Chapter 1

Methods

1.1 Purpose and goals

In this thesis the goal is to examine the influence of uncertainty in models and parameters with regards to r-process nucleosynthesis. This in turn, is used to estimate the influence of uncertainties with regards to the Rhenium-Osmium cosmic clock. This will be done by comparing a semianalyical galactic chemical evolution model (`Omega`) with data from a high resolution smoothed particle hydrodynamics simulations (`Eris`).

`Omega` is used because of it's simplicity and versatility; it can be executed as a python script on most computers, takes relatively little time to run and easily modifiable parameters. `Omega` also demonstrates a much larger resolution in mass, by dividing mass into a wide range of nuclei.

The smoothed particle hydrodynamics simulation `Eris` is a high resolution simulation that resembles the Milky Way Galaxy in many aspects, and is therefor a great candidate for a Milky Way Proxy. Assuming that the evolution of `Eris`also resembles the evolution of the Milky Way allows us to use the star formation history and baryonic content data from `Eris` in order to match the generated data from `Omega`. Assuming that `Eris` is a good approximation to the Milky Way morphological history, the data from `Eris` are assumed to represent the mass and abundances of the Milky Way galaxy. By finding a model of `Omega` that most accurately reproduce the mean-values of `Eris`[1], a Milky Way-proxy (that resolves $^{187}_{75}$Re and $^{187}_{76}$Os isotopes is found.

After an appropriate parameter-space is estimated, the yields of some relevant nuclei (Eu-, Re-, Os-, W-isotopes available) can be varied within their observational uncertainties. An semianalytical model, like `Omega`, will merely calculate

[1]Since `Omega` is a one-zone model without spatial dimensions, it can only approximate the mean of `Eris`. Not the spatial resolution.

the total amount of ejecta from stars (yield tables) and integrate the ejecta in time. Finding the effect of the observational uncertainties is done by changing the yields according to the observational uncertainties and tracking the difference through galactic time.

1.2 Fitting Omega to Eris-data

To fit the Omega model to Eris, the star formation rate calculated for each timestep is used as the star formation history in Omega. This ensures that for sufficiently large timesteps Omega will use the same star formation rate as Eris (however interpolated onto the timesteps used in Omega). Otherwise, the parameters of Omega are examined, in order to best reproduce the total baryonic content of the simulation, as well as the spectroscopic abundance of [O/H], [Fe/H], and [Eu/H].

In this section *Omega-MilkyWay* refers to the built-in options for a Milky Way-like model in Omega, while *Omega-MilkyWay-c.sfr.* is similar, but assumes a constant star formation rate. *Fiduccial Omega-model* will be represent the parameter-space of Omegathat will reproduce data from Eris.

1.2.1 Insertion (rename!)

Galactic age, initial mass function, astrophysical effects and star formation are so essential to chemical evolution models that they will be taken from Eris and applied directly to Omega. These values are listed in table 1.1.

'Eris' refers to data directly from Eris-simulation. *'Omega' default* (or *Vanilla Omega-model*) refers to the Omega model with no change to the initial parameters (see description in section ??). *'Omega' MW* (or *Omega-MilkyWay*)refers to the Omega model with the Milky Way parameter (see description in section ??), and *'Omega' MW cte* (or *Omega-MilkyWay-c.sfr.*) refers to the same but with a constant one-solar-mass-per-year star formation rate. *'Omega' w/'Eris'-SFR* (or *Fiduccial Omega-model*) refers to the Omega model with the star formation and mass function from Eris. This model, *Fiduccial Omega-model*will be exaned upon throughout the fitting process.

It is clear from figures 1.1 and 1.2 that star-formation is suppressed before 2Gyr for all models except *Vanilla Omega-model*.

footnote not showing

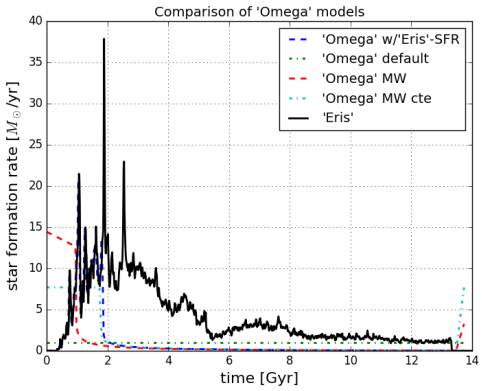


Figure 1.1: Star formation rate(measured in solar masses of stars formed from gas each year) for four models of `Omega` versus time.

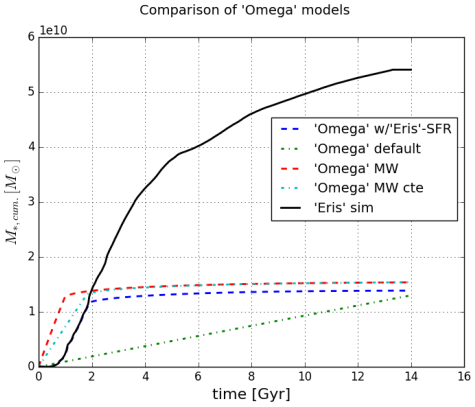


Figure 1.2: Total accumulated stellar mass(cumulutaive sum of stellar mass produces from gas, measured in solar masses) for four models of `Omega` versus time.

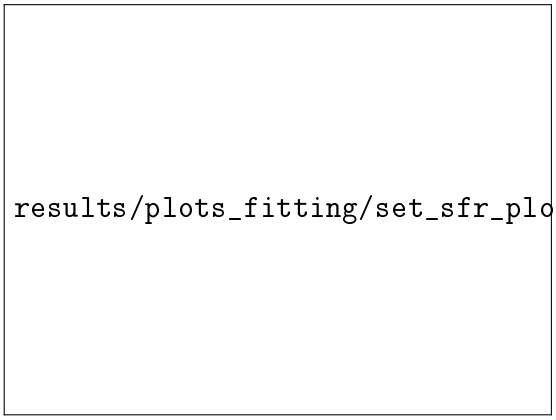


Figure 1.3: Replot this image, looks sketchy!

Mass of gas in the interstellar medium for four different models of `Omega`, and the `Eris` simulation against time in Gyrs.

Parameter	`Omega`-argument	Value taken from `Eris`
Galactic age	`tend`	14 Gyr
Star formation rate	`sfh_array`	Star-formation rate $[M_\odot/yr]$[2]
Initial mass function	`imftype`	Kroupa93[3]
Type Ia supernovae	`sn1a_on`	yes
Binary neutron star mergers	`nsmerger_on`	yes
Yield-table for binary neturon star mergers	`ns_merger_table`	*[Arnould et al., 2007]*
Neutron star - black hole merger	`bhns_merger_on`	no

Table 1.1: The yield-table for binary neutron star mergers was chosen because it was the only r-process yield-table (out of two available) with the presence of $^{187}_{75}$Re . In `Omega`, r-process yield tables are applied to neutron star mergers, and the yield-table chosen is calculated from the r-process abundance of the Solar systemat present day (*[Arnould et al., 2007]*). This means that all neutron star mergers are assumed to produce a r-process pattern like the one found in the Solar system.

1.2.2 Mass

f_b [ ]	z[ ]	$M_{vir}[10^{11}M_\odot]$	$M_b[10^{10}M_\odot]$	t[Gyr]
0.121	0.0	7.9	9.6	13.724
0.126	1.0	5.4	6.8	6.075

Table 1.2: From Guedes10 table 1 , f_b is the baryonic mass fraction of the galaxy, z is the redshift in the simulation, M_{vir} is the virial mass of the halo, M_b is the total baryonic mass within the halo(multiplication of f_b and M_{vir}), t is the time of the corresopnding redshift. Time is calculated from redshift using Ned Wright's cosmology calculator(February 12th 2018) reference to cosmology calculator article here with the cosmological parameters, $H_0 = 73[kms^{-1}Mpc^{-1}]$, $\Omega_M = 0.24$, and $\Omega_\Lambda = 1 - \Omega_M = 0.76$ for a flat universe as stated in *[Guedes et al., 2011]*.

A galaxy's mass consist of gas and stars (not counting dark matter from section **??**), and in a one-zone model there is only gas and stars inside the box and gas outside the box (see figure **??**). The total baryonic mass inside `Eris` (at redshift z=0 and z=1) is taken from *[Guedes et al., 2011]* and can be found in table 1.2.

Supernova feedback will drive an outflow from the galaxy into the surrounding medium find appropriate reference . Adding outflow proportional to the supernova rate adds some realisim to the model, and might reproduce some of the spectroscopic features. In `Omega`this is activated with the parameters `mass_loading`(which ejects a amount of gas relative to the stellar mass formed), and `out_follows_E_rate`(which adds a timedelay to the outflow, making the outflow proportional to the supernova rate instead of the star formation rate). Outflow removes gas from the galaxy, or interstellar medium, lowering the total amount of mass in the galaxy. Therefore the initial primordial gas and constant inflow must be increased as well. Setting the initial mass, inflow and outflow, gives the desired mass content. A final comparison of the *Fiduccial `Omega`-model*model, the predefined models (*`Omega`-MilkyWay*, *`Omega`-MilkyWay-c.sfr.*, and `Omega`with all default parameters), and the data from the `Eris` simulation. For the predefined models, the initial mass of primordial gas have been increased to $9.6 \times 10^{10}M_\odot$(the final value baryon-mass from `Eris`) to show the full evolution of star formation. Two prominent features in the spectroscopic data of `Eris` is the two 'dips' around universal time t=5 and t=8 Gyrs. These dips might be reproduced by adding primordial inflow, hydrogen, and enriched outflow, concentrated on on those periods($t \simeq 5Gyr$ and $t \simeq 8Gyr$), since these periods might coincide with the death of stars from the star forming peak in figure 1.1. Varying supernova-related outflow(known as the `mass_loading` parameter) gives an expected result. In figure 1.8 variation in the spectroscopic iron abundance can

be seen the desired region, around the two 'dips', but the effect is too small to reproduce the two dips. The effect is also too wide and more closely similar to one big 'dip'. One unexpected result is that the smalles `mass_loading` parameter yields the lowest dip(not really a dip at all, but more flat in the desired direction). suggesting that the outflow from supernovae occur later than the two 'dips'. This means that the two 'dips' cannot be reproduced by outflow and inflow.

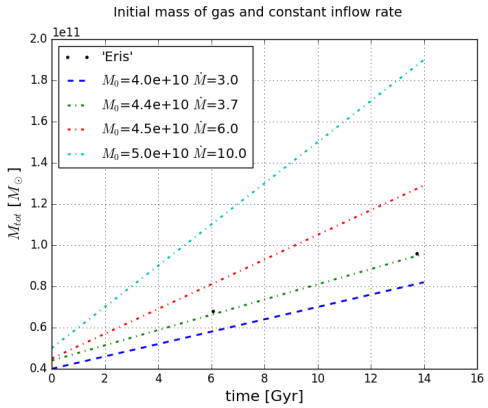


Figure 1.4: The total baryonic mass of the `Omega`-model for four different initial/inflow parameters. M_0 is the initial primordial gas of the galaxy(in $M_\odot$), $\dot{M}$ is the inflow (in $M_\odot$ /yr). This visualization shows that 44G $M_\odot$ and 3.7 $M_\odot$ /yr are the optimal parameters to reproduce the two baryonic data-points from `Eris`, although more then these four were tried.

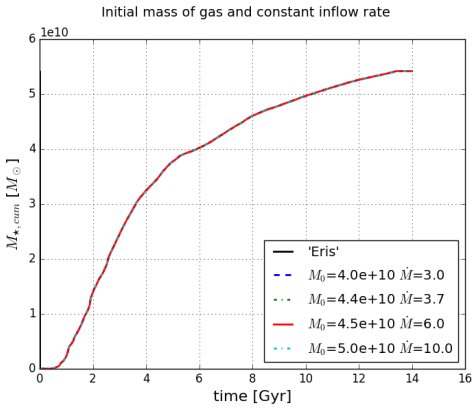


Figure 1.5: Plotting the cumulative stellar mass formed in the inflow-`Omega`-models. All four reproduce the `Eris` cumulative star formation, because these models have enough gas to form the stars.

Table 1.3: add table of parameter

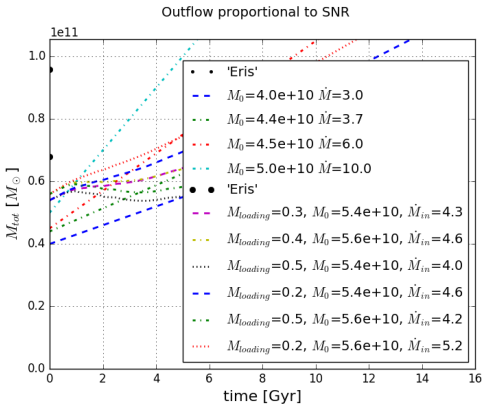


Figure 1.6: Adding outflow from supernovae feedback. Total baryonic mass of galaxy over time. what is the initial mass of gas and inflow rate? The outflow adds a non-linear effect to the total mass.

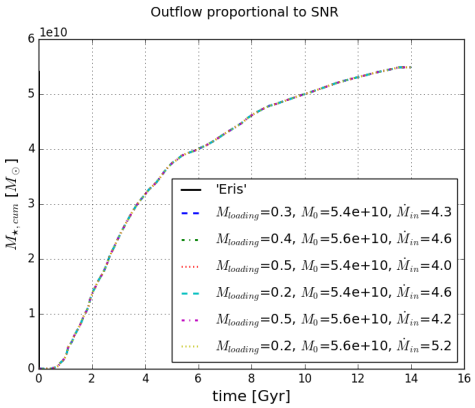


Figure 1.7: Adding outflow from supernovae feedback. Cumulative stellar mass formed against time for X Omega models, and the Eris-simulation. The outflow removes mass, but there is still enough gas to form stars from the Eris star formation rate.

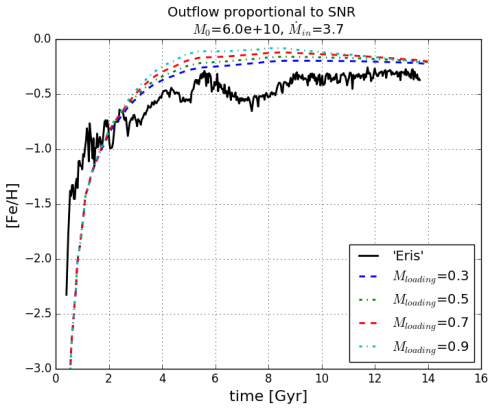


Figure 1.8: Considering a different initial mass, constant inflow of gas andn adding outflow from supernovae feedback. Iron abundance in the models with varying mass-loading parameters(solar masses of outflow per solar mass of supernova). The data from Eris show two 'dips' from the increasing tendency. The dips could not be reproduced by outflow of enriched material and inflow of hydrogen. Outflow peaks over the 'dips', reducing the spectroscopic abundance, however the effect is wide and smeared out over a time range beyond both 'dips'.

1.2.3 Effect of stars and supernovae

Chemical enrichment of galactic gas (the interstellar medium), comes from stars. Hydrogen and helium from the primordial gas is locked into a star, where fusion processes transmutates the elements into heavier elements up to iron. In the process some heavier elements are created, mostly by neutron capture processes. At the end of the stars life some of the material will be ejected back into the interstellar medium. Asymptotic giant branch stars are low mass stars at the end of their life, they eject mass via episodes known as helium flashes, leaving a white dwarf behind. Massive stars end their life as typeII supernovae, ejecting most of their enriched material leaving a neutron star or black hole behind. The very first stars, with no initial chemical enrichment, or metallicity, are called population III stars. They are generally believed to have a slightly different initial mass distribution function, and could produce slightly different distributions of metals. The exact science of population III stars is not well defined, as none has been observed, but the stellar population is one of the options of the `Omega` model and should therefor be taken into consideration when comparing `Eris`and `Omega`. The remnants, white dwarves, neutron stars and black holes, are not the end of the story, binary star systems can bring new life to these dead bodies. A white dwarf accreting plasma from the envelope of a binary star can accumulate enough mass to ignite a core-collapse that ejects more enriched matter into the interstellar medium. Two neutron stars in orbit around eachother can loose gravitational energy to gravitational waves and merge. Such an energetic event will create alot of heavy elements and eject alot of the mass of the binary system with great velocity. Similar gravitational events can occur between two black holes and a black hole and a neutron star. The last two event will be ignored because two black holes do not create or eject any heavy elements (or any elements at all), while black hole neutron star merger is not included in `Eris`.

This section will look at the effect of yield-tables for massive stars and asymptotic giant branch stars, the transitionmass between medium mass stars and massive stars, the boundaries of the initial mass function of population III stars (the first metall-less stars) and their yield tables. The values and tables available does not significantly alter the spectroscopic abundance of oxygen, iron, or europium, and the plots are therefor not shown here. In order to look at the effect of type 1a supernvoae, the number of supernovae, their delay-time distribution are varied. The yield-tables available for type 1a supernvoae give relatively similar results for oxygen and iron, so the plots are not shown here.

The delay-time distribution can be chosen from a gaussian, exponential or powerlaw distribution. All were tried, but only the exponential distribution is shown in this section.

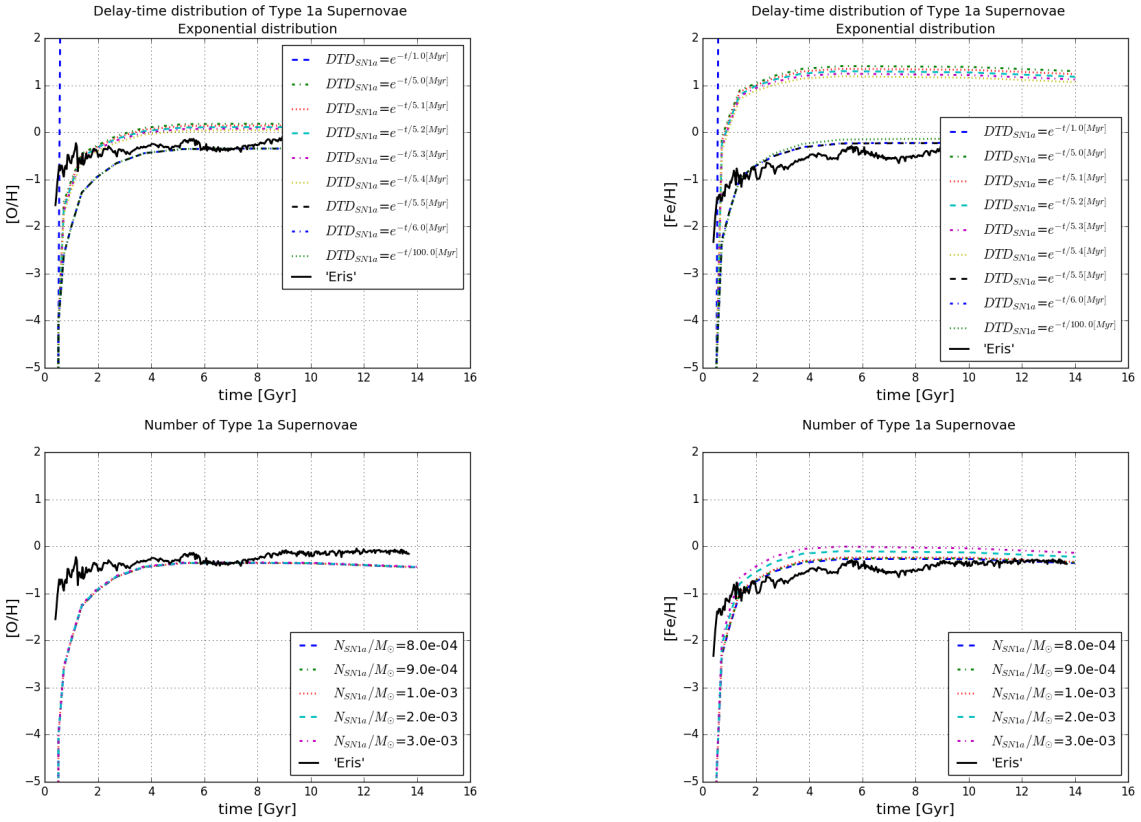


Figure 1.9: Spectroscopic abundance of oxygen and iron in `Omega`compared to the data from `Eris`.
TODO! Add more description

Table 1.4: add table of parameter

1.2.4 Nucleosynthesis from r-process

Only binary neutron star mergers are considered, because this is the same site as *[Shen et al., 2015]*

Table 1.5: add table of parameter

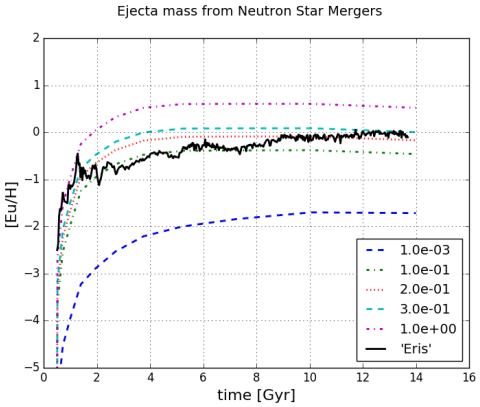


Figure 1.10: Spectroscopic europium abundance against galactic time for `Eris`-data and several `Omega`models. In the models the mass ejected from each neutron star merger have been modified. Modifying the mass ejected from each event will just scale the total europium content up and down. Ejecting 0.2-0.3 $M_{\odot}$ per event gives a pretty decent fit to late time europium and early time europium. However for the 'dips' between 2 and 8 Gyrs, the `Omega` model overshoots the `Eris` data.

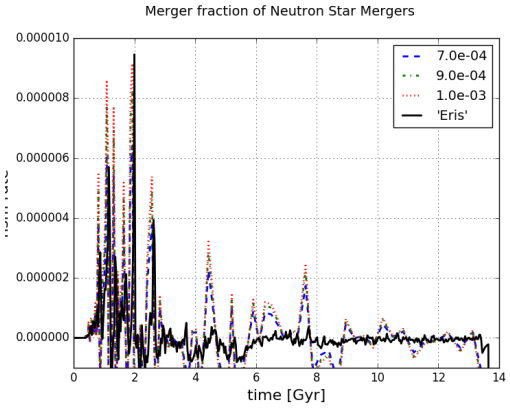


Figure 1.11:]
TODO! Add more descriptive text

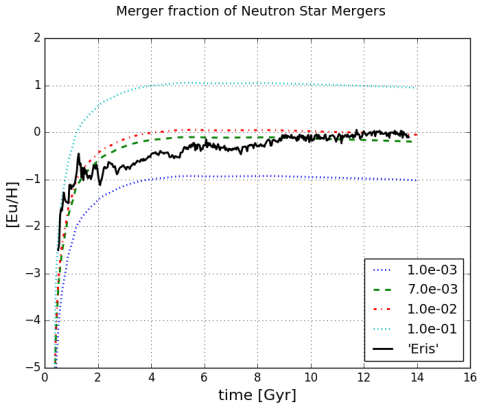


Figure 1.12: TODO! Add more descriptive text

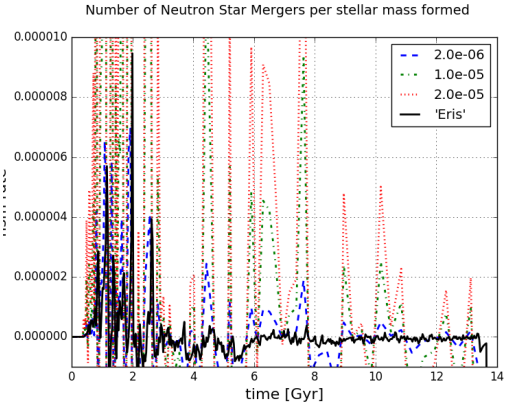


Figure 1.13: TODO! Add more descriptive text

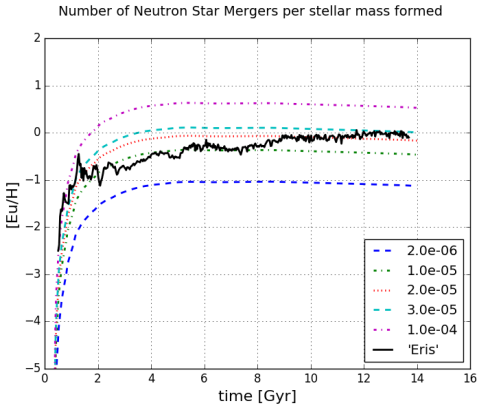


Figure 1.14: TODO! Add more descriptive text

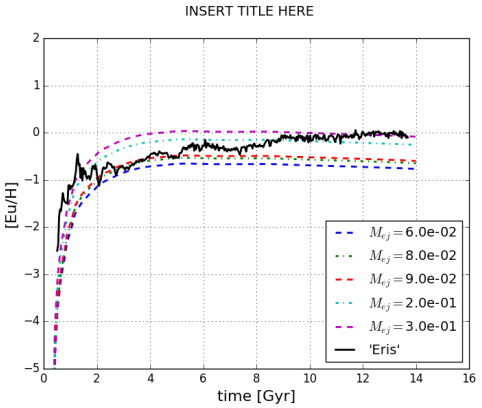


Figure 1.16: TODO! Add more descriptive text

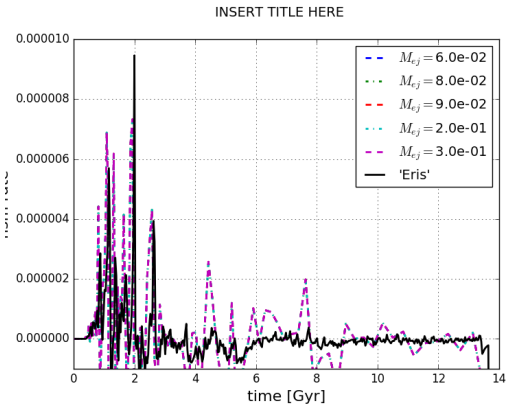


Figure 1.15: TODO! Add more descriptive text

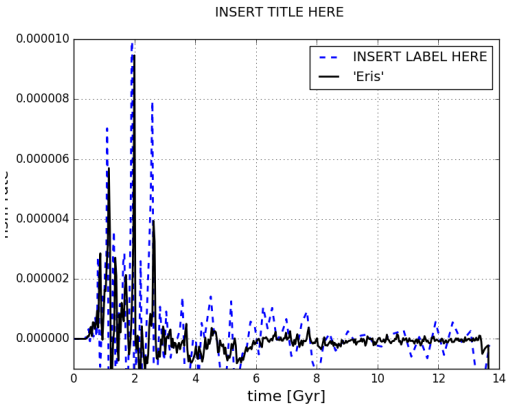


Figure 1.17: TODO! Add more descriptive text

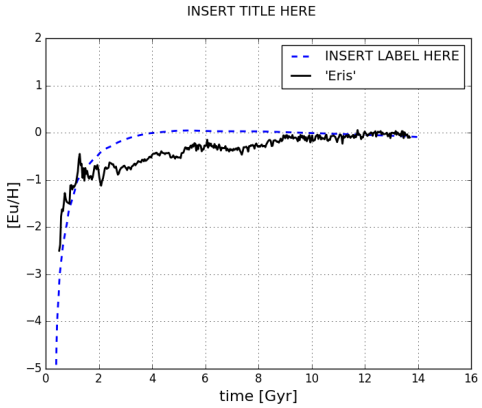


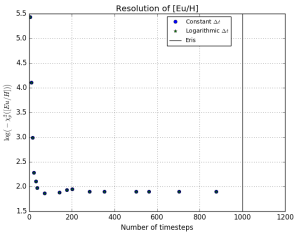
Figure 1.18: TODO! Add more descriptive text

1.2.5 Size of timesteps

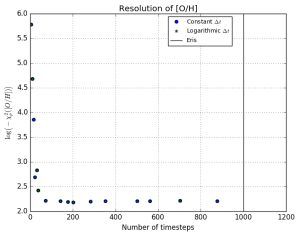
In the fitting process the timestep was chosen somewhat arbitrarily in order to give the best result in a realistic amount of time for working purposes. The size of the timesteps are important for accuracy of the result of a numerical calculation, longer and fewer timesteps give unaccurate results, but too short and too many timesteps take much memory and comupting time. numerical presicion and floating point arithmetic can also come into play at the shorter timestep simulations.

In order to check the accuracy of `Omega` against `Eris` a comparison is needed. `Omega` uses two different timestepping tecnhiques, one with n logarithmic timesteps between start and end, and one with constant timesteps between start and end. By varying the number of timesteps in the `Omega` data for [O/H], [F/H], [Eu/H] in time will be created. These data will be interpolated onto the time-data for `Eris` and compared to the `Eris`-data with a pearson chi-squared test. The pearson static is only used to compare the difference with increasing time-resolution, the chi-squared statistic will not be used to compare what is, in essence, two different simulations.

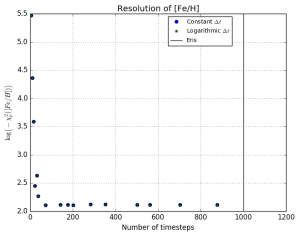
It should be noted that there were some peculiar difficulties discovered at this point. `Omega` cannot create a stable list of timestep-values when the timesteps are shorter then the time-array in the input star formation rate (which, in this case belongs to `Eris`). The same problem also occured for some timestep-sizes close to the timestepsize of `Eris`. This is the reason for the sparse datapoints at higher resolutions.



(a) TODO!



(b) TODO!



(c) TODO!

1.2.6 Final parameters of Omega

Omega-parameter	Description	Value

Table 1.6: TODO! Parameters that produce best fit between Omegaand Eris-data.

1.3 Impact of stellar yields

The Omega model with Eris bestfit parameters is used to calculate the amount on $^{187}_{75}$Re . A single function of Omega is overwritten in order to change the stellar yields of $^{187}_{75}$Re . The function multiplies the stellar yield of $^{187}_{75}$Re ($Y_{^{187}_{75}\mathrm{Re}}$) with a constant, the new yield is denoted $\hat{Y}_{^{187}_{75}\mathrm{Re}}$.

The yield table for binary neutron star mergers is taken from [Arnould et al., 2007] and is the calculated distribution of r-process isotopes in the Solar system. The Solar system-distribution of isotopes is measured from the solar photosphere and chondrite meteorites in [Palme and Beer, 1993]. The s-process distribution can be calculated from nuclear reaction networks, and the r-process/p-process distributions are calculated by subtracting the s-process distribution after fitting to the Solar system-abundances. The basic assumption is that all neutron star mergers eject material with the same isotopic distribution, and this distribution matches the observed solar distribution. In short, the yield-table used is the *observed* r-process distribution of the Solar system, and this distribution is applied to every neutron star mergers.

In the Eris-postprocessing (*[Shen et al., 2015]*) the yield-table applied to all neutronstar mergers is a different one. The yield table was based on simulations by (some reference to Rosswog) of neutron star mergers. Those yield-tables however does not contain any output of $^{187}_{75}$Re , which is essential for the purpose of this thesis.

What are the underlying uncertainties of the yield-tables that we so generously apply to our simulations? The yield-tables represent the distribution of isotopes and elements ejected from a star during the end of it's lifetime. This information comes from stellar evolution codes like [Paxton et al., 2011] and the nuclear reaction networks applied to them, like [Pignatari et al., 2016]. Uncertainties in nuclear reaction rates, stellar interior environments and ejecta composition all add together to create a total uncertainty of yield tables which are hard to separate. In this case, the neutron star mergers are *assumed* to be the dominant contributer to r-process isotopes in the galaxy, and the distribution is *assumed* to follow observed r-process distribution in our Solar system(this is ultimately

an assumption of homogeneity). Following these assumptions, the uncertainties in nucleosynthesis of r-process isotopes will simply be the uncertainty Solar system-observations[Arnould et al., 2007]. Simulations suggest (some reference to Rosswog) that the ejecta yield of binary neutron star mergers are somewhat dependant on the electron-fraction of the initial neutron stars, not taking inhomogeneities, rotation, and magnetic field of the initial neutron stars into account. Also given the rarity of kilonovas (compact object mergers) the assumption of homogeneity is questionable.

In order to manipulate the program `Omega` without changing the source code, which can be found on website to omega , inheritance of python-classes was used.

By creating a new python-class that inherits all methods from `Omega` (which inherits many methods from `Chem_Evol`), and making a new function with the same name as the function that sets the yield tables, the old funciton is overridden by the new. The old function in `Chem_Evol` is called `__set_yield_tables`, and the new function, `chem_evol__set_yield_tables`, has all the same content and overrides the old function. By adding a few lines of code to the end of the yield-table function, a 'fudge factor' is multiplied to the yield of a single isotope across all yield-tables used. The extra lines of code are shown in listings 46.

```
##########################################################
### End of function as written in 'chem_evol.py' ###
"""
Change ytables(multiply yields of 'isotope' with 'factor') with
value of self.experiment_factor to isotope corresponding to self.
    experiment_isotope
"""
##########################################################

#AGB + massive stars, and pop3 stars
#loop over the different objects
for table_object, table_name in zip([self.ytables, self.ytables_pop3], ["agb/
    massive", "pop3"]):
    #get list of available metalicities
    loa_metallicities = table_object.metallicities
    for Z in loa_metallicities:
        #get list of masses for each metallicity
        loa_masses = table_object.get(Z=Z, quantity="masses")
        for M in loa_masses:
            #get current yield
            try:
                present_yield = table_object.get(M=M, Z=Z, quantity="Yields",
    specie=self.experiment_isotope)
            except IndexError: #isotope doesn't exist for this table
                continue
            #modify yield by some factor
            new_yield = present_yield*self.experiment_factor
            #"insert" new yield back into table
            table_object.set(M=M, Z=Z, specie=self.experiment_isotope, value=
    new_yield)

# SN1a, NS-NS merger, BH-NS merger
# get index of isotope
index_iso = self.history.isotopes.index(self.experiment_isotope)
```

```
#loop over different objects
for table_object, table_name in zip([self.ytables_1a, self.ytables_nsmerger,
    self.ytables_bhnsmerger], ["sn1a", "nsm", "bhnsm"]):
    #get list of available metalicities
    loa_metallicities = table_object.metallicities
    #loop over metallicities
    for i_Z, Z in enumerate(loa_metallicities):
        #get current yield
        try:
            present_yield = table_object.yields[i_Z][index_iso]
        except IndexError: #this means that isotope doesn't exist for this table
            continue
        #modify yield by some factor
        new_yield = present_yield*self.experiment_factor
        #"insert" new yield back into table
        table_object.yields[i_Z][index_iso] = new_yield
```

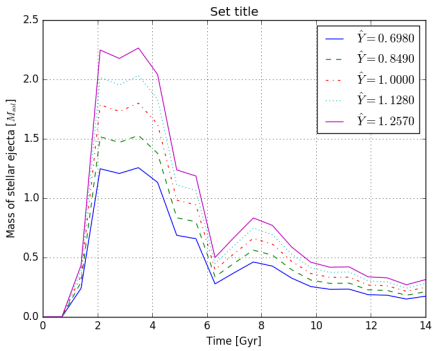
Listing 1.1: Snippet of code added to the existing function `__set_yield_tables` in `Chem_Evol` in `Omega`-framework. The code-snippet multiplies the yield of isotope `self.experiment_isotope` with a factor `self.experiment_factor` for all yield-tables where the isotope can be found.

isotope	standard	min	max	σ_{lower}	σ_{upper}
Re-187	0.0318	0.027	0.0359	-0.1509	0.1289
Re-185	0.0151	0.011	0.0176	-0.2715	0.1656
Os-188	0.0707	0.0633	0.0781	-0.1047	0.1047
Os-189	0.103	0.0961	0.109	-0.067	0.0583
Os-190	0.152	0.137	0.168	-0.0987	0.1053
Os-192	0.273	0.252	0.289	-0.0769	0.0586
Eu-151	0.0452	0.0267	0.0482	-0.4093	0.0664
Eu-153	0.0495	0.046	0.0526	-0.0707	0.0626

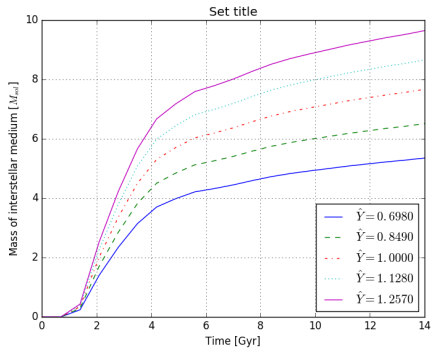
Table 1.7: Values and uncertainties of r-process nuclei near $^{187}_{75}$Re from *[Arnould et al., 2007]*. The relative uncertainty, σ-values, are calculated on the assumption that min/max are the one-sigma standard deviations in either direction. These values are expanded upon in appendix **??**.

The r-process yields (which applies to neutron star mergers in `Omega`) are taken from *[Arnould et al., 2007]*. These yields are the observed r-process abundances of the Solar system, and the standing assumption is that the distribution of r-process nuclei from neutron star mergers follow the observed r-process abundances. In this spirit, the yield-tables for neutron star mergers are modified with the uncertainty of observed r-process abundances in the Solar system. The values, absolute uncertainty and relative uncertainty of the selected nuclei are shown in table 1.7. The minimum and maximum values are interpreted as one-sigma deviations in either direction, and the relative uncertainty, σ is calculated thereafter.

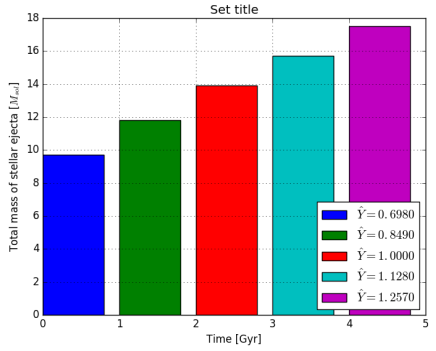
The effect of each nuclei in table 1.7 was calculated. The results were very similar, so only $^{187}_{75}\mathrm{Re}$ will be shown. In table 1.8 the output deviation, in



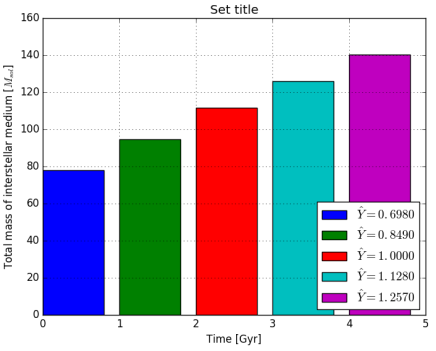
(a) Time evolution of $^{187}_{75}\mathrm{Re}$ ejected from supernovae and neutron star mergers.



(b) Time evolution of the total mass of $^{187}_{75}\mathrm{Re}$ in the galaxy over time.



(c) Sum total of mass of $^{187}_{75}\mathrm{Re}$ ejected from supernovae and neutron star mergers.



(d) Sum total of mass of $^{187}_{75}\mathrm{Re}$ in the interstellar medium of the galaxy

Figure 1.20: Resulting evolution of $^{187}_{75}\mathrm{Re}$ -mass in a chemical evolution galaxy model.

ejected mass and total mass, is compared to the input uncertainty. The input uncertainty is the standard deviation from table 1.7 for one and two sigma deviation in either direction. The table represent the deviation in figure 1.20. The relative uncertainty of the input is nearly the same as the output deviation. This suggests that the uncertainty of the stellar yield-tables affect the total abundance linearly (the relative uncertainty remains the same throughout the calculation).

σ_{init}	$\sigma_{ISM}(z=0)$	$\Sigma\sigma_{ISM}$	$\sigma_{\dot{m}}(z=0)$	$\Sigma\sigma_{\dot{m}}$
-0.302	-0.301887	-0.301887	-0.301887	-0.301887
0.128	0.128931	0.128931	0.128931	0.128931
0.257	0.257862	0.257862	0.257862	0.257862
0	0	0	0	0
-0.151	-0.150943	-0.150943	-0.150943	-0.150943

Table 1.8: Comparison of output mass-uncertainty with input yield-uncertainty. σ_{init} is the relative uncertainty from table 1.7 for zero, one, and two sigmas in either direction. σ_{ISM} is the relative deviation between the resulting mass of the interstellar medium, at current time (Z=0) and integrated for all time (Σ). $\sigma_{\dot{m}}$ is the relative deviation between the ejected mass per timestep into the interstellar medium, at current time (Z=0) and integrated for all time (Σ).

1.4 Uncertainty from multiple parameter

For varying different parameters simultaneously a similar method was used as in section results yields 1.3? , where a "fudge-factor" was applied to the *Fiduccial* `Omega`*-model* parameter-values. The "fudge-factor" was randomly sampled from a gaussian distribution with mean 1.0 and standard deviation equal to the relative uncertainty σ, from table ref table rncp arnould uncertianty 1.7? . The "fudge-factor" is then multiplied with the parameter value found from the bestfit *Fiduccial* `Omega`*-model* fitted to data from `Eris`.

Monte Carlo experiment

In order to manipulate several yield-values and other parameters in `Omega` at once, a modification to `__set_yield_tables` in `Chem_Evol` was implemented. This is similar to the modification in section 1.3 (as seen in listing 46), but includes a list for all isotopes and "fudge factors". The extra lines of code that replace the modification in section 1.3 are shown in listings 58.

Other input variables are multiplied with a similar factor before the `Omega`-simulation is executed.

```
1  ##########################################################
2  ### End of function as written in 'chem_evol.py' ###
3  """
4  Change ytables(multiply yields of 'isotope' with 'factor')
5  This step requires
6  'self.loa_manip_isotope' and 'self.loa_manip_yields'!
7  """
8  ##########################################################
9
10 #AGB + massive stars, and pop3 stars
11 #loop over the different objects
12 for table_object, table_name in zip([self.ytables, self.ytables_pop3],
```

```
                                        ["agb/massive", "pop3"]):
    #get list of available metalicities
    loa_metallicities = table_object.metallicities
    for Z in loa_metallicities:
        #get list of masses for each metallicity
        loa_masses = table_object.get(Z=Z, quantity="masses")
        for M in loa_masses:
            #loop over all isotopes to manipulate
            for manip_isotope, manip_factor in zip(self.loa_manip_isotopes,self.
    loa_manip_yields):
                #get current yield
                try:
                    present_yield = table_object.get(M=M, Z=Z, quantity="Yields"
    ,
                                                     specie=manip_isotope)
                except IndexError: #this means that isotope doesn't exist for
    this table
                    continue
                #modify yield by some factor
                new_yield = present_yield*manip_factor
                #"insert" new yield back into table
                table_object.set(M=M, Z=Z, specie=manip_isotope, value=new_yield
    )
                #print "Fixed new yield(%s): from %1.4e to %1.4e"%(table_name,
    present_yield, new_yield)

# SN1a, NS-NS merger, BH-NS merger
#loop over different objects
for table_object, table_name in zip(
        [self.ytables_1a, self.ytables_nsmerger, self.ytables_bhnsmerger],
        ["sn1a", "nsm", "bhnsm"]):
    #get list of available metalicities
    loa_metallicities = table_object.metallicities
    #loop over metallicities
    for i_Z, Z in enumerate(loa_metallicities):
        #loop over all isotopes to manipulate
        for manip_isotope, manip_factor in zip(self.loa_manip_isotopes,self.
    loa_manip_yields):
            # get index of isotope
            index_iso = self.history.isotopes.index(manip_isotope)
            #get current yield
            try:
                present_yield = table_object.yields[i_Z][index_iso]
            except IndexError: #this means that isotope doesn't exist for this
    table
                continue
            #modify yield by some factor
            new_yield = present_yield*manip_factor
            #"insert" new yield back into table
            table_object.yields[i_Z][index_iso] = new_yield
            #print "Fixed new yield(%s): from %1.4e to %1.4e"%(table_name,
    present_yield, new_yield)
return
```

Listing 1.2: Snippet of code added to the existing function `__set_yield_tables` in `Chem_Evol` in `Omega`-framework. The code-snippet multiplies the yield of a list of isotopes, `self.loa_manip_isotope`, with a corresponding factor from a list of factors `self.loa_manip_yields` for all yield-tables where the isotopes can be found.

Postprocessing

The output files for each `Omega`-run consists of time-arrays for a multitude of measurables, e.g. the mass of $^{187}_{75}\mathrm{Re}$ in the interstellar medium. Postprocessing of all the datafiles must be done in order to account for the β^--decay of $^{187}_{75}\mathrm{Re}$ to $^{187}_{76}\mathrm{Os}$ [4]. This is done, for each timestep, by calculating the amount of decayed material from parent nucleus to daughter nucleus. The amount of decayed material is calculated from the timestep length and half-life of the radioactive parent nucleus, and applied to the current and all following timesteps for parent and daughter nuclei. Add reference to section of β^--decay calculations The new data is then saved to file in the same format. The function for applying the decay to parent nucleus and daughter nucleus ($^{187}_{75}\mathrm{Re}$ and $^{187}_{76}\mathrm{Os}$, respectively, in our case) is found in listings 19.

```
def apply_decay(self, time_array, parent_array, daughter_array, halflife):
    """ Apply decay from parent to daughter with
    the corresponding time-array and nuclear halflife.
    Halflife in same units as time_array. """

    decay_constant = np.log(2)/halflife

    for i in range(len(time_array)-1):
        #calculate time
        dt = time_array[i+1] - time_array[i]
        #calculate decay
        dN = - decay_constant * parent_array[i] * dt
        #apply decay to parent forall indeces greater then i
        parent_array[i+1:] += dN
        #same for daughter, but negative decay
        daughter_array[i+1:] -= dN

    return parent_array, daughter_array
```

Listing 1.3: Snippet of code implementing β^--decay in postprocessing on data calculated by `Omega`.

1.5 Summary of methods

1.6 Discussion

Attempting to probe the uncertainties of nuclear observables is difficult, because both `Eris` and `Omega` both use the observed r-process abundance from the Solar system as the yield tables for all r-process events (neutron star mergers). This mean that all r-process events assume to generate the same distribution of heavy

[4] At the time of writing, `Omega`does not account for β^--decay of radioactive nuclei, so this is implemented in postprocessing. The effect of β^--decay in `Omega`is minimal as the total metallicity does not change, which again does not change the stellar yields considered.

metals, and that distribution is the same as the component measured in the Solar system. In order to vary these values in `Omega` a single isotope is chosen, along with a factor that applies to the yield of that isotope. When using `Omega`, all yield tables will multiply with this factor for that specific isotope alone. E.g. ('C-14', 1.3) would mean that the yields for $^{14}_{6}\mathrm{C}$ will be multiplied with 1.3 in all yield tables. This method does not only probe the nuclear uncertianties, varying the yield-tables of a chemical evolution code finds the effect of the accumulated uncertainty of the stellar evolution applied to chemical evolution methods. add more details of stellar evolution codes in theory part I

There are many parameters that affect r-process production alone. Ejecta distribution and size from neutron star mergerand type II supernovae, and the ratio of ejecta and frequency between the two. Even in relatively simple galactic chemical evolution models, these parameters can interact in unpredictable ways. In order to get a complete picture of errors and uncertainty in numerical models, a common approach has been Monte Carlo style simulation do I need a reference for this? . Each parameter is randomly drawn from a distribution correspondin g to their "determined" uncertainty (usually the chosen distributions are gaussian), and after many repetitions of randomly drawing, the result can be viewed statistically in order to determine uncertainty behavior. This thesis will such an experiment with list parameters here and see how the uncertainties of the r-process affect the uncertainty of the Re-Os dating method.

Bibliography

[Arnould et al., 2007] Arnould, M., Goriely, S., and Takahashi, K. (2007). The r-process of stellar nucleosynthesis: Astrophysics and nuclear physics achievements and mysteries. *Phys.Rep.*, 450:97–213.

[Guedes et al., 2011] Guedes, J., Callegari, S., Madau, P., and Mayer, L. (2011). Forming Realistic Late-type Spirals in a ΛCDM Universe: The Eris Simulation. *ApJ*, 742:76.

[Kroupa et al., 1993] Kroupa, P., Tout, C. A., and Gilmore, G. (1993). The distribution of low-mass stars in the galactic disc. *Monthly Notices of the Royal Astronomical Society*, 262(3):545–587.

[Palme and Beer, 1993] Palme, H. and Beer, H. (1993). 3.4.6 classification of solar system materials: Datasheet from landolt-börnstein - group vi astronomy and astrophysics · volume 3a: "instruments, methods, solar system" in springermaterials. Copyright 1993 Springer-Verlag Berlin Heidelberg.

[Paxton et al., 2011] Paxton, B., Bildsten, L., Dotter, A., Herwig, F., Lesaffre, P., and Timmes, F. (2011). Modules for experiments in stellar astrophysics (mesa). *The Astrophysical Journal Supplement Series*, 192(1):3.

[Pignatari et al., 2016] Pignatari, M., Herwig, F., Hirschi, R., Bennett, M., Rockefeller, G., Fryer, C., Timmes, F. X., Ritter, C., Heger, A., Jones, S., Battino, U., Dotter, A., Trappitsch, R., Diehl, S., Frischknecht, U., Hungerford, A., Magkotsios, G., Travaglio, C., and Young, P. (2016). Nugrid stellar data set. i.stellar yields from h to bi for stars with metallicities z = 0.02 and z = 0.01. *The Astrophysical Journal Supplement Series*, 225(2):24.

[Shen et al., 2015] Shen, S., Cooke, R. J., Ramirez-Ruiz, E., Madau, P., Mayer, L., and Guedes, J. (2015). The History of R-Process Enrichment in the Milky Way. *ApJ*, 807:115.