Reconstructing the phylogenetic network for galaxy formation

In the network,

Each node in the network is a star cluster or a star formation event.

Say, there are in total *K* clusters, where the *k*th cluster has age (with uncertainties)*,* gas mass used to form stars (hidden variables), and the relative abundance of species (a vector for multiple species with uncertainties) before star forms.

Multiple edges can lead to a node and multiple edges can come out of a node. The weight of an edge linking two nodes are the proportion of gas in the younger star cluster that was ejected from the older star cluster.

Since the younger cluster may receive gas from any older cluster, then is a linear combination of , the amount of species ejected from an older cluster , , where the coefficient is the proportion of gas in cluster , which, divided by gives the edge weight from node to 𝑘 in the network.

is pristine gas (75% H and 25% He), so gives the proportion of gas in the cluster that is from the infall.

As a result, the network is completely determined by the ages of clusters. For example, if we rank clusters from oldest to youngest, say T1 > T2 >…> TK, and summarise the weight of edges as a matrix with same order of row and column corresponding to 1,2,…K clusters, so that row column gives , where , then all the edge weights we need to estimate are all the elements in the upper diagonal of the matrix. Adding as the first row, then in total, we have (K+1)K/2 edges.

For a cluster, to calculate its from its age , gas mass and species abundance , we use the yield table. See SN\_and\_AGB\_yields\_Wiersma+09a.zip file.

Given a bin of star mass , we have the following from tables:

1) IMF for the mass bin:

2) the yield of species *i*:

3) the total ejected mass

4) the delay time , or the min time of DTD

5) cumulative function of DTD, for between 0.85 and 8 .

\*note that 4) and 5) needs to add another time delay for gas cooling.

In the cluster, the expected total number of stars of mass bin is , and the number of stars of mass bin , ~. Since is a large number, we can approximate this Poisson distribution as a normal distribution, so ~.

Then the amount of a species’ mass ejected by stars at time (count backwards in time) are weighted sum of normals:

If , other wise .

For change in species ’s mass:

, where and is probability of star mass between 3 and 16.

See Yates et al. 2013 eqn 5 for details.

So, the amount of all the species generated till time are a multivariate normal:

, where has diagonal and the off-diagonal element being .

Some clusters have inhomogeneous gas, reflected by the variation in species abundance in different stars in the clusters. To account for this inhomogeneity, say observed stars in the cluster with abundance of species ,,…, , our best guess is proportion of gas has abundance proportion of gas has abundance , etc. So,

and

, where has diagonal and the off-diagonal element being .

This is the function to calculate the distribution the amount of species ejected from a star cluster given the current time , the cluster age , the cluster gas mass , and the cluster species abundance . Let’s call this function calY(t, T, G, X) in the pseudo-code below.

In the pseudo-code, we use MCMC to integrate out the hidden variables G and estimate T and matrix A.

Data: see abusolar\_cluster\_giants.csv for T\_m and T\_sd, and see abusolar\_cluster\_giants\_individual stars.csv for X[i]

T\_m= A vector of mean cluster age.

T\_sd= A vector of sd cluster age.

X[i] = A matrix of species abundance of stars in cluster i with row as stars and col as species.

Variables:

for K clusters:

T= a vector of cluster age

G= a vector of cluster gas mass

A= matrix of edge weights

Y\_m[i]= a vector of mean species ejected from a cluster i

Y\_sig[i]= covariance matrix of species ejected from a cluster i

logLik = log likelihood value

logPr= log prior value

Prior:

Pr\_T[i]= Prior for T[i] is a normal distribution with mean T\_m[i] and T\_sd[i], truncated so that T[i]>=0.

Pr\_G= Prior for all Gs shares the same prior, say a lognormal with mean G\_m and G\_sd. To facilitate convergence, let’s discretise the prior into N bins of equal probability, so that we only need to sample bin index for each cluster, and the corresponding G for the ith bin is (i-0.5)/N quantile of the lognormal distribution.

Pr\_A= Prior for each element in A is an exponential distribution with rate A\_m, with larger A\_m corresponding to stronger coefficient of L2 norm. If we assume clusters with similar radius are more likely to share gas, then we can set A\_m as inverse of the absolute difference in radius between two clusters.

Initialization:

T=T\_m

G\_idx= a vector of a random sample of bin index from Pr\_G for a cluster

G = the (G\_idx-0.5)/N quantiles of Pr\_G

A= matrix (0, nrow=K+1, ncol=K), with the first row as pristine gas and the remaining the cluster index ranked by T, so oldest cluster first, col are cluster index ranked by T.

To initialise A:

For each cluster i,

Set Y\_m[1] as the species abundance in pristine gas (75% H and 25% He)

Calculate Y\_m[j+1] and Y\_sig[j] of each cluster j that is older than i, using calY(T[i], T[j], G[j], X[j])

Calculate the upper diagonal part of the ith col of A as the solution of b in linear equations: mean of X[i]=Y\_m\*b, which is the Moore-Penrose inverse for linear equations.

Set all b<0 to 0.

This initialisation of A approximates its maximum likelihood estimates

Calculate logLik of cluster i as the sum of log of normal pdf over each star in cluster i, i.e.,

Sum logLik over all clusters as the final logLik.

Calculate logPr as

Recursion:

In each MCMC,

1. Update age of each cluster in T using Metropolis Hasting and sliding window proposal:

For each T[i],

Draw a random value u from a uniform distribution on interval (0,1).

Proposal a new T[i], T’[i]=T[i]+(u-0.5)\*delta, where delta is a tuning parameter to adjust the how different the new value is from the old value.

If T’[i] <0, reflect if as T’’[i] = -T’[i]

If T’[i] change the order of cluster age, then reorder the row and column of A matrix to generate a new A’ matrix according to the new order of cluster age; for element falls in the lower diagonal in A’, transpose the element to the upper diagonal.

Calculate new logLik’ and new logPr’ from this new T[i]’ and/or new A’

Calculate acceptance probability

Draw a random value from a uniform distribution on interval (0,1),

If , then accept the new value, set T[i]=T’[i] and A=A’ if changed.

Otherwise, reject the new value.

1. Update each element in the upper diagonal of A using Metropolis Hasting and sliding window proposal:

Draw a random value u from a uniform distribution on interval (0,1).

Proposal a new A[i,j], A’[i,j]=A[i,j]+(u-0.5)\*delta, where delta is a tuning parameter to adjust the how different the new value is from the old value.

If A’[i,j] <0, reflect if as A’’[i,j] = -A’[i,j]

Calculate new logLik’ and new logPr’ from this new A’[i,j]

Calculate acceptance probability

Draw a random value from a uniform distribution on interval (0,1),

If , then accept the new value, set A[i,j]=A’[i,j].

Otherwise, reject the new value.

1. Update bin index of gas mass of each cluster in G

Pick a random pair (i,j) of clusters and swap their bin index.

Calculate new G’[i],G’[j]

Calculate new logLik’ from this new G’[i],G’[j]

Calculate acceptance probability

Draw a random value from a uniform distribution on interval (0,1),

If , then accept the new value, set G[i]=G’[i] and G[j]=G’[j]

Otherwise, reject the new value.

Run a large number of MCMC until convergence.

Draw a network from each MCMC sample of T and A, with T determines the location of a node along time axis and the upper diagonal of A determines the edge weight between any two nodes.

When all the networks are overlayed with each other, we may find a dominant network with the highest density. This is the network we want.