A

GCE nova resources

As part of Chapters 4 and 5, we produced a set of nova ejecta delay-time distributions suitable for use in galactic chemical evolution models. We have compiled these ejecta delay-time distributions, together with several other useful resources, with the intent to provide a useful tool to future GCE studies seeking to include novae. Here, we present a reference intended to assist efforts to use this set of resources.

The resources can be accessed here: DOI:10.5281/zenodo.6898161

Works that use these resources should cite (Kemp et al., 2022a) and Kemp et al. (2022b). Works that use the yield tables should also cite the relevant theoretical work which computed the table (see below).

A.1 What's included

1. **Yield sets.**

   We have included the nova yield files we compiled in the resources, for convenience. The files included are formatted for use in the JINAPyCEE/NuPyCEE environment (Côté et al., 2017, 2018), but will likely still be useful to other codes after some modification.

2. **Delay-time distributions.**

   We have included both ejecta delay-time distributions and regular nova delay-time distributions; how to distinguish the two is described in Section A.3.

3. **Example Jupyter notebook**

   We have included a (detailed) example notebook illustrating the use of the resources within JINAPyCEE/NuPyCEE.

   Functions that may be useful to non-JINAPyCEE/NuPyCEE users are highlighted below.

A.2 The yield tables

There are 7 directories, each containing formatted yield files and some some unformatted tables. The unformatted tables are not important, but might be useful for some users. The yield files contain the mass fractions of the relevant isotopes. Below, we describe which yield tables are in each of the 7 directories found within '/yields':

1. /jose1998yields

   Yield file structure: Jose1998_<WD composition>_MWD<$M_{\rm WD}$ with decimal ('.') replaced with 'p']_<mixing fraction (%)>mx.txt

   Example: Jose1998_CO_MWD0p8_50mx.txt is the yield table for a C/O WD of 0.8 $M_\odot$, assuming 50% pre-mixng.

   Comments: From José & Hernanz (1998), and used in the J1998 yield profile in Chapters 4 and 5. WD composition is denoted 'CO' for C/O WDs, 'ONe' for O/Ne WDs.

2. /jose2020yields

   Yield file structure: Jose2020_MWD<WD composition>_MWD<$M_{\rm WD}$ with decimal ('.') replaced with 'p']_123321.txt

   Comments: Jose2020_MWD1p0_123321 is the 1.0 $M_\odot$ C/O WD model from José et al. (2020), Jose2020_MWD1p25_123321 is the 1.25 $M_\odot$ model. Used in the J2020 yield profile in Chapters 4 and 5

3. /rukeya2017_table1

   Yield file structure: Rukeya2017_<WD composition>MWD_MWD<$M_{\rm WD}$ with decimal ('.') replaced with 'p']_<mixing fraction (%)>mx.txt

Example: Rukeya2017_MWD0p8_50mx.txt is the yield table for a C/O WD of 0.8 $M_\odot$, assuming 50% pre-mixng.

Comments: From Rukeya et al. (2017), and used in the R2017simple yield profile in Chapter 4. WD composition is C/O unless specified 'ONe', in which case the WD was of O/Ne composition. Note that Rukeya et al. (2017) only reported Li yields.

4. /rukeya2017_table2

   Yield file structure: Rukeya2017_<WD composition>MWD_MWD<$M_{\rm WD}$ with decimal ('.') replaced with 'p']_Mdot<$\dot{M}$ in scientific notation with decimal ('.') replaced with 'p', '-' replaced with 'm', and E replaced with 'e'>.txt

   Examples: Rukeya2017_MWD1p1_Mdot1p0em10.txt is the yield table for a 1.1 $M_\odot$ C/O WD accreting at 1.1×10^{-10} $M_\odot$ yr^{-1}. Rukeya2017_ONeMWD1p34_Mdot3p7em7.txt is the yield table for a 1.34 $M_\odot$ O/Ne WD accreting at 3.7×10^{-7}.

   Comments: From Rukeya et al. (2017), and used in the R2017 yield profile in Chapter 4. WD composition is C/O unless specified 'ONe', in which case the WD was of O/Ne composition. Note that Rukeya et al. (2017) only reported Li yields.

5. /starrfield2009yields

   There are only two yield tables provided here:

   Starrfield2009_MWD1p25_I2005A.txt and Starrfield2009_MWD1p35_I2005A.txt, for the 1.25 and 1.35 $M_\odot$ O/Ne WDs from Starrfield et al. (2009), respectively. Used in the S2009/2020 yield profile in Chapters 4 and 5.

6. /starrfield2020yields

   Yield file structure: Starrfield2020_MWD<$M_{\rm WD}$ with decimal ('.') replaced with 'p'>_<mixing fraction (%)>mx.txt

   Example: Starrfield2020_MWD0p8_50mx.txt is the yield table for a C/O WD of 0.8 $M_\odot$, assuming 50% pre-mixng.

   Comments: From Starrfield et al. (2020), and used in the S2009/2020 yield table in Chapters 4 and 5. All of these are C/O models, even the massive WDs.

7. obs_simple Yield file structure: obs_simple_<mixing fraction (%)>mx_<tag, see comments>.txt

   Example: obs_simple_25mx_curupper.txt is the 1-σ upper bound for the observationally derived mass fraction for ^{7}Be using all available ^{7}Be nova yields, but not the derived Li yield of V1369 Cen.

   Any 'tag' that does NOT have 'cur' in it included the derived Li yield of V1369 Cen in the calculation (see Izzo et al. (2015); Molaro et al. (2022), and is NOT recommended for use other than comparison of the effect of including it. If 'lower' is in the tag, its a 1-σ lower bound. If neither 'lower' nor 'upper' is in the tag, it is the mean. These yield tables were used in the 'obs' cases in Chapter 4.

In each yield file, the relevant isotope names and their mass fractions can be found. Each column corresponds to a different metallicity, specified in the header, but note that the data is the same in each column. It is in principle possible to have metallicity-dependent nova yields. However, such yields have never been computed, so the solar metallicity yields are filled for all metallicity columns. The metallicity columns are necessary for the JINAPyCEE/NuPyCEE implementation in order to include metallicity-dependent ejecta delay-time distributions.

A.3 (Ejecta) delay-time distributions

Found within the '/dtds' directory. Three different binary physics cases are included, which have the same structure for the files. These cases are:

'standard', the binary physics case that most of this thesis deals with (see Chapters 2-5);

'eta0p01': As above, but with the accretion efficiency (η) set to 0.01;

'lambda0p5': The alternate common envelope physics ($\lambda_{CE} = 0.5$) case we used to test the dependency of binary physics on Li yields in Chapter 4.

Within each of these different cases, there is a directory '/jsonfiles'. Within /jsonfiles lie a large number of different delay-time and ejecta delay-time distributions suitable for different purposes, computed at various metallicities and time-binning.

A.3.1 File name structure

The general structure of the file names is

<DTD, DTDproc, or DTDtimecenters>_z<metallicity>_bw<time bin size in Myr>_<mwdbin>_<mdotbin>.json

Where:

DTD indicates a nova delay-time distribution, and DTDproc denotes a ejecta delay-time distribution. DTD tracks nova events, DTDproc tracks nova ejecta. DTDtimecenters is not a delay-time distribution, but instead provides the time coordinates (in Myr) for the relevant file.

The metallicity is denoted by one of the following tags: 0p03 (0.03), 0p025 (0.025), 0p02 (0.02), 0p0175 (0.0175), 0p015 (0.015), 0p0125 (0.0125), 0p01 (0.01), 0p007 (0.007), 0p005 (0.005), 0p003 (0.003), 0p0025 (0.0025), 0p002 (0.002), 0p001 (0.001), 0p0005 (0.0005), 0p00025 (0.00025), 1em4 (0.0001).

The time bin size (in Myr) is either 500, 100 (what we use), 50, 10, or 1. It is purely a numeric property of the process of generating the distributions. Using a small bin size will slow your simulation considerably; we find 100 Myr binning to be adequate.

'mwdbin' is the WD mass range that determines what part of the $M_{\rm WD}$ parameter space we are counting within that particular distribution file. The range string is structured: <lower bound><mwd< <upperbound> (eg. 0.9<mwd<1.075 is the range string that counts novae that occur on WDs with masses between 0.9 and 1.075 $\rm M_{\odot}$).

'mdotbin' is the WD accretion rate range that determines what part of the $\dot{M}$ parameter space we are counting within that particular distribution file. Note that files with a specified 'mdotbin' range are only useful with the R2017 yield profle, as only this profile has $\dot{M}$ dependent yield tables. The range string is structured: <lower bound><Mdot< <upperbound> (eg. 3e-09<mdot<3e-08 is the range string that counts novae that occur with accretion rates between 3×10^{-9} and 3×-8 $\rm M_{\odot}\,yr^{-1}$).

Finally, note that omitting the 'mdotbin' and 'mwdbin' components of the structure gives you the file names for 'all' novae, without distinguishing between different WD masses or accretion rate regimes. For example, DTD_z0p01_bw10.json is the delay-time distribution at $Z = 0.01$ with 10 Myr time binning for all novae, irrespective of the $M_{\rm WD}$

or $\dot{M}$ at the time of nova.

Each delay-time (or ejecta delay-time) presents the nova counts or ejecta mass in a given bin, normalised per unit mass of star forming material. It is important to emphasise that it is the nova count and ejecta mass, NOT the nova count per year and the ejecta mass per year.

A.4 Using the files

The details of how these files can be implemented into a given GCE code will of course be code dependent. The notebook included with the resources gives a working example of how we did this with `OMEGA+`. Here, we attempt provide some general guidance that we hope will be useful to a most GCE codes.

The first requirement we will discuss is the creation of a yield–delay-time distribution map. Depending on how sophisticated you want your nova model to be, this could be quite complex. In our models, we allowed for metallicity-dependent ejecta delay-time distributions, and we *could* have included metallicity-dependent nova yield tables, had any existed. Further, as we had access to $M_{\rm WD}$-dependent theoretical nova yields (and even $\dot{M}$-dependent yields for Li, thanks to work of Rukeya et al., 2017) we computed, for each metallicity, individual delay-time distributions for many sections of the nova parameter-space. We then mapped the appropriate delay-time distributions to the most appropriate nova yields.

The point is that, irrespective of GCE code, the determination of which ejecta delay-time distributions should be paired with which yield table is necessary. The creation of such a map can be labour intensive, depending on the complexity of the model. For this reason, we wish to highlight some of the useful functions within the jupyter notebook included in the resources that may make this easier.

There are several 'dtdmappingarr' functions included that compute mapping arrays which have the structure:

[[dtd_paths][mwd_coords][mdot_coords][yield_paths]]

The most important elements are the dtd_paths and yield_paths; depending on your implementation, mwd_coords and mdot_coords may not be needed, but are intended to

```
[['/Users/alexkemp/Desktop/GCE_NOVA_RESOURCES/nova_resources/dtds/standard/jsonfiles/DTDproc_z0p02_bw100_0.9<mwd<1.
075.json'
  '/Users/alexkemp/Desktop/GCE_NOVA_RESOURCES/nova_resources/dtds/standard/jsonfiles/DTDproc_z0p02_bw100_1.2<mwd<1.
3.json'
  '/Users/alexkemp/Desktop/GCE_NOVA_RESOURCES/nova_resources/dtds/standard/jsonfiles/DTDproc_z0p02_bw100_mwd<0.7.js
on'
  '/Users/alexkemp/Desktop/GCE_NOVA_RESOURCES/nova_resources/dtds/standard/jsonfiles/DTDproc_z0p02_bw100_0.7<mwd<0.
9.json'
  '/Users/alexkemp/Desktop/GCE_NOVA_RESOURCES/nova_resources/dtds/standard/jsonfiles/DTDproc_z0p02_bw100_1.075<mwd
<1.2.json'
  '/Users/alexkemp/Desktop/GCE_NOVA_RESOURCES/nova_resources/dtds/standard/jsonfiles/DTDproc_z0p02_bw100_1.3<mwd.js
on']
 ['1.0' '1.25' '0.8' '0.8' '1.15' '1.35']
 ['0.0' '0.0' '0.0' '0.0' '0.0' '0.0']
 ['/Users/alexkemp/Desktop/GCE_NOVA_RESOURCES/nova_resources/yields/jose1998yields/Jose1998_CO_MWD1p0_50mx.txt'
  '/Users/alexkemp/Desktop/GCE_NOVA_RESOURCES/nova_resources/yields/jose1998yields/Jose1998_ONe_MWD1p25_50mx.txt'
  '/Users/alexkemp/Desktop/GCE_NOVA_RESOURCES/nova_resources/yields/jose1998yields/Jose1998_CO_MWD0p8_50mx.txt'
  '/Users/alexkemp/Desktop/GCE_NOVA_RESOURCES/nova_resources/yields/jose1998yields/Jose1998_CO_MWD0p8_50mx.txt'
  '/Users/alexkemp/Desktop/GCE_NOVA_RESOURCES/nova_resources/yields/jose1998yields/Jose1998_ONe_MWD1p15_50mx.txt'
  '/Users/alexkemp/Desktop/GCE_NOVA_RESOURCES/nova_resources/yields/jose1998yields/Jose1998_ONe_MWD1p35_50mx.txt']]
```

Figure A.1 Example mapping array, generated by the 'dtdmappingarr_jose1998' function in the included notebook. This example is the mapping array used for the J1998 yield profile (José & Hernanz, 1998). Structure: [[dtd_paths][mwd_coords][mdot_coords][yield_paths]]. Note that, as there is no $\dot{M}$-dependency to the yields for this yield profile, the 'mdot_coords' are all set to zero.

provide a useful reference of what part of the parameter space the entry is referring to. Figure A.1 shows the mapping array generated by the 'dtdmappingarr_jose1998' function.

Note that in the mapping array, ejecta delay-time distributions of only a single metallicity ($Z = 0.02$) are included. That is because this is *not* fed directly into `OMEGA+`. Additional processing propagates the mappings of this table to the rest of the metallicities and loads in the actual DTD nova data. In `OMEGA+`, metallicity-dependence of the yields are contained within each yield table. Its worth repeating that none of our yield tables have metallicity-dependence anyway, but if this sort of data became available in the future and you wanted to include it in your own models, that's how it would be done in the framework of the NuPyCEE environment.

The precise format required for the final inputs will be GCE-dependent. In our implementation for OMEGA+, most of the heavy-lifting is done by the 'read_nova_data' function, which reads in and formats (including metallicity-dependency, controlled by the 'othersimtags' variable) the appropriate ejecta delay-time distributions and computes normalisation factors and time arrays. The precise normalisation factors that are needed will also be code dependent.

Another function which might be useful is 'readandreturnselected' in the 'utility funcs' cell, which can be used to read in the .json files. Most of the other functions are related to the processing and visualisation of the resulting data.

A.5 Closing thoughts

The included resources are expected to improve the ability of GCE codes to model novae, providing more detailed information that can be used to model them more accurately. The above information should hopefully provide a starting point for someone seeking to use the provided materials within another GCE code. Ultimately the technical details of the implementation will, however, be dictated by the capabilities and structure of the code in question. The existing functions within the included notebook should provide a starting point for prospective users to form their own dtd-yield maps. Users of the JINAPyCEE/NuPyCEE environment should find the notebook particularly useful, and able to include novae within their simulations with relatively little effort by using the provided functions.

A.6 Useful links

- The resources: DOI:10.5281/zenodo.6898161
- https://github.com/becot85/JINAPyCEE (JINAPyCEE github, needed for `OMEGA+`)
- https://github.com/NuGrid/NuPyCEE (NuPyCEE github, needed for `OMEGA` and `SYGMA`,)which are needed to run `OMEGA+`
- https://pypi.org/project/iniabu/ (helpful module for solar abundance data, used in parts of the notebook)