

University of Saskatchewan
Subatomic Physics Internal Report
SPIR 152

BFRoot: Blowfish Root Analysis package

Version 1.0

Rob Pywell

May 2017, Last Update: May 11, 2017

Abstract

This report describes the root analysis codes for analysis of *Blowfish* data and experiments that may use a subset of *Blowfish* cells.

Contents

Contents	2
1 Introduction	3
2 Design Choices	3
3 Installation	3
4 Using Blowfish Root	5
5 Work Remaining	8
6 Lesser GNU Public License	9
References	13

1 Introduction

This is the first release of a suite of root analysis codes for analysis of *Blowfish* data and experiments that may use a subset of *Blowfish* cells.

The package includes a number of c++ codes that run within the Root [2] analysis environment. Has only been tested with root 5. Most of the codes make use of a visual interface that simplifies the analysis tasks. It provides for the conversion of the Lucid[1] data files to root files (using the RLucid[3] package) and the subsequent analysis to produce neutron yields for each *Blowfish* cell.

2 Design Choices

The codes make use of the some of design philosophy used by Ward Wurtz for his original BFRoot suite of codes. Some of the code's structure has been borrowed liberally from Ward's original code. However there have been a number of significantly different choices, and that has resulted in most codes being written from scratch.

One design choice has been that each of the code elements should be able to be run completely independently of the other element. There is no overarching management code that must be in place for other elements to work. This greatly facilitates debugging, and allows for the possibility that one element could be modified for a specialized application, without affecting other elements.

Another design choice was that all information and calibration files should be in a human readable form (i.e. are text files rather than being embedded within a root file.) Again this greatly facilitates debugging and allows simple changes to be made easily. As well these files are in a format that can be used by the Lucid data acquisition system. The time overhead in reading and decoding these text files is minimal since such files are generally read or written only once in the analysis of a run.

A third design choice was to introduce an intermediate step in the analysis of a run. After the original Lucid file has been converted to a root file, a file called a "filtered" file is created for later analysis steps. This filtered file contains only multiplicity one events, since only these have the proper ADC gate timing and are therefore the only events that can be used in the later analysis. The filter process also produces a "multiplicity file" that allows correction, in the final step of analysis, for the number of events lost by the filter. As well as only needing to record information from one cell per event, the filter also removes event types that are no longer needed, e.g. pedestal events. This very significantly reduced file size very much speeds up the later analysis steps. This is especially important in steps, such as fitting to find PSD parameters and applying final cuts, where frequent reprocessing of data is needed. The filter process can calculate light output and PSD values once the gains and PSD parameters have been determined.

3 Installation

Installation requires that RLucid, the package that converts lucid files to root files, be first installed. The latest RLucid is RLucid-1.1. This can be obtained from <http://nucleus.usask.ca>. Follow the installation steps given with it.

Unpack the gzipped tar file `BFRoot-1.0.tgz` in any appropriate place in your file system.

```
> tar zxvf BFRoot-1.0.tgz
```

This will create the directory `BFRoot-1.0`. It is probably useful to rename this directory something that is more indicative of the particular experiment you want to analyze. For the purposes of this guide it will assumed that it has been renamed simply `Blowfish`.

Change to that directory.

```
> cd Blowfish
```

A number of files and sub-directories have been created. If you will be analyzing standard 88 cell *Blowfish* data, either in the normal or GDH orientation, no changes need to be made before compiling.

If you will be analyzing data where the *Blowfish* cells are in some other arrangement (e.g. for data taken in the UTR in May 2016 or June 2015) you will need to need to modify `BFGlobal.h` to indicate the cell numbers used. As well, you may need to use a different “Looker” code. The “Looker” code is the process used by `RLucid` to convert `Lucid` data to a root file. The looker code "`BlowfishLooker.cxx`" and "`BlowfishLooker.h`" is for the standard *Blowfish* configuration. Additional looker codes are in the sub-directory `lookers`. (These may not have all the bells and whistles that are in the `BlowfishLooker` code however.)

The `rootlogon.C` is set up to load all the needed files when root is started. But these libraries need to be compiled on your system first. To compile, first start root.

```
> root
```

Root will complain that files are not available to be loaded. That is OK.

Then execute the code `compileall.C` that will build the needed libraries and leave them in this directory. Then quit root and start it again. i.e. do...

```
root [0] .x compileall.C
... output for successful library creation...
(int)0
root [1] .q
> root
(int)0
root [0]
```

Root should start without any errors.

The `BFRoot` GUI main menu can be started simply by typing new `BFMainMenu` then the analysis can be started.

(*Note:* If you modify any of the code and you recompile (e.g. with `.L PSDGui.cxx+`) it is usually necessary to quit root and start it again to avoid root crashes. If crashes persist, I have found that is sometimes necessary to remove all the libraries and dependency files (e.g. by `> rm *.d *.so`) and then recompile them all using `.x compileall.C`.)

Several sub-directories are needed by the analysis. They are created when you unpack the tar file. They are:

- `filtered` This is where the filtered root data files are stored.
- `histograms` This is where root files containing histograms are stored when a
- `Save Histograms` button is pressed in most of the GUI interfaces.

- `param` This folder contains many sub-directories where configuration and calibration files are stored. The structure mimics the directory structure use by the Lucid data acquisition system for online analysis of data as it comes in. But more information is stored in these files than is needed during data taking. Many of the files are run specific, in which case the filename will begin with the run number. e.g. in the sub-directory PSD might be the file `0139.psd_parameters.dat` which are the PSD parameters derived from run 139. Also in that directory might be the file `0139.psd_fitting_params.dat` which contain the variables used by the PSDGui to obtain the PSD parameters. This file would be written when pressing a `Save Fitting Parameters` button. In general the variables used by each analysis step can be saved in such a file to be read in automatically when you restart the program with that run number. Then you can continue where you left off.
- `root_data` This folder is where you might store the root files that result from using RLucid to convert the original Lucid files into Root format. This folder does not have to have this particular name. The folder could also be a link to somewhere else on your computer.

You will find that it is very convenient to have two other folders or links.

- `lucid_data` This would be a link to the folder contain all of the original lucid data files. This folder/link does not have to have this particular name. This is where you will select files to be converted by RLucid and written to the folder `root_data`. Files read by RLucid can be gzipped or not.
- `sim_data` This would be a link to the folder containing the output lucid files from the Geant4 simulation of the experiment. This folder/link does not have to have this particular name. These simulation files will be converted to root files by RLucid in the same way as for lucid data files.

Files written by the BlowfishDeuteron simulation[4], for example, will be automatically detected by BFRoot as being simulation data and the analysis method adjusted accordingly. (I have found it useful to adopt the convention that simulation files have run numbers 8000+ to differentiate them from experiment data files. BlowfishDeuteron follows this convention.)

4 Using Blowfish Root

When you start Blowfish Root by typing `new BFMainMenu` a GUI will be created with many options available.

- `RXLucid` will start the RLucid GUI which will allow conversion of Lucid data to Root data. Use of the GUI is described in the RLucid manual. It is also possible to batch process a number of lucid to root conversions without using the GUI. The file `convertlist.C` is included for this purpose. It can be modified if needed for your situation. Execute it using the `Root .x` command.

As distributed RLucid will use the translation code `BlowfishLooker`. This code makes use of the following files. `Cell_Map.dat` This is the map indicating which cells are connected to which ADC and TDC channels. This should be the same `Cell_Map.dat` that was used by Lucid when the data was taken. This should also be the same as the `Cell_Map.dat` used by the Geant4 simulation of the experiment. This file resides in the directory "param/Cells". (The `Cell_Map.dat`

is generated from the master cellmap file by the Perl script `makecellmap`). When RLucid converts the data using BlowfishLooker it produces a pedestals file which is stored in `param/Ped`. BlowfishLooker also creates some histograms of the raw data in each ADC and TDC channel which is stored within the root file created. If needed these histograms may be viewed later by the View Histograms GUI.

- **Open File** Once you have converted root files, you can open a root file for analysis. This will normally be selected from your `root_data` directory. Once opened the file name will be displayed in the status bar at the bottom of the GUI.

An open file can then be processed by any of the remaining analysis GUIs listed in the Main Menu. However the Find Gains, Find PSD Parameters, and Calculate Yields steps require the existence of a filtered root file. Therefore the usual first step is to create a filtered file.

- **Filter Data** This GUI will create a filtered file. Creating a filtered file needs several parameter files, but not all are needed depending on the type of data file to be process. A pedestal file is needed for all data files. However a pedestal file is normally already created by RLucid Looker process and will be recognized automatically. The Filter GUI does provide the facility to check that pedestals are correct and can regenerate them if needed.

If the data file is a simulation run, no other parameter files are needed unless the arrangement of the cells is not the standard *Blowfish* arrangement (in which case a "Cell Information" file is needed.)

If the data file is a source gain calibration run no other Parameter Files are needed, since the filtered file will only be read by the Find Gains analysis step.

If the data file is an AmBe source run (for example) that will be used for finding PSD parameters no other Parameter Files are required.

If the data file is a data production run with beam then all the parameter files are needed (except the Cell Information file if it is non-standard *Blowfish* arrangement). TDC offsets may be found using the Filter Gui itself. A TDC calibration file is usually obtained by examining the TDC offsets from two runs. One taken with a cable of known time length insert in the TDC start. If no TDC calibration file is provided the nominal TDC calibration of 100 ps/channel is used for all cells.

Cell gains may be loaded by selecting the appropriate file. This may be (a) a file generated directly by the Find Gains GUI, (b) a gain file obtained by fitting the results of several gain runs, or (c) a file created by the Track Gains function using the gain monitoring system.

PSD parameters are usually created from the Find PSD Parameters function.

Before writing the filtered file, select the appropriate check boxes in the Run Information section. In particular the *Blowfish* arm that is up is not included in the Lucid data stream so it would have to be set for each run.

The Rebuild Histograms can be used to update histograms after parameter files are read in. After you are satisfied, hitting the Write Filtered Root File and Multiplicity Fraction File button will run through all events and write the filtered file which will be stored in the filtered

directory. A Multiplicity Fraction file is also written to be used later by the Calculate Yields function.

Other functions included in the Blowfish Root Main Menu are:

- **Fit Flasher/Source Peaks** This find the peaks for each run needed by the gain monitoring systems.
- **Track Gains** This takes the Flasher and Source information along with the gains from source runs, to determine gains for each individual run.
- **View Scalers** This allows view of the scaler information contained in each Lucid run. It makes use of the information in the `Scaler_Map.dat` file stored in the "param/Scaler" directory to map scalers to the names of the signals. Also "View Scalers" can calculate the number of photons incident on the target during the run. To do this it makes use of the calibration factors stored in the file `CalibrationFactors.dat`, or the calibration factor can be entered in the GUI.
- **Calculate Yields** This function calculates the yield for each cell based on cuts that can be entered in the GUI or stored in a cut file. It provides the facility to make corrections to the yield based on randoms, events lost by the PSD cut, and the multiplicity fraction.

In the `View Objects` section are functions to view histograms saved in root files. More details on the functionality of each process is still to be written.

5 Work Remaining

More Testing... Add option to batch process the filtering process. Finish documentation...

6 Lesser GNU Public License

GNU LESSER GENERAL PUBLIC LICENSE

Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software—to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages—typically libraries—of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) The modified work must itself be a software library.
- b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
- c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
- d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

References

- [1] Doug Murray, et al., SPIR-100, *LUCID User's Guide*.
available at <http://nucleus.usask.ca> under "Reports".
- [2] ROOT Collaboration, *The ROOT System Home Page*. <http://root.cern.ch/>.
- [3] Ward Wurtz and Rob Pywell, SPIR-151, *RLucid: Lucid to Root Conversion Package*,
available at <http://nucleus.usask.ca> under "Reports".
- [4] Rob Pywell, *Blowfish Simulation for Deuteron experiments*,
available at <http://nucleus.usask.ca> under "Software".