Lomb-Scargle Algorithm Documentation

Matt James

1 Description of Lomb-Scargle Periodogram

Everything in this document is pieced together from *Lomb* (1976); *Scargle* (1983); *Hocke* (1998). This algorithm will calculate the power, P, amplitude, A, and phase, ϕ, of a wave with a given frequency, f, within an irregularly sampled time series of length n samples.

Let us consider sinusoidal waves of the form

$$y_f(t_i) = a\cos\omega(t_i - \tau) + b\sin\omega(t_i - \tau), \tag{1}$$

where $\omega = 2\pi f$ and $1 \leq i \leq n$.

τ is defined in *Scargle* (1983) as

$$\tan 2\omega\tau = \frac{\sum_{i=1}^{n}\sin 2\omega t_i}{\sum_{i=1}^{n}\cos 2\omega t_i}, \tag{2}$$

which can easily be rewritten as

$$\tau = \frac{\mathrm{arctan2}\left(\sum_{i=1}^{n}\sin 2\omega t_i, \sum_{i=1}^{n}\cos 2\omega t_i\right)}{2\omega} \tag{3}$$

The constants a and b are defined as

$$a = \frac{\sqrt{\frac{2}{n}}\sum_{i=1}^{n} y_i \cos\omega(t+i-\tau)}{\left(\sum_{i=1}^{n}\cos^2\omega(t+i-\tau)\right)^{\frac{1}{2}}} \tag{4}$$

and

$$b = \frac{\sqrt{\frac{2}{n}}\sum_{i=1}^{n} y_i \sin\omega(t+i-\tau)}{\left(\sum_{i=1}^{n}\sin^2\omega(t+i-\tau)\right)^{\frac{1}{2}}}. \tag{5}$$

The periodogram is calculated using

$$P(\omega) = \frac{1}{2\sigma^2}\frac{n}{2}(a^2 + b^2), \tag{6}$$

where $\sigma = \frac{1}{n-1}\sum_{i=1}^{n} y_i^2$ is the variance (assuming that the mean has been subtracted from the data already).

The amplitude is calculated using

$$A(\omega) = \sqrt{\frac{4\sigma^2}{n}P(\omega)} \tag{7}$$

or, equivalently,

$$A(\omega) = \sqrt{a^2 + b^2}. \tag{8}$$

Equation 1 can be expressed differently:

$$y_f(t_i) = A(\omega)\cos\left[\omega(t_i - \tau) + \phi\right], \tag{9}$$

where the wave phase is given by,

$$\phi = -\arctan2(b, a). \tag{10}$$

2 Example Python Code

The code used in this module is based on the following Python code.

```python
def LombScargle(t,x,f):
  '''
  Calculates the Lomb-Scargle periodogram using the method defined in
  Hocke 1998. This method assumes that any mean is removed from the
  data.
  '''

  #preformat the input variables
  t = np.array([t],dtype='float64').flatten()
  x = np.array([x],dtype='float64').flatten()
  f = np.array([f],dtype='float64').flatten()

  #get array sizes
  nf = np.size(f)
  n = np.size(t)

  #create output arrays (Power, amplitude, phase, a, b)
  P = np.zeros((nf,),dtype='float64')
  A = np.zeros((nf,),dtype='float64')
  phi = np.zeros((nf,),dtype='float64')
  a = np.zeros((nf,),dtype='float64')
  b = np.zeros((nf,),dtype='float64')

  #convert f to omega
  w = 2*np.pi*f

  #calculate variance (sigma**2)
  o2 = np.sum(x**2)/(n-1)

  #loop through each frequency
  for i in range(0,nf):
    #calculate sums to get Tau
    ss2w = np.sum(np.sin(2*w[i]*t))
    sc2w = np.sum(np.cos(2*w[i]*t))

    #calculate Tau
    Tau = np.arctan2(ss2w,sc2w)/(2*w[i])

    #calculate w(t - Tau)
    wtT = w[i]*(t - Tau)

    #calculate some more sums
    syc = np.sum(x*np.cos(wtT))
    sys = np.sum(x*np.sin(wtT))
    sc2 = np.sum(np.cos(wtT)**2)
```

```
    ss2 = np.sum(np.sin(wtT)**2)

    #get a  and b
    rt2n = np.sqrt(2/n)
    a[i] = rt2n*syc/np.sqrt(sc2)
    b[i] = rt2n*sys/np.sqrt(ss2)

  #calculate the periodogram
  a2b2 = a**2 + b**2
  P[:] = (n/(4*o2))*a2b2

  #calculate amplitude
  A[:] = np.sqrt(a2b2)

  #calculate phase
  phi[:] = -np.arctan2(b,a)

  return P,A,phi,a,b
```

References

Hocke, K. (1998), Phase estimation with the lomb-scargle periodogram method, *Ann. Geophys.*, *16*, 356–358.

Lomb, N. R. (1976), Least-squares frequency analysis of unequally spaced data, *Astrophysics and Space Science*, *39*(2), 447–462, doi:10.1007/BF00648343.

Scargle, J. (1983), Studies in astronomical time series analysis. ii - statistical aspects of spectral analysis of unevenly spaced data, *The Astrophysical Journal*, *263*, doi: 10.1086/160554.