LA-UR-17-20567
Approved for public release; distribution is unlimited.

Title: User Manual for Whisper-1.1

Author(s): Brown, Forrest B.
Rising, Michael Evan
Alwin, Jennifer Louise

Intended for: Whisper release with MCNP6.2

Issued: 2017-01-26

User Manual for Whisper-1.1

Forrest Brown, Michael Rising, Jennifer Alwin

Monte Carlo Methods, Codes, & Applications Group, XCP-3, LANL

Contents

1.0 Introduction

Whisper is a statistical analysis package developed in 2014 to support nuclear criticality safety (NCS) validation [1-3]. It uses the sensitivity profile data for an application as computed by MCNP6 [4-6] along with covariance files [7,8] for the nuclear data to determine a baseline upper-subcritical-limit (USL) for the application. Whisper version 1.0 was first developed and used at LANL in 2014 [3]. During 2015-2016, Whisper was updated to version 1.1 and is to be included with the upcoming release of MCNP6.2.

This document describes the user input and options for running *whisper-1.1*, including 2 *perl* utility scripts that simplify ordinary NCS work, *whisper_mcnp.pl* and *whisper_usl.pl*. For many detailed references on the theory, applications, nuclear data & covariances, SQA, verification-validation, adjoint-based methods for sensitivity-uncertainty analysis, and more – see the *Whisper – NCS Validation* section of the MCNP Reference Collection at mcnp.lanl.gov. There are currently over 50 Whisper reference documents available.

1.1 Overview of Whisper usage and distribution files

Figure 1 provides an overview of the use of MCNP6 to generate sensitivity profiles for an application problem, and the use of Whisper to select similar benchmarks and perform the statistical analysis for determining a baseline USL for the application. It goes without saying, and will not be repeated throughout this document, that the baseline USL provided by Whisper is not necessarily the number that should be used in NCS evaluations; NCS analyst judgment should always be a central factor in determining USLs. Specifically, analyst judgment is needed in assessing operations, conservatively modeling an operation, and assessing whether additional margin is needed. MCNP6-Whisper cannot do those things. However, the use of MCNP6-Whisper provides repeatable, quantitative, physics-based information to NCS analysts for determining USLs, replacing much of what used to be carried out based solely on expert judgment. The combination of MCNP6-Whisper calculations plus the experience and expert judgment of NCS analysts has potential for significantly improving the determination of USLs for NCS evaluations.

See the Whisper-1.1 Release Notes for instructions on installing Whisper-1.1 and its associated data files. **Figure 2** shows the directory structure and selected files present after installing Whisper-1.1.

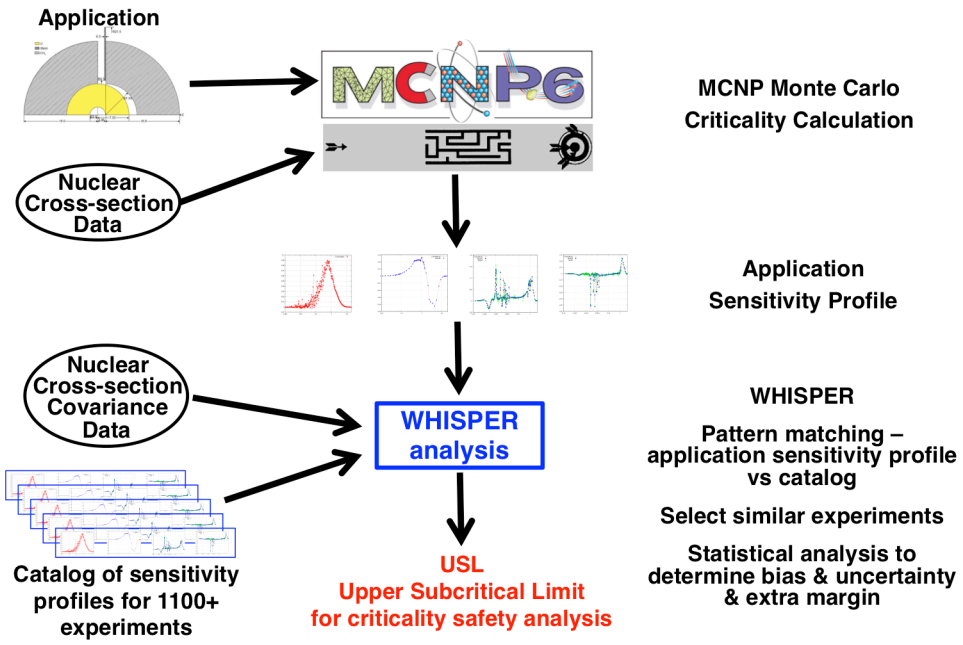

Figure 1. MCNP-Whisper methodology for NCS validation

```
WHISPER/
      Makefile
      bin/
            linux/
            macosx/
            windows/
      Benchmarks/
            Correlations/
                  BenchCorrel.dat
            Exclude/
                  ExcludedBenchmarks.dat
            Inputs/
                  MCNP6 input files for benchmarks...
            Sensitivities/
                  Sensitivity profiles from benchmarks...
            TOC/
                  BenchmarkTOC.dat
      CovarianceData/
            BLO-44g/
                  Data/
                        ACE covariance data files...
                  Adjusted/
                        ACE covariance files after GLLS adjust...
      Documentation/
            whisper_docs.html
      Source/
            Makefile
            Source code files...
      Testing/
            Makefile
            Installation/
      Utilities/
```

Figure 2. Directory structure for Whisper-1.1

2.0 Whisper-1.1 environment variables

Only 2 environment variables need to be set in order to use Whisper-1.1: PATH and WHISPER_PATH. These can be set in user startup files (e.g., *.bash_profile, .bashrc, .tcshrc,* etc.) or explicitly using the following commands:

bash shell:	`export`	`WHISPER_PATH="install-path/WHISPER"`
		`PATH="$WHISPER_PATH/bin:$PATH"`
csh or tcsh:	`setenv`	`WHISPER_PATH "install-path/WHISPER"`
	`setenv`	`PATH "$WHISPER_PATH/bin:$PATH"`
windows:	`set`	`WHISPER_PATH=install-path\WHISPER`
	`PATH`	`%WHISPER_PATH%\bin;%PATH%`

In the above command, `install-path` is the absolute pathname for the directory where Whisper-1.1 was installed. In standard installations with MCNP6, Whisper-1.1 is installed into the `MCNP/MCNP_CODE` directory, so that examples of the `install-path` would be `$HOME/MCNP/MCNP_CODE` for Linux or Mac, and `C:\MCNP\MCNP_CODE` for Windows. `install-path` must start with a "/" on Linux or Macs, and must start with a drive-letter on Windows, e.g., "`C:\`". For the LANL HPC clusters, authorized MCNP users can access Whisper-1.1 by using `/usr/projects/mcnp/ncs` for the `install-path` above.

Note that normally the MCNP6 environment variables DATAPATH, PATH, and DISPLAY must also be defined. In setting the PATH variable above, it is assumed that the WHISPER_PATH is set after the PATH is adjusted to include MCNP. That is, the Whisper bin directory should appear in the PATH environment variable before the MCNP bin path appears.

After setting the above environment variables, an easy check is to type (in the same command window where the environment variables were set): `whisper --version`

This command displays the version information for whisper-1.1

3.0 Whisper-1.1 command-line arguments

Table 1 lists the command-line arguments used by Whisper-1.1. For most arguments, users can use a shortened option name to save typing, or a longer meaningful option name for clarity. Examples for typical applications using these arguments are presented in Section 7. When using the *whisper_mcnp.pl* and *whisper_usl.pl* scripts to simplify NCS work, the scripts set the proper arguments listed in Table 1 for Whisper, and users do not need to directly run whisper-1.1.

Table 1. Whisper-1.1 Command-line Arguments

Application Specifications

-a file, --applications file

file is the table of contents (TOC) file for applications. It contains a list of the filenames and the calculated keff and standard-deviation for each, one line per application. For an application, the filename is the name of the MCNP6 input file (256 character or less), and the calculated keff and standard-deviation follow. One or more blanks should separate the entries on a single line. Filenames should be relative to the current working directory. The first line of the file is considered a comment and is ignored. Lines that are all blank are ignored. Lines that begin with a "#" are comments and are ignored.

Default: none

-ap dir, --appsenpath dir

dir is the directory containing the sensitivity profile data for each of the applications listed in the application TOC file. dir may be relative to the current working directory, or an absolute pathname. This directory should contain files with a "k" appended to the name of each application file.

Default: none

Benchmark Specifications

-b file, --benchmarks file

file is a table of contents (TOC) file for the catalog of benchmark results. It has the same format as the applications TOC file: The first line is ignored; blank lines are ignored; lines beginning with "#" are comments & ignored. There should be one line for each benchmark with the filename, keff, and standard-deviation.

Default: `$WHISPER_PATH/Benchmarks/TOC/BenchmarkTOC.dat`

-bp dir, --benchsenpath dir

dir is the directory containing the sensitivity profile data for each of the applications listed in the application TOC file. dir may be relative to the current working directory, or an absolute pathname. This directory should contain files with a "k" appended to the name of each application file.

Default: `$WHISPER_PATH/Benchmarks/Sensitivities`

-x file, --exclude file

file contains of list of benchmarks (from the Benchmark TOC file) that should NOT be used in subsequent whisper-1.1 calculations, 1 filename per line. The entry "`-x none`" or "`--exclude none`" can be used to override any exclusions and use all benchmarks listed in the BenchmarkTOC file.

Default: `$WHISPER_PATH/Benchmarks/Exclude/ExcludedBenchmarks.dat`

Covariance Data Specifications

-c dir, --covlibpath dir

dir is the directory containing the ACE covariance files

Default: `$WHISPER_PATH/CovarianceData/BLO-44g/Data`

-d dir, --adjusted dir

dir is the directory containing ACE covariance files after GLLS adjustments

Default: `$WHISPER_PATH/CovarianceData/BLO-44g/Adjusted`

--ace_suffix nn

nn is a 2-digit identifier to use for the ACE covariance files. The default is 01, so that ACE covariance files will have names of the form 92235.01v.

Default: 01

Table 1. Whisper-1.1 Command-line Arguments (Continued)

Benchmark Correlation Specifications

-k file, --bench_correl file

file is the filename containing correlations between experimental benchmark results. These correlations are only used during the GLLS adjustment of covariance data, or during benchmark rejection. This file is normally used during the whisper-1.1 setup or modification of benchmark files, and is not normally used in user applications.
Default: `$WHISPER_PATH/Benchmarks/Correlations/BenchCorrel.dat`

Benchmark Rejection Specifications

-r file, --reject file

file is the filename where whisper lists the names of rejected benchmarks. This file is an output file generated during the setup or modification of benchmark files, and is not normally used in user applications. After running whisper-1.1 to determine the outliers to reject, the contents of this file should be appended to the Benchmark Exclusion file.
Default: `none`

User Options Specifications

-u file, --user_options file

file is a filename containing a list of user options to use instead of the default whisper-1.1 values. See Section X.X for details. This file is not normally used in user applications.
Default: none

General Options

-h, --help

Display the help information contained in this table to the screen.

-o file, --output file

file is the filename to use for the detailed Whisper output.
Default: whisper.out.

-t n, --threads n

Maximum number of threads to use for whisper-1.1 parallel threading.
Default: min(16, number-of-cpu-cores)

-v, --version

Display Whisper version information to the screen, then stop.

4.0 *whisper_mcnp.pl*

whisper_mcnp.pl is a *perl* script portable to Linux, Mac OS X, and Windows that simplifies the task of running MCNP6 to generate keff and the sensitivity profile for one or more application problems. This section describes the requirements and usage of *whisper_mcnp.pl*.

4.1 Preliminaries

4.1.1 *perl* is required

If a user is invoking this script and running MCNP6 on a Windows platform, some version of *perl* must first be installed. For Windows systems where the Cygwin package is installed, users should make sure that *perl* was included in the installation, and if not, add it. To check, open a command window and type "`perl --version`". For Windows systems without Cygwin, the ActivePerl Community Edition package can be downloaded for free from www.activestate.com/downloads, and is easy to install. After installation, type "`perl --version`" to verify that it was installed.

4.1.2 Environment variables must be set

The WHISPER_PATH, PATH, DATAPATH, and DISPLAY environment variables must be set properly, as discussed in Section 2.0.

4.1.3 Create Working directory & MCNP6 input files for application problems

In some working directory, create the MCNP6 input files for one or more application problems. As usual, make some test runs with MCNP6, including plotting the geometry. As a reminder, the file- and directory names for MCNP-related problems should not contain blanks. For illustration, the directions below assume the working directory name is `WD`, and `WD` contains 2 MCNP6 input files, `myjob1.i` and `myjob2.i`.

The MCNP6 input files should be created normally, but should not include input cards for `kopts` or `ksen`, since they will be added by *whisper_mcnp.pl*. The `kcode` card should be present, but it will be replaced by *whisper_mcnp.pl*. A `prdmp` card, if present, will also be replaced by *whisper_mcnp.pl*.

4.2 Actions by *whisper_mcnp.pl*

To run *whisper_mcnp.pl* in a command window, after changing directories to WD, simply type the script name followed by the names of the MCNP6 input files:

```
whisper_mcnp.pl     myjob1.i   mjob2.i
```

The *whisper_mcnp.pl* script then does the following:

1. Create a directory (in `WD`) named **`Calcs`**. This directory will be used to hold modified input files, and any files created by running MCNP6.
2. Each of the input files is read, and modified input files are written (with the same names) into the **`Calcs`** directory.
    a. By default, the `kcode` cards are replaced by:

    ```
    kcode    100000   1.0    100    600
    ```

    The entries on the `kcode` card may be overridden by additional options to *whisper_mcnp.pl* (see below). It is recommended that the last 2 entries not be changed. For benchmark calculations, it is recommended to use 100000 neutrons/cycle, while for most applications 10000 neutrons/cycle or 20000 neutrons/cycle should be adequate.

b. A `kopts` card is inserted:

```
kopts   blocksize= 5
```

c. A `ksen` card is inserted:

```
ksen1 xs
      rxn = +2 +4 -6 +16 102 103 104 105 106 107 -7 -1018
      erg =
       1.0000e-11 3.0000e-09 7.5000e-09 1.0000e-08 2.5300e-08 3.0000e-08
       4.0000e-08 5.0000e-08 7.0000e-08 1.0000e-07 1.5000e-07 2.0000e-07
       2.2500e-07 2.5000e-07 2.7500e-07 3.2500e-07 3.5000e-07 3.7500e-07
       4.0000e-07 6.2500e-07 1.0000e-06 1.7700e-06 3.0000e-06 4.7500e-06
       6.0000e-06 8.1000e-06 1.0000e-05 3.0000e-05 1.0000e-04 5.5000e-04
       3.0000e-03 1.7000e-02 2.5000e-02 1.0000e-01 4.0000e-01 9.0000e-01
       1.4000e+00 1.8500e+00 2.3540e+00 2.4790e+00 3.0000e+00 4.8000e+00
       6.4340e+00 8.1873e+00 2.0000e+01
```

This `ksen` card directs MCNP6 to compute sensitivity tallies for all isotopes and the 12 reactions considered by Whisper, for a set of energy tally bins that precisely match those of the ACE covariance data used with whisper-1.1

d. A `prdmp` card is inserted:

```
prdmp  j  9999999
```

3. Create a file (in `WD`) called **MCNPInputlist.toc**, that contains the names of the modified MCNP6 input files in the **`Calcs`** directory, 1 filename per line.

4. Create a directory (in `WD`) named **`KeffSenLib`**. This directory will be used to hold sensitivity profiles later, when the *whisper_usl.pl* script is run.

5. Run the MCNP6 calculations for each of the application input files:

   a. If *whisper_mcnp.pl* is being invoked on the front-end of an HPC Linux cluster, the **`-submit`** option should be supplied on the *whisper_mcnp.pl* command line. This results in the MCNP6 jobs being submitted to the compute nodes on the cluster using ordinary Moab commands. (The script will need to be modified if other batch job submittal systems are being used on the HPC cluster.) One job is submitted for each of the applications, with a runtime limit of 1 hour. (The runtime limit can be changed by other arguments to whisper_mcnp.pl, see below.) Users should check the batch queueing system for status (e.g., `showq –u username`) and wait for the jobs to complete.

   b. On an office or laptop system, the MCNP6 calculations will be run locally, 1 job at a time

   c. The `name=` option is used on the MCNP6 command line, so that output, runtpe, and srctp files in the `Calcs` directory will have the same name as the input file, with “o”, “r”, “s” appended respectively.

   d. For an HPC system using the **`-submit`** option, by default 16 OpenMP threads are used per job (i.e., “tasks 16” is supplied to MCNP6). For other cases, by default 4 OpenMP threads are used per job. With either approach, the number of threads that MCNP6 uses can be specified explicitly to *whisper_mcnp.pl* on the command line (see below).

After using *whisper_mcnp.pl* to set up and run the MCNP6 jobs, the following files and directories will exist, where names in **black** are the original files/directories, names in **green** were created by *whisper_mcnp.pl*, and names in **red** were created by MCNP6:

```
WD/
   myjob1.i    myjob2.i
   MCNPInputlist.toc
   Calcs/
        myjob1.i   myjob1.io   myjob1.ir   myjob1.is
        myjob2.i   myjob2.io   myjob2.ir   mjob2.is
   KeffSenLib/
```

4.3 Options for *whisper_mcnp.pl*

Table 2 lists all of the command-line options for *whisper_mcnp.pl*. These may be used as desired to override the default values.

Table 2. whisper_mcnp.pl usage & options

```
whisper_mcnp.pl [Options]  Filelist
 Options:
      -help             print this information
      -local            run MCNP jobs locally, on this computer [default]
      -submit           submit batch MCNP jobs, using msub
      -walltime x       walltime limit for submit batch jobs (eg, 01:00:00)
      -mcnp x           pathname for MCNP6 executable
      -xsdir x          pathname for MCNP6 xsdir file
      -data x           pathname for MCNP6 data, DATAPATH
      -threads x        number of threads for MCNP6
      -neutrons x       number of neutrons/cycle for MCNP6
      -discard x        number of inactive cycles for MCNP6
      -cycles x         total number of cycles for MCNP6

 Filelist:
      Names of MCNP6 input files. The names should not contain blanks.
      The files must include a KCODE card (that will be replaced), &
      must not contain KSENn, KOPTS, or PRDMP cards (they will be supplied)

 Defaults:        **for local**                **for submit**
      -mcnp       mcnp6                        mcnp6
      -xsdir      $DATAPATH/xsdir_mcnp6.2      $DATAPATH/xsdir_mcnp6.2
      -data       $DATAPATH                    $DATAPATH
      -walltime         -                      01:00:00
      -threads          4                      16
      -neutrons         10000                  100000
      -discard          100                    100
      -cycles           600                    600
```

5.0 *whisper_usl.pl*

whisper_usl.pl is a *perl* script portable to Linux, Mac OS X, and Windows that simplifies the task of running *whisper-1.1* for one or more application files. It must be run after the *whisper_mcnp.pl* script. The script should be run from the same working directory as the *whisper_mcnp.pl* script (WD). It is required that *perl* is available, environmental variables are set properly, and that *whisper_mcnp.pl* and associated MCNP6 runs completed successfully.

5.1 Actions by *whisper_usl.pl*

To run *whisper_usl.pl* in a command window, after changing directories to WD, simply type the script name. If it is desired to not use the default Benchmark Exclude file, then either "none" or a different filename to use can be included on the command line.

```
whisper_usl.pl      [excludefile]
```

The *whisper_usl.pl* script then does the following:

1. For each application input file listed in `MCNPInputList.toc` (denoted myjob.i below):
   a. Extract sensitivity profiles from `Calcs/myjob.io` and write the profile data into a file `myjob.ik` in directory `KeffSenLib`/
   b. Create (or add to) file `KeffSenList.toc`, adding 1 line per application containing the name of the MCNP input file, the benchmark experiment keff and standard-deviation (two 0.0 values for application problems), and the calculated keff & standard-deviation.
2. Run whisper-1.1, with arguments

   ```
   $WHISPER_PATH/bin/whisper  -a KeffSenList.toc  -ap KeffSenLib
   ```

   If an exclude file was specified on the *whisper_usl.pl* command-line, then the entry "`-x excludefile`" is also included.
3. The output from whisper-1.1 is put into the `WD/Whisper.out` file.

After running *whisper_mcnp.pl* & *whisper_usl.pl*, the following files/directories will exist, where names in black are the original files/directories, names in green were created by *whisper_mcnp.pl*, names in red were created by MCNP6, names in blue were created by *whisper_usl.pl*, and names in magenta were created by whisper-1.1:

```
WD/
   myjob1.i    myjob2.i
   MCNPInputlist.toc
   Calcs/
        myjob1.i   myjob1.io   myjob1.ir   myjob1.is
        myjob2.i   myjob2.io   myjob2.ir   mjob2.is
   KeffSenList.toc
   KeffSenLib/
        myjob1.ik  myjob2.ik
   Whisper.out
```

6.0 *whisper_get_sens.pl*

This *perl* script will parse an MCNP6 output file, locate the sensitivity profile data, and write the sensitivity profile data into a file. It is a utility script that may be needed if users want to directly run MCNP6 and *whisper-1.1* without using the *whisper_mcnp.pl* & *whisper_usl.pl* scripts.

Usage is straightforward:

```
whisper_get_sens.pl   mcnp_out_file   sens_file
```

The script will read the MCNP6-generated output file and write the sensitivity profile data into the sens_file. It will also print to the screen the values of k_{eff} ± std, EALF, and ANECF found in the MCNP6 output file.

Example:

```
bash: whisper_get_sens.pl  Calcs/in-28-2-1.txto  senfile

whisper_get_sens.pl
        ..... Read mcnp6 output  file: Calcs/in-28-2-1.txto
        ..... Create sensitivity file: senfile
        .....  KeffCalc = 0.96744 +- 0.00035
        .....  EALF     = 1.2228E-01 MeV
        .....  ANECF    = 1.4923E+00 MeV
```

7.0 Examples – Applications

7.1 Sample screen output & Whisper.out

Figure 3 illustrates the screen output produced for a test run of MCNP6-Whisper-1.1 using the *whisper_mcnp.pl* & *whisper_usl.pl* scripts, and **Figure 4** illustrates the Whisper.out file produced. For this example (which is used in classroom hands-on training), only 246 benchmarks were used. Many similar output lines were deleted for brevity (denoted by …… in Figures 3 & 4.)

7.2 Application problem, using *whisper_mcnp.pl* & *whisper_usl.pl* on a laptop or office computer

```
whisper_mcnp.pl   -neutrons  10000   myjob.i
whisper_usl.pl
```

7.3 Application problem, using whisper_mcnp.pl & whisper_usl.pl from the front-end node of an HPC cluster

```
whisper_mcnp.pl   -neutrons 20000 -submit   myjob.i
      . . . . . (wait for mcnp6 job to finish on compute node)
whisper_usl.pl
```

7.4 Application problem, running MCNP6 & whisper-1.1 maually, without using *whisper_mcnp.pl* & *whisper_usl.pl*

Prepare an MCNP6 input file for an application, xxx.i, explicitly including the `kcode, ksen1, kopts, prdmp` cards that are shown in Section 5.2 step 2.

Run MCNP6 using input file xxx.i.

```
mcnp6  name=xxx.i    tasks 16
```

Create the sensitivity file, xxx.ik

```
whisper_get_sens.pl   xxx.io   xxx.ik
```

Create an application TOC file, including the calculated keff & std-dev (from previous step) inserted below (in place of keff & std)

```
echo "#my explicit test…"           > appTOC.txt
echo "xxx.i  0.0 0.0  keff  std"   >> appTOC.txt
```

Run whisper-1.1 directly, with whatever benchmark libraries are desired

```
whisper   --applications   appTOC.txt    --appsenpath  "."  \
          --benchmarks     my-bench-TOC-file  \
          --benchsenpath   my-bench-sen-dir   \
          --exclude        my-exclude-file    \
          --threads 24   --output  my-out-file.txt
```

7.5 Benchmark setup or revision – adding new benchmarks

This example is relevant to setting up the collection of benchmark files, their associated sensitivity profiles, testing for outliers to be excluded, and generating adjusted covariance files.

These tasks are generally done by an experienced user or code developer in setting up the MCNP6-Whisper installation, and are not generally done by ordinary NCS analysts in their work.

It is important to note that the steps below alter the contents of the installed Whisper directories and datafiles. A backup copy of everything in the $WHISPER_PATH directory should be made before proceeding with the steps below.

Steps for adding new benchmarks & configuring Whisper:

1. Create the new benchmark files.
    a. Each file should be set up for use with the *whisper_mcnp.pl* script as described in Section 4. Specifically, the `kcode` card should be included (but will be replaced), and the `ksen1, kopts,` and `prdmp` cards should not be supplied (since they will be created).
    b. In addition, each of the new benchmark input files should contain a comment line of the form
    ```
    c   k(bmk) = 1.23456  +-  0.12345
    ```
    that provides the experimental measurement for the benchmark keff and standard-deviation, as obtained from the ICSBEP Handbook [9] section for that benchmark. (Of course the above numbers should be replaced by the appropriate ICSBEP values.)
2. Use the *whisper_mcnp.pl* & *whisper_usl.pl* scripts to run MCNP6-Whisper for the benchmarks:

    ```
    whisper_mcnp.pl    new-bench-1.i   new-bench-2.i
    whisper_usl.pl
    ```

    Ignore the *whisper_usl.pl* results; the sensitivity files created will be used in the next step.

3. Copy files to the Whisper benchmark directories:

```
cp  new-bench-1.i               $WHISPER_PATH/Benchmarks/Inputs
cp  new-bench-2.i               $WHISPER_PATH/Benchmarks/Inputs

cp  KeffSenLib/new-bench-1.ik   $WHISPER_PATH/Benchmarks/Sensitivities
cp  KeffSenLib/new-bench-2.ik   $WHISPER_PATH/Benchmarks/Sensitivities
```

4. Append the list of new benchmarks (with the experimental & calculated keff & std) to the Benchmark TOC file:

```
cat  KeffSenList.toc  >>  $WHISPER_PATH/Benchmarks/TOC/BenchmarkTOC.dat
```

5. If there are any benchmark correlations (typically obtained from DICE) that involve the newly added benchmarks, edit & add the new correlations to the file:

```
$WHISPER_PATH/Benchmarks/Correlations/BenchCorrel.dat
```

6. Regenerate the Benchmark Exclusion file using whisper-1.1, and replace the previous Exclusion file:

```
whisper  --exclude none  --reject  new-reject-file

cp  new-reject-file  \
    $WHISPER_PATH/Benchmarks/Exclude/ExcludedBenchmarks.dat
```

7. Update & replace the Adjusted Covariance files using whisper-1.1:

   First, copy all of the covariance files from BLO-44g/Data to BLO-44g/Adjusted:

```
cp    $WHISPER_PATH/CovarianceData/BLO-44g/Data/*  \
      $WHISPER_PATH/CovarianceData/BLO-44g/Adjusted
```

   Any covariance files that are adjusted in the step below will be overwritten with the adjusted version from Whisper-1.1; any covariance files that are <u>not</u> adjusted (because they are not used in any of the benchmarks) will be unchanged.

   Then, run whisper-1.1 to create adjusted covariance files:

```
whisper  --adjusted  $WHISPER_PATH/CovarianceData/BLO-44g/Adjusted
```

Figure 3. Sample screen output from using whisper_mcnp.pl & whisper_usl.pl

```
bash: whisper_mcnp.pl   -submit   myjob.i

******************
*                *
*  whisper_mcnp  *      a utility script to set up input & run MCNP for Whisper
*                *
******************
      Input File TOC          = MCNPInputList.toc
      Calculation directory   = Calcs
      Sensitivity directory   = KeffSenLib
      Neutrons/cycle          = 100000
      Cycles to discard       = 100
      Total Cycles to run     = 600
      MCNP6 executable        = /usr/projects/mcnp/mcnpexe -be
      XSDIR file              = /usr/projects/mcnp/MCNP_DATA/xsdir_mcnp6.2
      DATAPATH                = /usr/projects/mcnp/MCNP_DATA
      Threads                 = 16
      Wall-clock time for job = 01:00:00
      All jobs will be submitted using moab
      ...process  mcnp input file:  myjob.i
      ...modified mcnp input file:  Calcs/myjob.i
      ...submit mcnp job to cluster using moab:  myjob.i


bash: whisper_usl.pl

******************
*                *
*  whisper_usl   *      set up & run Whisper validation calculations
*                *
******************

=====> setup files for whisper
   ---> setup for problem myjob.i
      ...extract sensitivity profile data from:  Calcs/myjob.io
      ...copy    sensitivity profile data to:    KeffSenLib/myjob.ik
      ...extract calc Keff & Kstd    data from:  Calcs/myjob.io
      ...  KeffCalc= 0.96802 +- 0.00052,  ANECF= 1.4904E+00 MeV,  EALF= 1.2150E-01 MeV

=====> run whisper
/usr/projects/mcnp/ncs/WHISPER/bin/whisper -a KeffSenList.toc -ap KeffSenLib
 whisper-1.1                    2016-12-22   (Copyright 2016 LANL)
 WHISPER_PATH                 = /usr/projects/mcnp/ncs/WHISPER
 Benchmark TOC File           = /usr/projects/mcnp/ncs/WHISPER/Benchmarks/TOC/BenchmarkTOC.dat
 Benchmark Sensitivity Path   = /usr/projects/mcnp/ncs/WHISPER/Benchmarks/Sensitivities
 Benchmark Correlation File   =
 Benchmark Exclusion File     =
 Benchmark Rejection File     =
 Covariance Data Path         = /usr/projects/mcnp/ncs/WHISPER/CovarianceData/BLO-44g/Data
 Covariance Adjusted Data Path =
 Application TOC File         = KeffSenList.toc
 Application Sensitivity Path  = KeffSenLib/
 User Options File            =
 Output File                  = Whisper.out
       ........
 Reading benchmark data ...
 Reading application data ...
 Reading covariance data ...
 Reading adjusted covariance data ...
 Calculating application nuclear data uncertainties ...
 Calculating upper subcritical limits ...
......case     1  Ck=  0.41263
......case     4  Ck=  0.36554          ← all Ck's printed in Whisper.out,
......case     3  Ck=  0.63497             only a few printed to the screen
       ........
......case   246  Ck=  0.18901
                                    calc        data unc    baseline    k(calc)
    application                     margin      (1-sigma)   USL         > USL
    myjob.i                         0.01334     0.00209     0.97623    -0.00686
```

Figure 4. Sample Whisper.out output file associated with Figure 3

```
whisper-1.1                    2016-12-22   (Copyright 2016 LANL)
 WHISPER_PATH                  = /usr/projects/mcnp/ncs/WHISPER
 Benchmark TOC File            = /usr/projects/mcnp/ncs/WHISPER/Benchmarks/TOC/BenchmarkTOC.dat
 Benchmark Sensitivity Path    = /usr/projects/mcnp/ncs/WHISPER/Benchmarks/Sensitivities
 Benchmark Correlation File    =
 Benchmark Exclusion File      =
 Benchmark Rejection File      =
 Covariance Data Path          = /usr/projects/mcnp/ncs/WHISPER/CovarianceData/BLO-44g/Data
 Covariance Adjusted Data Path =
 Application TOC File          = KeffSenList.toc
 Application Sensitivity Path  = KeffSenLib/
 User Options File             =
 Output File                   = Whisper.out

 Reading benchmark data ...
   benchmark               k(bench)    unc         k(calc)     unc          bias        unc
   pu-met-fast-001-001.i   1.00000     0.01100     1.01174     0.00007    -0.01174     0.01100
             ..........

   246 benchmarks read,      0 benchmarks excluded.

 Reading application data ...
                               application     k(calc)     unc
                                   myjob.i     0.96802     0.00052

 Reading covariance data ...
  Reading covariance data for 1001 ...
             ..........
 Reading adjusted covariance data ...
  Reading covariance data for 1001 ...

 Calculating application nuclear data uncertainties ...
    application                                 adjusted    prior
    myjob.i                                     0.00209     0.01221

 Calculating upper subcritical limits ...
                                                calc        data unc    baseline    k(calc)
    application                                 margin      (1-sigma)   USL         > USL
    myjob.i                                     0.01334     0.00209     0.97623    -0.00686

    Benchmark population    =   48
    Population weight       =  28.56732
    Maximum similarity      =   0.96434

    Bias                    =   0.00850
    Bias uncertainty        =   0.00484
    Nuc Data uncert margin  =   0.00209
    Software/method margin  =   0.00500
    Non-coverage penalty    =   0.00000

    benchmark                                   ck          weight
    pu-met-fast-011-001.i                       0.9643      1.0000
    pu-met-fast-044-002.i                       0.9641      0.9958
    pu-met-fast-021-002.i                       0.9618      0.9545
    pu-met-fast-003-103.i                       0.9602      0.9252
    pu-met-fast-026-001.i                       0.9594      0.9099
    pu-met-fast-025-001.i                       0.9584      0.8912
    pu-met-fast-032-001.i                       0.9572      0.8699
    pu-met-fast-016-001.i                       0.9546      0.8221
    pu-met-fast-027-001.i                       0.9546      0.8217
             ........
    pu-met-fast-012-001.i                       0.9167      0.1283
    pu-met-fast-040-001.i                       0.9166      0.1269
    pu-met-fast-045-003.i                       0.9163      0.1209
    pu-met-fast-045-004.i                       0.9147      0.0909
    pu-met-fast-002-001.i                       0.9145      0.0874
```

For this application, 48 of the 246 benchmarks used in the test run were selected as neutronically similar & sufficient for valid statistical analysis

Benchmark rankings shown below

Acknowledgments

This work was supported and encouraged by the US DOE/NNSA Nuclear Criticality Safety Program. The original version of Whisper-1.0 was developed by B.C. Kiedrowski, now a Professor at the University of Michigan.

References

All references with a LANL report number, LA-UR-….., are available in the MCNP Reference Collection at mcnp.lanl.gov.

1. B.C. Kiedrowski, F.B. Brown, et al., "Whisper: Sensitivity/Uncertainty-Based Computational Methods and Software for Determining Baseline Upper Subcritical Limits", Nucl. Sci. Eng. **181**, 17-47, [also LA-UR-14-26558] (2015).
2. B.C. Kiedrowski, "User Manual for Whisper (v1.0.0), Software for Sensitivity- and Uncertainty-Based Nuclear Criticality Safety Validation", LA-UR-14-26436 (2014).
3. B.C. Kiedrowski, et al., "Validation of MCNP6.1 for Criticality Safety of Pu-Metal, -Solution, and -Oxide Systems", LA-UR-14-23352 (2014).
4. J.T. Goorley, et al., "Initial MCNP6 Release Overview," *Nuclear Technology*, **180**, 298 (2012).
5. J.T. Goorley, "MCNP6.1.1-Beta Release Notes," LA-UR-14-24680 (2014).
6. M.B. Chadwick, et al., "ENDF/B-VII.1 Nuclear Data for Science and Technology: Cross Sections, Covariances, Fission Product Yields and Decay Data," LA-UR-11-05121, Los Alamos National Laboratory, also: *Nuclear Data Sheets*, **112** Issue 12, pg. 2887-2996 (Dec 2011).
7. R.C. Little, T.Kawano, et al., "Low-fidelity Covariance Project", *Nuclear Data Sheets* **109**, 2828-2833 [also LA-UR-08-7732] (2008).
8. F.B. Brown, M.E. Rising, "Covariance Data File Formats for Whisper-1.0 & Whisper-1.1", LA-UR-17-20098 (2017).
9. International Handbook of Evaluated Criticality Safety Benchmark Experiments, NEA No. 7231, OECD Nuclear Energy Agency (2014).