

## SOFTWARE METAPAPER

# Vespucci: A Free, Cross-Platform Tool for Spectroscopic Data Analysis and Imaging

Daniel P. Foose<sup>1</sup> and Ioana E. P. Sizemore<sup>1</sup>

<sup>1</sup> Department of Chemistry, Wright State University, Dayton, Ohio, United States  
foose.3@wright.edu, ioana.pavel@wright.edu

Vespucci is a software application developed for imaging and analysis of hyperspectral datasets. Vespucci offers several advantages over other software packages, including a simple user interface with a small learning curve, no cost, and less restrictive licensing. Vespucci expands several analysis techniques including univariate imaging, principal components analysis, partial-least-squares regression, and vertex components analysis with endmember extraction, and k-means clustering. Additionally, Vespucci can perform a number of useful data-processing operations, including filtering, normalization, baseline correction, and background subtraction. Datasets that consist of spatial or temporal data with a corresponding digital signal, including spectroscopic images, mass spectrometric images, and X-ray diffraction data can be processed in this software. A few use cases for Raman and surface-enhanced Raman spectroscopies are provided. Vespucci is written in C++ and makes use of the MLPACK [3], Armadillo [9], Qt, and QCustomPlot libraries. Vespucci is a graphically-driven package that is designed with ease-of-use in mind and is equally capable to other available tools. Vespucci's capabilities are extended by interfaces to Octave and R to allow existing research code to be run from a common environment. Additionally, Vespucci's C++ classes can be used to construct more specialized programs when an application programming interface (API) is desired. The source code and a Windows binary distribution can be accessed at <https://github.com/dpfoose/Vespucci>.

**Keywords:** Spectroscopy; Chemometrics; Raman Spectroscopy; Infrared Spectroscopy; Biomedical Imaging; Spectroscopic Imaging; Univariate Analysis; Multivariate Analysis; Data Processing

## (1) Overview

### Introduction

#### Motivation

The main goal of this research was to develop a free, cross-platform tool for spectroscopic mapping and analysis, entitled Vespucci after the Renaissance cartographer Amerigo Vespucci. Vespucci offers several main advantages in comparison with other available instrument software and chemometrics packages such as Solo from Eigenvector Research, the scikit-spectra Python library, and the hyperSpec and chemoSpec R packages.

**Licensing of commercial products.** The restrictive licensing of numerous proprietary instrument software and chemometrics packages precludes the use of the software on devices owned by individual researchers without the purchase of an additional license. The expense and availability may make the implementation of advanced analysis techniques inaccessible to researchers. By releasing this software on the Internet at no cost, with no proprietary dependencies, barriers to use due to software licensing are removed [6].

**Ease-of-use.** Existing software packages for spectroscopic data analysis are generally written with advanced users in mind. These packages come as a library of functions, which must be called from a command-line interface. This interface affords a great deal of customization at the expense of ease-of-use for less advanced users. Vespucci is driven by a graphical user interface (GUI) that is intuitive to use even by beginners. No programming knowledge is necessary to use the software, but extensions written in Octave, which is mostly code-compatible with MATLAB®, and R may be used by more advanced users.

#### Supported Methods

##### Data Processing

Vespucci is capable of several of the most common data pre-processing techniques in chemometrics, as described below.

**Smoothing.** Vespucci supports a number of smoothing methods including moving average filters, median filters, Savitzky-Golay smoothing and Whittaker smoothing.

**Data selection.** Individual spectra are easily viewed and manipulated. Spectra may be removed by threshold to reject clipped or poorly-focused spectra. Spectral data beyond a certain spatial range in a spectroscopic image may also be removed.

**Normalisation.** Vespucci supports min/max normalisation (subtraction of the minimum of each spectra followed by division by the maximum), unit area normalisation (dividing each spectra by its sum), standard normal variate normalisation, Z-score normalisation, normalization by the maximum intensity at a particular spectral abscissa range, scaling by a particular number, vector normalisation of each spectrum vector, and mean-centring.

#### Analysis and Imaging

Vespucci supports a variety of powerful and commonly-used methods for spectral data analysis, as illustrated below.

**Univariate.** Vespucci is capable of univariate analysis and imaging by peak intensity, peak area, and peak width (estimated full-width-at-half maximum), area ratio between two peaks, and intensity ratio between two peaks.

**Multivariate.** Vespucci is capable of classical principal component analysis (PCA), a widely-used method for dimension reduction, both using the singular value decomposition, and partial-least-squares (PLS) regression (Vespucci, like MATLAB, uses the SIMPLS algorithm for PLS regression [4]). Vespucci is capable of Vertex Component Analysis (VCA), an algorithm for dimension reduction and endmember extraction [8]. This algorithm finds the spectra in the dataset that are responsible for the most variance. Images can be created from the component scores generated by both methods.

**Peak detection.** Vespucci uses a peak-finding method based on convolution with a Mexican hat kernel, mathematically identical to the continuous wavelet transform (CWT). This method was first developed by Du for analysing proteomic mass spectrometry data [5], and was then applied to Raman spectroscopy by Zhang [12]. It uses a signal smoothed by Mexican hat kernels of varying width to determine the peak centres of the unsmoothed data. This facilitates the determination of the local extrema with smooth signals. A “chemical barcode” can then be constructed from these results, allowing the researcher to identify spectral signatures within the data and to determine which spectra contain which peaks of interest.

**Imaging.** Vespucci utilizes several colour scales for different purposes. The traditional, “rainbow” colour scale is implemented, along with a rainbow colour scale (“VespucciSpectral”) that runs from black to white. Vespucci implements modified versions of the ColorBrewer [2] colour scales, which are designed to increase linearly in perceived luminosity and overcome problems identified with the rainbow colour scales [1]. Scale bars of arbitrary size can be added, and images can be saved in both vector (PDF, SVG) and raster (TIFF, BMP, PNG, GIF, JPEG) formats.

#### External Code Interface

Code written in R, Octave, and most code written in MATLAB® can be executed from Vespucci using the external code interface. This allows researchers to use

existing codebases with the software. Vespucci depends on an installation of R to execute code on objects stored in shared memory. Vespucci is distributed with a copy of the Octave interpreter as a DLL file. All data objects associated with each dataset can be exposed to the interface at the discretion of the user.

#### User Interface

Vespucci is designed so that a user with an understanding of basic GUI paradigms can easily utilize the software (**Figure 1**). The main window of the program consists of two panes. The left pane consists of a list of datasets on which operations can be performed. The right pane is a list of images created through the various available imaging techniques.

Vespucci supports the two most common ASCII (\*.txt) formats: the “wide text” format in which each row of the file represents a spectrum, and the “long text” format in which spectra are concatenated sequentially. Data import is handled by a simple dialog. Because text files do not contain metadata about the abscissa or ordinate labels, the user can specify labels in the import dialog.

#### Applications

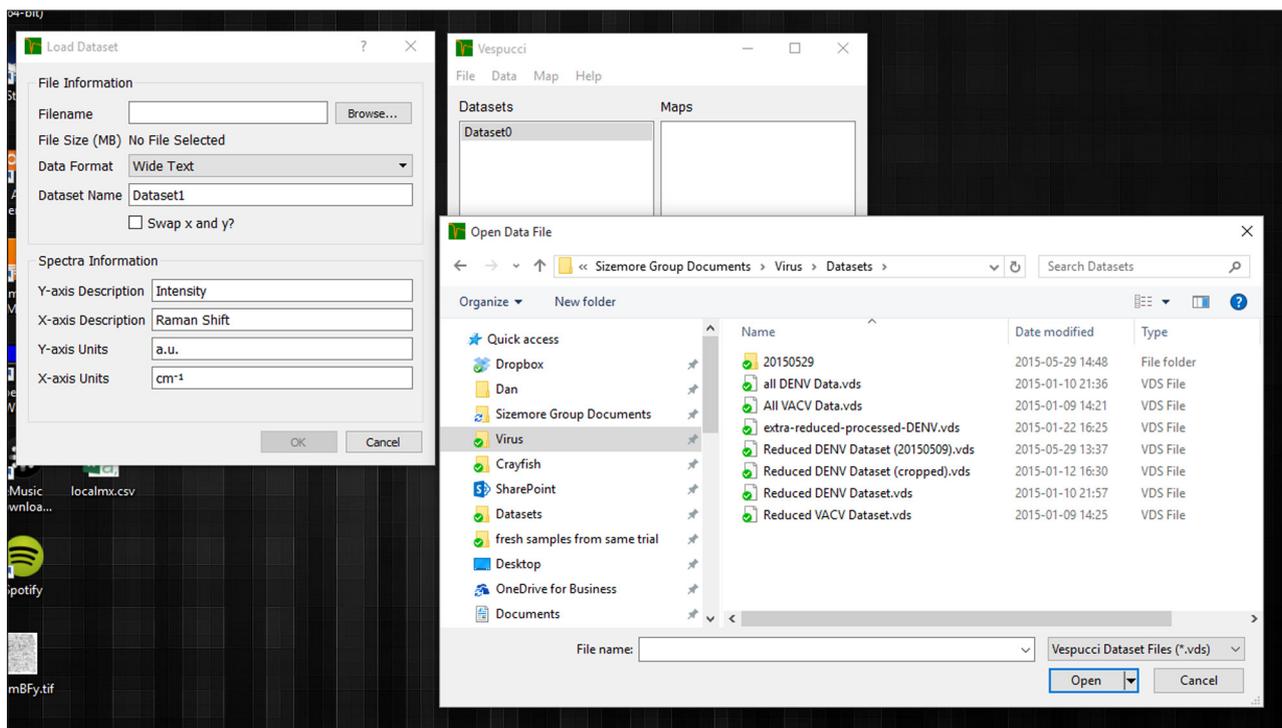
Vespucci is capable of handling a wide variety of spectroscopic data, including infrared spectroscopy, ultraviolet-visible spectroscopy and frequency-domain terahertz spectroscopy, with spatial or temporal metadata. The use of Vespucci for several tasks in the analysis of Raman and surface-enhanced Raman spectroscopic data has been demonstrated below.

##### Surface-Enhanced Raman Spectroscopy (SERS)

The utility of Vespucci for several SERS studies has been shown in unpublished works and several replications of previous works. SERS is a Raman spectroscopic technique that utilises plasmonic nanomaterials, such as silver and gold nanoparticles, to enhance the intensity of Raman signals for the qualitative and quantitative determination of analytes at trace concentrations [7].

**Probing nanoparticle-virion interactions.** Spectroscopic data analysis is primarily concerned with the extraction of chemical information based on the presence and profile of “peaks” in spectra. When a dataset is very large, manual peak finding and determination become time consuming and are prone to human error. Automatic evaluation of peaks in vibrational spectroscopy, with non-linear baselines and substantially broadened peaks, can be difficult. A rigorous, baseline-independent method to determine peaks of varying widths is needed in these cases. The “CWT” method fits these criteria [12], but was previously only available as a command-line R package. Vespucci provides the first implementation of this algorithm with a graphical user interface.

Vespucci’s peak detection methods have been used to determine spectral regions of interest for subsequent analysis. The goal of this work is to use the chemical information found in the peaks to determine the chemical environment inhabited by AgNPs when interacting with virions. *Dengue virus* samples were incubated with AgNPs then inactivated and deposited on glass slides. All Raman signals were smoothed with a median filter of window

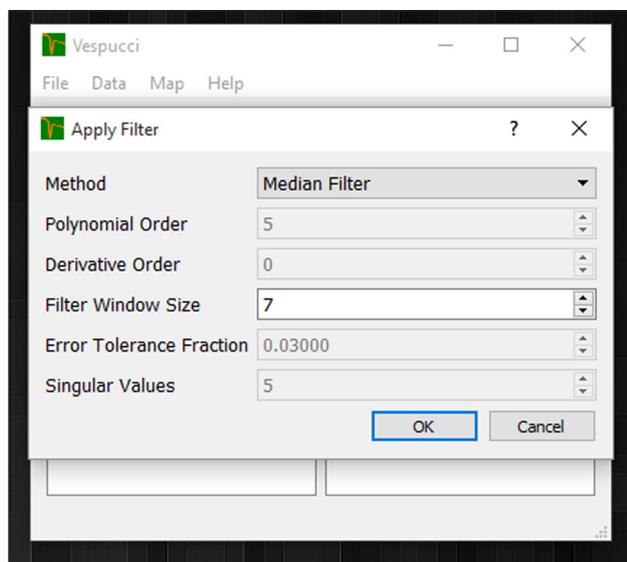


**Figure 1:** The Vespucci main window and file import dialog. Vespucci uses the user's native file dialog to select the file for import. Metadata can be specified by the user.

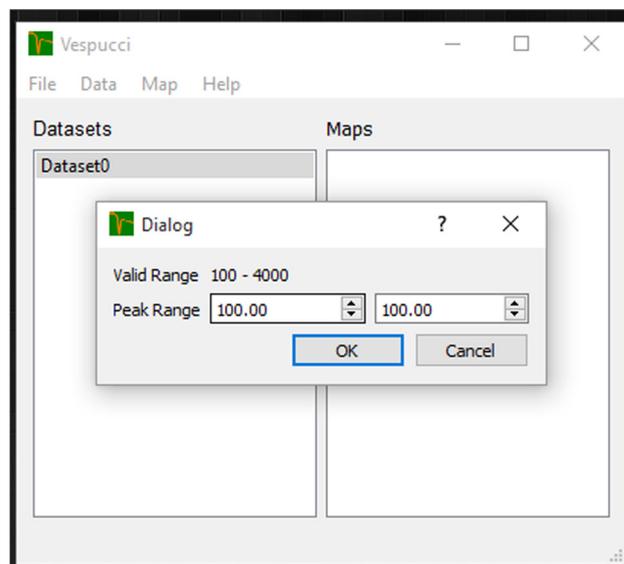
size 7 (**Figure 2**) and normalized to the glass signal centred near 2600  $\text{cm}^{-1}$  (**Figure 3**). Spectra of glass slides without sample were recorded as a control measurement. The average glass spectrum was fitted with a Voigt curve in Origin 8.0. The glass signal fit was subtracted from all samples, and those samples whose maximum signal intensity was less than one half the intensity of the glass spectrum were removed. The CWT-based peak finding method (**Figure 4**) was then applied to produce a “chemical barcode”, a bar graph plotting the total detected peak centres against wavenumber (**Figure 5**). The general peak regions corresponding to each peak were then determined by estimating the width of the peaks in the bar graph. Vespucci

provides, for the first time, an automatic and reproducible system for determining potential spectral regions of interest for the interaction of AgNPs and virions. Pre-processing and subsequent analysis were completed with only a few clicks of the mouse. This approach may also be applicable for other nanobiological studies (*e.g.*, cell-nanoparticle and biomatrix-nanoparticle interactions) when analyte concentrations are very low.

**SERS substrate assessment.** The univariate imaging features of Vespucci can be used to assess the SERS capabilities of silver nanorods (AgNRs). AgNRs were constructed by the vapour deposition of silver onto silicon platforms at two different temperatures (100 and 300 K).



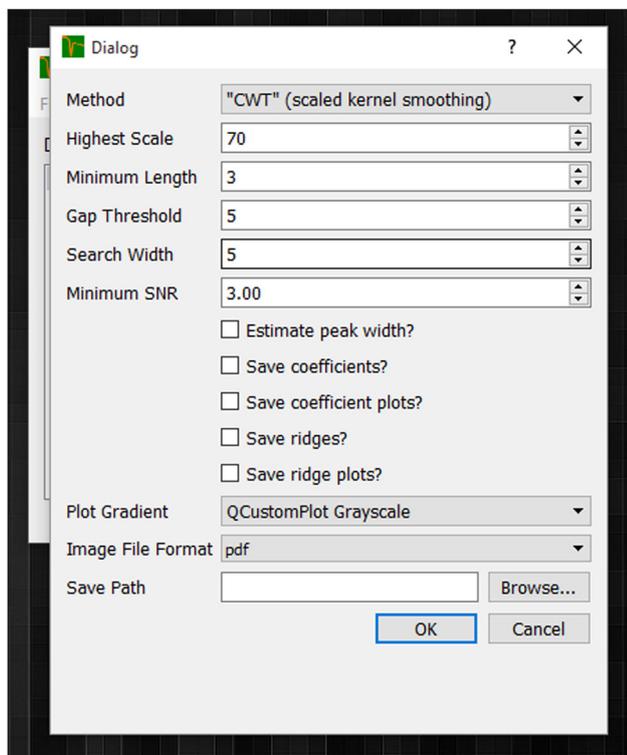
**Figure 2:** The Vespucci dialog used to perform smoothing.



**Figure 3:** The Vespucci dialog used to perform normalization by peak intensity.

The resulting SERS substrates were then exposed to a solution of rhodamine-6G (R6G) as a test probe and imaged. In addition to determining the spatial distribution of

SERS “hotspots” (areas where R6G may experience higher enhancement) on each substrate, the two samples could be easily compared by using a common colour scale between them. The data was processed with a median filter of window size 7, followed by standard normal variate normalization. Univariate images of the Riemann sum of the spectral region from 1625 to 1675  $\text{cm}^{-1}$ , corresponding to a xanthine breathing marker mode (Figures 6 and 8), were compared to determine differences in overall enhancement. The values were larger for the substrates synthesized at 100 K than for the substrates synthesized at 300 K (Figure 7), because the colder temperature provided favourable kinetics to produce denser and better-aligned AgNR surfaces [10].

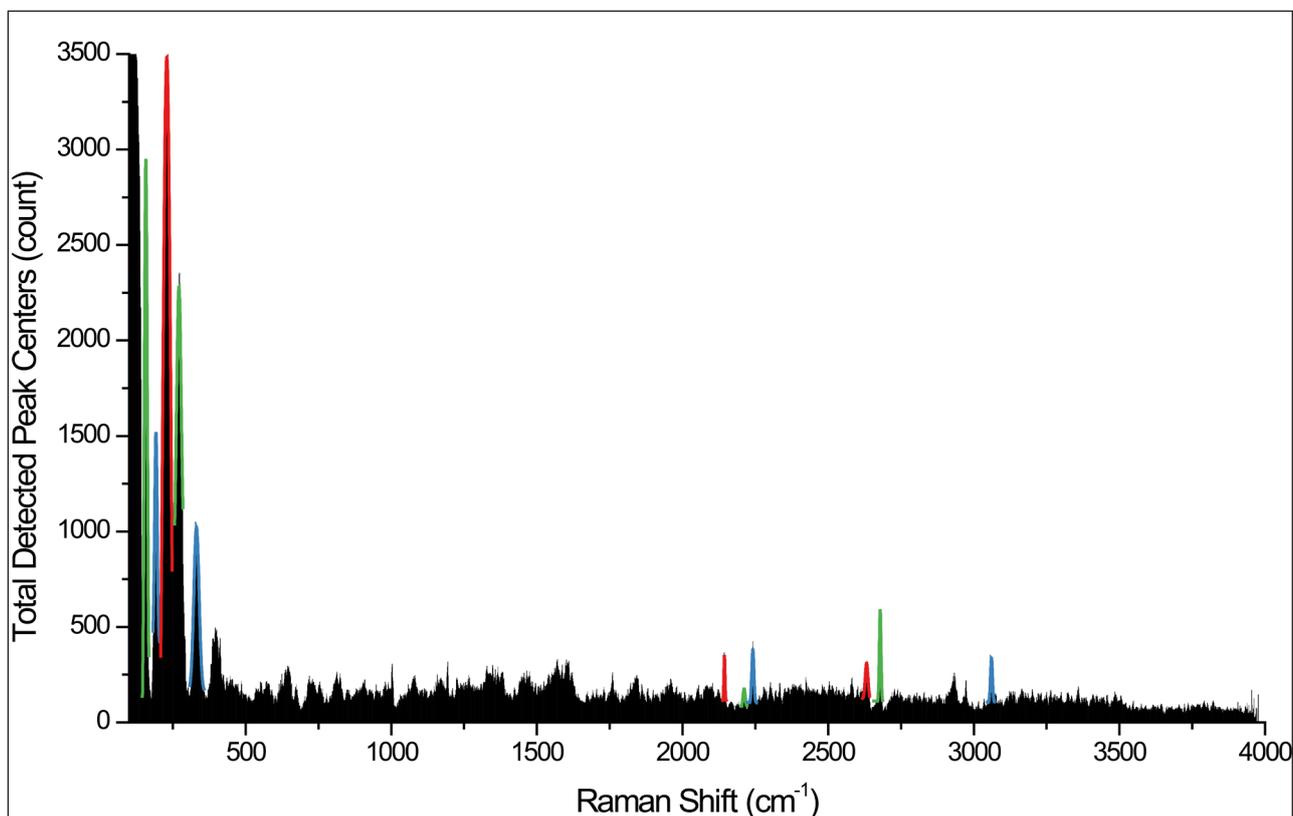


**Figure 4:** The Vespucci dialog for performing CWT peak detection. Vespucci includes the first implementation of the CWT peak detection algorithm with a graphical user interface.

#### Raman Spectroscopy

**Bone mineralisation.** Bone consists of both organic (bone marrow, collagen) and inorganic (hydroxyapatite) components. While increase in inorganic components and subsequent crystallisation corresponds to bone strengthening, abnormal bone mineralisation interferes with vascularization, inhibiting growth. The intensity of spectral signatures corresponding to inorganic moieties can be used to assess the degree of mineralisation of biomaterials. A toxicological study was devised to examine the interaction between platinum group metal (PGM) salts and developing chick embryos [11]. This work utilises the data processing and univariate analysis features of Vespucci to demonstrate how PGM salts interfere with the mineralisation process.

Raman spectral scans were performed on slices of chick embryo tibiotarsi. Embryos were exposed to PGM solutions *in ova* on the 7<sup>th</sup> and 11<sup>th</sup> day of incubation in order

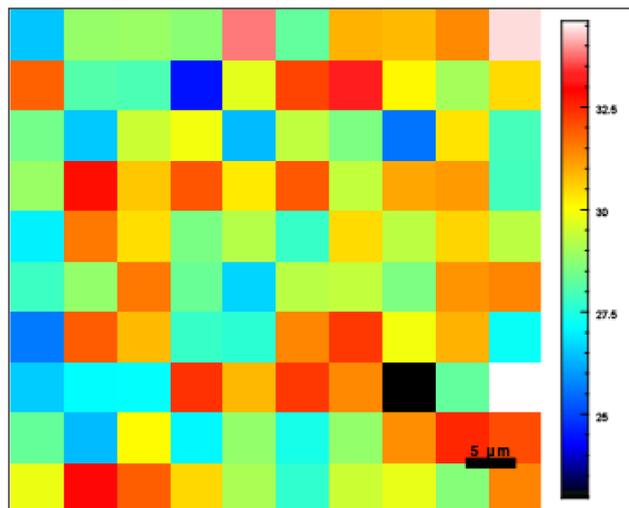


**Figure 5:** The “chemical barcode” produced by applying the peak finding method to all spectra in the dataset.

to study potential bone structural changes due to exposure to PGMs. Crystallisation was assessed and spatially determined by observing intensity near the  $\nu_1$  band of phosphate (search range 946–976  $\text{cm}^{-1}$ ). The crystalline structure can be observed (**Figure 9**) in multiple colour scales.

#### Use of C++ API

The Vespucci C++ API allows for the creation of specialty programs to perform the same task on multiple datasets. Here, a simple program to pre-process and perform VCA



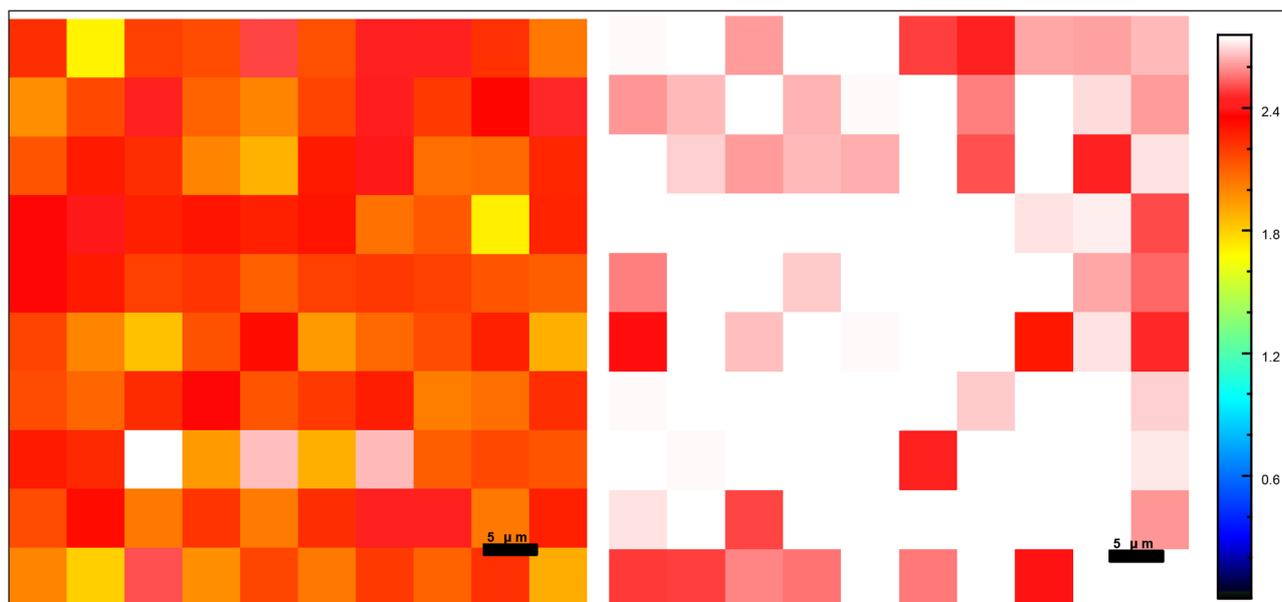
**Figure 6:** Using Vespucci to examine the spatial distribution of SERS hotspots. The colour scale corresponds to the Riemann sum of the xanthine breathing-mode peak. Spots higher on the colour scale (e.g., white or red) indicate better surface-enhancement capability.

of all datasets in a particular folder is demonstrated in the Examples folder in the source tree. The procedural maths library, combined with a few Qt classes, was used to automate the data analysis and pre-processing workflow.

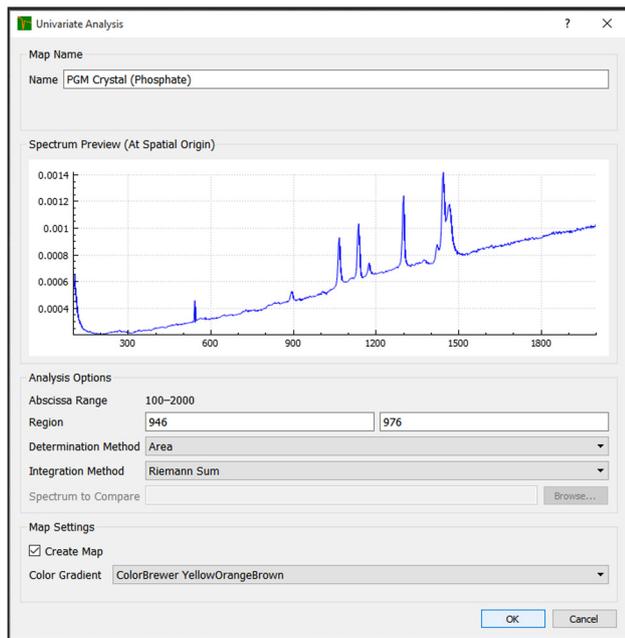
#### Implementation and architecture

In Vespucci, datasets are stored as `VespucciDataset` objects, which contain an Armadillo matrix containing the spectra as columns, metadata (including spatial or temporal coordinates and the spectral abscissa) and associated processing and analysis methods. Math functions are handled by the `VespucciMath` namespace, which contains basic algorithms for the analysis and processing methods (some simple processing methods are handled in the `VespucciDataset` class). Data import is handled by the `TextImport` and `BinaryInput` namespaces. `VespucciMath` can thus also be used as a procedural API for dealing with spectroscopic data files. The output of analysis methods are stored in `AnalysisResults` objects (or in `MLPACKPCADData`, `PLSDData`, `PrincipalComponentsData`, `UnivariateData` or `VCADData` objects), which are heap allocated using smart pointers and accessed through the `VespucciDataset` parent object. The GUI form classes interact with datasets entirely through smart pointers to `VespucciDataset` objects, which are managed through a `VespucciWorkspace` object that contains information about the operating environment and the currently open datasets.

R integration is handled through the `RInside` library. A local instance of R is created, and the variables requested by the user to send to the environment are added. The code specified by the user is then executed, and the variables requested by the user are returned to Vespucci, either



**Figure 7:** A comparison of the surface-enhancing capabilities of two AgNR synthesis methods in Vespucci. The image on the left is from an AgNR SERS substrate synthesized in cryogenic conditions. On the right, an AgNR substrate synthesized in ambient conditions. The colour scale is mapped to the Riemann sum of the spectral region from 1625 to 1675  $\text{cm}^{-1}$ . The presence of more white-coloured spots in the right spectral map indicates an overall improvement in enhancement compared to the left spectral map.



**Figure 8:** The univariate imaging dialog.

to the `VespucciDataset` object (in the case of spectral or abscissa data) or to a new `AnalysisResult` object.

Octave integration is handled through the Octave C++ API via a helper process, using the Boost Interprocess Communication library (this is due to the fact that Octave for Windows is distributed only as 32-bit binaries). A shared memory object consisting of copies of the variables requested by the user is created by Vespucci then accessed by the helper process, which creates Octave objects from

the memory used by the Armadillo objects, writes an Octave function containing the code specified by the user, then calls the function. The variables requested by the user are then converted back into Armadillo objects and placed in a shared memory object that is accessed by Vespucci, which creates `AnalysisResult` objects or modifies members of the `VespucciDataset` object.

### Quality control

Vespucci's output was compared to the output of existing packages for the included methods (MATLAB, R and Octave implementations), and found to be identical (with the exception of the VCA method, where initial endmembers are selected at random). Unit tests for the mathematics namespace are included in the source tree. Vespucci has been tested for ease-of-use and stability with favourable results.

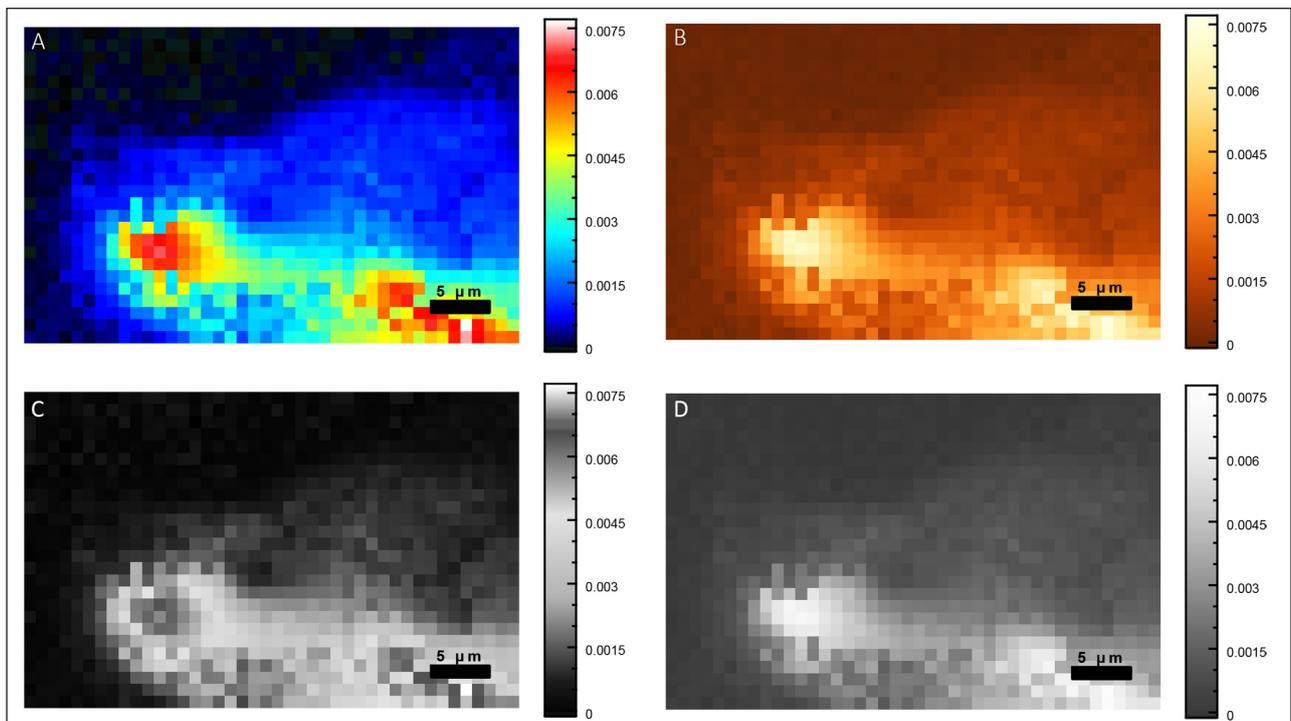
## (2) Availability

### Operating system

Vespucci has been tested on Windows 7, Windows 8, Windows 10, and Ubuntu 15.04 ("Vivid Vervet"). The Windows binary distribution is only compatible with 64-bit versions of Windows. Vespucci depends only on cross-platform libraries, so compilation on other Unix-like operating systems, including Mac OS X, is possible.

### Programming language

Vespucci utilizes initialiser lists and lambda functions, and will compile on most C++11-compliant compilers. Vespucci has been compiled successfully on GCC 4.9.1 and Clang 3.6.0.



**Figure 9:** A crystal found in a chick embryo tibiotarsus. **A** and **C** depict the colour and greyscale representations of a divergent “rainbow” colour scheme. **B** and **D** depict the colour and greyscale representations of a colour scheme that increases linearly in perceived brightness.

### Additional system requirements

Proper performance requires an Intel Core 2 or better processor, and at least 4 GB of RAM. Since datasets are stored entirely in RAM, computers with more RAM are capable of dealing with larger numbers of larger datasets (A single spectrum with 4096 spatial points has a size of about 50 kB).

### Dependencies

Vespucci for Windows is distributed with all necessary DLL files for normal operation. R integration requires R 3.1.3 or higher and the R packages RCpp, RCppArmadillo and RInside (which are also linked as C++ libraries). While the Windows binary distribution is compiled in  $\times 86\_64$  bytecode, Octave integration is handled by a helper process compiled in 32-bit  $\times 86$  bytecode, because Octave for Windows is distributed as 32-bit binaries.

Compiling Vespucci from source requires MLPACK and its dependencies (Armadillo, HDF5, BLAS (or replacement), LAPACK (or replacement), ARPACK, and Boost (only the Program Options, Random, Math and Unit Test Framework must be compiled)), Qt 5.0 or higher, CMinpack, and a fork of QCustomPlot distributed with the Vespucci source.

### List of contributors

The primary author of the Vespucci source code is Daniel P. Foose.

Vespucci integrates code from several other projects under the GPL license. The SIMPLS algorithm implementation was translated from the Octave Statistics package written by Fernando Damian Niewveldt. The VCA [8] and hySime algorithms were ported from code written by José Nascimento and José Bioucas Dias.

Vespucci is inspired by an unpublished MATLAB package written by Adam C. Stahler. User testing has been provided by M. Ali Akbar, Kevin A. O'Neil, Seth W. Brittle and Sessa A. Paluri.

### Code repository

**Name:** GitHub

**Identifier:** <https://github.com/dpfoose/Vespucci>

**Licence:** GNU General Public License Version 3

**Date published:** 28/07/2015

### Language

English.

### (3) Reuse potential

Vespucci is useful to researchers in any field that involves spectroscopy, including chemistry, biochemistry, molecular and cellular biology, biomedical engineering, materials science and engineering, and physics. Vespucci is designed to be useful for both novices and advanced scientists.

### Competing Interests

The authors declare that they have no competing interests.

### Acknowledgements

Dr David Dolson and Dr Michael Raymer are acknowledged for their advice and direction regarding the creation of the software. Dr Andrew Sarangan and Dr Piyush Shah are acknowledged for the AgNR samples.

The authors would like to acknowledge Award #14383405: National Science Foundation: Environmental Health & Safety of Nanotechnology.

### References

1. **Borland, D** and **Taylor Russell, M I** 2007 Rainbow Color Map (Still) Considered Harmful. *IEEE Computer Graphics and Applications*, 27, 14–17. DOI: <http://dx.doi.org/10.1109/MCG.2007.323435>
2. **Brewer, C A** 2015 *ColorBrewer* [Online]. Available at: <http://colorbrewer2.org> [Accessed 17 Augustus 2015].
3. **Curtin, R R**, et al. 2013 MLPACK: A Scalable C++ Machine Learning Library. *Journal of Machine Learning Research*, 14: 801–805. Available at: <http://www.jmlr.org/papers/v14/curtin13a.html>.
4. **de Jong, S** 1993 SIMPLS: An alternative approach to partial least squares regression. *Chemometrics and Intelligent Laboratory Systems*, 14: 251–263. DOI: [http://dx.doi.org/10.1016/0169-7439\(93\)85002-X](http://dx.doi.org/10.1016/0169-7439(93)85002-X)
5. **Du, P, Kibbe, W M** and **Lin, S M** 2006 Improved peak detection in mass spectrum by incorporating continuous wavelet transform-based pattern matching. *Bioinformatics*, 22: 2059. DOI: <http://dx.doi.org/10.1093/bioinformatics/btl355>
6. **Ince, D C, Hatton, L** and **Graham-Cumming, J** 2012 *Nature*, 482: 485–488. DOI: <http://dx.doi.org/10.1038/nature10836>. PMID: 22358837.
7. **Moskovits, M** 2006 Surface-Enhanced Raman Spectroscopy: a Brief Perspective. In Kniepp, K, Kneipp, H and Moscovits, M (Eds.) *Surface-Enhanced Raman Scattering: Physics and Applications*. Berlin: Springer-Verlag, pp. 1–17. DOI: [http://dx.doi.org/10.1007/3-540-33567-6\\_1](http://dx.doi.org/10.1007/3-540-33567-6_1)
8. **Nascimento, J M P** and **Bioucas Dias, J M** 2005 Vertex component analysis: A fast algorithm to unmix hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing*, 43: 898–910. DOI: <http://dx.doi.org/10.1109/TGRS.2005.844293>
9. **Sanderson, C** 2010 *Technical Report*. NICTA. Available at: [http://arma.sourceforge.net/armadillo\\_nicta\\_2010.pdf](http://arma.sourceforge.net/armadillo_nicta_2010.pdf).
10. **Shah, P J** 2012 *Nanostructured Columnar Thin Films Using Oblique Angle Deposition: Growth, SERS Characterization and Lithographic Processing*. Dayton, OH: Wright State University. Available at: [https://etd.ohiolink.edu/!etd.send\\_file?accession=wright1341852929&disposition=inline](https://etd.ohiolink.edu/!etd.send_file?accession=wright1341852929&disposition=inline).
11. **Stahler, A C**, et al. 2013 Evaluating the abnormal ossification in tibiotarsi of developing chick embryos exposed to 1.0 ppm doses of platinum group metals by spectroscopic techniques. *Bone*, 53(2), 421–429. DOI: <http://dx.doi.org/10.1016/j.bone.2012.12.051>
12. **Zhang, Z-M**, et al. 2009 An intelligent background-correction algorithm for highly fluorescent samples in Raman spectroscopy. *J. Raman Spectrosc.* DOI: <http://dx.doi.org/10.1002/jrs.2500>

**How to cite this article:** Foose, D P and Sizemore, I E P 2016 Vespucci: A Free, Cross-Platform Tool for Spectroscopic Data Analysis and Imaging. *Journal of Open Research Software*, 4: e4, DOI: <http://dx.doi.org/10.5334/jors.91>

**Submitted:** 09 September 2015 **Accepted:** 05 January 2016 **Published:** 29 January 2016

**Copyright:** © 2016 The Author(s). This is an open-access article distributed under the terms of the Creative Commons Attribution 4.0 International License (CC-BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited. See <http://creativecommons.org/licenses/by/4.0/>.