Clustering Similar Points

Overview

One goal of machine learning is to cluster data points by identifying similar points. A common method of clustering is k-means, which we have implemented for our data. To get better clusters, we first ran principal components analysis to compress the data to axes that represent almost all of the data.

K-Means Algorithm

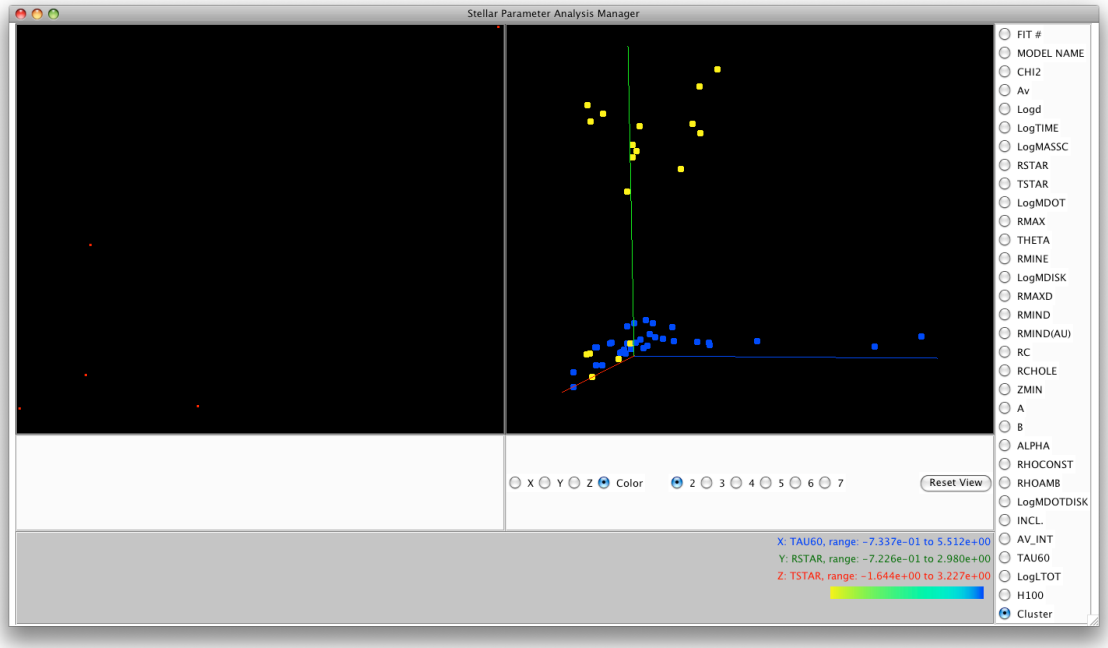
- instantiate k means to start with: our algorithm divides the data into k equal parts, and selects a random point from each to serve as a starting mean. The idea behind this is that the data may already be sorted by some dimension, so this helps to distribute the means.
- while the sum of the distances the means moved is greater than .00001
    - create a list of new cluster means, initially all set to 0
    - for each point in the data
        - set its marker to the cluster it is closest to (based on Euclidean distance)
        - update that cluster's new mean, and its count
    - calculate the distances between each cluster's mean and its new mean
- return list of means and list of markers

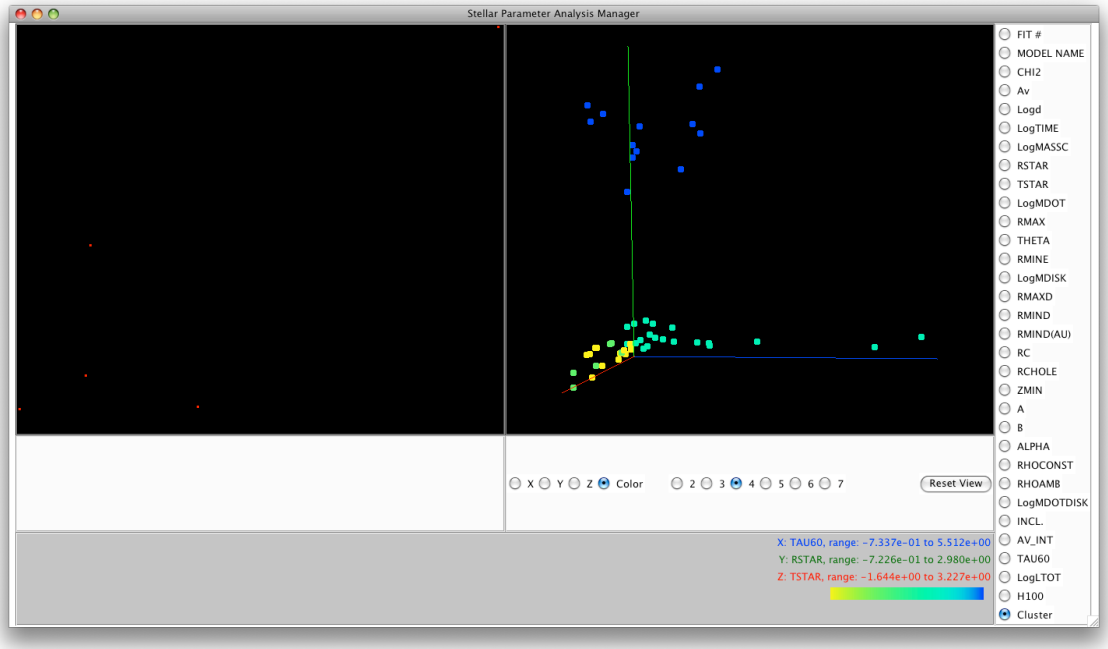
What is Input

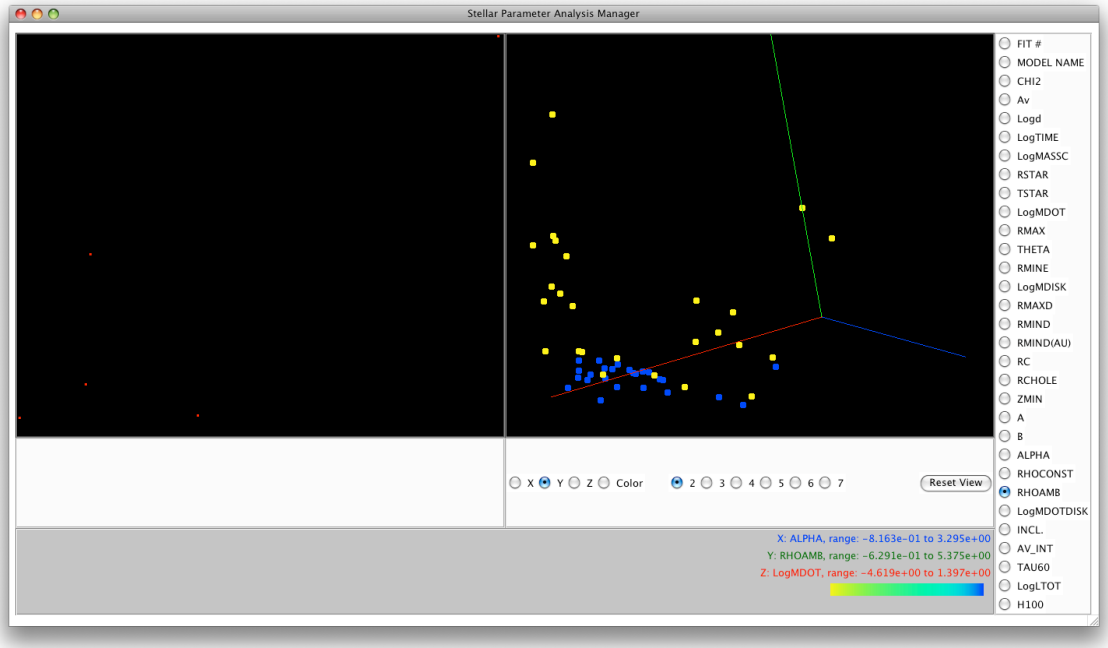
Instead of having the k-means distance metric scale the distances each time (a feat which requires it to remember the mean and standard deviation of each dimension throughout the method), our function expects pre-processed data in which the distances between features are all the same. This means that we subtract off the mean and divide by the standard deviation for each feature before we send the data to be clustered. To simplify things further, the standardized data is run through principal components analysis, and the resulting data in the eigenspace is what is actually clustered. This makes the means returned by the function not very informative, but the labels still correspond to the correct points, which is all that is needed to identify to which cluster each point belongs.

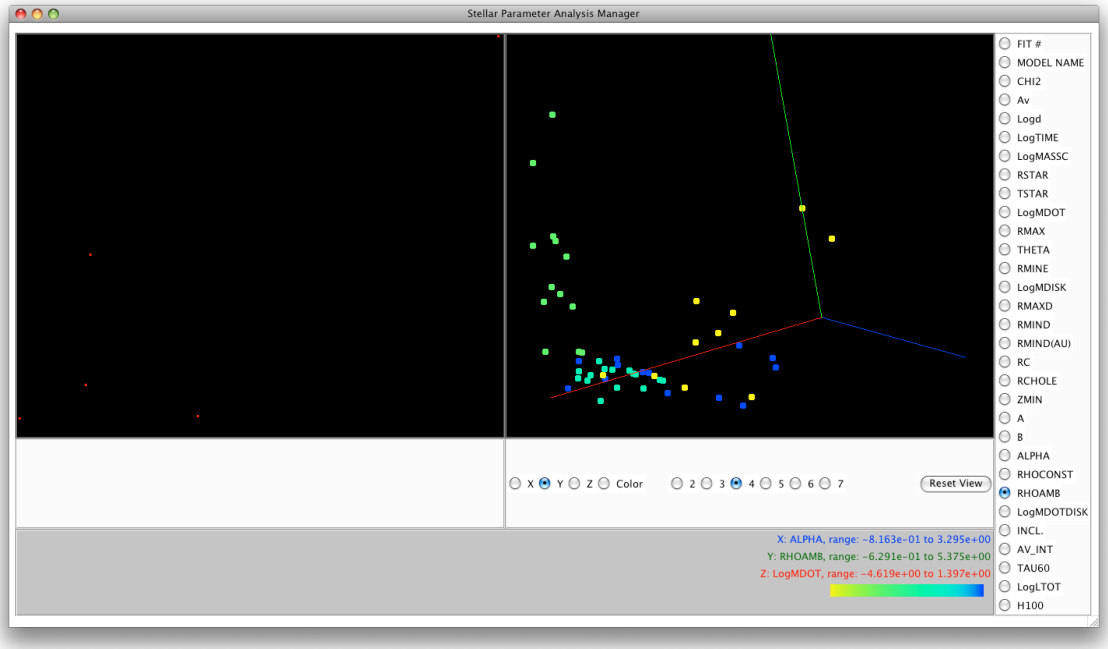
Visualizations

Clusters are categorical, so it is intuitive to visualize them with color. We made "cluster" an option for the color 'axis,' and when selected, more radio buttons appear that allow the user to select how many clusters they would like to see (2-7).




Stellar Parameter Analysis Manager
FIT #
MODEL NAME
CHI2
Av
Logd
LogTIME
LogMASSC
RSTAR
TSTAR
LogMDOT
RMAX
THETA
RMINE
LogMDISK
RMAXD
RMIND
RMIND(AU)
RC
RCHOLE
ZMIN
A
B
ALPHA
RHOCONST
RHOAMB
LogMDOTDISK
INCL.
AV_INT
TAU60
LogLTOT
H100
Cluster
X Y Z Color 2 3 4 5 6 7
Reset View
X: TAU60, range: -7.337e-01 to 5.512e+00
Y: RSTAR, range: -7.226e-01 to 2.980e+00
Z: TSTAR, range: -1.644e+00 to 3.227e+00


Stellar Parameter Analysis Manager
FIT #
MODEL NAME
CHI2
Av
Logd
LogTIME
LogMASSC
RSTAR
TSTAR
LogMDOT
RMAX
THETA
RMINE
LogMDISK
RMAXD
RMIND
RMIND(AU)
RC
RCHOLE
ZMIN
A
B
ALPHA
RHOCONST
RHOAMB
LogMDOTDISK
INCL.
AV_INT
TAU60
LogLTOT
H100
X Y Z Color 2 3 4 5 6 7
Reset View
X: ALPHA, range: -8.163e-01 to 3.295e+00
Y: RHOAMB, range: -6.291e-01 to 5.375e+00
Z: LogMDOT, range: -4.619e+00 to 1.397e+00


Stellar Parameter Analysis Manager
X Y Z Color
2 3 4 5 6 7
Reset View
X: ALPHA, range: -8.163e-01 to 3.295e+00
Y: RHOAMB, range: -6.291e-01 to 5.375e+00
Z: LogMDOT, range: -4.619e+00 to 1.397e+00
FIT #
MODEL NAME
CHI2
Av
Logd
LogTIME
LogMASSC
RSTAR
TSTAR
LogMDOT
RMAX
THETA
RMINE
LogMDISK
RMAXD
RMIND
RMIND(AU)
RC
RCHOLE
ZMIN
A
B
ALPHA
RHOCONST
RHOAMB
LogMDOTDISK
INCL.
AV_INT
TAU60
LogLTOT
H100