ProDuSe Documentation

Release 0.9.1

Christopher Rushton

Mar 24, 2018

Contents

Authors

Christopher Rushton
Marco Albuquerque

Contact ckrushto@sfu.ca.

Source Code Github

ProDuSe is a variant caller designed for use with libraries prepared and sequenced using semi-degenerate barcoded adapters. These barcodes allow PCR and optical duplicates which were derived from the same starting molecule to be flagged and merged into a single consensus sequence. The addition of a strand-specific tag also allows molecules which are derived from different parental strands to be flagged, allowing variants at extremely low frequency (VAF~=0.1%) to be called confidently.

CHAPTER 1

Quick Links

1.1 Quick Start

1.1.1 Check and install dependencies

To run ProDuSe successfully, ensure the following non-python dependencies are installed on your system:

- Samtools (>=1.3.1)
- Burrows-Wheeler Aligner (>=0.7.0)
- Python (>=3.4)

1.1.2 (Optional) Install ProDuSe

To run ProDuSe as a command line command, install ProDuSe as follows:

From the command line:

```
cd /path/to/ProDuSe/clone/
python setup.py install
```

This will automatically install all python dependencies.

1.1.3 Setup ProDuSe configuration

To run ProDuSe, you will need the following:

1. Sequence (fastq) files from your sample of interest 3. ProDuse configuration file

If you are unsure of the barcode sequence, it can be estimated using adapter_predict

To start, modify the ProDuSe configuration file with your sample's information:

```
[Pipeline]
barcode_sequence=NNNWSMRWSYWKMWWT
barcode_position=0001111111111110
max_mismatch=3
family_mask=0001111111111110
family_mismatch=3
duplex_mask=0000000001111110
duplex_mismatch=2
reference=test/reference/GRCh38_TNFSF14.fa
sample_config=etc/sample_config.ini
filter=etc/default_filter.p
```

See run_produse for a detailed description of each argument

1.1.4 Run ProDuSe

Finally, run ProDuSe with the specified information.

If you installed ProDuSe:

```
produse run_produse -c /path/to/produse/config/file --fastqs /path/to/fastq.R1 /path/
↪to/fastq.R2
```

If you did not install ProDuSe:

```
/path/to/ProDuSe/ProDuSe/ProdusePipleine.py -c /path/to/produse/config/file --fastqs /
↪path/to/fastq.R1 /path/to/fastq.R2
```

All results will be placed in the current working directory (this can be changed using *-d*), under *produse_analysis_directory*.

1.1.5 Running multiple samples at once

ProDuSe can process multiple samples sequentially with the same command. You will need a *sample configuration* file to use this feature.

Specify each sample in this configuration file:

```
[Sample1]
fastqs=/path/to/Sample1/fastq.R1,/path/to/Sample1/fastq.R2

[Sample2]
fastqs=/path/to/Sample2/fastq.R1,/path/to/Sample2/fastq.R2
```

If you would like to specify different parameters for each sample, they can be specified in the sample_config file:

```
[Sample1]
fastqs=/path/to/Sample1/fastq.R1,/path/to/Sample1/fastq.R2
barcode_sequence=NNNWSMRWSYWKMWW
barcode_position=000111111111111

[Sample2]
fastqs=/path/to/Sample2/fastq.R1,/path/to/Sample2/fastq.R2
max_mismatch=2
```

Arguments specified in the sample config file will only be used for that sample

Note: Parameters specified in the sample configuration file take precidence for that sample

When running ProDuSe, this sample configuration file will be specified instead of fastq files.

If ProDuSe is installed:

```
produse run_produse -c /path/to/produse/config/file *-sc /path/to/sample/
↪configuration/file*
```

If ProDuSe is not installed:

```
/path/to/ProDuSe/ProDuSe/ProdusePipleine.py -c /path/to/produse/config/file *-sc /
↪path/to/sample/configuration/file*
```

All results will be outputted in individual sample directories under 'produse_analysis_directory'.

1.2 The ProDuSe Pipeline

The ProDuSe pipeline consists of multiple stages, each of which is described breifly here.

1.2.1 Trim

Trims the barcode sequence off each read, and stores it in a FASTQ comment. Any reads where the barcode deviates significantly from the expected degenerate range are discarded

- Input: Paired fastq files
- Output: Trimmed paired fastq files

Note: Trim can be used to demultiplex samples, assuming the barcodes used in each sample are sufficiently distinct

See also:

Trim

1.2.2 Mapping

Maps provided reads to a reference genome using the Burrows-Wheeler Aligner (mem algorithm). The resulting SAM file is converted into a BAM file and sorted, with the FASTQ comment stored as a read tag.

- Command: bwa mem <reference> <trimmed_fastq.R1.fastq> <trimmed_fastq.R2.fastq> | samtools view -b | samtools sort > out.trim.bam

1.2.3 Collapse

Collapses duplicate reads into a consensus sequence. In addition, reads which are in "duplex" (i.e. originate from the same parental molecule) are flagged here

- Input: Trimmed BAM file

- Output: Collapsed BAM File

See also:

Collapse

1.2.4 ClipOverlap

Idenfies bases that overlap between each read pair, and generates a consensus from the overlap. This consensus is then assigned to only one read in the read pair, thus removing overlapping bases.

- Input: Collapsed BAM file
- Output: Clipped BAM file

See also:

ClipOverlap

1.2.5 Call

Identifies all possible variants in the specified file, then filters those variants based upon capture-space and locus-specific characteristics.

- Input: Clipped BAM file
- Output: Two VCF files, listing raw (all) and filtered variants

See also:

Call

CHAPTER 2

Main Commands

2.1 Run ProDuSe

2.1.1 Purpose

Wrapper script which runs each step of the ProDuSe Pipeline sequentially on each sample. Can process multiple samples simultaneously.

2.1.2 Run Using

```
produse run_produse
```

or

```
/path/to/ProDuSe/ProdusePipeline.py
```

2.1.3 Parameters

General Parameters for running analysis

-c, –config An INI configuration file, which can specify any of the following arguments. Any arguments passed at the command line override those specified in the configuration file, which override defaults.

–fastqs A pair of filepaths to a set of paired-end FASTQ files. Mutually exclusive with -sc–sample_config.

-sc, –sample_config An INI configuration file, specifying one or more samples to analyze. This file can also provide sample-specific arguments.

-d, –outdir Base output directory for intermediate files and results. –directory_name will be created inside this directory. Default is the current working directory.

-r, –reference A filepath to a reference genome FASTA file. Ideally, BWA and .fai indexes should be present in the same directory. If they are not, they will be generated automatically.

-j, –jobs Number of samples to analyze in parallel. Use 0 or a negative number to process as many samples as possible simultaneously. Default is 1.

Additional Analysis Parameters

–bwa A filepath to a bwa executable. Default is $(which bwa).

–samtools A filepath to a samtools executable. Default is $(which samtools)

–directory_name Name of the directory to create inside -d/–outdir to store intermediate files and results. Default is "produse_analysis_directory".

–append_to_directory If –directory_name already exists inside -d/–outdir, place the intermediate files and results for this analysis inside this directory. If any samples have the same name as those inside –directory_name, they will not be analyzed.

–cleanup Following analysis, remove all files present in the "tmp" directory of each sample

Barcode Trimming Parameters

-b, –barcode_sequence The sequence of the degenerate barcode, specified in IUPAC bases.

-p, –barcode_position Positions in the -b/–barcode sequence to use when comparing expected and actual barcode sequences (1=Yes, 0=No). The entire barcode will still be trimmed, regardless of which positions are flagged with 0.

-mm, –max_mismatch Maximum number of positions (specified using -p/–barcode_position) which can fall outside the expected degeneracy range before the read pair is discarded. This is the total between both the forward and reverse read.

–trim_other_end Examine the end of the read for the presence of a barcode (for example, in the case of read-through). Will not remove partial barcodes

Family Collapsing Parameters

-fm, –family_mask Barcode positions to consider when determining if two reads are members of the same family (1=Yes, 0=No).

-fmm, –family_mismatch Maximum number of mismatched positions (specified using -fm/–family_mask) permitted before two reads are considered members of different families.

-dm, –duplex_mask Barcode positions to consider when determining if two families are in duplex (1=Yes, 0=No).

-dmm, –duplex_max_mismatch Maximum number of mismatched positions (specified using -dm/–duplex_mask) permitted before two families are not considered a duplex

-t, –targets A filepath to a BED file listing the capture regions of interest. Read pairs that do not overlap these positions will be discarded

–tag_family_members Store the name of each read incorporated into a family in the read tag "Zm"

Filtering Parameters

-f, –filter A filepath to a pickled Random Forest Classifier, used to filter variants. Can be generated using train

Note: If no -b/–barcode is specified for one or more samples, adapter_predict will be run on those samples automatically

2.1.4 Pipeline Stages

This command will run the following stages of the ProDuSe Pipeline sequentially on each sample:

- adapter_predict (Estimates the degenerate barcode sequence, only run if no barcode is specified)
- Trim (Trim Barcodes)
- Align reads to reference (Burrows-Wheeler Aligner)
- Collapse (Identify and merge duplicate reads into a consensus)
- ClipOverlap (Identifies positions which overlap between read pairs, and generates a consensus)
- Call (Flag and filter variants)

2.1.5 Configuration Files

In lieu of specifying all arguments at the command line, arguments can be specified in **either** the sample configuration file, or the main produse configuration file. This approach is recommended, as it improves reproducibility between runs.

If a single argument is specified multiple times, the following orders of precedence apply:

1. -sc/–sample_config file
2. Command line
3. -c/–config file

Arguments specified in the -sc/–sample_config file only apply to the specified sample. More information on configuration files can be found here.

2.1.6 Directory Layout

When run_produse is called, it creates a directory structure for each sample inside of –directory_name, as follows:

```
directory_name
        Sample_Name
                config
                tmp
                results
        ProDuSe_Task.log
```

The contents of each folder are as follows:

config Stores configuration files for each step of the pipeline, and files indicating when a pipeline stage is complete

tmp Stores intermediate files (ex. Trimmed FASTQ files, raw BAM files)

results Stores the final BAM file and variant calls

All parameters used to run a given instance, as well as software versions, are specified in ProDuSe_Task.log

2.2 Adapter Predict

Predicts the barcoded adapter sequence range used in a given file.

2.2.1 Run Using

```
produse adapter_predict
```

or

```
python /path/to/ProDuSe/ProDuSe/adapter_predict.py
```

2.2.2 Parameters

-i –input Paired fastq files. Two files must be specified.

-m –max_adapter_length Limit the adapter sequence prediction to this length

2.2.3 Additional Considerations

Warning: Ensure your samples are de-multiplexed prior to running Adapter Predict. The adapter sequence is predicted from ALL reads in the fastq files.

2.3 Trim

2.3.1 Purpose

Trims the barcode sequence from each read, and stores the barcode in a FASTQ tag Reads where the barcode does not fall within specified barcode range and mismatch threshold are discarded.

2.3.2 Run Using

```
produse trim
```

or

```
python /path/to/ProDuSe/ProDuSe/Trim.py
```

2.3.3 Parameters

-c, –config A configuration file which can provide any of the following arguments. See the config page for more details.

-i –input Two FASTQ files to be trimmed. These files may be gzipped.

-o –output

Two output FASTQ files

These files can automatically be gzipped by appending '.gz' to the output file name.

The output file order corresponds with the input file order (i.e -i read1.fastq read2.fastq -> -o read1.out.fastq read2.out.fastq)

-b –barcode_sequence The barcode sequence range, described using IUPAC bases. Can be determined using adapter_predict

-p –barcode_position

The positions in the adapter sequence to consider when comparing reference (i.e. -b) and actual adapter sequences

0 = Do not consider this position

1 = Consider this position

Note that the entire barcode sequence will be trimmed, even if a position is labeled 0.

–mm –max_mismatch The maximum number of mismatches allowed between the reference and actual adapter sequences before a read pair is discarded

—reverse

Instead of discarding reads with mismatching barcode sequences, do the opposite (save mismatching reads, discard all others).

Useful for debugging.

–no_trim Do not trim the adapter sequence. Only store the adapter sequence a read tag.

–trim_other_end

Examine the other end of the read for barcode sequences as well.

Will not remove partial barcodes

Useful when read lengths are significantly longer than the median fragment length

2.3.4 Helpful Tips

If the read discard rate is extremely high, check the supplied barcode sequence.

2.3.5 Additional Notes

If the supplied fastqs are multiplexed, a single sample can be extracted if no other samples use barcoded adapters, or if the barcoded adapter sequences between are distinct enough (i.e. the difference between the two barcode sequences exceeds the maximum mismatch threshold). Note that there may be some spillover if non-barcoded reads start with a sequence that falls within the barcode sequence range by chance, or if the differences between barcoded sequences is only slightly higher than the maximum mismatch threshold.

2.4 Collapse

Identifies the start position, barcode sequence, and mapping strand for each read pair in the supplied BAM file. If both reads in one or more reads share the same start position, mapping strand, and barcode sequence (within mismatch tolerance), they are flagged as a "family", and merged into a single consensus sequence. If family members disagree at a given position, the most common base is used as a consensus. In the case of a tie, the base with the highest aggregated quality score across all family members is used. The quality of each base set to the highest quality base at that position.

2.4.1 Run Using

```
produse collapse
```

or

```
python /path/to/ProDuSe/ProDuSe/Collapse.py
```

2.4.2 Parameters

-c –config A configuration file which can supply any of the arguments below. See the config page for more details.

-i –input Input SAM/CRAM/BAM file. Each read must contain a read tag which stores adapter sequences

-o –output Output SAM/CRAM/BAM file containing collapsed reads (Use "-" for stdout). The output file format will be chosen based upon the supplied file extension, or if "-" is used, will be the same as the input file format. Will be unsorted.

-fm –family_mask

Positions in the barcode sequence to use when comparing barcode sequences between reads which originate from the same parental strand.

1=Use this position, 0=Do not use this position.

-dm –duplex_mask

The positions in the adapter sequence to use when comparing adapter sequences for reads of opposing types (i.e. forward vs reverse reads).

1=Use this position, 0=Do not use this position.

-fmm –family_max_mismatch The maximum number of mismatches allowed between the barcode sequence of two read pairs before two read pairs are considered members of different families (See -fm).

-dmm –duplex_max_mismatch The maximum number of mismatches allowed between the barcode sequence of two families before they are considered as not in duplex (See -dm).

-r –reference Reference genome, in FASTA format. Must be the same genome version that the reads were aligned against.

-t –targets A BED3 file or better listing regions of interest. Any read pairs which fall entirely outside these regions will be discarded

–tag_family_members Store the original name of all reads which were incorporated into a family in the read tag "Zm"

2.4.3 Additional Considerations

The runtime of Collapse depends not only on the absolute number of reads, but the proportion of reads which are duplicates. BAM files with high duplicate rates will take significantly longer than BAM files with a lower duplicate rate.

Currently, this version of Collapse does not perform local realignment of soft-clipped regions.

2.5 ClipOverlap

2.5.1 Description

Identifies all positions which overlap inside of a given read pair. If any bases overlap, a consensus is generated and that consensus is assigned to only one read in the read pair.

2.5.2 Run Using

```
produse clip
```

or

```
/path/to/ProDuSe/Clone/ProDuSe/ClipOverlap.py
```

2.5.3 Parameters

-c –config A configuration file which can provide any of the following arguments. See the config page for more info.

-i –input An input SAM/BAM file (use "-" to read from standard in). Does not need to be sorted.

-o –output An output SAM/BAM file in which to write clipped reads (use "-" to write to stdout). Will be UNSORTED. The file type is determined from the file extension, or the input file type if stdout is specified.

–tag_origin Add a read tag indicating which read a consensus base originated. S=Both reads agree.

Warning: The output BAM file will be unsorted, even if the input BAM file is sorted.

2.5.4 Helpful Tips

The output BAM file can be sorted easily by pipeline the output to "samtools sort". For example:

```
produse clip -i inFile.bam -o - | samtools sort > outFile.bam
```

2.5.5 Additional Info

Any overlap between read pairs is determined solely using the alignments in the BAM file: No realignment is performed. If the bases at a position disagrees between a read pair, the base with the highest mapping quality is used. In the case of a tie, the base from the read which starts later is used.

When obtaining a consensus between INDELs, the read with the lowest number of INDELs in the overlapping region is used as the consensus. No realignment is performed.

The consensus overlap is assigned to either read 1 or read 2. For the other read, the positions coresponding to the consensus are soft clipped.

Warning: Overlap between read pairs is removed via soft-clipping. This may cause problems for some structural variant callers which examine soft-clipped bases

2.6 Call

Identifies all possible positions which support an alternate allele (no matter how weakly) across the entire capture space. These variants are then filtered using a random forest filter, based upon numerous characteristics

2.6.1 Run Using

```
produse call
```

or

```
python /path/to/ProDuSe/ProDuSe/Call.py
```

2.6.2 Parameters

-c –config An optional configuration file which can specify any of the arguments described below

-i –input An input BAM containing collapsed (and ideally clipped) reads. This file must be sorted by position.

-o –output Output VCF file containing filtered variants

-u, –unfilted Output VCF file containing ALL possible variants

-r –reference Reference genome, in FASTA format. A reference index should be present in the same directory

-t –target_bed A BED3 file specifying a capture space to restrict variant calling

-f, –filter A piclke containing a tranined Random Forest Classifier

CHAPTER 3

Additional Commands

3.1 Resume ProDuSe

3.1.1 Description

Restarts a previously terminated instance of run_produse

3.1.2 Run Using

```
produse resume_produse
```

or

```
/path/to/ProDuSe/Clone/ProDuSe/ResumePipeline.py
```

3.1.3 Parameters

-d –produse_dir Path to the base ProDuSe analysis directory (usually named produse_analysis_directory)

-j –jobs Number of samples to process in parallel

3.1.4 Additional Information

run_produse automatically generates <task>_Complete file when each pipeline component is completed for each sample. These files are placed in the "config" directory of the coresponding sample. This script identifies samples in which not all <task>_Complete files have been generated, and resumes the analysis from there.

Note: If you wish to re-run a stage of the pipeline, simply remove the coresponding <task>_Complete file

3.2 Update Config

3.2.1 Description

Coverts an older produse config file to a format compatible with this version of ProDuSe

3.2.2 Run Using

```
produse update_config
```

or

```
python /path/to/ProDuSe/ProDuSe/UpdateConfig.py
```

3.2.3 Parameters

-i/–input The configuration file to be reformatted

-o/–output Path to the new, reformatted configuration file

3.2.4 Additional Info

Duplicate parameters are not permitted in config files. If two parameters specify the same argument, one will be removed automatically. If they specify different arguments, they will both be retained. One of the arguments must be removed manually.

Any parameters with no arguments are removed automatically

3.3 Train

3.3.1 Description

Trains ProDuSe's variant calling filter using one or more sets of validated variants. A random forest classifier will be trained using this data

3.3.2 Run Using

```
produse train
```

or

```
/path/to/ProDuSe/Clone/ProDuSe/Train.py
```

3.3.3 Parameters

-c –config A configuration file which can provide any of the following arguments. See the config page for more details.

-b –bam One or more post-clipoverlap BAM files, which were used for variant filtering and validations. Must be specified in an order that coresponds to the order specified by -v/–validations

-v –validations One or more VCF files listing validated variants. The number and order of file must corespond to files specified in -b/–bam.

-o –output Output file which will store the random forest classifier

-r –reference Reference genome, in FASTA format. An index should also be present in the same directory.

-t –targets Optional. One or more BED files specifying regions in which to restrict candidate variant identification. Must be specified in an order coresponding to the order specified in -b/–bam.

3.3.4 Additional Information

For each sample provided, ProDuSe will identify all candidate variants in a manner identical to Call. If a candidate variant is flagged in the validation VCF file with VALIDATED=TRUE, it will be considered a true variant, while variants flagged with VALIDATED=UNKNOWN will be ignored completely. All remaining variants will be flaged as artifacts. False variants will be randomly subset so there are the same number of false and true variants prior to training the variant filter.

CHAPTER 4

Additional Links

4.1 How ProDuSe Works

4.1.1 Summary

Coming Soon.

4.1.2 Additional Links

Error Supression Using Degenerate Barcodes

4.2 Configuration Files

Most ProDuSe commands allow arguments to be specified using a configuration INI file. In most cases, this is a much more convenient method of specifying arguments if ProDuSe is to be run multiple times. While you can view more detailed info on config files here, a brief overview will be described here.

4.2.1 Standard Config Files

All config files provided to produse commands (using -c/–config) use the following format:

```
[Script_Name]
argument=parameter
argument2=parameter2
```

Where Script_Name is the name of the tool which will use those arguments, with the arguments listed in the section below. Short form argument names are not supported.

Note that config file sections and arguments not relevant to a given tool will safely be ignored, so it is possible to merge config files:

```
[Trim]
barcode_sequence=NNNWTWYYT
max_mismatch=3

[Pipeline]
reference=genome.fa
```

A demo sample configuration file can be obtained here.

Note: Argument provided at the command line will override arguments specified in the config file

4.2.2 Sample Configuration Files

When running multiple samples using run_produse, a special sample configuration file is specified instead:

```
[Sample_Name]
fastqs=/some/path/read.R1.fastq.gz,/some/path/read.R2.fastq.gz

[Sample_Name_2]
fastqs=/some/other/path/read.R1.fastq.gz,/some/other/path/read.R2.fastq.gz
```

Where Sample_Name is the name used for that sample (this is prepended to all intermediate and output files). Any parameters supported by run_produse can be specified, and will be used for that sample **only**. For instance:

```
[Sample_Name_1]
fastqs=/some/path/read.R1.fastq.gz,/some/path/read.R2.fastq.gz
reference=GRCh37.fa

[Sample_Name_2]
fastqs=/some/other/path/read.R1.fastq.gz,/some/other/path/read.R2.fastq.gz
reference=GRCh38.fa
```

GRCh37 will be used as the reference genome for Sample 1, while GRCh38 will be used for sample 2. Click here to view a demo sample config file

Note: Argument specified in config superseded those specified at either the command line, or by normal config files

4.3 License

TERMS AND CONDITIONS

0. Definitions.

“This License” refers to version 3 of the GNU General Public License.

“Copyright” also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

“The Program” refers to any copyrightable work licensed under this License. Each licensee is addressed as “you”. “Licensees” and “recipients” may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or

running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

> a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
>
> b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".
>
> c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.
>
> d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

> a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

> a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
>
> b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
>
> c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
>
> d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
>
> e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
>
> f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

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, you do not

qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If 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 convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU 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 Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "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 PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM 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 PROGRAM (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 PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS