Open-Software Tools for the Analysis of Electrochemical Impedance Spectra

AiMES ECS Data Science Showcase

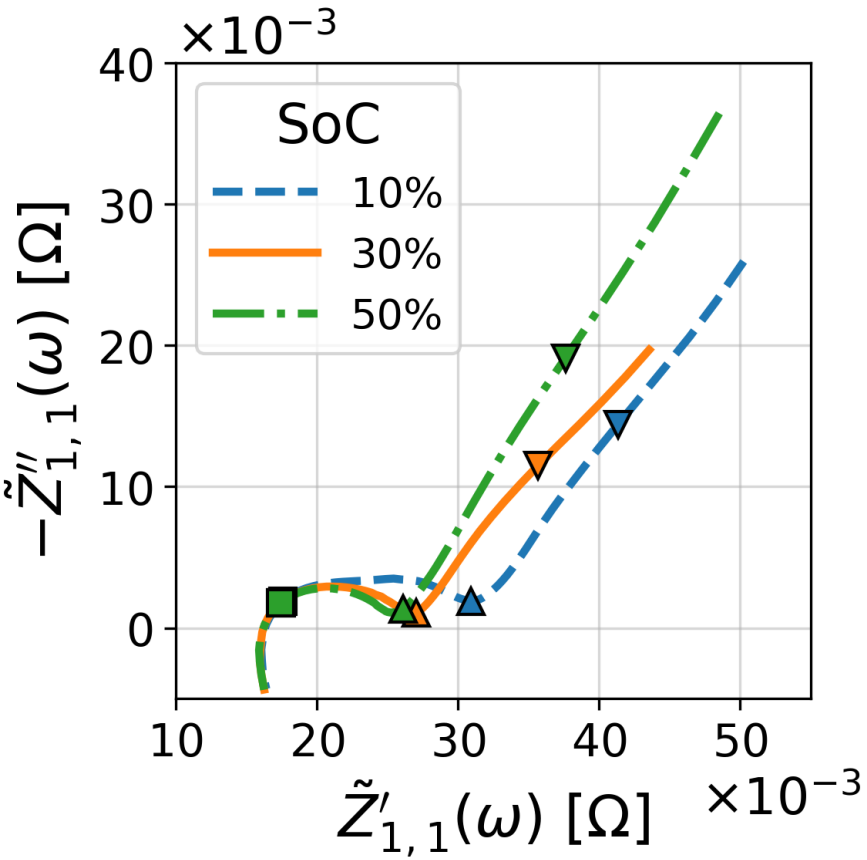
Matthew D. Murbach and Daniel T. Schwartz

Department of Chemical Engineering and Clean Energy Institute
University of Washington, Seattle

1 October 2018

mmurbach@uw.edu  matt_murbach  mdmurbach

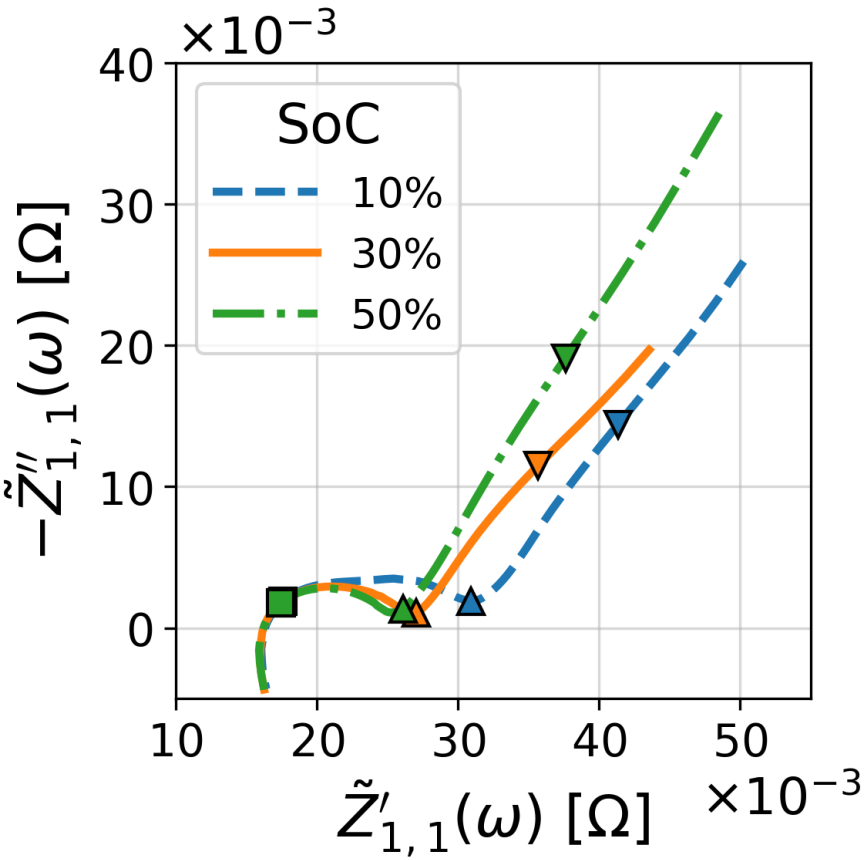
Goal: extract *physically meaningful* and *reliable* insights from data



- Battery health
- Relative impact of kinetics/transport
- Quantitative parameters
- ...

Murbach, M. D. et al. *J. Electrochem. Soc.* 165, A2758 (2018)

Goal: extract *physically meaningful* and *reliable* insights from data



Current analysis options:

- Potentiostat software
- Proprietary 3^{rd} party software
- Code written in individual labs

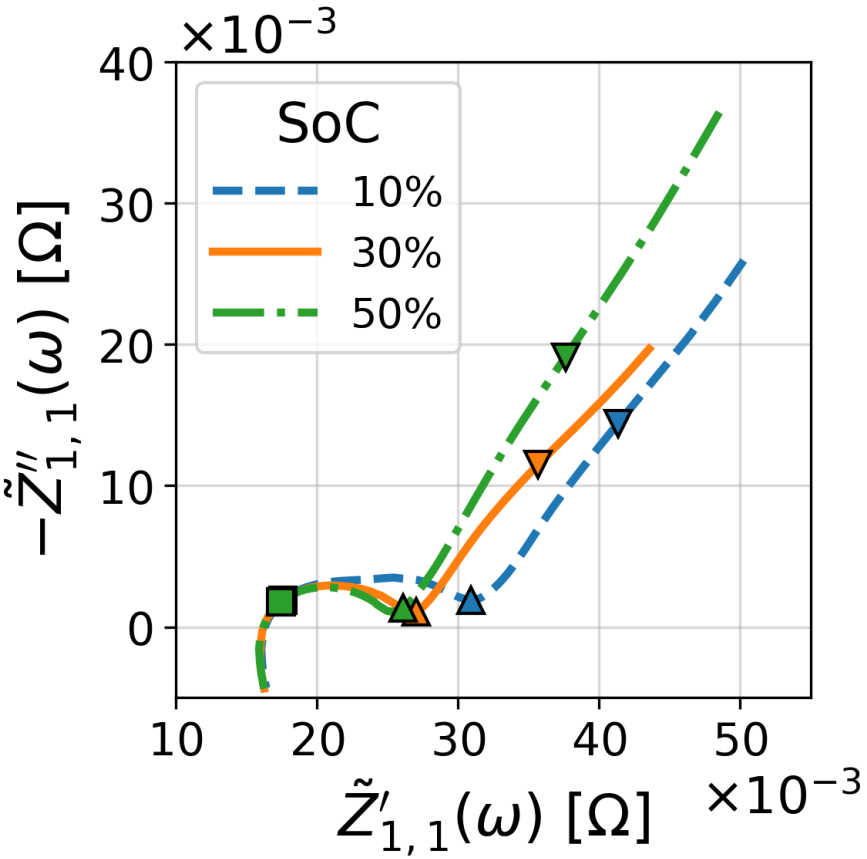
Metrohm Autolab

ZView®

EIS Spectrum Analyser

Murbach, M. D. et al. *J. Electrochem. Soc.* 165, A2758 (2018)

Goal: extract *physically meaningful* and *reliable* insights from data



Current analysis options:

Metrohm Autolab

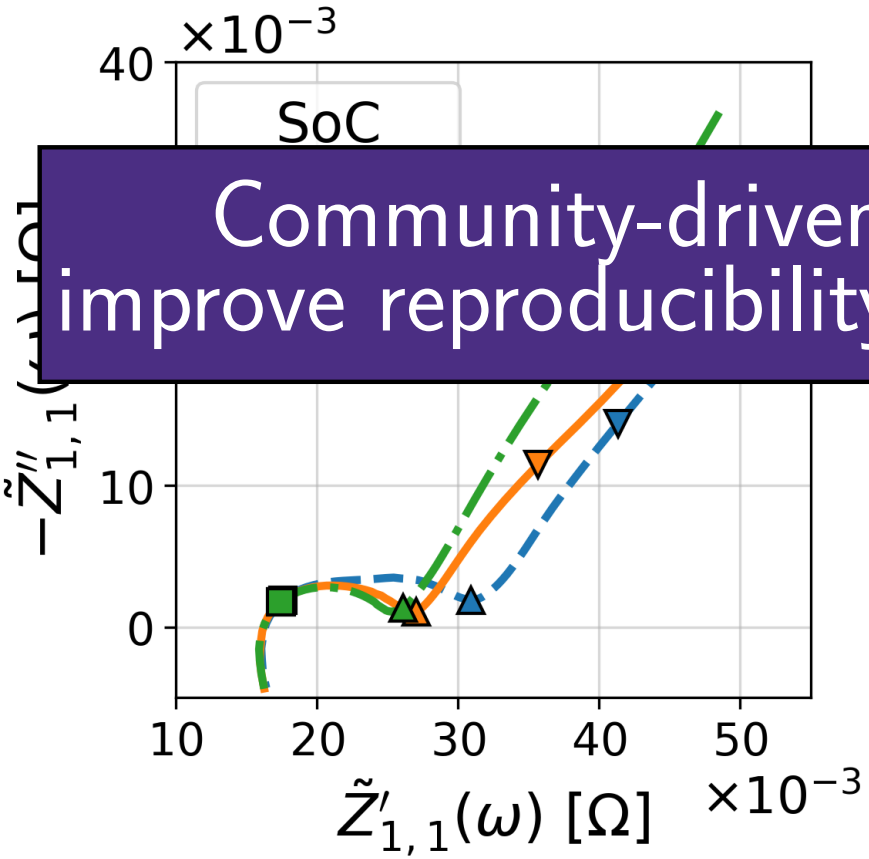
ZView®

EIS Spectrum Analyser

- Typically clunky to use
- Proprietary/hidden algorithms (cant reproduce!)
- Difficult to share and improve

Murbach, M. D. et al. *J. Electrochem. Soc.* 165, A2758 (2018)

Goal: extract *physically meaningful* and *reliable* insights from data



Current analysis options:

Community-driven, open-source tools improve reproducibility and accelerate science

- Typically clunky to use
- Proprietary/hidden algorithms (cant reproduce!)
- Difficult to share and improve

Murbach, M. D. et al. *J. Electrochem. Soc.* 165, A2758 (2018)

What will the future of impedance analysis look like?

- Fast, **reproducible** analysis of impedance spectra
- Easy-to-use software that encourages **best practice**
- **Community-driven** toolkit grows with improved methods for interpretation and analysis

An example: Scikit-learn has made high-quality machine learning available to all

machine learning made easy-to-use

- Easy to apply and compare different models, open and powerful enough to accomplish real tasks
- The key is the *community-driven api*

1^{st} steps: Python package + web-based GUI

impedance.py

```
In [1]: import matplotlib.pyplot as plt
        import numpy as np

        from impedance.circuits import CustomCircuit

In [2]: # read data
        data = np.genfromtxt('../../../data/exampleData.csv', delimiter=',')

        frequencies = data[:,0]
        Z = data[:,1] + 1j*data[:,2]

        frequencies = frequencies[np.imag(Z) < 0]
        Z = Z[np.imag(Z) < 0]

In [3]: circuit = CustomCircuit(initial_guess=[.01, .005, .1, .005, .1, .001, 200], circuit='R_0-p(R_1,C_1
        )-p(R_1,C_1)-W_1/W_2')

        circuit.fit(frequencies, Z)

        print(circuit)

        ------------------------------
        Circuit: None
        Circuit string: R_0-p(R_1,C_1)-p(R_1,C_1)-W_1/W_2
        Algorithm: leastsq
        Fit: True
        Fit parameters:
                R_0 = 1.65e-02
                R_1 = 8.77e-03
                C_1 = 3.28e+00
                R_1 = 5.31e-03
                C_1 = 2.32e-01
                W_1 = 6.37e-02
                W_2 = 2.38e+02

        ------------------------------
```

ImpedanceAnalyzer

Murbach, M. D. et al. *J. Electrochem. Soc.* 165, A297 (2018)

impedance.py demo

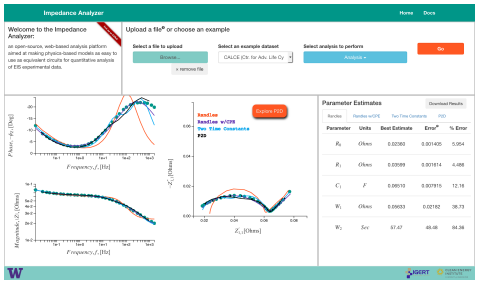
https://github.com/ECSHackWeek/impedance.py

ImpedanceAnalyzer demo

https://github.com/mdmurbach/ImpedanceAnalyzer

Murbach, M. D. et al. *J. Electrochem. Soc.* 165, A297 (2018)

Moving from beta to ImpedanceAnalyzer v1.0



- Login/save data and settings
- Additional physics-based datasets
- Implement confidence interval estimation and visualization
- Drag-n-drop circuit creation
- Desktop vs. web application

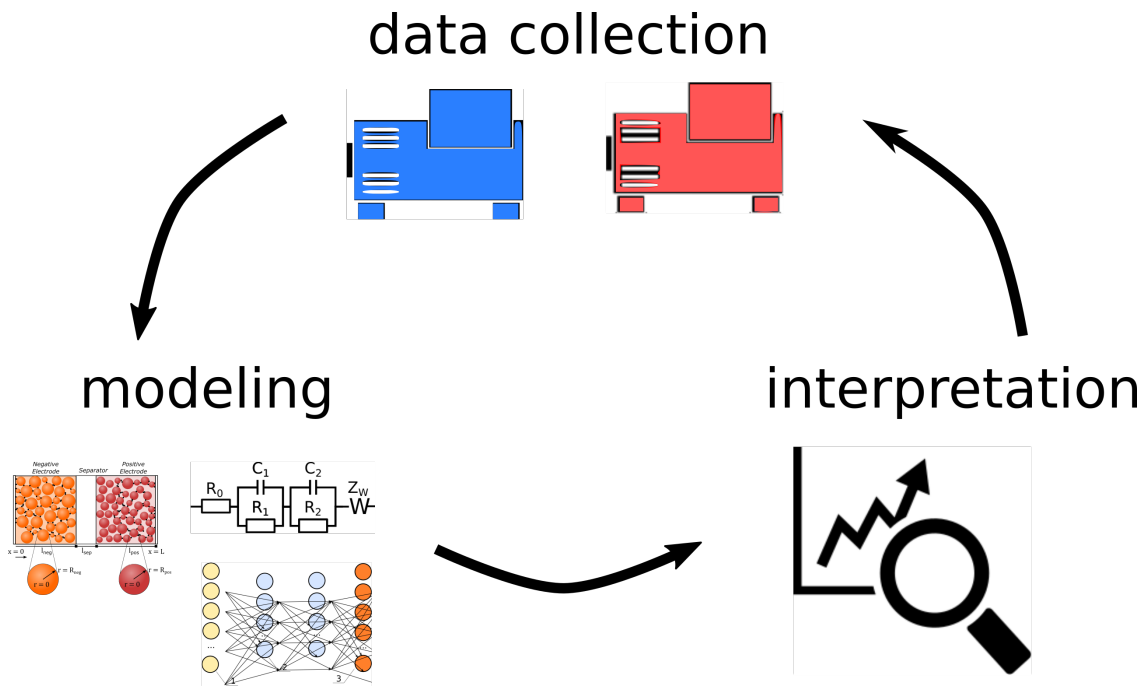
Expanding usability and features of impedance.py

```
In [1]: import matplotlib.pyplot as plt
        import numpy as np

        from impedance.circuits import CustomCircuit

In [2]: # read data
        data = np.genfromtxt('../../../data/exampleData.csv', delimiter=',')

        frequencies = data[:,0]
        Z = data[:,1] + 1j*data[:,2]

        frequencies = frequencies[np.imag(Z) < 0]
        Z = Z[np.imag(Z) < 0]

In [3]: circuit = CustomCircuit(initial_guess=[.01, .005, .1, .005, .1, .001, 200], circuit='R_0-p(R_1,C_1
        )-p(R_1,C_1)-W_1/W_2')

        circuit.fit(frequencies, Z)

        print(circuit)

        ------------------------------
        Circuit: None
        Circuit string: R_0-p(R_1,C_1)-p(R_1,C_1)-W_1/W_2
        Algorithm: leastsq
        Fit: True
        Fit parameters:
                R_0 = 1.65e-02
                R_1 = 8.77e-03
                C_1 = 3.28e+00
                R_1 = 5.31e-03
                C_1 = 2.32e-01
                W_1 = 6.37e-02
                W_2 = 2.38e+02


        ------------------------------
```

- Incorporate interactive visualizations
- Additional equivalent circuit and physics-based models and elements
- Improve initialization of smart parameter guesses
- Improve data validation methods
- ...

longer-term vision:

open API for data, analysis, and interpretation pipeline



Thank you!

- **Neal Dawson-Elli**
- Qin Pang
- Simon Timbillah
- Prince Sarfo
- Jason Bonezzi
- Prof. David Beck
- Prof. Dan Schwartz
- **Victor Hu**
- Erica Eggleton
- Linnette Teo
- Yanbo Qi

CLEAN ENERGY INSTITUTE
UNIVERSITY of WASHINGTON

eScience Institute
ADVANCING DATA-INTENSIVE DISCOVERY IN ALL FIELDS