

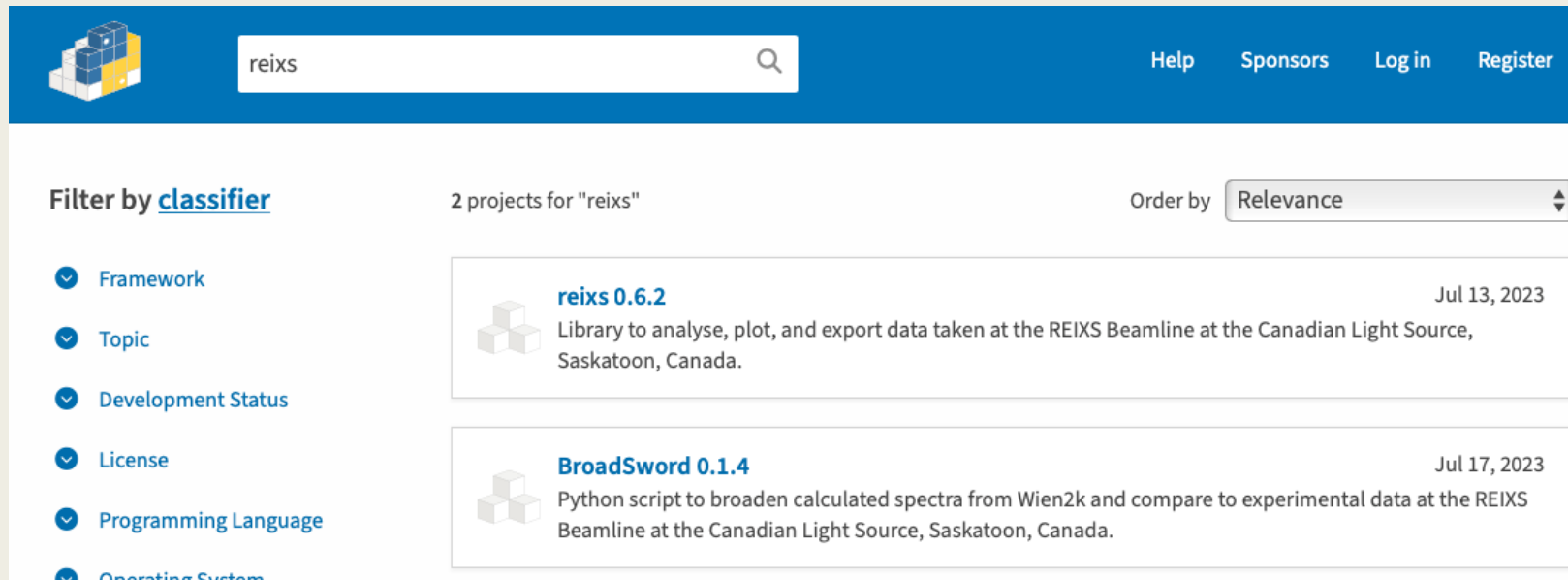
BroadSword Motivations

- Having the ability to save and reload a broadened calculation
- Something more compatible with the .csv output of the reixs software
- The ability for our group to modify the code as needed
- Something written purely in python while still being reasonably fast



Downloading BroadSword

- Similar to downloading Patrick's reixs package through the terminal
 - *pip install BroadSword (First time)*
 - *pip install --upgrade BroadSword (To update to newest version)*
- Found on PyPi by searching for BroadSword or reixs
 - <https://pypi.org/project/BroadSword/>



The screenshot shows the PyPI search results for the query 'reixs'. The header is blue with the PyPI logo, a search bar containing 'reixs', and links for Help, Sponsors, Log in, and Register. Below the header, the search results are displayed. On the left, there are filter options: Framework, Topic, Development Status, License, Programming Language, and Operating System. The main content area shows '2 projects for "reixs"'. The results are ordered by Relevance. The first result is 'reixs 0.6.2' by the REIXS Beamline at the Canadian Light Source, Saskatoon, Canada, dated Jul 13, 2023. The second result is 'BroadSword 0.1.4' by the REIXS Beamline at the Canadian Light Source, Saskatoon, Canada, dated Jul 17, 2023.

reixs

Help Sponsors Log in Register

Filter by [classifier](#)

2 projects for "reixs" Order by Relevance

reixs 0.6.2 Jul 13, 2023
Library to analyse, plot, and export data taken at the REIXS Beamline at the Canadian Light Source, Saskatoon, Canada.

BroadSword 0.1.4 Jul 17, 2023
Python script to broaden calculated spectra from Wien2k and compare to experimental data at the REIXS Beamline at the Canadian Light Source, Saskatoon, Canada.

Input Parameters

- Specify the base directory
- Enter name of input file
 - LoadExp takes in .txt or .csv, and you can specify the number of header lines to ignore, allowing direct input from the reixs software into broadsword
- Some parameters can be left empty (Ex: sites) and it will assume a value for them. (sites=1)
 - If XANES calculation file is not specified, it will simply duplicate the XAS into that parameter. (See bottom line for minimum required inputs)
- GS = Ground state, ES = Excited state

Upload XES

4Sample/FinalSpectrum/N_test_XES.txt

Upload XANES

P2N4Sample/FinalSpectrum/N_test_XAS.txt

Upload Calculated XES

Ef 0.44996547 Elem. N

Upload Calculated XAS

/Users/cas003/Downloads/2022
Work/GeP2N4Sample/FinalSpectrum/N1_emis.txspec
/Users/cas003/Downloads/2022
Work/GeP2N4Sample/FinalSpectrum/N2_emis.txspec
/Users/cas003/Downloads/2022
Work/GeP2N4Sample/FinalSpectrum/N3_emis.txspec
/Users/cas003/Downloads/2022
Work/GeP2N4Sample/FinalSpectrum/N4_emis.txspec

Original BroadSword

Upload Calculated XANES

/Users/cas003/Downloads/2022
Work/GeP2N4Sample/FinalSpectrum/N1_half.txspec
/Users/cas003/Downloads/2022
Work/GeP2N4Sample/FinalSpectrum/N2_half.txspec
/Users/cas003/Downloads/2022
Work/GeP2N4Sample/FinalSpectrum/N3_half.txspec
/Users/cas003/Downloads/2022
Work/GeP2N4Sample/FinalSpectrum/N4_half.txspec

Ef	Ebind	Edge	Sites
0.45062079	27.176237	K	1
0.45091878	27.177975	K	1
0.45090808	27.122234	K	1
0.45088602	27.177070	K	1

```
# Create an instance of the class
```

```
broad = Broaden()
```

```
# Load the experimental and calculations
```

```
broad.loadExp(basedir,XES="N_test_XES.txt",XANES="N_test_XAS.txt",GS_fermi=0.44996547,headerlines=[2,2])
```

```
broad.loadCalc(basedir,XES="N1_emis.txspec",XAS="N1_abs.txspec",GS_bindingEnergy=27.176237,XANES="N1_half.txspec",ES_fermi=0.45062079,sites=1,edge="L2")
```

```
broad.loadCalc(basedir,XES="N2_emis.txspec",XAS="N2_abs.txspec",GS_bindingEnergy=27.177975,XANES="N2_half.txspec",ES_fermi=0.45091878)
```

```
broad.loadCalc(basedir,XES="N3_emis.txspec",XAS="N3_abs.txspec",GS_bindingEnergy=27.122234,XANES="N3_half.txspec",ES_fermi=0.45090808)
```

```
broad.loadCalc(basedir,XES="N4_emis.txspec",XAS="N4_abs.txspec",GS_bindingEnergy=27.177070,XANES="N4_half.txspec",ES_fermi=0.45088602)
```

```
# broad.loadCalc(basedir,XES="N1_emis.txspec",XAS="N1_abs.txspec",GS_bindingEnergy=27.177070) # Minimum required inputs to broaden a spectra.
```

New BroadSword

Broadening Parameters

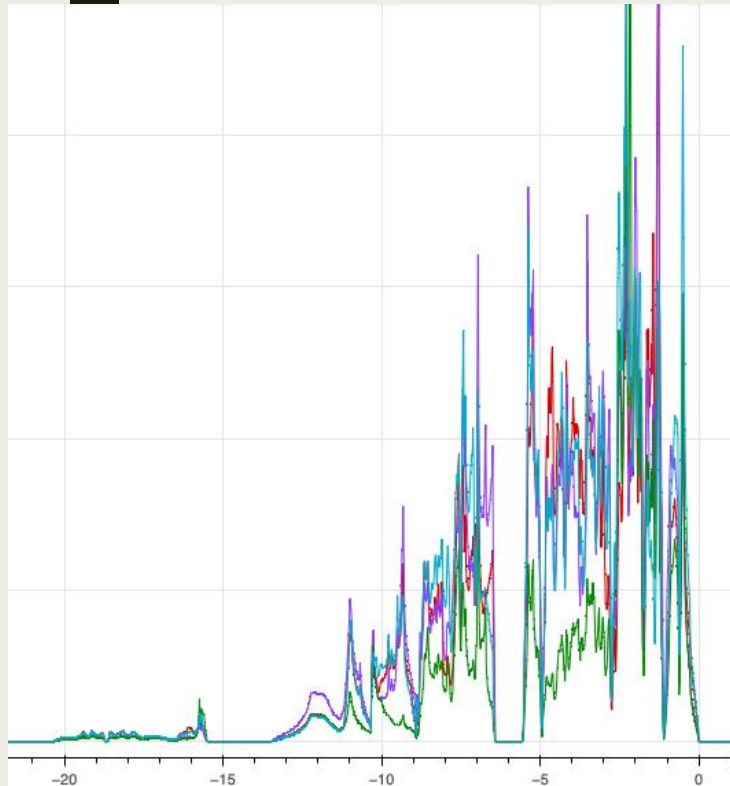
- Specify the broadening parameters the same way as in original program
- Specify the shifting parameters
 - Will produce an output graph to show the shift
- Broaden the spectra
 - 'separate' simply refers to plotting the XES in a different plot window than the XAS for visual purposes

The screenshot displays the 'BroadSword' software interface. On the left, a 'Bands' list contains ten energy ranges, each repeated twice: -20.3396 to -18.7196, -18.6196 to -15.4995, -13.4395 to -6.41937, -5.53935 to 0.00075, and -20.3396 to -18.7196. Below this is a 'Shift' input field set to 20.2 and a 'Band Gap' input field set to 4.21. A 'Shift' button is at the bottom. On the right, the 'Original BroadSword' section has two columns: 'XES' and 'XAS'. Under 'XES', 'E_Spec' is 1200 and 'Scale' is 0.5. Under 'XAS', 'E_Mono' is 5000. Below these, 'E_Core' is 0.15 and 'Disorder' is 0.5. A 'Broaden' button is positioned to the right of the 'Disorder' field.

```
# Initialize the broadening parameters
broad.initResolution(corelifetime=0.15,specResolution=1200,monoResolution=5000,disorder=0.5,XESscaling=0.5,XASscaling=0.5)
# Shift the spectra until the calculation aligns with the experimental
broad.Shift(XESshift=19.2,XASshift=20.2,separate=False)
# Broaden the spectra
broad.broaden(separate=False)
```

New BroadSword

Targeting Specific XES Bands



Output of .printBands()

```
In inequivalent atom #0
Band #0 is located at -20.33963 to -18.7196
Band #1 is located at -18.6196 to -15.49954
Band #2 is located at -13.4395 to -6.41937
Band #3 is located at -5.53935 to 0.00075
In inequivalent atom #1
Band #0 is located at -20.33963 to -18.7196
Band #1 is located at -18.6196 to -15.49954
Band #2 is located at -13.4395 to -6.41937
Band #3 is located at -5.53935 to 0.00075
In inequivalent atom #2
Band #0 is located at -20.33963 to -18.7196
Band #1 is located at -18.6196 to -15.49954
Band #2 is located at -13.4395 to -6.41937
Band #3 is located at -5.53935 to 0.00075
In inequivalent atom #3
Band #0 is located at -20.33963 to -18.7196
Band #1 is located at -18.6196 to -15.49954
Band #2 is located at -13.4395 to -6.41937
Band #3 is located at -5.53935 to 0.00075
```

Bands	Shift	Scale
-20.3396 to -18.7196	30	0.1
-18.6196 to -15.4995	33	0.2
-13.4395 to -6.41937	30	0.2
-5.53935 to 0.00075	20	0.4
-20.3396 to -18.7196	15	0.2
-18.6196 to -15.4995	19.2	0.2
-13.4395 to -6.41937	19.2	0.4
-5.53935 to 0.00075	19.2	0.2
-20.3396 to -18.7196	30	0.3
-18.6196 to -15.4995	33	0.2
-13.4395 to -6.41937	30	0.1
-5.53935 to 0.00075	20	0.5
-20.3396 to -18.7196	15	0.3

Original BroadSword 0 0

- Print the bands to the screen
- Add in a new parameter into the .initResolution() and .Shift() functions that were described last page
- Specify the scaling/shifting for each band
 - This is an array of arrays. Each inner array contains the band scale/shift values for a single inequivalent site

```
# Optionally you can scale and shift specific bands in XES. Use printBands() to determine where the bands are located.
# Then add the new argument XESbandScaling into initResolution() and XESbandshift int Shift()
broad.printBands()
broad.initResolution(corelifetime=0.15,specResolution=1200,monoResolution=5000,disorder=0.5,XESScaling=0.5,XASScaling=0.5,XESbandScaling=[[0.1,0.2,0.2,0.4],[0.2,0.2,0.4,0.2],[0.3,0.3,0.3,0.3],[0.1,0.1,0.1,0.1]],XESbandshift=[[30,33,30,20],[15,19.2,19.2,19.2],[30,33,30,20],[15,19.2,19.2,19.2]])
broad.Shift(XESshift=19.2,XASshift=20.2,separate=False,XESbandshift=[[30,33,30,20],[15,19.2,19.2,19.2],[30,33,30,20],[15,19.2,19.2,19.2]])

# Broaden the spectra
broad.broaden(separate=False)
```

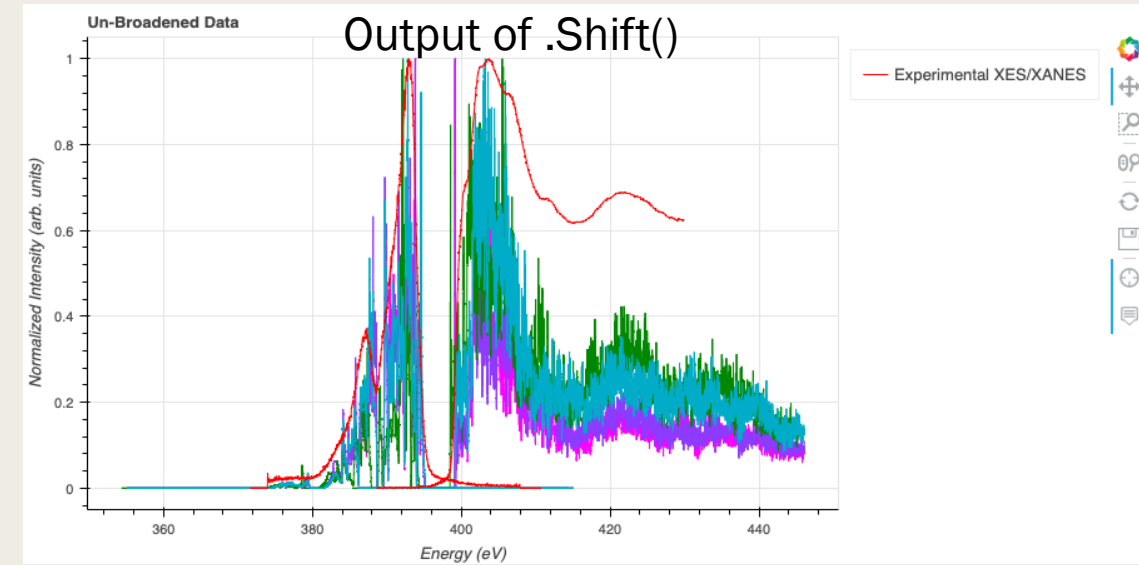
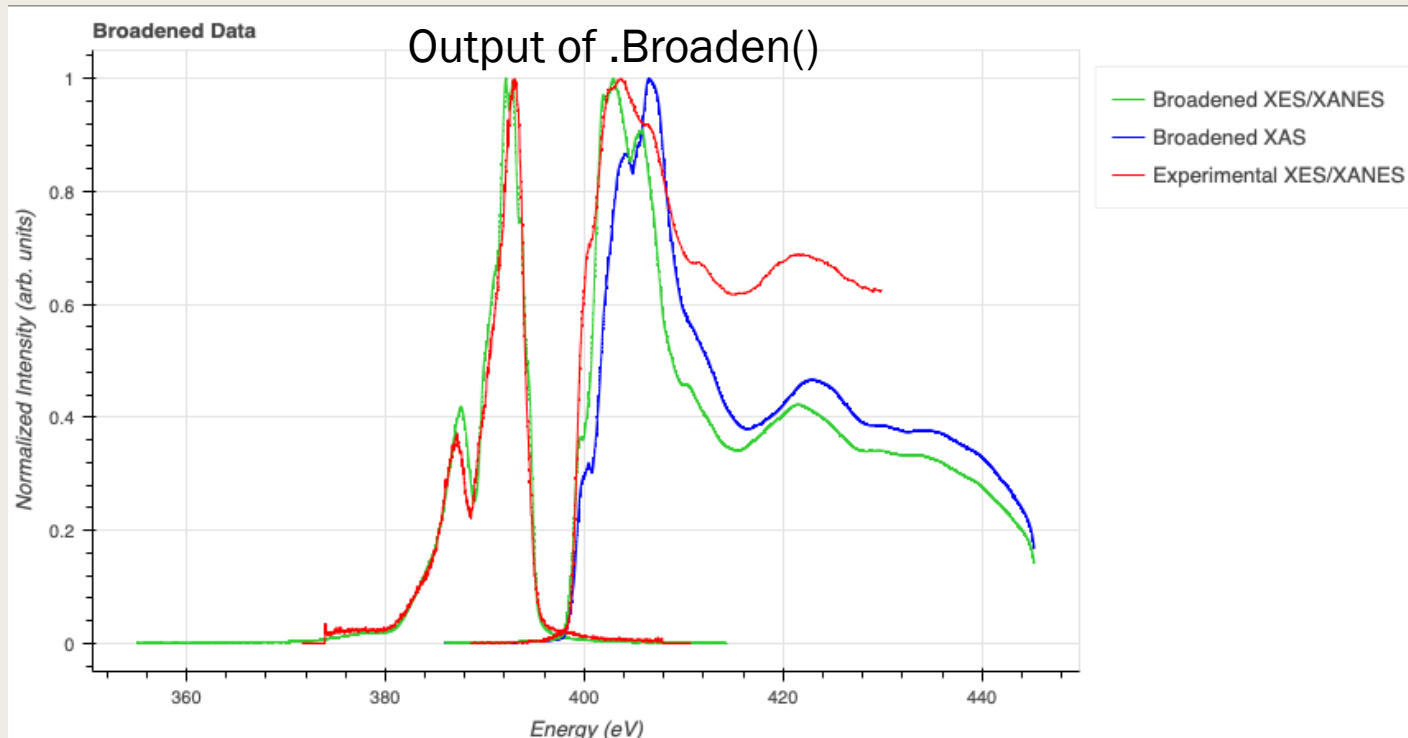
Shift value becomes obsolete

A single inequivalent atom, and the four bands associated with it

New BroadSword

Broadened Output

```
# Export the broadened calculated spectra  
broad.export(filename="GeP2N4",element="N",individual=False)
```



- Returns values accurate to ~3 decimal places to original broadsword program
 - Python is using more precision in its floating-point numbers
 - (Perhaps reduce this for speed improvement???)
- Individual = False
 - Returns only the broadened sum of the inequivalent sites
 - Ex: GeP2N4_N_XES.csv
- Individual = True
 - Returns the individual broadened contributions as well
 - This is the default in original broadsword

General Comments/Future Work

- Bring back usability of the Ctypes version, as that was slightly faster, but more difficult for the user to set up, and not functional on windows yet.
- Various miscellaneous updates to the comments and example program to make it clearer to understand
- Update the plotting functions to be more code friendly
- The python code is almost a direct copy from the C program, and so it follows some very poor python practices such as having everything in a global variable.
 - *The only major difference is the use of vectorization, which is what allows it to run within a reasonable timeframe.*
- Testing has been fairly limited, so please do a couple trial runs to confirm that the output is the same from the original before assuming everything is correct