

Variance

April 30, 2018

```
In [1]: import fabio
```

```
/mntdirect/_scisoft/users/jupyter/jupy35/lib/python3.5/site-packages/h5py/___init___py:36: Futur
from ._conv import register_converters as _register_converters
```

```
In [2]: import numpy
```

```
In [3]: data = fabio.open("calib/n104_saxs_00003.edf").data
        dark = fabio.open("calib/n104_saxs_00003_dark.edf").data
        print(data.dtype, dark.dtype)
```

```
uint16 uint16
```

```
In [4]: #Numpy
```

```
def formula_np(data, dark):
    return 0.1*(1.0*data-dark)+1.0
```

```
variance_np = formula_np(data, dark)
print(variance_np.max(), variance_np.min(), variance_np.mean(), variance_np.std())
```

```
2.4000000000000004 -0.9000000000000001 1.0004278700086806 0.14051968965805456
```

```
In [5]: %timeit formula_np(data, dark)
```

```
14.2 ms ± 64.9 µs per loop (mean ± std. dev. of 7 runs, 100 loops each)
```

```
In [6]: #Numexpr approach
```

```
import numexpr
formula_ne = numexpr.NumExpr("0.1*(data-dark)+1.0", [("data", float), ("dark", float)])
print(formula_ne.input_names)
variance_ne = formula_ne(data, dark)
print(variance_ne.max(), variance_ne.min(), variance_ne.mean(), variance_ne.std() )
```

```
('data', 'dark')
```

```
2.4000000000000004 -0.9000000000000001 1.0004278700086806 0.14051968965805456
```

```
In [7]: %timeit formula_ne(data, dark)
```

```
1.86 ms ± 26.1 µs per loop (mean ± std. dev. of 7 runs, 100 loops each)
```

```
In [8]: variance_ne - variance_np
```

```
Out[8]: array([[0., 0., 0., ..., 0., 0., 0.],  
              [0., 0., 0., ..., 0., 0., 0.],  
              [0., 0., 0., ..., 0., 0., 0.],  
              ...,  
              [0., 0., 0., ..., 0., 0., 0.],  
              [0., 0., 0., ..., 0., 0., 0.],  
              [0., 0., 0., ..., 0., 0., 0.]])
```