Sogol Sanjaripour

Amirreza Bahrami Moghaddas

Yousef Jamshidi

The UK Infrared Telescope M33 monitoring project. Variable red giant stars in the central square kiloparsec

In this project the main aim was to identify stars in the very final stage of their evolution, and for which the luminosity is more directly related to the birth mass than the more numerous less-evolved giant stars that continue to increase in luminosity. The most extensive data set was obtained in the *K* band with the UIST instrument for the central 4 × 4 arcmin2 (1 kpc2) – this contains the nuclear star cluster and inner disc. These data, taken during the period 2003–2007, were complemented by *J*- and *H*-band images. Photometry was obtained for 18 398 stars in this region; of these, 800 stars I found to be variable, most of which are asymptotic giant branch (AGB) stars. In this report I present the methodology of the variability survey.

Messier 33 is one of three stereotypical spiral galaxies that inhabit the Local Group. Located in the constellation of Triangulum, it spans about a degree on the sky. Its favourable inclination angle of 56◦ makes it a prime subject for the study of the detailed structure and stellar content of a spiral galaxy like our own. The distance to M33 is approximated to be 2.723 million light years.

Tracing stellar populations of a wide range in ages, from as recently formed as 30 Myr ago to as ancient as 10 Gyr, asymptotic giant branch (AGB) stars (Marigo et al. 2008) are exquisite probes of the star formation history of galaxies. With their high luminosity (≈1000–60 000 L) and low temperature (T ∼ 3000–4500 K), AGB stars dominate the appearance of galaxies at near-infrared (near-IR) wavelengths. This is aided further by the low extinction at IR wavelengths compared to that at optical wavelengths. More massive stars, up to ∼30 M, become red supergiants (RSGs; Levesque et al. 2005; Levesque 2010) and they can be used to trace the more recent star formation history over about 10–30 Myr ago.

Stars at this advanced level of evolution exhibit strong radial pulsations on time-scales of typically 150–1500 d (e.g. Wood et al. 1992; Wood 1998; Pierce et al. 2000; Whitelock et al. 2003). As a result of this pulsation, AGB stars lose up to 80 per cent of their mass to the interstellar medium.

Detecting *variable* AGB stars is a powerful tool in reconstructing the star formation history of a galaxy as these stars are in the final stages of their evolution and hence their luminosity is more directly related to their birth mass than that of less-evolved AGB stars that still undergo significant evolution in luminosity.

There are several ways to identify variable AGB stars, I used equations in Stetson 1993 and 1996 papers to find variable stars among all observations. As it is mentioned observations were in K or J filter depends on each star. Based on data I gathered I designed an algorithm using Python to find variable stars and pick them for next steps in this project.

J index:

$$J = \frac{\sum_{k=1}^{n} w_k \operatorname{sign}(P_k)\sqrt{|P_k|}}{\sum_{k=1}^{n} w_k}.$$

Here, observations i and j have been paired and each pair k has been given a weight w_k; the product of the normalized residuals, $P_k = (\delta_i \delta_j)_k$, where $\delta_i = (m_i - \langle m \rangle\ ))$ is the deviation of measurement i from the mean, normalized by the error on the measurement, i. Note that δ_i and δ_j may refer to measurements taken in different filters.[2] The J index has a large positive value for variable stars and tends to zero for data containing random noise only.
In circumstances where we are dealing with a small number of observations or corrupt data, we gain from also calculating the kurtosis index:

K index:

$$K = \frac{\frac{1}{N}\sum_{i=1}^{n} |\delta_i|}{\sqrt{\frac{1}{N}\sum_{i=1}^{N} \delta_i^2}}.$$

The value of K depends on the shape of the light curve: $K = 0.9$ for a sinusoidal light variation, where the source spends most of the time near the extrema; $K = 0.798$ for a Gaussian distribution, which is concentrated towards the average brightness level (as would random noise); and $K \rightarrow 0$ for data affected by a single outlier (when $N \rightarrow \infty$).

2 Following Stetson (1996), $P_k = \delta^2 - 1$ if $i = j$.

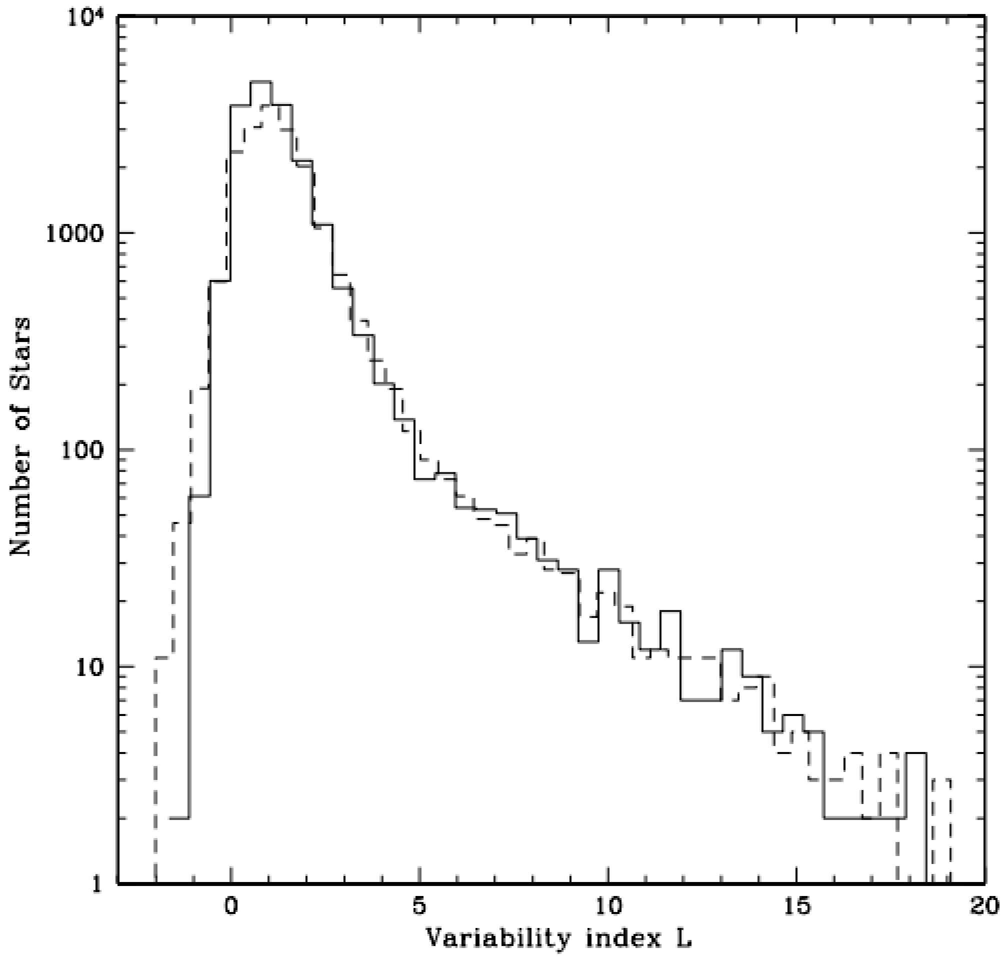
The variability index that I calculate in this project depends on both the *J* and *K* indices and is defined by (Stetson 1996)

$$L = \frac{J \times K}{0.798}.$$

Measurements can be paired if they are taken close in time compared to the (expected) period of variability, which, for the type of stars we search for within the context of this program, is of the order of 100 d or longer. If within a pair of observations only one measurement is available for a particular star, then the weight of the pair for that star is set to 0.5.

I calculated L index for all stars in catalogue, if a star is variable, P_k will be positive; because in observing a star when magnitude in filter K is more than <K>, the same happens in J filter and J is more than <J>. In another case both K - <K> and J - <J> are negative and lead to positive $P_{k.}$ To generalize the amount of L index for variable stars is not zero.

In conclusion by choosing appropriate weight W for each pair in observation based on the quality of data, I plotted the histogram of variability index L using all measured magnitudes for all 18398 observed stars. The final result was incredible, for a galaxy that its stars are all non-variable, the histogram is Gaussian; on the contrary, for M33 galaxy the histogram is not completely Gaussian and for L >4 we can see the distribution of variable stars.



10⁴
1000
100
10
1
Number of Stars
0
5
10
15
20
Variability index L

Programming code used in this project:

```
# -*- coding: utf-8 -*-
"""
Created on Sat May 22 12:46:22 2021

@author: Sogol
"""

import numpy as np
import math



'''
K -> 0
J -> 1
H -> 2
'''

cat1 = np.loadtxt( 'Catalogue1.txt' , dtype = str )
cat2 = np.loadtxt( 'Catal2.txt' , dtype = str )


new_cat = np.zeros((len(cat2) , 5))
new_cat_index = 0

for index,number  in enumerate(cat2[: , 2]):
    if number == '0':# this is K
        #print('here')
        new_cat[new_cat_index , 0] = float(cat2[index , 0])
        new_cat[new_cat_index , 1] = cat2[index , 3]
        new_cat[new_cat_index , 2] = 0
        i = np.where(cat1[: , 0] == cat2[index , 0])
        new_cat[new_cat_index , 3] = float(cat1[i , 7])
        new_cat[new_cat_index , 4] = float(cat1[i , 3] )



        new_cat_index += 1

    elif number == '1':# this is J
        new_cat[new_cat_index , 0] = float(cat2[index , 0])
        new_cat[new_cat_index , 1] = 0
        new_cat[new_cat_index , 2] = cat2[index , 3]
        i = np.where(cat1[: , 0] == cat2[index , 0])
        new_cat[new_cat_index , 3] = float(cat1[i , 7])
```

```
        new_cat[new_cat_index , 4] = float(cat1[i , 3] )



        new_cat_index += 1

    elif number == '2': # this is H
        pass

    #new_cat[: , 0] = cat2[: , 0]

np.savetxt('data.txt', new_cat, delimiter='   ',  fmt='%.3f' )



#calculating delta index for each star
i = 0
delta = np.zeros((len(new_cat) , 2))
observation_number = np.zeros((len(delta) , 2))
K = np.zeros((len(delta) , 2))
for i in range(len(new_cat)-1):
    delta[i , 0] = new_cat[i , 0] #reporting the ID of each star
    observation_number[i , 0] = new_cat[i , 0]
    K[i , 0] = new_cat[i , 0]
    if(new_cat[i , 1] != 0): #observed in filter k
        delta[i , 1] = new_cat[i , 1] - new_cat[i , 3]
    else:
        delta[i , 1] = new_cat[i , 2] - new_cat[i , 4]





#calculating K variable index
i = 0

while(i < 183437):
    sigmadelta = 0
    sigmadelta2 = 0
    n = 1
    while(delta[i , 0] == delta[i+n , 0]):
        n+=1 #The number of observations for each star
    observation_number[i, 1] = n
    for a in range(n-1):
        sigmadelta = sigmadelta + abs(delta[i+a , 1])
        sigmadelta2 = sigmadelta2 + math.pow(delta[i+a , 1] , 2)
    if(sigmadelta2 != 0):
        K[i , 1] =  ((1/n)*sigmadelta)/math.sqrt((1/n)*sigmadelta2)
    i = i + n


P = np.zeros((len(new_cat) , 2))
```

```
x = 0
while(x < 183437):
    if(delta[x , 0] == delta[x+1 , 0]): #observations for the same
star!
        if(new_cat[x , 1] != 0 and new_cat[x+1 , 1] !=0): #both
observations were in filter K
            deltaav = (delta[x ,1] + delta[x+1 , 1]) / 2  #average of
2 deltas in filter K
            P[x , 1] = math.pow(deltaav , 2) - 1
            P[x , 0] = delta[x , 0]
            x+=2
        elif(new_cat[x , 1] == 0 and new_cat[x+1 , 1] == 0): #both
observations were in filter J
             deltaav = (delta[x ,1] + delta[x+1 , 1]) / 2  #average of
2 deltas in filter J
             P[x , 1] = math.pow(deltaav , 2) - 1
             P[x , 0] = delta[x , 0]
             x+=2
        else:
            P[x , 1] = delta[x , 1]*delta[x+1 , 1]
            P[x , 0] = delta[x , 0]
            x+=2
    else:
        x+=1


J = np.zeros((len(delta) , 2))
j = 0
y = 0
while(y < 183435):
    if(P[y , 0] != 0):
        if(P[y , 0] == P[y+2 , 0]): #more than 1 group of observation
            while(P[y , 0] == P[y+2 , 0]):
                j = j + (np.sign(P[y , 1])*(math.sqrt(abs(P[y , 1]))))
                y+=2
            j = j + (np.sign(P[y , 1])*(math.sqrt(abs(P[y , 1]))))
            J[y , 1] = j
            J[y , 0] = P[y , 0] #transfering star id into J matrix
            j = 0
            y+=1
        else: #only 1 group of observation
            J[y , 1] = (np.sign(P[y , 1])*(math.sqrt(abs(P[y , 1]))))
            J[y , 0] = P[y , 0]
            y+=1
    else:
        y+=1
```

```
'''
array1 = np.zeros((len(cat1) , 5))
# id from cat1
array1[: , 0] = cat1[: , 0]
# Jmag from cat1
array1[: , 1] = cat1[: , 3]
# Kmag from cat1
array1[: , 2] = cat1[: , 7]
print ( array1 )
#arrayfloat = array.astype(np.float)
'''
```

References:

Atefeh Javadi,1,2 Jacco Th. van Loon2 and Mohammad Taghi Mirtorabi
1*Physics Department, Alzahra University, Vanak, Tehran 1993891176, Iran*
2*Astrophysics Group, Lennard-Jones Laboratories, Keele University, Staffordshire ST5 5BG*

Stetson P. B., 1993, in Butler C. J., Elliott I., eds, IAU Coll. Ser. 136, Stellar Photometry – Current Techniques and Future Developments. Cambridge Univ. Press, Cambridge, p. 291

Stetson P. B., 1996, PASP, 108, 851