QGIS are focused on processing and analyzing field-scouting
data with aerial or satellite remote sensing imagery. The
Drop Pins algorithm creates a grid of point geometries, with
optional attributes taken from a spreadsheet file (The GIMP
Development Team, 2019). The Locate Pins algorithm takes a
set of points, duplicates them, and assigns them row and col-
umn index attributes within a local user-defined coordinate
system. The Value Grabber algorithm takes a set of points and
a raster and assigns each point the attributes corresponding
to the raster band values at that point. The Grid Aggregator
algorithm takes a set of points and overlays a grid on top of
them, assigning each grid cell a set of attributes that are the
aggregates (e.g., mean, median, sum, SD, etc.) of the points
within that cell. The Drop, Grab, and Aggregate algorithm
applies Drop Pins, followed by Value Grabber, followed by
Grid Aggregator. Similarly, the Locate, Grab, and Aggregate
algorithm applies Locate Pins, followed by Value Grab-
ner, followed by Grid Aggregator. With these algorithms,
QScout allows users to efficiently process high volumes of
non-geotagged scouting data and imagery in a format that
can be easily incorporated into analyses, modeling, and down-
stream decision-making.

3.1 | The Drop Pins algorithm

The Drop Pins workflow begins with a data spreadsheet file in
.csv or .xlsx format (as illustrated in Table 1) and an approxi-
mate understanding of the geography of the plot of land where
the data were collected. The user begins by drawing a bound-
ing polygon around the plot of land in QGIS to indicate where
pins should be dropped. Within the bounding polygon, the
user then draws a row vector, indicating the ordinal direction
of the rows, and a reference point for the location of plants.
The user then opens the Drop Pins graphic user interface from
the QGIS processing plugins panel and sets the Bounding
Polygon, Row Vector, and Input Data to the appropriate val-
ues. The user also specifies the row spacing and the interval
between plants within a row with the Row Spacing and Point
Interval parameters. If the user does not provide the algorithm
with a data file, the algorithm will drop a pin at every point
in the field where it thinks a plant will be and will assign row
and plant number attributes to those fields (see Figure 2).
FIGURE 4 The mechanism of the Value Grabber algorithm. The circles at the top of the figure represent pin geometries, and each grid represents a raster band. The pin geometries are copied and the values of the raster pixels occupying the same location as each pin are set as the attributes of the pins, as shown at the top of the figure.

Once the data have been transferred from the spreadsheet to the point layer, the user can easily perform any number of geographic analyses on them. For the purposes of this article, the focus has been placed on an analysis comparing scouting data with satellite raster datasets. Using Drop Pins and a datasheet, QScout can drop pins on a field at a close approximation of the individual plants’ locations, allowing the user to georeference plants and the corresponding scouting data easily. When used without a datasheet, the Drop Pins functionality can be used to collect a random sample of points for downstream analyses or to identify all individual plants within a field, or as a foundation for using other tools such as Grid Aggregator to obtain individual plant raster values.

3.2 The Locate Pins algorithm

The Locate Pins algorithm estimates the location of each pin in a field in terms of the row number and plant number within a row, or some other specified designator (such as “panel” for trellised crops). It takes the same set of inputs as the Drop Pins algorithm, but with a pin geometry layer instead of a data spreadsheet, and it produces a duplicate of the pin geometry layer provided, with plant and row number attributes added to each pin (see Figure 3). Thus, when the user starts with previously geotagged or dropped pins in a field, the Locate Pins function can identify plant locations in a manner that facilitates easy communication with field workers. This allows the user to see how the scouted data correspond to satellite-derived plant health assessments and to use and/or return that information to individuals on the ground for making decisions about crop management.

3.3 The Value Grabber algorithm

The Value Grabber algorithm transfers values from a raster layer to a pin geometry layer (see Figure 4). The user provides the algorithm with a pin geometry layer and a raster layer, and the algorithm produces a duplicate of the pin geometry layer with attributes added from the raster file. The raster value attributes for each pin are set to the value of the raster pixel at the location of the pin. The algorithm also has more advanced optional parameters that allow the user to grab values from a circular area around the pin, with customizable options for how multiple pixel values should be weighted and combined. The user also has the option to specify their own grab function by passing the algorithm a Python script. Value Grabber is similar to the Sample raster values plugin that is packed with QGIS (QGIS Development Team, 2021), but can be used on a raster that is not a QGIS layer, allowing for the analysis of large rasters that may cause issues for QGIS on less powerful machines. In addition, Value Grabber allows for greater user specificity in the method of sampling through the optional Grab Radius and Grab Function parameters. The functionality of Value Grabber allows the user to easily export data files for downstream analysis, create summary statistics, and share plant health data with stakeholders.

3.4 The Grid Aggregator algorithm

The Grid Aggregator algorithm functions as the near reverse of the Value Grabber algorithm (see Figure 5). Whereas the
FIGURE 5  Mechanics of Grid Aggregator. The algorithm creates a grid of rectangular polygon geometries over the points layer (or an area specified by the user) with the user-specified cell width and height, then assigns values to each grid cell based on the values of the points within that cell, using one of the specified aggregation functions or a custom aggregation function provided by the user. In this figure, the aggregation function used is the “mean”, so the value of each cell (large text) is set to the average of all the points within the cell (small text).

FIGURE 6  (a) The QGIS viewport after running Drop Pins for six rows of the Cornell Pathology Vineyards at Crittenden Research Farm in Geneva, NY. The blue box is the user-defined bounding polygon; the red line (lower left corner) is the user-defined row vector. The points were generated by Drop Pins. Each point corresponds to a panel of four vines. (b) Attribute table for the points dropped by Drop Pins. These features include (from left to right): row, panel number, treatment, and percentage of symptomatic leaf area for four disease categories. The highlighted entry corresponds to the yellow point in the lower right corner of Panel a.
Value Grabber algorithm transfers data from a raster grid to points, the Grid Aggregator algorithm transfers data from points to a grid. The user provides a points layer and width and height values for the grid cells and specifies which attributes of the points’ features to aggregate. The algorithm produces a vector layer of consecutive rectangular polygon geometries forming a grid aligned north–south–east–west-aligned grid. Each rectangle, which represents a grid cell, is assigned the attributes of all the pins within the grid cell. The user is able to decide whether to set the values to the mean, median, minimum and maximum, sum, SD, or weighted mean of the pin values. The user also has the option to specify their own aggregation function by passing the algorithm a Python script. Varying the size of the grid allows the user to accommodate datasets at different resolutions and to determine which resolution is best suited for constructing a model for their variables of choice.

In addition to the QGIS polygon geometry layer, the user can enter the name of an excel file, to which Grid Aggregator will write the data. This function makes it easy to aggregate ground-scouted information (e.g., disease severity or incidence) to the same spatial scale as the imagery (see Figure 6).

3.5 Combination algorithms

The combination algorithm Drop, Grab, and Aggregate allows the user to quickly georeference scouting data, combine it with raster data, and aggregate the points into a grid for easy analysis. Similarly, the combination algorithm Locate, Grab, and Aggregate makes it easy for the user to identify areas of poor plant health for targeted scouting by field workers. In both cases, the same parameters are inputted, and the function is the same as in the individual algorithms, but they are contained in the same interface.

4 CONCLUSION

With the growing interest in agricultural remote sensing, it is more important now than ever to provide users with tools to link these data with on-the-ground validation in order to provide a full analysis of agricultural systems on multiple scales and to look for biologically relevant patterns. QScout provides a tool suite that is appropriate for GIS users to georeference scouting data, determine the location of georeferenced data relative to a larger layout, combine scouting data with remote sensing data, and form geometric vector data into formats that are easier to process. As remote sensing datasets— particularly high-resolution multispectral and hyperspectral datasets—continue to proliferate in the coming years, tools such as QScout and other QGIS plugins will provide an invaluable organizational aid for plant pathologists, plant phenomicists, and others in the agricultural sciences.

ACKNOWLEDGMENTS

Funding for the QScout development was provided by USDA-ARS Grape Genetics Research Unit CRIS project 8060-21220-007-00D. Support for Fernando E. Romero Galvan comes from NASA JPL Strategic University Research Program and NASA project #80NSSC21K1605. We thank Graham Trolley, Nikita Gambhir, and Shivranjani Baruah for their work as early beta-testers of the program. David Combs for providing scouting data, and the rest of the members of the Grape Sensing, Pathology, and Extension Lab at Cornell University (GrapeSPEC) for their support and encouragement. We thank Terry Bates and Nicholas Gunner for their advice and feedback on plugin functionality.

The QScout plugin suite can be downloaded from the QGIS plugins dialog. The source code and a tutorial are publicly available at https://github.com/GoldLabGrapeSPEC/QScout.

AUTHOR CONTRIBUTIONS

Joshua R. Evans: conceptualization, formal analysis, investigation, methodology, project administration, resources, software, visualization, writing—original draft, writing—review and editing. Fernando E. Romero Galvan: conceptualization, data curation, validation, visualization, writing—review and editing. Kathleen Kanaley: visualization. Lance Cadle-Davidson: funding acquisition, resources, supervision, writing—review and editing. Kaitlin M. Gold: conceptualization, investigation, project administration, resources, supervision, writing—review and editing.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

ORCID

Joshua R. Evans https://orcid.org/0000-0001-7961-7676
Lance Cadle-Davidson https://orcid.org/0000-0003-4241-4634
Kaitlin M. Gold https://orcid.org/0000-0002-0306-9244

REFERENCES

GDAL/OGR contributors. (2021). GDAL/OGR geospatial data abstraction software library. Open Source Geospatial Foundation. https://gdal.org

GRASS Development Team. (2020). Geographic Resources Analysis Support System (GRASS GIS) software, Version 7.8. Open Source Geospatial Foundation. http://grass.osgeo.org

Martinelli, F., Scalenghe, R., Davino, S., Panno, S., Scuderi, G., Ruisi, P., Villa, P., Stroppiana, D., Boschetti, M., Goulart, L. R., Davis, C. E., & Dandekar, A. M. (2015). Advanced methods of plant disease detection. A review. Agronomy for Sustainable Development, 35(1), 1–25. https://doi.org/10.1007/s13593-014-0246-1

Nagel, G. W., Novo, E. M. L. De M., & Kempel, M. (2020). Nanosatellites applied to optical Earth observation: A review. Ambiente e Agua—An Interdisciplinary Journal of Applied Science, 15(3), 1. https://doi.org/10.4136/ambi-agua.2513
Nutter, F. W., Jr., Teng, P. S., & Shokes, F. M. (1991). Disease assessment terms and concepts. *Plant Disease, 75*, 1187–1188.

Planet Team. (2017). *Planet application program interface: In space for life on Earth*. https://api.planet.com

Polgar, C. A., & Primack, R. B. (2011). Leaf-out phenology of temperate woody plants: From trees to ecosystems. *New Phytologist, 191*(4), 926–941. https://doi.org/10.1111/j.1469-8137.2011.03803.x

QGIS Development Team. (2021). *QGIS geographic information system*. QGIS Association. https://www.qgis.org/

The GIMP Development Team. (2019). *GIMP (2.10.12)* [Computer software]. GIMP. https://www.gimp.org

Zeng, L., Wardlow, B. D., Xiang, D., Hu, S., & Li, D. (2020). A review of vegetation phenological metrics extraction using time-series, multispectral satellite data. *Remote Sensing of Environment, 237*, 111511. https://doi.org/10.1016/j.rse.2019.111511

How to cite this article: Evans, J. R., Romero Galvin, F. E., Kanaley, K., Cadle-Davidson, L., & Gold, K. M. (2022). QScout: A QGIS plugin tool suite for georeferencing and analysis of field-scouted and remote sensing data. *Plant Phenome Journal, 5*: e20031. https://doi.org/10.1002/ppj2.20031