Webnucleo Technical Report: Input XML for libnucnet

Bradley S. Meyer

January 10, 2010

This technical report describes some details of XML input to libnucnet.

1 Libnucnet__Nuc

The Libnucnet__Nuc structure handles data about nuclei in a collection of nuclear species. The data for each species are the atomic number Z, the mass number A, the atomic mass excess Δ, the ground state spin of the species, and data for the nuclear partition function.

The nuclear mass excess is defined as

$$M(Z,A)c^2 = 931.478A + \Delta(Z,A), \tag{1}$$

where $M(Z,A)c^2$ is the rest mass energy of nuclide (Z,A) and the rest mass energy and mass excess are measured in MeV. The scaling for the mass excess is chosen such that $\Delta(6,12)$, the mass excess for ^{12}C, is zero.

The nuclear partition function $G(T)$ is a sum over nuclear levels weighted by the Boltzmann factor:

$$G(T) = \sum_i \left(2J(E_i)+1\right) e^{-E_i/kT}, \tag{2}$$

where the sum over i is over all nuclear levels. Because data on the ground-state nuclear spin ($J_{g.s.}$) is already present, and because $G(T)$ can grow rapidly with temperature, Libnucnet__Nuc actually stores the quantity $F(T)$:

$$F(T) = \log_{10}\left(\frac{G(T)}{2J_{g.s.}+1}\right). \tag{3}$$

Some nuclear species have long-lived meta-stable states that must be treated as separate nuclear species (e.g., ^{26}Al). These species have their own mass excess, spin, and partition functions (e.g., [Gupta & Meyer(2001)]). The ground state is labeled "g" while the meta-stable state is labeled "m" (e.g., $^{26}\mathrm{Al}_g$ and $^{26}\mathrm{Al}_m$). If a nuclide has more than one meta-stable state to be treated as a separate species, the labels are "g", "m1", "m2", ...

The libnucnet API allows a user to input nuclear data directly; however, the most convenient method to read in the relevant information is via an XML file. The schema that defines the grammar for such an input file is libnucnet_nuc.xsd in the xsd_pub for a version of libnucnet.

An example of an input XML file is the following, which contains data for ^{1}H and ^{12}C:

```
<?xml version="1.0" encoding="UTF-8"?>

<nuclear_data>

  <!--h1-->

  <nuclide>
    <z>1</z>
    <a>1</a>
    <source>Tuli (2000)</source>
    <mass_excess>7.289</mass_excess>
    <spin>0.5</spin>
    <partf_table>
      <point>
        <t9>0.1</t9>
        <log10_partf>0</log10_partf>
      </point>
      <point>
        <t9>0.15</t9>
        <log10_partf>0</log10_partf>
      </point>
      <point>
        <t9>0.2</t9>
        <log10_partf>0</log10_partf>
      </point>
      <point>
        <t9>0.3</t9>
        <log10_partf>0</log10_partf>
      </point>
      <point>
        <t9>0.4</t9>
        <log10_partf>0</log10_partf>
      </point>
      <point>
        <t9>0.5</t9>
        <log10_partf>0</log10_partf>
      </point>
      <point>
        <t9>0.9</t9>
```

```
      <log10_partf>0</log10_partf>
    </point>
    <point>
      <t9>1</t9>
      <log10_partf>0</log10_partf>
    </point>
    <point>
      <t9>1.5</t9>
      <log10_partf>0</log10_partf>
    </point>
    <point>
      <t9>2</t9>
      <log10_partf>0</log10_partf>
    </point>
    <point>
      <t9>2.5</t9>
      <log10_partf>0</log10_partf>
    </point>
    <point>
      <t9>3</t9>
      <log10_partf>0</log10_partf>
    </point>
    <point>
      <t9>3.5</t9>
      <log10_partf>0</log10_partf>
    </point>
    <point>
      <t9>4</t9>
      <log10_partf>0</log10_partf>
    </point>
    <point>
      <t9>5</t9>
      <log10_partf>0</log10_partf>
    </point>
    <point>
      <t9>10</t9>
      <log10_partf>0</log10_partf>
    </point>
  </partf_table>
</nuclide>

<!--c12-->

<nuclide>
  <z>6</z>
  <a>12</a>
```

```
<source>Tuli (2000)</source>
<mass_excess>0.</mass_excess>
<spin>0.</spin>
<partf_table>
  <point>
    <t9>0.1</t9>
    <log10_partf>0</log10_partf>
  </point>
  <point>
    <t9>0.15</t9>
    <log10_partf>0</log10_partf>
  </point>
  <point>
    <t9>0.2</t9>
    <log10_partf>0</log10_partf>
  </point>
  <point>
    <t9>0.3</t9>
    <log10_partf>0</log10_partf>
  </point>
  <point>
    <t9>0.4</t9>
    <log10_partf>0</log10_partf>
  </point>
  <point>
    <t9>0.5</t9>
    <log10_partf>0</log10_partf>
  </point>
  <point>
    <t9>0.9</t9>
    <log10_partf>0</log10_partf>
  </point>
  <point>
    <t9>1</t9>
    <log10_partf>0</log10_partf>
  </point>
  <point>
    <t9>1.5</t9>
    <log10_partf>0</log10_partf>
  </point>
  <point>
    <t9>2</t9>
    <log10_partf>0</log10_partf>
  </point>
  <point>
    <t9>2.5</t9>
```

```
      <log10_partf>0</log10_partf>
    </point>
    <point>
      <t9>3</t9>
      <log10_partf>0</log10_partf>
    </point>
    <point>
      <t9>3.5</t9>
      <log10_partf>0</log10_partf>
    </point>
    <point>
      <t9>4</t9>
      <log10_partf>0</log10_partf>
    </point>
    <point>
      <t9>5</t9>
      <log10_partf>0</log10_partf>
    </point>
    <point>
      <t9>10</t9>
      <log10_partf>0</log10_partf>
    </point>
  </partf_table>
</nuclide>

</nuclear_data>
```

In this file, the root tag is **nuclear_data**. The data are a sequence of nuclides (**nuclide** tag). The lines like

```
<!-- h1 -->
```

are optional comments that are ignored by the XML parser. There is no bound of the number of nuclides that may be present in the file. Data for a nuclide are a sequence of **z**, **a**, **source**, **mass_excess**, **spin**, and **partf_table** tags. The data for these tags are:

z: The nuclide's atomic number (required). It must be a non-negative integer.

a: The nuclide's mass number (required). It must be a positive integer.

source: A string giving the data source for the file (usually a reference to a paper from which the data are taken). This tag is optional.

mass_excess: The nuclide's mass excess in MeV (see Eq. (1). This is a floating point number, and the tag is required. Note that this tag was changed from **mass** to **mass_excess** as of version 0.2 of libnucnet.

spin: The nuclide's spin (required). This is a non-negative floating point number.

The partition function data are those contained between the **partf_table** tags. The data are to be thought of as a table with each table entry between the **point** tags. The data for each table point are:

t9: The temperature in billions of Kelvins (required). It must be a non-negative float.

log10_partf: The $F(T)$ factor [see Eq. (3)] evaluated at the T_9 of this point (required). It is a float.

As of version 0.4, it is no longer necessary for the partition function data to be sorted in ascending order by temperature. The routine to update the input data sorts them before storing.

The libnucnet API routine that computes the nuclear partition function for a nuclide interpolates the $F(T)$ from the values in the table using the GNU Scientific Library spline interpolation routine. It then computes the nuclear partition function as

$$G(T) = (2J_{g.s.} + 1)\, 10^{F(T)}.$$

If the input temperature is less than the first **t9** of the table, the routine simply uses the lowest temperature point's F. Similarly, if the input temperature is greater than the last **t9** of the table, the routine uses the highest temperature point's F. In other words, the partition function routine does not extrapolate. The user should therefore supply the partition function table data for the fully desired temperature range. If only two partition function data points are provided, spline interpolation does not work. Instead, as of version 0.4, the routine interpolates linearly between the two points.

The routine that parses in the nuclear data will replace data for a given nuclide. Thus, the input data may contain multiple entries for a given nuclide. Suppose, for example, the input XML file has multiple nuclear data entries for ^{28}Si. The output from the parsing routine Libnucnet__Nuc__new_from_xml() will use the last entry (i.e., closest to the end of the file) for ^{28}Si. The user can thus add data to the end of the nuclear data file (but within the **nuclear_data** tags) to replace the original data for a nuclide with his or her own; however, a better solution is to use the API routine Libnucnet__Nuc__updateFromXml(), as demonstrated in the Libnucnet__Nuc Examples Tutorial.

When a nuclide has multiple states, as discussed above, the data for the mass excess, spin, and partition function are contained in the **states** tags. The data for each state is between **state** tags. The **id** tag is label for the state ("g" for the ground state; "m" for a single meta-stable state; "m1", "m2", ... for multiple meta-stable states. An example of XML data for a nuclide with multiple states is the following:

```
<!--al26-->
```

```
<nuclide>
  <z>13</z>
  <a>26</a>
  <states>
  <state id="g">
    <source>Tuli (2000) + Gupta and Meyer (2001)</source>
  <mass_excess>-12.21</mass_excess>
  <spin>5</spin>
  <partf_table>
    <point>
      <t9> 0.0100</t9>
      <log10_partf>0.000000</log10_partf>
    </point>
    <point>
      <t9> 0.1000</t9>
      <log10_partf>0.000000</log10_partf>
    </point>
    <point>
      <t9> 0.2000</t9>
      <log10_partf>0.000000</log10_partf>
    </point>
    <point>
      <t9> 0.3000</t9>
      <log10_partf>0.000000</log10_partf>
    </point>
    <point>
      <t9> 0.4000</t9>
      <log10_partf>0.000000</log10_partf>
    </point>
    <point>
      <t9> 0.5000</t9>
      <log10_partf>0.000017</log10_partf>
    </point>
    <point>
      <t9> 0.6000</t9>
      <log10_partf>0.000087</log10_partf>
    </point>
    <point>
      <t9> 0.7000</t9>
      <log10_partf>0.000274</log10_partf>
    </point>
    <point>
      <t9> 0.8000</t9>
      <log10_partf>0.000651</log10_partf>
    </point>
    <point>
```

```
    <t9> 0.9000</t9>
    <log10_partf>0.001279</log10_partf>
  </point>
  <point>
    <t9> 1.0000</t9>
    <log10_partf>0.002183</log10_partf>
  </point>
  <point>
    <t9> 2.0000</t9>
    <log10_partf>0.013907</log10_partf>
  </point>
  <point>
    <t9> 3.0000</t9>
    <log10_partf>0.005404</log10_partf>
  </point>
  <point>
    <t9> 4.0000</t9>
    <log10_partf>0.003288</log10_partf>
  </point>
  <point>
    <t9> 5.0000</t9>
    <log10_partf>0.003779</log10_partf>
  </point>
  <point>
    <t9> 6.0000</t9>
    <log10_partf>0.007458</log10_partf>
  </point>
  <point>
    <t9> 7.0000</t9>
    <log10_partf>0.015913</log10_partf>
  </point>
  <point>
    <t9> 8.0000</t9>
    <log10_partf>0.030094</log10_partf>
  </point>
  <point>
    <t9> 9.0000</t9>
    <log10_partf>0.049427</log10_partf>
  </point>
  <point>
    <t9>10.0000</t9>
    <log10_partf>0.072309</log10_partf>
  </point>
</partf_table>
</state>
<state id="m">
```

```
  <source>Tuli (2000) + Gupta and Meyer (2001)</source>
<mass_excess>-11.982</mass_excess>
<spin>0</spin>
<partf_table>
  <point>
    <t9> 0.0100</t9>
    <log10_partf>0.000000</log10_partf>
  </point>
  <point>
    <t9> 0.1000</t9>
    <log10_partf>0.000000</log10_partf>
  </point>
  <point>
    <t9> 0.2000</t9>
    <log10_partf>0.000000</log10_partf>
  </point>
  <point>
    <t9> 0.2500</t9>
    <log10_partf>0.000000</log10_partf>
  </point>
  <point>
    <t9> 0.3000</t9>
    <log10_partf>0.000000</log10_partf>
  </point>
  <point>
    <t9> 0.4000</t9>
    <log10_partf>0.000000</log10_partf>
  </point>
  <point>
    <t9> 0.5000</t9>
    <log10_partf>0.000000</log10_partf>
  </point>
  <point>
    <t9> 0.6000</t9>
    <log10_partf>0.000000</log10_partf>
  </point>
  <point>
    <t9> 0.7000</t9>
    <log10_partf>0.000004</log10_partf>
  </point>
  <point>
    <t9> 0.8000</t9>
    <log10_partf>0.000013</log10_partf>
  </point>
  <point>
    <t9> 0.9000</t9>
```

```
      <log10_partf>0.000056</log10_partf>
    </point>
    <point>
      <t9> 1.0000</t9>
      <log10_partf>0.000187</log10_partf>
    </point>
    <point>
      <t9> 2.0000</t9>
      <log10_partf>0.306116</log10_partf>
    </point>
    <point>
      <t9> 3.0000</t9>
      <log10_partf>0.623279</log10_partf>
    </point>
    <point>
      <t9> 4.0000</t9>
      <log10_partf>0.729541</log10_partf>
    </point>
    <point>
      <t9> 5.0000</t9>
      <log10_partf>0.806769</log10_partf>
    </point>
    <point>
      <t9> 6.0000</t9>
      <log10_partf>0.875325</log10_partf>
    </point>
    <point>
      <t9> 7.0000</t9>
      <log10_partf>0.939000</log10_partf>
    </point>
    <point>
      <t9> 8.0000</t9>
      <log10_partf>0.999050</log10_partf>
    </point>
    <point>
      <t9> 9.0000</t9>
      <log10_partf>1.057029</log10_partf>
    </point>
    <point>
      <t9>10.0000</t9>
      <log10_partf>1.114257</log10_partf>
    </point>
  </partf_table>
  </state>
  </states>
</nuclide>
```

2 Libnucnet__Reac

The Libnucnet__Reac structure handles data about a collection of nuclear reactions. The data for each reaction are the source of the data, the names of the reactants and products, and the reaction rate data. The rate data are in the form of a single rate, a rate table, or a NON SMOKER fit. An example of a libnucnet reaction file is the following:

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<reaction_data>

<!-- h1 + n to h2 + gamma -->

     <reaction>
       <source>Smith et al. (1993)</source>
       <reactant>h1</reactant><reactant>n</reactant>
       <product>h2</product><product>gamma</product>
       <rate_table>
         <point>
           <t9>0.001</t9>
           <rate>4.6168E+04</rate>
           <sef>1.000</sef>
         </point>
         <point>
           <t9>0.002</t9>
           <rate>4.5663E+04</rate>
           <sef>1.000</sef>
         </point>
         <point>
           <t9>0.003</t9>
           <rate>4.5281E+04</rate>
           <sef>1.000</sef>
         </point>
         <point>
           <t9>0.004</t9>
           <rate>4.4963E+04</rate>
           <sef>1.000</sef>
         </point>
         <point>
           <t9>0.005</t9>
           <rate>4.4684E+04</rate>
```

```
  <sef>1.000</sef>
</point>
<point>
  <t9>0.006</t9>
  <rate>4.4435E+04</rate>
  <sef>1.000</sef>
</point>
<point>
  <t9>0.007</t9>
  <rate>4.4208E+04</rate>
  <sef>1.000</sef>
</point>
<point>
  <t9>0.008</t9>
  <rate>4.3997E+04</rate>
  <sef>1.000</sef>
</point>
<point>
  <t9>0.009</t9>
  <rate>4.3801E+04</rate>
  <sef>1.000</sef>
</point>
<point>
  <t9>0.010</t9>
  <rate>4.3617E+04</rate>
  <sef>1.000</sef>
</point>
<point>
  <t9>0.020</t9>
  <rate>4.2172E+04</rate>
  <sef>1.000</sef>
</point>
<point>
  <t9>0.030</t9>
  <rate>4.1112E+04</rate>
  <sef>1.000</sef>
</point>
<point>
  <t9>0.040</t9>
  <rate>4.0252E+04</rate>
  <sef>1.000</sef>
</point>
<point>
  <t9>0.050</t9>
  <rate>3.9519E+04</rate>
  <sef>1.000</sef>
```

```
</point>
<point>
  <t9>0.060</t9>
  <rate>3.8875E+04</rate>
  <sef>1.000</sef>
</point>
<point>
  <t9>0.070</t9>
  <rate>3.8300E+04</rate>
  <sef>1.000</sef>
</point>
<point>
  <t9>0.080</t9>
  <rate>3.7778E+04</rate>
  <sef>1.000</sef>
</point>
<point>
  <t9>0.090</t9>
  <rate>3.7299E+04</rate>
  <sef>1.000</sef>
</point>
<point>
  <t9>0.100</t9>
  <rate>3.6857E+04</rate>
  <sef>1.000</sef>
</point>
<point>
  <t9>0.200</t9>
  <rate>3.3649E+04</rate>
  <sef>1.000</sef>
</point>
<point>
  <t9>0.300</t9>
  <rate>3.1600E+04</rate>
  <sef>1.000</sef>
</point>
<point>
  <t9>0.400</t9>
  <rate>3.0131E+04</rate>
  <sef>1.000</sef>
</point>
<point>
  <t9>0.500</t9>
  <rate>2.9022E+04</rate>
  <sef>1.000</sef>
</point>
```

```
<point>
  <t9>0.600</t9>
  <rate>2.8161E+04</rate>
  <sef>1.000</sef>
</point>
<point>
  <t9>0.700</t9>
  <rate>2.7481E+04</rate>
  <sef>1.000</sef>
</point>
<point>
  <t9>0.800</t9>
  <rate>2.6942E+04</rate>
  <sef>1.000</sef>
</point>
<point>
  <t9>0.900</t9>
  <rate>2.6514E+04</rate>
  <sef>1.000</sef>
</point>
<point>
  <t9>1.000</t9>
  <rate>2.6175E+04</rate>
  <sef>1.000</sef>
</point>
<point>
  <t9>2.000</t9>
  <rate>2.5518E+04</rate>
  <sef>1.000</sef>
</point>
<point>
  <t9>3.000</t9>
  <rate>2.7018E+04</rate>
  <sef>1.000</sef>
</point>
<point>
  <t9>4.000</t9>
  <rate>2.9256E+04</rate>
  <sef>1.000</sef>
</point>
<point>
  <t9>5.000</t9>
  <rate>3.1766E+04</rate>
  <sef>1.000</sef>
</point>
<point>
```

```
            <t9>6.000</t9>
            <rate>3.4347E+04</rate>
            <sef>1.000</sef>
          </point>
          <point>
            <t9>7.000</t9>
            <rate>3.6902E+04</rate>
            <sef>1.000</sef>
          </point>
          <point>
            <t9>8.000</t9>
            <rate>3.9381E+04</rate>
            <sef>1.000</sef>
          </point>
          <point>
            <t9>9.000</t9>
            <rate>4.1759E+04</rate>
            <sef>1.000</sef>
          </point>
          <point>
            <t9>10.000</t9>
            <rate>4.4025E+04</rate>
            <sef>1.000</sef>
          </point>
        </rate_table>
      </reaction>

<!-- ne15 + n to ne16 + gamma -->

<reaction>
   <source>ADNDT (2001) 75, 1 (non-smoker)</source>
   <reactant>ne15</reactant>  <reactant>n</reactant>
   <product>ne16</product>  <product>gamma</product>
   <non_smoker_fit>
      <Zt> 10</Zt>
      <At> 15</At>
      <Zf> 10</Zf>
      <Af> 16</Af>
      <Q>     8.071000</Q>
      <spint> 0.0000</spint>
      <spinf> 0.0000</spinf>
      <TlowHf>-1.0000</TlowHf>
      <Tlowfit> 0.0100</Tlowfit>
      <Thighfit> 10.</Thighfit>
      <acc> 1.900000e-06</acc>
      <a1> 6.225343e+00</a1>
```

```
      <a2> 1.023384e-02</a2>
      <a3>-1.272184e+00</a3>
      <a4> 3.920127e+00</a4>
      <a5>-1.966720e-01</a5>
      <a6> 1.394263e-02</a6>
      <a7>-1.389816e+00</a7>
      <a8> 2.983430e+01</a8>
   </non_smoker_fit>
</reaction>

<!-- o15 to n15 + positron + neutrino_e -->

<reaction>
   <source>Tuli (2000)</source>
   <reactant>o15</reactant>
   <product>n15</product>  <product>positron</product>
   <product>neutrino_e</product>
   <single_rate>5.6704e-3</single_rate>
</reaction>

<!--ne24 + positron -> na24 + anti-neutrino_e-->
<reaction>
  <source>ffn rates</source>
  <reactant>ne24</reactant>
  <reactant>positron</reactant>
  <product>na24</product>
  <product>anti-neutrino_e</product>
  <user_rate key="two-d weak rates">
    <properties>
      <property name="Extra information">
        Data are from Fuller, Fowler, and Newman (1982).  The user
        interpolates the 2-d table given input t9 and electron density
        (Ye * mass density).
      </property>
      <property name="log10_rhoe" tag1="0">1.000</property>
      <property name="log10_rhoe" tag1="1">2.000</property>
      <property name="log10_rhoe" tag1="2">3.000</property>
      <property name="log10_rhoe" tag1="3">4.000</property>
      <property name="log10_rhoe" tag1="4">5.000</property>
      <property name="log10_rhoe" tag1="5">6.000</property>
      <property name="log10_rate" tag1="0" tag2="0">-.1000E+03</property>
      <property name="log10_rate" tag1="0" tag2="1">-.1000E+03</property>
      <property name="log10_rate" tag1="0" tag2="2">-.1000E+03</property>
      <property name="log10_rate" tag1="0" tag2="3">-.1000E+03</property>
      <property name="log10_rate" tag1="0" tag2="4">-.1000E+03</property>
      <property name="log10_rate" tag1="0" tag2="5">-.1000E+03</property>
```

```
      <property name="log10_rate" tag1="1" tag2="0">-.5453E+02</property>
      <property name="log10_rate" tag1="1" tag2="1">-.5553E+02</property>
      <property name="log10_rate" tag1="1" tag2="2">-.5655E+02</property>
      <property name="log10_rate" tag1="1" tag2="3">-.5770E+02</property>
      <property name="log10_rate" tag1="1" tag2="4">-.6005E+02</property>
      <property name="log10_rate" tag1="1" tag2="5">-.6824E+02</property>
      <property name="log10_rate" tag1="2" tag2="0">-.2767E+02</property>
      <property name="log10_rate" tag1="2" tag2="1">-.2867E+02</property>
      <property name="log10_rate" tag1="2" tag2="2">-.2968E+02</property>
      <property name="log10_rate" tag1="2" tag2="3">-.3073E+02</property>
      <property name="log10_rate" tag1="2" tag2="4">-.3224E+02</property>
      <property name="log10_rate" tag1="2" tag2="5">-.3642E+02</property>
      <property name="log10_rate" tag1="3" tag2="0">-.1371E+02</property>
      <property name="log10_rate" tag1="3" tag2="1">-.1471E+02</property>
      <property name="log10_rate" tag1="3" tag2="2">-.1571E+02</property>
      <property name="log10_rate" tag1="3" tag2="3">-.1673E+02</property>
      <property name="log10_rate" tag1="3" tag2="4">-.1790E+02</property>
      <property name="log10_rate" tag1="3" tag2="5">-.2017E+02</property>
      <property name="log10_rate" tag1="4" tag2="0">-.7939E+01</property>
      <property name="log10_rate" tag1="4" tag2="1">-.8352E+01</property>
      <property name="log10_rate" tag1="4" tag2="2">-.9301E+01</property>
      <property name="log10_rate" tag1="4" tag2="3">-.1031E+02</property>
      <property name="log10_rate" tag1="4" tag2="4">-.1137E+02</property>
      <property name="log10_rate" tag1="4" tag2="5">-.1292E+02</property>
      <property name="log10_rate" tag1="5" tag2="0">-.6462E+01</property>
      <property name="log10_rate" tag1="5" tag2="1">-.6483E+01</property>
      <property name="log10_rate" tag1="5" tag2="2">-.6689E+01</property>
      <property name="log10_rate" tag1="5" tag2="3">-.7511E+01</property>
      <property name="log10_rate" tag1="5" tag2="4">-.8542E+01</property>
      <property name="log10_rate" tag1="5" tag2="5">-.9846E+01</property>
      <property name="t9" tag1="0">.010</property>
      <property name="t9" tag1="1">.100</property>
      <property name="t9" tag1="2">.200</property>
      <property name="t9" tag1="3">.400</property>
      <property name="t9" tag1="4">.700</property>
      <property name="t9" tag1="5">1.000</property>
    </properties>
  </user_rate>
</reaction>

</reaction_data>
```

The root tag for the file is **reaction_data**. The data are a sequence of reactions (**reaction** tags). There is no bound on the number of reactions that can be present in the file. For each reaction, the following tags are present:

source: A string giving the reaction source for the data. This is an optional tag, and the string is usually a reference to a publication with the data.

reactant: A string giving the name of one of the reactants (required). The number of reactants that may be present is unbounded, but there needs to be at least one reactant.

product: A string giving the name of one of the products (required). The number of products that may be present is unbounded, but there needs to be at least one product.

The reaction string is constructed from the reactants and products. The nuclear species are named with small letters, a number, and, if present, the state label. For example, ^{16}O is o16 as a reactant or product. The reaction string constructed from the reactants and products should be valid, that is, it should satisfy baryon number, lepton number, and charge conservation. To do this, the reactants or products are not just nuclear species but can also be "gamma" (for gamma-rays), "electron" (for an electron, that is, an e^-), "positron" (for a positron, that is, an e^+), and "neutrino_e" (for an electron-type neutrino, that is, a ν_e), "anti-neutrino_e" (for an electron-type anti-neutrino, that is, a $\bar{\nu}_e$). Other allowed leptons are muons ("mu"), anti-muons ("anti-mu"), tauons ("tau"), anti-tauons ("anti-tau"), and their corresponding neutrinos ("neutrino_mu", "anti-neutrino_mu", "neutrino_tau", and "anti-neutrin_tau").

The rate data follow the reactants and products. At present, the rate data can be one of four types. Only one type can be present for a particular reaction. For a rate that is the same at all temperatures, the data are between **single_rate** tags:

single_rate: The rate per nuclide per second. This is the data for a rate that is the same at all temperatures.

For a rate whose data are contained in a table (**rate_table** tag), the data are in table entries (**points**):

t9: The temperature in billions of Kelvins (required). This is a non-negative float.

rate: The rate per interaction pair or triplet (or higher multiplet) per second (required). This is a non-negative float.

sef: The stellar enhancement factor (required). This is a non-negative float. If the **rate** datum is for the ground state, libnucnet multiplies the rate by the sef to correct for the excited state rate. If the rate for the given temperature is already corrected for excited states, an sef of 1.0 should be used.

As of version 0.4, it is no longer necessary for the table data to be sorted in ascending order by temperature. The routine to update the input data sorts them before storing.

libnucnet computes rates from rate table data by interpolating the $\log_{10}$ of the product of the rate and the sef with the GNU Scientific Library spline interpolation routines. The result is then exponentiated. libnucnet does not extrapolate beyond the table. If the input temperature is lower than the lowest temperature in the table, the lowest temperature value for the rate times sef is used. If the input temperature is higher than the highest temperature in the table, the highest temperature value for the rate times sef is used. The user should therefore supply rate data for the full temperature range expected in the problem. If only two points are provided for the table, the rate is computed between those points by linear interpolation, not by the spline routines.

For a rate that is given by a non-smoker fit (**non_smoker_fit**) [Rauscher & Thielemann(2000)], the data are between the following tags:

Zt: Atomic number of the target nucleus (optional). It is a non-negative integer.

At: Atomic mass of the target nucleus (optional). It is a positive integer.

Zf: Atomic number of the final nucleus (optional). It is a non-negative integer.

Af: Atomic mass of the final nucleus (optional). It is a positive integer.

Q: Q value of the reaction (optional). It is a float. libnucnet does not use this but rather calculates it from data from the nuclear data file.

spint: Spin of the target nucleus (optional). It is a non-negative float. libnucnet does not use this but rather uses data from the nuclear data file.

spinf: Spin of the final nucleus (optional). It is a non-negative float. libnucnet does not use this but rather uses data from the nuclear data file.

TlowHf: Lowest value for which the Hausher-Feshbach fit works (optional). It is a float.

Tlowfit: Lowest value for which the fit works (required). It is a float.

Thighfit: Highest value for which the fit works (optional). If not supplied, it is assumed to be a default value, initially set to 10. It is a float.

acc: Accuracy of the fit (optional). It is a non-zero float.

a1: First non-smoker fit parameter (required). It is a float.

a2: Second non-smoker fit parameter (required). It is a float.

a3: Third non-smoker fit parameter (required). It is a float.

a4: Fourth non-smoker fit parameter (required). It is a float.

a5: Fifth non-smoker fit parameter (required). It is a float.

a6: Sixth non-smoker fit parameter (required). It is a float.

a7: Seventh non-smoker fit parameter (required). It is a float.

a8: Eighth non-smoker fit parameter (optional). It is a float. libnucnet does not actually use this parameter.

The rate per interacting pair, triplet, or higher multiplet is computed by the Non-Smoker fit as

$$exp\left(a1 + a2/T_9 + a3/T_9^{1/3} + a4T_9^{1/3} + a5T_9 + a6T_9^{5/3} + a7\log T_9\right) \quad (4)$$

As of version 0.2, if the input temperature in billions of K is less than **Tlowfit**, libnucnet computes the reaction rate at **Tlowfit**. Version 0.1 of libnucnet set the rate to zero for an input temperature less than **Tlowfit**. As of version 0.3, if the temperature in billions of K is greater than **Thighfit**, libnucnet computes the reaction rates at **Thighfit**. If the user does not supply **Thighfit**, it is taken as a default value, which is set by D_THIGHTFIT_DEFAULT in Libnucnet__Reac.h and, in the distribution, is set to 10.

Also as of version 0.2, libnucnet is able to handle multiple non-smoker fits. Here the format is similar to the non-smoker fit data above except that there are now **fit** tags between the **non_smoker_fit** tags. An example is the following:

```
<!--he4 + he4 + he4 -> c12 + gamma-->
<reaction>
  <source>non smoker example</source>
  <reactant>he4</reactant>
  <reactant>he4</reactant>
  <reactant>he4</reactant>
  <product>c12</product>
  <product>gamma</product>
  <non_smoker_fit>
    <fit note="non-resonant part">
      <spint>0</spint>
      <spinf>0</spinf>
      <TlowHf>-1</TlowHf>
      <Tlowfit>0.01</Tlowfit>
      <Thighfit>12.</Tlowfit>
      <acc>1.9e-06</acc>
      <a1>5.3463</a1>
      <a2>0</a2>
      <a3>-37.1289</a3>
      <a4>14.2705</a4>
      <a5>-92.885</a5>
      <a6>-20.4254</a6>
      <a7>-0.666667</a7>
      <a8>0</a8>
```

```
    </fit>
    <fit note="The first resonance part">
      <spint>0</spint>
      <spinf>0</spinf>
      <TlowHf>-1</TlowHf>
      <Tlowfit>0.01</Tlowfit>
      <Thighfit>10.</Tlowfit>
      <acc>1.9e-06</acc>
      <a1>-11.8694</a1>
      <a2>-4.32998</a2>
      <a3>0</a3>
      <a4>-6.24062</a4>
      <a5>0.715957</a5>
      <a6>-0.0561058</a6>
      <a7>-1.5</a7>
      <a8>0</a8>
    </fit>
    <fit note="The second resonance part">
      <spint>0</spint>
      <spinf>0</spinf>
      <TlowHf>-1</TlowHf>
      <Tlowfit>0.01</Tlowfit>
      <Thighfit>8.</Tlowfit>
      <acc>1.9e-06</acc>
      <a1>-121.677</a1>
      <a2>-1.36658</a2>
      <a3>0</a3>
      <a4>1.86071</a4>
      <a5>-130.231</a5>
      <a6>-7.77528</a6>
      <a7>-1.5</a7>
      <a8>0</a8>
    </fit>
  </non_smoker_fit>
</reaction>
```

Each fit has the same form as a single non-smoker fit. The optional note attribute allows the user the identify the different fits. The total reaction rate is the sum of the rate from each fit, as computed by Eq. (4).

As of version 0.5, it is possible for users to supply their own function to be applied during calculation of a reaction rate. The data for the function are supplied between **user_rate** tags, which have a required **key** attribute that identifies the user-supplied function associated with the data. The data for the function are "properties", so the data are enclosed between **properties** tags and each datum is enclosed in **property** tags, which is a string representing

any kind of data type. Each **property** has the following attributes:

name: The name associated with the property (required). It is a string.

tag1: A tag associated with the property (optional). It is a string and helps distinguish properties with the same name.

tag2: Another tag associated with the property (optional). It is a string and helps distinguish properties with the same name and tag1.

A libnucnet technical report describes application of user-supplied rate functions in more detail, and examples in the libnucnet distribution demonstrate their use.

3 Libnucnet__Net

The Libnucnet__Net structure consists of a collection of nuclei and reactions among them. The input is the nuclear data and the reaction data, and the XML file is therefore simply a combination of the nuclear data and the reaction data. It has the form then

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<nuclear_network>

  <nuclear_data>
  .
  .
  .
  </nuclear_data>

  <reaction_data>
  .
  .
  .
  </reaction_data>

</nuclear_network>
```

An xslt stylesheet provided with the libnucnet distribution allows the user to combine a nuclear data and reaction data input XML file to construct a Libnucnet__Net input XML file easily.

A valid reaction is defined as one that satisfies conservation of baryon number, lepton number, and charge and occurs between nuclei included in the network (that is, nuclei that have data in the **nuclear_data** part of the input

file). Routines attached to Libnucnet__Net compute reverse reaction rates by detailed balance. To compute reverse rates, libnucnet uses the nuclear data from the **nuclear_data** part of the Libnucnet__Net input XML file. In particular, the reverse reaction rate routines use the Q value of the reaction computed from the mass excesses and nuclear partition functions of the reactants and products. The reverse rate is computed by detailed balance as described, for example, in [Fowler et al.(1967)Fowler, Caughlan, & Zimmerman]. Clearly data for all reactants and products in a reaction must be present for the reverse rate to be calculated. Thus libnucnet only computes reverse rates for valid reactions.

4 Zone Data

The time-dependent data in libnucnet is contained in zones. Data for each zone can be read in from an input XML. As of version 0.3, the XML data for zones is placed between **zone_data** tags. For example, for a single-zone calculation, the input XML file would look like:

```
<zone_data>

  <zone>

    <mass_fractions>
      <nuclide name="n">
        <z>0</z>
        <a>1</a>
        <x>0.5</x>
      </nuclide>
      <nuclide>
        <z>1</z>
        <a>1</a>
        <x>0.3</x>
      </nuclide>
      <nuclide>
        <z>6</z>
        <a>12</a>
        <x>0.2</x>
      </nuclide>
    </mass_fractions>

  </zone>

</zone_data>
```

The data between the **zone** tags are for a set of **mass_fractions**, which consist of a sequence of nuclides (**nuclide** tag). For each nuclide, the data are:

name: The name of the nuclide as an attribute (optional).

z: The atomic number of the nuclide (required). This is a non-negative integer.

a: The mass number of the nuclide (required). This is a positive integer.

x: The mass fraction of the species (required). This is a float whose value lies between 0 and 1.

The sum of all the mass fractions for all the nuclides should be unity, although this is not required. If both the nuclide **name** and the **z** and **a** are both provided, libnucnet will assign the abundance to the species as determined by the **name**.

When there are multiple zones in the calculation, the zone data part of the input data file has a form like the following example:

```
<zone_data>

  <zone label1="x1" label2="y1" label3="z1">
    <mass_fractions>
      <nuclide>
        <z>0</z>
        <a>1</a>
        <x>0.5</x>
      </nuclide>
      <nuclide>
        <z>1</z>
        <a>1</a>
        <x>0.5</x>
      </nuclide>
    </mass_fractions>
  </zone>

  <zone label1="x2" label2="y2" label3="z2">
    <mass_fractions>
      <nuclide name="n">
        <z>0</z>
        <a>1</a>
        <x>0.5</x>
      </nuclide>
      <nuclide name="h1">
        <z>1</z>
        <a>1</a>
        <x>0.4</x>
```

```
      </nuclide>
      <nuclide name="he4">
        <z>2</z>
        <a>4</a>
        <x>0.1</x>
      </nuclide>
    </mass_fractions>
  </zone>

</zone_data>
```

For each **zone**, there is a set of **mass_fractions**, which consist of a sequence of nuclides (**nuclide** tag). As before, the data for each nuclide is the name, atomic number, mass number, and mass fraction of each species with non-zero abundance in the zone. Also, the mass fractions within a given zone should all sum to unity.

libnucnet sets up a number of zones equal to the number of zones in the input file. The zones are labeled by the supplied label attributes. A zone may have up to three labels, which are strings. If no labels are provided, the first zone will be given labels such that **label1** = "0", **label2** = "0", **label3** = "0", the second zone will have **label1** = "1", **label2** = "0", **label3** = "0", the third zone will have **label1** = "2", **label2** = "0", **label3** = "0", etc. If some labels are provided for a zone but others are not, those labels not provided are given the value "0". If identical labels are provided for two zones, libnucnet error handling will be invoked.

As of version 0.3, it is possible to assign optional properties to a zone. An optional property's value is stored as a string that is identified by another string giving the property's name. The user may also supply two additional tags to identify the property. The property can be assigned in the input XML file. An example would be:

```
<zone_data>

  <zone label1="1">
    <optional_properties>
       <property name="t9" tag1="0">10.</property>
       <property name="t9" tag1="1">9.</property>
       <property name="t9" tag1="2">8.</property>
       ...
       <property name="t9" tag1="20">0.1</property>
       <property name="time" tag1="0" tag2="seconds">0.</property>
       <property name="time" tag1="1" tag2="seconds">1.e-3</property>
       <property name="time" tag1="2" tag2="seconds">2.e-3</property>
       ...
```

```
      <property name="time" tag1="20">1.</property>
   </optional_properties>
   <mass_fractions>
     ...
   </mass_fractions>
  </zone>

  ...

</zone_data>
```

5 Libnucnet

Nuclear data, reaction data, and zone data can be combined into a single input XML file, which can be read in with libnucnet API routines. The combined file looks like:

```
<libnucnet_input>

  <nuclear_data>
  ...
  </nuclear_data>

  <reaction_data>
  ...
  </reaction_data>

  <zone_data>
  ...
  </zone_data>

</libnucnet_input>
```

This input can be read in with the libnucnet API routine Libnucnet__new__from__from__xml().

References

[Fowler et al.(1967)Fowler, Caughlan, & Zimmerman] Fowler, W. A., Caughlan, G. R., & Zimmerman, B. A. 1967, Ann. Rev. Astron. Astrophys., 5, 525, Thermonuclear Reaction Rates

[Gupta & Meyer(2001)] Gupta, S. S. & Meyer, B. S. 2001, Phys. Rev., 64, 25805, Internal equilibration of a nucleus with metastable states: ^{26}Al as an example

[Rauscher & Thielemann(2000)] Rauscher, T. & Thielemann, F. 2000, At. Data Nucl. Data Tables, 75, 1, Astrophysical Reaction Rates From Statistical Model Calculations