CHARISMA

# RamanChada

# Cheat Sheet

## Loading & displaying data

```
R = RamanChada(‘C:\file.spc’)
```

*Spectrum object info:* **R**

```
RamanChada with 3526 points generated Wed Sep 15 …
```

*Get metadata as dict*: `R.meta`

*Add metadata (dict):*

```
R.add_metadata({‘Power[mW]’:5})
```

*Get processing log as list:*

`R.log` *or* `R.show_log()`

*Undo last processing step:* `R.rewind(-1)`

*Revert to original data:* `R.rewind(0)`

*Save .cha file with all changes:*

```
R.commit(‘commit message’)
```

*Load .cha file:*

```
R = RamanChada(‘C:\file.cha’)
```

*Plot spectrum:* `R.plot()`

*Get the raw data from the same file:*

```
S = RamanChada(
         R.file_path, raw=True)
```

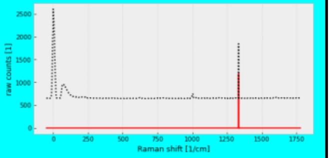
## Cosmic rays

*Fit x ray model*

```
R.fit_xrays()
```

*Plot x ray model*

```
R.plot_xrays()
```



```
Remove x rays: R.remove_xrays()
```

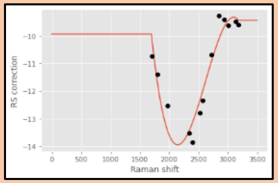
## Calibration

*Interpolate to x axis of reference spectrum:*

```
R.interpolate_x(reference_spectrum)
```

*Calibrate x with existing RamanCalibration* `cal`*:*

```
R.calibrate(cal)
```



*Show x calibration curve:*

```
cal.show()
```

*Get calibration time / date:*

```
cal.time
```

*Generate x calibration using reference spec:*

```
cal = R.make_x_calibration(ref)
```

*…and consider only peaks in interval:*

```
cal = R.make_x_calibration(
    ref,1700,3200)
```

*Generate x calibration using peak positions list:*

```
cal = R.make_x_calibration(
    [202.12,451.76,…,1809.28])
```

*Calibrate y with existing RamanCalibration:*

```
R.calibrate_y(y_cal)
```

*Generate y calibration using reference spectrum:*

```
y_cal = R.make_y_calibration(ref)
```

`Save` *RamanCalibration* `to disk:`

```
cal.save(‘C:\cal_filename.chacal’)
```

`Load` *RamanCalibration* `from disk:`

```
cal = read_x_calibration(
     ‘C:\cal_filename.chacal’)
```

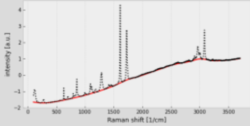
## Baseline separation

*Fit baseline using SNIP method*

```
R.fit_baseline(method='snip’)
```

*Plot baseline model*

```
R.plot_baseline()
```



*Fit and remove baseline*

```
R.remove_baseline(method='snip’)
```

## Pre-processing

*Smooth spectrum using Savitzky-Golay filter:*

```
R.smooth('sg',window=11,order=3)
```

*Normalize spectrum using vector norm:*

```
R.normalize(‘vector’)
```

*Area normalization using only an interval:*

```
R.normalize(‘area',500,1250)
```

*Crop spectrum on the x axis to 500-1.250 cm*$^{-1}$*:*

```
R.x_crop(500,1250)
```

*Add spectrum S to R:* `R.math(S,‘+')`

## Peaks search & fitting

*Find peaks with prominence>0.2 without fitting:*

```
R.peaks(prominence=0.2,fit=False)
```

*DataFrame with detected/fitted peaks:* `R.bands`

*Fit peaks within 2xFWHM w/ Voigt profile & plot:*

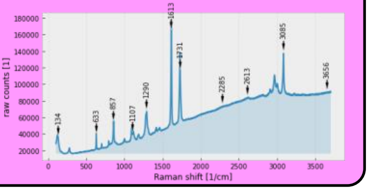
```
R.peaks(fitmethod='voigt’,
     interval_width=2,show=True)
```

*Find peaks using wavelets, fit & sort by position:*

```
R.peaks(cwt=True,sort_by='position’)
```

*Plot spectrum with peak positions*

```
R.show_bands()
```



## Batch processing

*List of RamanChada objects from path list* `files`*:*

```
SL = [RamanChada(f) for f in files]
```

*Normalize all spectra by standard normal variate:*

```
[s.normalize() for s in SL]
```

*Apply Wiener filter to all and then save to disk:*

```
[s.smooth(‘wiener’,7) for s in SL]
[s.commit(‘smoothed’) for s in SL]
```