

# *interlab: A Python Module for Analyzing Interlaboratory Comparison Data*

**David A. Sheen**

National Institute of Standards and Technology,  
Gaithersburg, MD 20899, USA

[david.sheen@nist.gov](mailto:david.sheen@nist.gov)

**Software DOI:** <https://doi.org/10.18434/t4/1503310>

**Software Version:** 0.1

**Key words:** interlaboratory comparison; metabolomics; outlier detection; uncertainty analysis.

**Accepted:** February 25, 2019

**Published:** March 15, 2019

<https://doi.org/10.6028/jres.124.006>

---

## 1. Summary

interlab was developed as a software tool to perform consensus analysis on spectral data from interlaboratory studies. It is designed to estimate the spread in the spectral data and to identify possible outliers among both spectral populations and facilities in the study. Use of this code allows researchers to identify laboratories producing data closest to the consensus values, thereby ensuring that untargeted studies are using the most precise data available to them. The software was originally developed for analyzing nuclear magnetic resonance spectroscopic data [1, 2] but can be applied to nearly any array data, including Raman or Fourier-transform infrared spectroscopy and gas or liquid chromatography. Details on the implementation of the code can be found in Ref. [1].

The input for the code consists of a set of sample labels identifying the physical objects measured in the interlaboratory study, facility labels that identify the facility of origin of the measurements, and the data themselves. It is the user's responsibility to format the data and metadata so that the code can read it. In addition, the user must specify the distance function that will be used to compare the spectra and the statistical distribution that these distances will be fit to; it is the user's responsibility to ensure the data are suitable for the distance function they choose, e.g., if the distance function is valid only for probability mass functions, then each sample's data must be sum-normalized and everywhere nonnegative. One example is given in the example notebook, [https://pages.nist.gov/interlab\\_py/analysis\\_demo.html](https://pages.nist.gov/interlab_py/analysis_demo.html).

Given the input data, the code will perform the following tasks:

- Calculates the interspectral distances based on the user-selected distance function;
- Fits the user-selected distribution function to the distance data and calculates the corresponding probability scores (for a normal distribution, this is the Z-score);
- Identifies outliers within each spectral population;

- Conducts a principal components analysis on the probability scores (e.g., Z-scores) and compute the projected statistical distance;
- Uses the projected statistical distance to determine the data set outliers.

The software cannot be used out of the box. Users must create an interface to their own software, and that interface will be specific to the user's application. The example notebook, [https://pages.nist.gov/interlab\\_py/analysis\\_demo.html](https://pages.nist.gov/interlab_py/analysis_demo.html), demonstrates one such possible interface.

## 2. Software Specifications

<b>NIST Operating Unit(s)</b>	Material Measurement Laboratory
<b>Category</b>	Uncertainty analysis
<b>Targeted Users</b>	PIs for large interlaboratory studies
<b>Operating System(s)</b>	Cross-platform, where Python, NumPy, and SciPy are installed
<b>Programming Language</b>	Python 3.6
<b>Inputs/Outputs</b>	This software is not usable out-of-the-box. The user must write additional code to interface their model with this software.
<b>Documentation</b>	<a href="https://pages.nist.gov/interlab_py">https://pages.nist.gov/interlab_py</a>
<b>Accessibility</b>	N/A small-scale research tool
<b>Disclaimer</b>	<a href="https://www.nist.gov/director/licensing">https://www.nist.gov/director/licensing</a>

## 3. Methods for Validation

The code largely consists of scipy [3] and scikit-learn [4] functions, which are already tested. The outputs of each stage of the code are analyzed and validated in the example notebook, [https://pages.nist.gov/interlab\\_py/analysis\\_demo.html](https://pages.nist.gov/interlab_py/analysis_demo.html).

## 4. References

- [1] Sheen DA, Rocha WFC, Lippa KA, Bearden DW (2017) A scoring metric for multivariate data for reproducibility analysis using chemometric methods. *Chemometrics and Intelligent Laboratory Systems* 162:10-20. <https://doi.org/10.1016/j.chemolab.2016.12.010>
- [2] Rocha WFC, Sheen DA, Bearden DW (2018) Classification of samples from NMR-based metabolomics using principal components analysis and partial least squares with uncertainty estimation. *Analytical and Bioanalytical Chemistry* 410(24):6305-6319. <https://doi.org/10.1007/s00216-018-1240-2>
- [3] Jones E, Oliphant T, Peterson P (2001) *SciPy: Open source scientific tools for Python*. Available at <http://www.scipy.org/>
- [4] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011) Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12:2825-2830.

**About the author:** David Sheen is a Physicist in the Chemical Sciences Division at NIST. His work is on uncertainty analysis in complex systems including high-temperature chemical kinetics and metabolomics. The National Institute of Standards and Technology is an agency of the U.S. Department of Commerce.