**A Nuclear Reaction Network tutorial for the Nuclei in the Cosmos XVII School**

M. R. Mumpower & E. M. Holmbeck

**PRISM: Portable Routines for Integrated nucleoSynthesis Modeling**

A Fortran-based nuclear reaction network that is run at the command line.

**Installation of PRISM**

1. Download the ~~math kernel libraries (MKL)~~ **Base ToolKit** from Intel for your operating system, [~~here~~](https://www.intel.com/content/www/us/en/developer/tools/oneapi/onemkl-download.html) [**here**](https://www.intel.com/content/www/us/en/developer/tools/oneapi/base-toolkit-download.html).
   1. Note that this download is roughly 1 GB. Please start it as soon as possible!
   2. Note that you will need roughly 10 GB of space for the installation size.
   3. If your OS hasn’t been updated recently, you may need to download an older version of MKL
   4. On Linux, ensure that the .sh file is executable by “chmod u+x <filename.sh>”
   5. On Mac: add the libiomp5.dylib to your DYLD\_LIBRARY\_PATH:  
      “export DYLD\_LIBRARY\_PATH=$DYLD\_LIBRARY\_PATH:/opt/intel/oneapi/compiler/2023.1.0/mac/compiler/lib”
2. Download the PRISM (1.6.0) directory from the NIC XVII School Google Drive link.
   1. For ease of this tutorial, note that the executables are pre-compiled, so you should not have to worry about compiling from source code directly.

**Testing installation of PRISM**

* Load the MKL libraries into your environment.
  1. For instance, using Bash on Linux this looks something like:
     + “. /opt/intel/oneapi/setvars.sh”
  2. On MAC OS it is the same.
     + “. /opt/intel/oneapi/setvars.sh”
     + -OR- “source /opt/intel/oneapi/setvars.sh”
* Move to the downloaded PRISM directory and issue command:
  1. “cp prism-<YOUR\_OS> input”
     + Copy the executable into the input directory; “<YOUR\_OS>” is either “linux” or “mac-m-proc”
  2. “cd input”
  3. “./prism -h”
     + You should see the help message displayed and version number printed (1.6.0).
     + If you see an error message, it is likely that the environmental variables are not properly set. Remember, environmental variables

**Still having issues? Use this! (Thanks, Claudio!)**

docker run -it -v <YOUR\_PRISM\_DIRECTORY>:/home/nic/PRISM clausualc/nic-prism

**Structure of input files**

* Decay files (“decay” and “prob-decay”):  
  1 | Number of decays  
  2 | list of Z of reactants(s)  
  3 | list of N of reactants(s)  
  4 | list of Z of product(s)  
  5 | list of N of product(s)  
  6 | rate (in seconds)  
  # 7 | probability associated with each product (same length as lines 4 and 5)  
  # Repeat lines 2-6 (decay) or 2-7 (prob-decay)
* Reaction files (“rxn” and “prob-rxn”):  
  1 | Number of reactions  
  2 | List of temperatures (in GK)  
  3 | list of Z of reactants(s)  
  4 | list of N of reactants(s)  
  5 | list of Z of product(s)  
  6 | list of N of product(s)  
  7 | rates (in seconds; same length as line 2)  
  # 8 | probability associated with each product (same length as lines 5 and 6)  
  # Repeat lines 3-7 (rxn) or 3-8 (prob-rxn)
* Astrophysical trajectory - list of time, temperature, and density  
  1 | Comments  
  2 | time(s) Temperature(GK) density(g cm^-3)  
  3 | time(s) Temperature(GK) density(g cm^-3)  
  …
* Initial composition - list of Z, A, and X (mass fraction). Remember ∑*iXi* = 1!  
  1 | Z1 A1 X1  
  2 | Z2 A2 X2  
  3 | Z3 A3 X3…  
  n | Zn An Xn

**Tutorial**

**Problem 1:** Recall, networks use both nuclear data and astrophysical data. PRISM encodes this information in a “control” (JSON formatted) file. Take a look at one of the control files in the “input/examples/control” directory. What sort of information is included? What nuclear data is included? What nuclear data isn’t included?

**Problem 2:** Open an astrophysical ‘trajectory’ file and plot the evolution of temperature and density. What can you say about the conditions of the file you chose? Predict what you think will happen when you run a nucleosynthesis simulation with this control file.

**Problem 3:** Perform your first PRISM calculation by choosing a control file to run with. This can be evoked at the command line using the incantation:

* “./prism -c <path-to-your-control-file>”

Once the simulation is complete, take a look at the output directory and plot the abundance information. Look for a file starting with “y”. Abundance information may be output per timestep, or at the final snapshot of the simulation. Abundance information is provided per nucleus (Z,A), isotopically (as a function of A only), or elemental (as a function of Z only).

**Problem 4:** Perform the same calculation, but add in the debugging flag:

* “./prism -c <path-to-your-control-file> -d debug”

This provides much more output of what the code is doing. Why do you think it is important to check at every timestep the preservation of mass fraction?

**Problem 5:** Try to write your own simple network with one species decaying. The starting formula is: dN/dt = -λ N . What type of functional form does the solution take? What happens when there’s more than one nucleus (for instance with nuclei decaying along an isotopic chain)?

**Problem 6:** Create your own reaction or decay file, add it to your control file, and run PRISM again to test the sensitivity on changing that rate. (Remember to rename your output files!)