

ndpac

March 18, 2020

```
In [1]: import tensorpac
```

```
data, _ = tensorpac.signals.pac_signals_tort(f_pha=4, f_amp=40, sf=100,
                                             rnd_state=42)
pha = tensorpac.spectral.spectral(data, 100, np.array([[2, 6]]),
                                  'pha', 'hilbert', 3, 7, 1)
amp = tensorpac.spectral.spectral(data, 100, np.array([[20, 30]]),
                                  'amp', 'hilbert', 6, 7, 1)
pha.shape, amp.shape
```

```
Out[1]: ((1, 10, 4000), (1, 10, 4000))
```

```
In [2]: tensorpac.methods.ndpac(pha, amp)
```

```
Out[2]: array([[ 0.          , 10.32921285,  0.          ,  5.53261113,
                 6.89715244,  5.42770039,  0.          ,  6.87300059,
                 12.29579344,  9.61393331]])
```

```
In [3]: from scipy.special import erfinv
```

```
def custom_ndpac(pha, amp, p=0.05):
    npts = amp.shape[-1]
    # Normalize amplitude :
    amp = np.subtract(amp, np.mean(amp, axis=-1, keepdims=True))
    amp = np.divide(amp, np.std(amp, axis=-1, keepdims=True))
    # Compute pac :
    pac = np.abs(np.einsum('i...j, k...j->ik...',
                          amp, np.exp(1j * pha)))

    s = pac**2
    pac /= npts
    # Set to zero non-significant values:
    xlim = npts * erfinv(1 - p)**2
    pac[s <= 2 * xlim] = 0.
    return pac
```

```
In [4]: custom_ndpac(pha, amp)
```

```
Out[4]: array([[ 0.          , 0.05081637,  0.          ,  0.03719076,  0.04152455,
                 0.03683646,  0.          ,  0.04145178,  0.0554432 ,  0.04902533]])
```