The SaSSY Manual

(Version 0.1.3)

Table of Contents

The SaSSY Manual (Version 0.1.2)

1 Copyright

2 Installing SaSSY

At the time of writing, SaSSY has an “academic” build environment. That means, it works, but it could be improved a whole lot more. It should just build on any Linux / Unix environment but there may be some bugs here and there. The end goal is to have a ./configure → make → make install setup. Anyone who wants to help me streamline the process and include libtool, autoconf, etc. is more than welcome to help, and any help will be GREATLY appreciated.

SaSSY has only been tested with different versions of the gcc compiler. Most work well, however, I recommend version 4.5.4 as this seems to have given the best results. See notes below for any compiler specific information.

2.1 Dependencies

There are a few dependencies which will need to be installed before SaSSY can be compiled and run. The versions given here are the versions I have tested but your mileage may vary.

Dependency	Version	Download
String-CamelCase	v-0.02	http://search.cpan.org/~hio/String-CamelCase-0.02/
pcre	v-8.11	http://www.pcre.org/
ncurses	v-5.7	http://ftp.gnu.org/pub/gnu/ncurses/
Google sparsehash	v-1.10	http://code.google.com/p/google-sparsehash/
readline	v-6.1	http://tiswww.case.edu/php/chet/readline/rltop.html

Table 1. SaSSY dependencies, versions and URLs

These need to be installed and tested. If you are installing SaSSY in your home directory then you may still be ok, it depends on what's installed on your system. It's best to just try and install SaSSY and deal with any errors as they occur. Otherwise, see section 2.4 below.

2.2 Get the source

Please visit sassy.mikeimelfort.com

2.3 System Install

Change into the src directory and type:

```
$ make
```

And go get a cup of coffee. This will produce an executable; either Sassy64 or Sassy32, depending on your architecture. Make symlinks from this folder to your bin directory. Also make sure you symlink the 'formatdefs' folder into the same directory as the executable.

2.4 Local Install

This section assumes that you will be installing SaSSY in your home directory which I'll call '$HOME'. You have extracted the code into a working directory called $SASSY. The first thing to do is simply try to build it using the 'make' command. However, this may not work because you may not have all the dependencies installed. Check the error messages which are printed when you try to make, to see which dependencies are missing. I'll assume that you have gcc or a similar compiler installed. Do some or all of the following, and then try to make again.

2.4.1 Install dependencies

First install any missing dependencies. I'll assume that you are going to install dependencies in '$HOME/local/'. If you are going to install them elsewhere, you'll need to modify these code snippets.

NOTE: Do not just type '$HOME'. On my system my home directory is located at '/home/uqmimelf' so I would write '$HOME/local/' as '/home/uqmimelf/local/'.

For all the dependencies EXCEPT String-CamelCase and readline use the following commands:

```
$ ./configure --prefix ~/local
$ make
$ make install
```

When you install readline you MUST link it against ncurses. So change the configure line to:

```
$ ./configure --prefix $HOME/local/ --with-curses
```

When you install String-CamelCase you need to use the 'perl' way of doing things:

```
$ perl Makefile.PL prefix=$HOME/local
$ make && make install
```

2.4.2 Modify Makefile.inc

Now you will need to modify a file called Makefile.inc. It is located in: $SASSY_DIR/src/.makefile_pieces. You need to modify the BASE_CCFLAGS and BASE_LIBS lines located at the top of this file to make SaSSY use the local versions of the dependencies. Originally, these lines should look like:

```
BASE_CCFLAGS := -O3 -Wall -Werror -Wno-unused-function
BASE_LIBS = -lreadline -lncurses -lpcre
```

If you installed any of the dependencies locally, you will need to change the BASE_CCFLAGS to this:

```
BASE_CCFLAGS := -O3 -Wall -Werror -Wno-unused-function -I $HOME/local/include
```

If you installed readline as one of the local dependencies, you will need to change the BASE_CCFLAGS to this:

```
BASE_CCFLAGS := -O3 -Wall -Werror -Wno-unused-function -static -I \
  $HOME/local/include
```

If you installed *any* of the dependencies locally, you will need to change the BASE_LIBS line to something like this:

```
BASE_LIBS  = -Bstatic $HOME/local/lib/libreadline.a -L$HOME/local/lib -lncurses \
  -lpcre
```

2.4.3 Set the PERL5LIB environment variable

If you needed to install String-CamelCase in your home directory you will need to tell the system where to find it. Type the following on the command line before you try to compile SaSSY. Your setup may be slightly different, but this should give you the main idea...

```
$ export PERL5LIB=$HOME/local/lib/perl5/site_perl/5.10.0/
```

3 Running SaSSY

SaSSY typically runs in two stages. First, there is a “load and save” stage where the raw data is loaded, naïve contigs are made and then all the info is saved into binary files. The second is a “re-load and extend” stage where the information is loaded from the binary files and then the naïve (or other) contigs are extended. The following describes a typical work flow.

3.1 Create “FileSet” tokens

If you're impatient:

```
$ SassyXX newFileSet
```

...and follow the prompts...

If that approach fails:

SaSSY loads raw data from a variety of different short read formats. Some of these formats store the read data in one text file, others use two. To avoid any confusion, I'll call each set of files a *FileSet*. Typically, each FileSet will contain paired short reads from a single library, with a given insert size and relative orientation. However, sometimes, for example with Illumina mate-paired data, there appears to be two distinct libraries; one with the intended insert size and relative orientation, and another *shadow library* with an insert size of about 200-400bp and a relative orientation in the opposite orientation. There are probably other good reasons for having multiple libraries in one FileSet but the shadow library is the most common one I've seen. SaSSY has been designed to handle FileSets which contain multiple libraries, you just need to tell it that they're there. This is done using a *FileSet token*.

Basically, a FileSet token wraps the raw data files and stores information about any libraries the FileSet contains. To make a FileSet:

```
$ Sassy64 newFileSet
***Sassy64*** Version: 0.1.0.0

       Copyright (c) 2008, 2009, 2010 Michael Imelfort and Dominic Eales.
       This is free software; see the source for copying conditions. There is NO
       warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

Creating a new file-set...
NOTE: If you get an error saying the format does not match when you are sure it does
then chances are you will need to convert the file from windows style new lines to dos

Please choose the format of the file-set:
1. FASTA-dualfiles (illumina style)
2. FASTA-dualfiles (plain vanilla)
3. FASTA-singlefile (TAGdb output)
4. FASTA-singlefile (illumina style shuffled)
5. FASTA-singlefile (plain vanilla shuffled)
6. FASTQ-dualfiles
7. FASTQ-singlefile
8. Illumina SCARF
9. Unknown file format
```

```
Choice:
```

Now enter a number from 1 – 9, depending on the type of input file you have.

Choice: 2

```
** All files must be in the same directory.
** All files must have the same number of lines.

Enter file 1 (FASTA_1 file):
```

We have chosen "`FASTA-dualfiles (plain vanilla)`" so the program expects two files to be entered, one for each end of the paired read. Enter the path to the first data file. This must be an exact path. You will be asked to enter the path to the second file after this.

```
Enter file 1 (Sequence File 1 file): PATH_TO_FILE_1
Enter file 2 (Sequence File 2 file): PATH_TO_FILE_2

Number of bases to trim from end of read: [0 for none]
```

If you want to do permanent trimming on this data file then do it here. NOTE: you can trim on a per-run basis on the command line when executing SaSSY so it may be better to enter 0 here. Next you will be asked to enter *estimates* for the insert size of the library. SaSSY will use these estimates to calculate more exact figures during runtime.

NOTE: SaSSY expects the insert size to be the distance BETWEEN the centres of the reads. This is different from all other assemblers and is something I will rectify when I find the time. For now you need to remember that if you have reads which have come from a library with a fragment size of 580bp using 100bp paired reads then the estimated insert size will be 480bp. In velvet you would use 580bp. This could/will be rectified in future versions of SaSSY.

```
Number of bases to trim from end of read: [0 for none] 0

Estimated library insert size #1: [0 to stop] 3500
Estimated library insert size #1: [0 to stop]
Orientation code for insert #1
0  <---  --->
1  <---  <---
2  --->  <---
3  --->  --->
: 0
```

```
Estimated library insert size #2: [0 to stop] 250
Estimated library insert size #2: [0 to stop]
Orientation code for insert #2
0  <---  --->
1  <---  <---
2  --->  <---
3  --->  --->
: 2
Estimated library insert size #3: [0 to stop] 0

File-set saved to `READS_DIRECTORY+FILENAME.sassy'
```

Enter the insert sizes and orientation types for all the libraries in this FileSet. Enter 0 to stop. You will find a FileSet token file in the directory which contains the reads. It has local references to the data files so it should stay in that folder. If you need to modify any of the values you entered, it is plain text so this should be straight forward. A typical FileSet token looks like this:

```
name:RANDOM_BAC_1_sequence.txt
format:Illumina SCARF
file_names:RANDOM_BAC_1_sequence.txt,RANDOM_BAC_2_sequence.txt
num_reads:7308118
read_length:50
trim_length:0
insert_lib:M:3500  St:0  O:0  Nr:0
insert_lib:M:250  St:0  O:2  Nr:0
--
```

This is the file which is passed to SaSSY with the '-f' flag in section 2.2

3.2 Load raw data and make naïve contigs

```
$ SassyXX -s -d OUTPUTDIR -f FileSet[,MAX_READS] [-f FileSet[,MAX_READS]]

Compulsory parameters:

 --dir -d WORKING_DIR              The directory to save output including where the
                                   assembled contigs are saved
 --save -s                         Load data from FileSets, do basic processing.
                                   Then save to output directory and exit
 --FileSet -f FileSet[,MAX_READS]  Add sequences described by 'FileSetfile'
                                   'MAX_READS' is maximum paired reads to load
Optional parameters:

 --loglevel -v LEVEL               Set log level. 0 for no logging [default: 0]
 --readlength READLENGTH           Set the effective readlength to this value
                                   (override values in the FileSet tag)
 --naive_off OFFSET                Maximum offset used to make naive contigs
                                          [default: 4]
```

```
--ext_off OFFSET                    Maximum offset used during extension ( this value
                                    cannot be exceeded during the extension step)
                                          [default: 12]
```

… And go make a cup of coffee.

NOTE: SaSSY can take any number of FileSet tokens.

NOTE: The readlength used in the assembly will be equal to the largest odd number which is less than or equal to the shortest readlength of any of the input FileSets. For example: If one FileSet has a readlength of 100bp and another has a readlength of 36bp, SaSSY will perform the assembly with a readlength of 35bp. The last base from reads in the later FileSet and the last 65bp from reads in the the former FileSet will be **DISCARDED**. It is recommended that if you have to use data with readlengths that vary this much then you should cut down the longer reads using the script called 'cutDown.pl' located in the scripts directory. Set –readlength to control this variable.

The output of this command is the saved binary files, and a file called 'out.naive_contigs.fa' which contains the naïve contigs. If a log level was specified, there will also be a file called sassy.log. Logging is not required, but if you like getting a log file, a level of 4 produces a manageable amount of information.

Loading part of a FileSet: If you would only like to load the first 1,000,000 reads of some FileSet, then modify the command line like this:

```
-f FileSet_1
```

to:

```
-f FileSet_1,1000000
```

Naïve and extension offsets:

```
--naive_off OFFSET
--ext_off OFFSET
```

Set the maximum allowable offsets for naïve contigs and extended contigs. The offset is the length of the non-overlapping parts of two overlapped reads. By default, SaSSY sets naïve offset to 4 and extension offset to 12. Thus, during naïve contig creation, two reads will be joined together if they are offset from each other by no more than 4 bases. During contig growing, this is extended to up to 12 bases. The greater the cut off, the more edges will be made between the reads (and hence more memory will be used).

You may override these parameters with any EVEN value between 2 and

16 inclusive. For example, if you were trying to assemble a data set with ridiculously high coverage, it may help to set these values to 2bp and 6bp respectively, which will reduce the number of edges made and hence the amount of memory used - hopefully without sacrificing the connectivity of the graph. If you had a very low coverage dataset, you may want to set them to to 8bp and 12bp to try induce extra edges in the naïve contig graph.

3.3 Extend naïve contigs / other sequences

```
    $ SassyXX -l -d OUTPUTDIR

 Compulsory parameters:

  --dir -d WORKING_DIR              The directory where saved data is stored and
                                    where the assembled contigs are saved
  --load -l                         Load the data from saved state

Optional parameters:

  --loglevel -v LEVEL               Set log level. 0 for no logging [default: 0]
  --no_strict                       Relax the growing algorithm, longer contigs
                                    but causes more over assemblies / chimeras
                                          [default: STRICT = true]
  --no_olap                         Skip overlaperation
                                          [default: OLAP = true]
  --upper_coverage_percent PERCENT  Set the maximum coverage allowable for a
                                    naive Contig to be used in stretching
                                          [default: true (use all contigs)]
  --ext_off OFFSET                  Maximum offset used during extension
                                          [default: 12]
  --olap_min_size MIN_SIZE          The minimum number of bases two Contexts
                                    must overlap [default: highest upper cut off]
  --olap_max_wastage MAX_WASTAGE    The maximum number of bases which can be
                                    wasted during an overlaperation event
                                          [default: 2 * read length]
  --scaff_link_cut_off NUM_LINKS    Minimum number of links needed to scaffold
                                    0 to skip scaffolding [default: 5]
  --scaff_insert_cut_off CUT_OFF    For scaffolding, use only reads with insert
                                    greater than CUT_OFF [default: 0]
  --ignore_trap TRAP_INSERT         When using a small supplementary insert size
                                    to clean the data, ignore this in extensions.
                                    0 to use all InsertLibs. [default: Ignore = 0]
  --graph                           Output the scaffold graph to stdout in
                                     format [default: no graph]
  --csv                             Output the information used in scaffolding
                                    to stdout in CSV format [default: no csv]
```

This should / will run significantly faster than the load & save run. This stage MUST be able to find a multiple fasta file, called out.naive_contigs.fa, in the working directory. It will output three files; out.advanced_contigs.fa, out.final_contigs.fa and out.scaffolds.fa. The first is the extended contigs before

final overlapping, the second contains the same contigs after overlapping and the final one contains the overlapped contigs after scaffolding. If a log level was specified, there will also be a file called sassy.log

Using “alien” seeds: This stage MUST be able to find a multiple fasta file, called out.naive_contigs.fa in the working directory. BUT this doesn't mean that this file has to contain contigs from SaSSY. You *could* use any fasta file here...

Extension options: These options are used during the extension phase of the algorithm.

`--no_strict`

By default, the extension phase runs in a “strict” mode. This means that during the growing phase it will try to extend the input contigs as best it can but it will not try to “push” things too far, as this can create chimeric contigs. If you have a low coverage data set then it may help to turn this feature off. To do this use the '--no_strict' flag

NOTE: This will nearly always let you produce longer contigs, you just need to spend more time after assembly to make sure that they are correct.

`--upper_coverage_percent PERCENT`

SaSSY uses the naïve contigs generated in the first stage to “stretch” extended contigs. For example, when an extension fails it may be that SaSSY has extended the seed contig and is part way through another long naïve contig. Because SaSSY “trusts” naïve contigs, especially longer ones, it will stretch the extension to the end of that contig and then try to continue extending. This can cause problems if SaSSY tries to stretch across a highly repetitive region. Setting this parameter will force SaSSY to use contigs for stretching only if they are within PERCENT percent of the mean coverage of the ten longest naïve contigs. This option will not work if you are using alien seeds (see above).

`--ignore_trap -I TRAP_INSERT`

Illumina paired-end data sets typically have the insert size distribution shown in Figure 1 below.

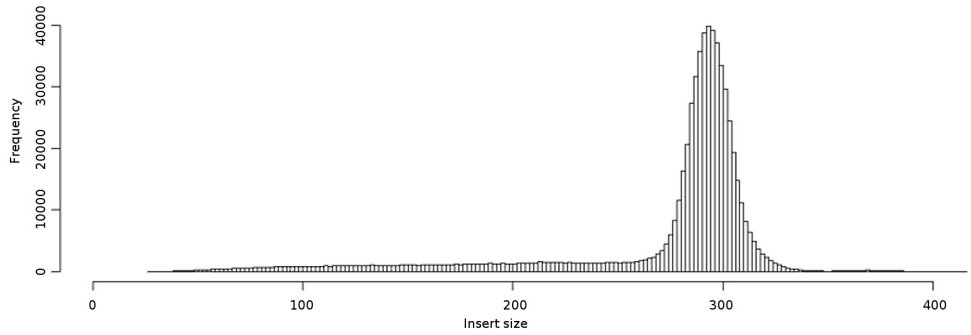

Figure 1. Typical distribution of inserts for Illumina Paired-end data.

Typically, there will be a lot of rubbish on the left hand side of the mean. Because SaSSY relies on insert size stability as opposed to coverage stability this can cause problems. To reduce the effect that this has on the assembly you can set a “trap” insert when defining the FileSet. For example, in the image above you would tell SaSSY that this data file has two insert sizes; one with a mean of approximately 300bp and one with a mean of approximately 50bp. The result is that many of the rubbish reads get separated from the “real” reads into their own little space, and the standard deviation about the upper mean gets a lot tighter. You can tell SaSSY to ignore these reads during the extension process by setting the ignore flag. For the diagram above, it may improve the assembly to ignore all reads with an insert size less than say 220bp.

NOTE: This is a global setting. Thus, for the example just given, if you had a “real” library with an insert size of less than 220bp, most of its reads would be ignored.

```
--ext_off OFFSET
```

You can override the value set for ext_off during naïve contig creation, however, it must be less than or equal to the value used for ext_off during the first stage.

Overlaperation options: These options concern the overlaperation stage which follows extension.

```
--no_olap -o
```

Overlaperation can sometimes be a little naïve. If your Contigs are looking very chimeric it may be useful to turn this feature off. During the scaffolding stage, the advanced contigs will be used instead of the final ones.

```
--olap_min_size MIN_SIZE
--olap_max_wastage MAX_WASTAGE
```

Overlaperation occurs when two contigs to have a **perfect overlap** (after end trimming) at least as long as the olap_min_size variable. By default this value is based on the largest insert size. If you are lucky enough to have a

data set with a number of different insert sizes, say 200bp, 2000bp and 10000bp you may want to set this parameter to something slightly larger than 2000bp so you can make use of these reads during overlaperation. Otherwise SaSSY will require two extended contigs to share a perfect overlap of about 10000bp for overlaperation to occur. Use olap_max_wastage to override the maximum number of bases which can be trimmed from the ends of the contigs to make a perfect overlap.

Scaffolding options: These options concern the scaffolding stage which is the final stage of the algorithm.

`--scaff -S NUM_LINKS`

Minimum number of links needed to make scaffold. By default this is set to 5. During scaffolding there needs to be a minimum of 5 paired links between two contigs for there to be any chance of a scaffold occurring. For lower or higher coverage datasets you may wish to lower or raise this amount, or let NUM_LINKS be 0 to skip scaffolding altogether.

`--scaff_insert_cut_off CUT_OFF`

Perhaps you don't want to reads from a 200bp insert size library to make scaffolds. You can control the minimum insert size used for scaffolding here.

Output options: Generating more output...

`--graph -g`

Output the scaffolds as a GraphViz formatted graph to STDOUT. Can be useful for seeing how the final contigs were formed into scaffolds. Using this flag will output two graphs. One containing all the possible ways SaSSY could have built the scaffolds and one containing the only the configuration which SaSSY used to create the final 'out.scaffolds' file.

`--csv`

Output all the links which could be used to make scaffolds in csv format to STDOUT. **This may generate a serious amount of output.** This will give every link between every contig which SaSSY identified during the scaffolding stage. It may be useful to examine this output if you feel that SaSSY made some bad scaffolding choices.

The fields of the csv file are given below:

Column	Description
Base	The first contig used in the link.
Linker	The second contig used in this link.
IsSameDirn	Is the Linker facing in the same direction as the Base?
IsBefore	Does the Linker lie before or after the Base?

FileId	Identifies which file this linking pair is from.
BasePos	The position of the read in the Base contig.
BaseDist	The Distance from the read to the end which closest to the Linker.
LinkerPos	The position of the read in the Base contig.
LinkerDist	The Distance from the read to the end which closest to the Base.
PairKey	A sorting key which is made from the IDs of the two contigs.
TypeKey	A sorting key which is made from the PairKey and the relative directionality of the Linker with respect to the base.

Table 2. Description of the scaffold csv columns.

NOTES:

1. The ID of the Base is always numerically lower than the ID of the Linker.
2. All directions are calculated relative to the directions that the contigs have been written in the file 'out.final_contigs.fa' (or out.advan_contigs.fa if –no_olap is set).
3. All distances are calculated relative to the centre of the reads. Thus, the first read in a contig lies at position 0 and the last read lies at position L – RL where L is the length of the contig and RL is the read length.

4 GNU Free Documentation License

GNU Free Documentation License
Version 1.3, 3 November 2008

```
 Copyright (C) 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.
     <http://fsf.org/>
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other
functional and useful document "free" in the sense of freedom: to
assure everyone the effective freedom to copy and redistribute it,
with or without modifying it, either commercially or noncommercially.
Secondarily, this License preserves for the author and publisher a way
to get credit for their work, while not being considered responsible
for modifications made by others.
```

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the

general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough

number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section

of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
D. Preserve all the copyright notices of the Document.
E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
H. Include an unaltered copy of this License.
I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the

violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation 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. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

> Copyright (c) YEAR YOUR NAME.
> Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
> A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

> with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.