An Open Source Software Solution for Data Acquisition, Management, and Analysis for Spectrometric Measurements

Kevin B. Martin, William J. Mills, III, Matthew Bikun
(Northern Illinois University, Dekalb, IL, USA)
Northern Illinois BEEEAM Lab

- Building Energy Efficiency, Ergonomics, and Management Lab
  - Building Management System (BMS)
  - Controlled Environment Agriculture
  - Indoor Environmental Impacts on Humans
  - Wearable Sensor Array
• Extracting value from data acquired from multiple sensors in a timely and reliable manner requires:
  • Organization
  • Consistent formatting
  • Integrity of data throughout its lifecycle

• Finding meaning in data becomes difficult and inefficient without an optimized approach to data acquisition, management, and analysis
Lesson’s Learned from Welding Lab Monitoring

- Indoor air-quality study conducted as an undergraduate research project
  - Multiple aerosol monitoring instruments were deployed over the course of the semester
  - 35+ parameters were acquired on numerous occasions for multiple hours

A proper data organization, management, and analysis strategy was **NOT** utilized resulting in:

1) Inconsistent formatting
2) Overall disorganization
3) Data loss
Consistency among datasets acquired by different spectrometers is achieved via Python programming. After commonality is achieved, the re-formatted data is sent to a SQLite database. The SQLite database is queried from within an R-Studio database and subsequently analyzed.
Python 3 Programming Language

• Easy readability and uncluttered simple-to-learn syntax
• Community Driven
  – Third-party packages and learning resources freely available
• Multi-platform
• Viable option for building complex multi-protocol network applications

https://www.python.org/community/logos/
SQLite Relational Database Management Software

- Low computational overhead
  - Ideal for on-device use or implementation in an Internet of Things (IOT) system
  - Serverless operability
  - 140 TB maximum file size
- Utilizes Structured Query Language (SQL)
  - ANSI standard
- Native Python support

https://sqlite.org/copyright.html
R-Statistical Programming Language

- Fully fledged statistical programming environment
- Multi-platform
- R-Studio: Integrated development environment for the R-language
  - Dynamic GUI’s
  - Free for personal use* (License tiers for commercial use)
- RSQlite
  - Interfaces SQLite databases with R workspaces
- Powerful
  - Quickly analyze “big-data”

https://www.r-project.org/logo/
Low Cost Micro-Controllers & Micro-Computers

- Raspberry-Pi Computers
  - Low-cost Linux based micro-computer
  - Operates within an open-source ecosystem
  - Full computer capabilities
    - Data logging and data transmitting
  - Permits connectivity with multiple spectrometers and other sensors
  - CO₂, particulate, temperature, humidity
- Arduino
  - Open source micro-controller platform that allows simple connectivity with hardware

https://www.raspberrypi.org/products/raspberry-pi-4-model-b/
Handling Multiple Spectrometers

Micro-Computer Controlled Sensors
Measurements are sent directly to the database in a standardized format using Python based code

Windows Desktop Sensors
Python scripts scrub CSV files output from desktop software and the re-formatted data is input into the database
### Data Format

<table>
<thead>
<tr>
<th>DeviceID</th>
<th>DateTime</th>
<th>Wavelength</th>
<th>Intensity</th>
</tr>
</thead>
<tbody>
<tr>
<td>GHZ_1</td>
<td>2019-10-02 12:45:15</td>
<td>401</td>
<td>0</td>
</tr>
<tr>
<td>GHZ_1</td>
<td>2019-10-02 12:45:15</td>
<td>402</td>
<td>0</td>
</tr>
<tr>
<td>GHZ_1</td>
<td>2019-10-02 12:45:15</td>
<td>403</td>
<td>0</td>
</tr>
<tr>
<td>GHZ_1</td>
<td>2019-10-02 12:45:15</td>
<td>404</td>
<td>0.0000531</td>
</tr>
<tr>
<td>GHZ_1</td>
<td>2019-10-02 12:45:15</td>
<td>405</td>
<td>0.0000993</td>
</tr>
<tr>
<td>GHZ_1</td>
<td>2019-10-02 12:45:15</td>
<td>406</td>
<td>0.00125</td>
</tr>
<tr>
<td>GHZ_1</td>
<td>2019-10-02 12:45:15</td>
<td>407</td>
<td>0.00137</td>
</tr>
<tr>
<td>GHZ_1</td>
<td>2019-10-02 12:45:15</td>
<td>408</td>
<td>0.00189</td>
</tr>
<tr>
<td>GHZ_1</td>
<td>2019-10-02 12:45:15</td>
<td>410</td>
<td>0.00215</td>
</tr>
<tr>
<td>GHZ_1</td>
<td>2019-10-02 12:45:15</td>
<td>411</td>
<td>0.00242</td>
</tr>
<tr>
<td>GHZ_1</td>
<td>2019-10-02 12:45:15</td>
<td>412</td>
<td>0.00247</td>
</tr>
<tr>
<td>GHZ_1</td>
<td>2019-10-02 12:45:15</td>
<td>413</td>
<td>0.000399</td>
</tr>
<tr>
<td>GHZ_1</td>
<td>2019-10-02 12:45:15</td>
<td>414</td>
<td>0.000467</td>
</tr>
<tr>
<td>GHZ_1</td>
<td>2019-10-02 12:45:15</td>
<td>415</td>
<td>0.000547</td>
</tr>
<tr>
<td>GHZ_1</td>
<td>2019-10-02 12:45:15</td>
<td>416</td>
<td>0.000556</td>
</tr>
<tr>
<td>GHZ_1</td>
<td>2019-10-02 12:45:15</td>
<td>417</td>
<td>0.000574</td>
</tr>
<tr>
<td>GHZ_1</td>
<td>2019-10-02 12:45:15</td>
<td>418</td>
<td>0.000941</td>
</tr>
</tbody>
</table>

**Standardized comma separated format**

```
DeviceID, DateTime, Wavelength, Intensity
```

- **DeviceID**
- **DateTime**
- **Wavelength**
- **Intensity**
Simple DB Connection

- **RSQLite**
  - Freely available and frequently maintained R Library package
  - Allows interfacing of SQLite databases from within R
  - Allows database editing and creation from within R
- Once in R, data can be analyzed using other community created packages
  - PCA
R-Based Lighting Calculations

- R based programs can be tailored to user needs
- Ideal for spectrometric data given the intensity values assigned to each wavelength

Values derived from example data
Principal Component Analysis

Transmission Glasses 6600K - 2700K PCA

PC1 (59.9% explained var.)

PC2 (22.6% explained var.)

Groups:
- 2700K
- 3000K
- 4100K
- 5000K
- 6500K
Principal Component Analysis

Transmission Glasses 6500K - 2700K PCA
Other R Realted Benefits

• Plotting libraries (ggplot & ggspectra)
• Photobiology (agricultural lighting parameter calculator)
Conclusions

• Combination of SQLite and R in conjunction with Python provides a flexible, highly capable pathway for data set management and analysis

• Conceptually simple
  – Allows for template like approach

• Implementation
  – Sensor integration still can be problematic
  – Python code not always supplied by manufacturer