ProDuSe Documentation

Release 0.2.3

Christopher Rushton

Jul 21, 2017

Contents

1 Useful Links 3

2 ProDuSe Commands 9

3 Other Commands 17

4 Additional Links 19

Authors

Marco Albuquerque
Christopher Rushton

Contact ckrushto@sfu.ca.

Source Code Github

ProDuSe is a Python command-line variant caller designed for use on Illumina sequencing data obtained from libraries constructing using barcoded adapters. Using these adapters, read duplicates are identified and collapsed into a single consensus sequence, correcting PCR and sequencing errors. In samples with a high duplication rate, this allows extremely rare variants be be called confidently.

CHAPTER 1

Useful Links

Quick Start

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 (>=0.7.12)
- Python (>=2.7)
- Stitcher (>=5.1.3)

(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.

Setup ProDuSe configuration

To run ProDuSe, you will need the following:

1. Sequence (fastq) files from your sample of interest
2. The barcode sequence of your sample
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:

```
[snv]
min_molecules = 2
mutant_molecules = 3
variant_allele_fraction_threshold = 0.01
min_reads_per_uid = 1

[adapter_qc]
adapter_sequence = NNNWSMRWSYWKMWWT
adapter_position = 0001111111111110
max_mismatch = 3

[collapse_bwa]

[collapse]
strand_position = 0001111111111110
adapter_max_mismatch = 3
duplex_position = 0000000001111110
duplex_max_mismatch = 2

[trim_bwa]

[trim]
adapter_sequence = NNNWSMRWSYWKMWWT
adapter_position = 0001111111111110
max_mismatch = 3

[filter_produse]
```

See the help page of each script if you are unsure what a specific parameter does.

Run ProDuSe

Finally, run ProDuSe with the specified information.

If you installed ProDuSe:

```
produse run_produse -x /path/to/Stitcher.exe -r /path/to/reference/genome/build -c /
↪path/to/produse/config/file -f /path/to/fastq.R1 /path/to/fastq.R2
```

If you did not install ProDuSe:

```
/path/to/ProDuSe/ProDuSe/ProdusePipleine.py -x /path/to/Stitcher.exe -r /path/to/
↪reference/genome/build -c /path/to/produse/config/file -f /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*.

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
```

Note: There is no space after the comma

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
adapter_sequence=NNNWSMRWSYWKMWW
adapter_position=000111111111111

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

These parameters will be used for that sample alone.

Note: Parameters specified in the sample configuration file override ProDuSe configuration file parameters for that sample, while ProDuSe configuration file parameters override command line parameters for ALL samples

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

If ProDuSe is installed:

```
produse run_produse -x /path/to/Stitcher.exe -r /path/to/reference/genome/build -c /
↪path/to/produse/config/file *-sc /path/to/sample/configuration/file*
```

If ProDuSe is not installed:

```
/path/to/ProDuSe/ProDuSe/ProdusePipleine.py -x /path/to/Stitcher.exe -r /path/to/
↪reference/genome/build -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'.

The ProDuSe Pipeline

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

Config

Sets up output directories, file symbolic links, and configuration files for each step in the ProDuSe pipeline

See also:

configure_produse.py

Trim

Trims the adapter sequence off of each read, and prepends the adapter sequence to the read name. In addition, any reads which do not match the barcode sequence (and are outside the mismatch range) are discarded

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

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

See also:

trim.py

bwa

Maps provided reads to a reference genome using the Burrows-Wheeler Aligner.

- Input: Trimmed paired fastq files
- Output: BAM file consisting of trimmed reads

See also:

bwa.py

Collapse

Collapses duplicate reads into a consensus sequence.

- Input: Trimmed BAM file
- Output: Paired collapsed (consensus) fastq files

See also:

collapse.py

bwa

Maps provided reads to a reference genome using the Burrows-Wheeler Aligner.

- Input: Collapsed paired fastq files
- Output: BAM file consisting of Collapsed reads

See also:

bwa.py

Stitcher

Merges forward and reverse reads into a consensus sequence if they overlap. Used to correct errors in overlapping bases.

- Input: Collapsed BAM file
- Output: Stitched (forward and reverse reads are merged) BAM file

See also:

Stitcher Documentation

SplitMerge

Splits reads merged by Stitcher back into forward and reverse reads. The shared bases are assigned to one of the two reads.

- Input: Stitched BAM file
- Output: BAM file with merged reads split. Unsorted

See also:

SplitMerge.py

SNV

Identifies positions whereby one or more bases support an alternate allele.

- Input: A BAM file containing de-stitched reads
- Output: A VCF file listing all positions with non-reference bases, and the count of these bases

See also:

snv.py

Filter

Filters candidate variant calls based upon overall capture space and locus characteristics.

- Input: A VCF file listing candidate variants
- Output: A VCF file listing filtered variants

See also:

filter_produse.py

CHAPTER 2

ProDuSe Commands

Config

Configures output directories, input and output file paths for each step of the ProDuSe pipeline, and configuration files for each script.

produse_config.py performs the following tasks:

- Creates a main output directory (produse_analysis_direcetory) in the specified directory
- Creates a log file listing the supplied arguments in the main output directory
- Creates a Makefile in the main output directory (Depreciated, has no purpose)
- (If necessary) Creates normal and bwa indexes for the designated reference genome

For each sample supplied:

- Creates a sample-specific directory and subdirectories in the main output directory using the sample's name
- Symbolically Links input files into this sample-specific directory
- Creates a config file for each step in the ProDuSe pipeline, containing sample-specific parameters

Run Using

```
produse configure_produse
```

or

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

Parameters

-f –fastqs

If running a single sample, the fastq files for the sample to be analyzed.

Two arguments must be supplied.

A single sample directory names 'Sample' will be created.

Mutually exclusive with -sc –sample_config.

-sc –sample_config

A configuration file listing sample names, fastq locations, and sample-specific parameters.

A directory will be created for each sample, using the name supplied.

Mutually exclusive with -f –fastqs.

-d –output_directory The directory to output ProDuSe results. Default is the current working directory.

-r –reference Reference genome build, in fasta format.

-c –config A configuration file listing parameters to be supplied to each stage of the ProDuSe pipeline.

Additional Considerations

While indexes will be automatically generated if none are present in the reference genome directory, this make take a significant amount of time.

Trim

Purpose

Trims adapter sequence from barcoded reads, and saves this information in the read name. Reads which do not fall within barcode range and mismatch threshold are discarded.

Run Using

```
produse trim
```

or

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

Parameters

-c, –config A configuration file which can provide any of the following arguments

-i –input A fastq file to be trimmed. This argument must be specified twice. This file may be gzipped.

-o –output

An output file for writing trimmed reads. This argument must be specified exactly twice.

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

The output file order corresponds with the input file order.

-as –adapter_sequence The barcode sequence range, described using IUPAC bases.

-ap –adapter_position

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

0 = Do not consider this position

1 = Consider this position

Note that the entire adapter 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.

-v

Instead of discarding reads with mismatching adapter sequences and saving all other read, do the opposite (save mismatching reads, discard all others).

Useful for debugging.

-u Do not trim the adapter sequence.

Helpful tips

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

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.

bwa

Maps supplied reads to a reference genome using the Burrows-Wheeler aligner. Sorts and converts the resulting SAM file into a sorted BAM file. Note that, for all intents and purposes, this script is simply a python wrapper around bwa, with some functionality restrictions.

Run Using

```
produse align
```

or

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

Parameters

-c –config A configuration file which can supply any of the parameters described below

-i –input A fastq file containing the reads to be mapped. This option must be provided exactly twice. These files may be gzipped.

-o –output Path and name of the output BAM file.

-t –threads Number of threads to use while running BWA.

-r –reference Reference genome, in fasta format. BWA indexes should exist in the same parent directory. If these do not exist, use *bwa index* to generate them.

-R –readgroup Read group information to add to the BAM file header. Useful for running some exterior tools, such as Picard.

Additional Considerations

To keep the terminal as clean as possible, many bwa output messages are suppressed. This can inadvertently include some error messages. If bwa.py exits abruptly, try running bwa directly to determine the cause of the failure.

Collapse

Identifies the start position, adapter sequence, and mapping strand for each read in the supplied BAM file. If two or more reads share the same start position, mapping strand, and adapter sequence (within mismatch tolerance), they are merged into a single consensus sequence. If there is a mismatch at a given position, the most common base is used as a consensus. The quality of each base set to the highest quality base at that position. If an individual read contains too many mismatches, it is discarded prior to collapsing.

Run Using

```
produse collapse
```

or

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

Parameters

-c –config A configuration file which can provide any of the parameters below

-i –input Input BAM file. The name of each read must contain the adapter sequence.

-o –output Path and name of output collapsed fastq files. This parameter must be specified exactly twice. The output fastqs can be gzipped automatically by appending ".gz" to the output name.

-sp –strand_position The positions in the adapter sequence to use when comparison adapter sequences for reads of the same type (i.e. between forward reads, or between reverse reads). 1=Use this position, 0=Do not use this position.

-dp –duplex_position 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.

-amm –adapter_max_mismatch The maximum number of mismatches allowed between the expected and actual adapter sequences when comparing reads of the same type (See -sp).

-dmm –duplex_max_mismatch The maximum number of mismatches allowed between the expected and actual adapter sequences when comparing molecules of different types (See -dp).

-smm –sequence_max_mismatch The maximum number of mismatches allowed in an individual read before it is discarded. This threshold should be adjusted based upon read length.

-oo –original_output Path and name of fastq files to write original (i.e. pre-collapse) reads. Reads exceeding mismatch thresholds will still be discarded. This option must be be specified exactly twice, or not at all. These fastqs can be gzipped automatically by appending ".gz" to the output name.

-sf –stats_file Path and name of a text file to store collapsing statistics.

Additional Considerations

The runtime of this script 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.

SplitMerge

Processes each read in the Stitched BAM file. Unstitched reads are simply saved to the output BAM file. If a read was stitched, the bases unique to the original forward and reverse read are identified and assigned to a replacement forward and reverse read. Bases shared between the two sequences are assigned to only one of the paired reads, to remove overlap. In addition, if Stitcher misaligned a portion of the Stitched sequence, the correct start position of each read is determined, and an INDEL is added as necessary.

Run Using

```
produse split_merge
```

or

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

Parameters

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

-i –input A BAM file containing reads merged (stitched) by Stitcher. This file must be sorted by read name.

-u –unstitched_input A BAM file containing reads immediately before they were merged by stitcher. If running the ProDuSe pipeline, this will be the collapsed BAM file. This file must be sorted by read name.

-o –output Path and name of the output BAM file.

Note: Input files must be sorted by read name, not position.

Additional Considerations

While the vast majority (>99.99%) of stitched reads will be successfully separated, some reads with very complicated stitching will be discarded. Note that while this step will identify some additional INDELs that bwa did not identify due to their size, ProDuSe does not currently call INDELs.

SNV

Identifies the nucleotides that overlap each position in the BAM file. All reads formed by a consensus less than the min_read_per_uid threshold are ignored. The type of each nucleotide at a position are assigned as follows:

1. If a read is in duplex, its' bases are assigned as duplex supported bases (D). Otherwise, they are assigned as singletons (S)
2. For each stand (+ and -), if a consensus read was generated by at least strong_supported_base_threshold number of original reads, that base is assigned as a strong base (P and N). Otherwise, they are assigned as weak bases (p and n).

Run Using

```
produse snv
```

or

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

Parameters

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

-i –input An input BAM containing collapsed reads. This file should be sorted by read position.

-o –output Path and name of an output VCF file containing candidate variants.

-s –molecule_stats Path and file name of an output tsv file containing position depth and molecule abundance information.

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

-tb –target_bed A tab-delineated BED3 file in which to restrict variant calling.

-vaft –variant_allele_fraction_threshold Minimum variant allele fraction to call a variant on each strand.

-mo –min_molecules Total molecules (i.e. actual depth) required to call a variant at this position. This threshold should be adjusted if you are restriction to sites expected to be mutated.

-mum –mutant_molecules Minimum number of supporting molecules on both strands required to call a variant.

-mrpu –min_reads_per_uid Each consensus read must have been formed from at least this number of original reads to use this read in variant calling.

-ssbt –strong_supported_base_threshold Bases from a consensus read collapsed from at this number of original reads will be classified as a strong supporting base.

-eds –enforce_dual_strand Only variants with support on both the positive and negative strand will be called.

-mq –min_qual Bases from weak supporting molecules with less than this quality threshold will be treated as an "N".

Filter

Filters candidate variants called by ProDuSe based upon the following characteristics:

- Minimum molecule counts: The number of molecules supporting a variant at a given position must deviate from the number expected from simple noise in order to call a variant.
- Molecule type: Stronger (duplex, strong bases) molecules will be examined for support (or lack of support) for a variant first. If too many strong molecule do not support a variant when support is expected, the variant will be discarded.
- (If designated) Dual strand support: Both the positive and negative strand must support a variant confidently before it is called.

Run Using

```
produse filter
```

or

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

Parameters

-c –config A configuration file which can supply any of the parameters described below.

-i –input Input VCF file, listing candidate variants and molecule counts.

-m –molecule_stats A tab-delineated file listing total depth and different molecule abundances across the entire capture space. Generated by snv.py.

-o –output Path and name to use for the output VCF file.

-sb –strand_bias_threshold Strand bias p-value threshold. If the strand bias of a variant is below this value, the variant will be discarded.

-st –strong_base_threshold Minimum number of strong bases required to call a variant. Note that this threshold will increase depending on locus and overall library characteristics for each molecule type.

-wt –weak_base_threshold Minimum number of weak bases required to call a variant. Note that this threshold will increase depending on locus and capture space characteristics for each molecule type.

Additional Considerations

If the duplicate rate of a library is extremely low, variants will be called based upon weak bases. Thus, a large number of false positive calls may be generated.

CHAPTER 3

Other Commands

Adapter Predict

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

Run Using

```
produse adapter_predict
```

or

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

Parameters

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

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

Additional Considerations

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

CHAPTER 4

Additional Links

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