



Long Interval Nucleotide K-mer Scaffolder

LINKS v1.6

René L. Warren, 2014-2016

email: rwarren at bcgsc.ca

Name

LINKS: Long Interval Nucleotide K-mer Scaffolder

Description

LINKS is a genomics application for scaffolding genome assemblies with long reads, such as those produced by Oxford Nanopore Technologies Ltd. It can be used to scaffold high-quality draft genome assemblies with any long sequences (eg. ONT reads, PacBio reads, another draft genomes, etc)

What's new in v1.6 ?

Incorporation of the BC Genome Sciences Centre custom Bloom filter with the rolling hash function. This new data structure supports the creation of Bloom filters from large genome assemblies (tested on assemblies of 3 Gbp human [writeBloom.pl ~3:30h @ 26GB RAM] and 20 Gbp white spruce [writeBloom.pl ~20h @ 178GB RAM]). The data

structure swap in v1.6 offers a 5-fold speed-up compared to v1.5.2, while supporting the creation of filters from large genome assembly drafts.

What's new in v1.5.2 ?

LINKS outputs a scaffold graph in gv format, with highlighted merges and edge attributes

What's new in v1.5.1 ?

Fixed a bug that prevented the creation of Bloom filters with a different false positive rate (FPR) than default. Using lower FPR does not influence scaffolding itself, only run time. For large genomes (>1Gbp), using a higher FPR is recommended when compute memory (RAM) is limiting.

What's new in v1.5 ?

LINKS uses a Bloom filter to limit hashed paired k-mers to only those found in the sequence file to re-scaffold. This feature decreases RAM usage by over 60%, while the run time is nearly unchanged. When ran iteratively, users can re-use Bloom filters with the -r options, which results in faster run times up to half compared to v1.3 and earlier.

<https://github.com/warrenlr/LINKS/blob/master/README.md> - [whats-new-in-v13](#)-**What's new in v1.3 ?**

Added support for fastq files. Added support for multiple long-reads files. With v1.3, the reads file is not supplied directly through -s, but with a file-of-filenameinstead, which is a text file listing the fullpath/FASTA or FASTQ on your system. The file-of-filenames supplied through the -s option could include a mixture of FASTA and FASTQ files.

What's new in v1.2 ?

Fixed bug that prevented reading traditional FASTA sequences (where a sequence is represented as a series of lines typically no longer than 120 characters)

What's new in v1.1 ?

Included offset option (-o option) - Enable LINKS to explore a wider k-mer space range when running iteratively
Minor fixes: IUPAC codes are now preserved

Implementation and requirements

LINKS is implemented in PERL and runs on any OS where PERL is installed. In v1.6, there is a single dependency to the BloomFilter.pm (included, see below for download/compile instructions) - BC Genome Sciences Centre's common Bloom filter
In v1.5, there is a single dependency to Bloom::Faster - an extension for the c library libbloom.

Install

Download the tar ball, gunzip and extract the files on your system using:
gunzip links_v1-6.tar.gz tar -xvf links_v1-6.tar

In v1.6, the use of the Bloom::Faster PERL library is deprecated

INSTRUCTIONS TO BUILD THE BloomFilter PERL module

1. DOWNLOAD the BC Genome Sciences Centre's BloomFilter: The BTL C/C++ Common Bloom filters for bioinformatics projects, as well as any APIs created for other programming languages.

cd ./links_v1.6/lib git clone git://github.com/bcgsc/bloomfilter.git cd swig

1. BUILD a PERL5 module

Make sure you have swig installed and included in your path.

<http://www.swig.org/>

TO BUILD a Perl5 module (run in swig/):

```
preinst-swig -Wall -c++ -perl5 BloomFilter.i g++ -c  
BloomFilter_wrap.cxx -I/usr/lib64/perl5/CORE -fPIC -Dbool=char g++ -  
Wall -shared BloomFilter_wrap.o -o BloomFilter.so
```

TO COMPILE, swig needs the following Perl5 headers:

```
#include "Extern.h" #include "perl.h" #include "XSUB.h"
```

If they are not located in /usr/lib64/perl5, you can run "perl -e 'use Config; print \$Config{archlib};'" to locate them.

1. VERIFY your install

```
[swig]$ ./test.pl
```

1. CHANGE the path to BloomFilter.pm in
LINKS/writeBloom.pl/testBloom.pl

You only need to change if you have re-built in a relative directory different from: use lib "\$FindBin::Bin/./lib/bloomfilter/swig"; (for LINKS)
use lib "\$FindBin::Bin/./lib/bloomfilter/swig"; (for
writeBloom.pl/testBloom.pl)

Documentation

Refer to the LINKS.readme file on how to run LINKS and the LINKS web site for information about the software and its

performance www.bcgsc.ca/bioinfo/software/links

Questions or comments? We would love to hear from you!

Email: rwarren at bcgsc.ca

Citing LINKS

Rene L. Warren, Chen Yang, Benjamin P. Vandervalk, Bahar Behsaz, Albert Lagman, Steven J. M. Jones and Inanç Birol. 2015. LINKS: Scalable, alignment-free scaffolding of draft genomes with long reads. GigaScience 4:35

DOI: 10.1186/s13742-015-0076-3© Warren et al. 2015

Thank you for using, developing and promoting this free software.

Credits

LINKS Rene Warren

SWIG/BloomFilter.pm Sarah Yeo Justin Chu

<https://github.com/bcgsc/bloomfilter> Justin Chu Ben Vandervalk Hamid Mohamadi (rolling/ntHash) Sarah Yeo Golnaz Jahesh

Running LINKS

e.g. `/usr/bin/time -v -o timeLINKS_ECK12singleTIG.txt ./LINKS -f
ecoliK12_abyss_illumina_contig_baseline.fa -s K12_F2D.fof -b ecoliK12-
ONT_linksSingleIterationTIG`

Usage: `./LINKS [v1.6] -f` sequences to scaffold (Multi-FASTA format, required) `-s` file-of-filenames, full path to long sequence reads (Multi-FASTA/fastq format, required) `-d` distance between k-mer pairs (eg. target distance to re-scaffold on. default `-d 4000`, optional) `-k` k-mer value (default `-k 15`, optional) `-t` step of sliding window when extracting k-mer pairs from long reads (default `-t 2`, optional) `-o` offset position for extracting k-mer pairs (default `-o 0`, optional) `-e` error (%) allowed on `-d` distance e.g. `-e 0.1 == distance +/- 10%` (default `-e 0.1`, optional) `-l` minimum number of links (k-mer pairs) to compute scaffold (default `-l 5`, optional) `-a` maximum link ratio between two best contig pairs (default `-a 0.3`, optional) *higher values lead to least accurate scaffolding* `-b` base name for your output files (optional) `-r` Bloom filter input file for sequences supplied in `-s` (optional, if none provided will output to `.bloom`) NOTE: BLOOM FILTER MUST BE DERIVED FROM THE SAME FILE SUPPLIED IN `-f` WITH SAME `-k` VALUE IF YOU DO NOT SUPPLY A BLOOM FILTER, ONE WILL BE CREATED (`.bloom`) `-p` Bloom filter false positive rate (default `-p 0.0001`, optional; increase to prevent memory allocation errors) `-x` Turn off Bloom filter functionality (`-x 1 = yes`, default = no, optional) `-v` Runs in verbose mode (`-v 1 = yes`, default = no, optional)

Notes: `-s K12_F2D.fof` specifies a file-of-filenames (text file) listing: `K12_full2dONT_longread.fa` (see `./test` folder) `-x` When turned on (`-x 1`),

LINKS will run with a behaviour equivalent to v1.3 (no Bloom filters). This may be useful for large genome assembly drafts and when long reads are extremely high quality.

Tips to minimize memory usage

The most important parameters for decreasing RAM usage are `-t` and `-d`. The largest dataset used for scaffolding by our group was a draft assembly of the white spruce genome (20 Gb)* - For this, a large sliding window, `-t` (200) was used and was decreased as the k-mer distance `-d` increased. *refer to `LINKSrecipe_pglaUCA29-WS77111.sh` in the `./test` folder

Because you want to start with a low `-d` for scaffolding, you have to estimate how many minimum links (`-l`) would fit in a `-d` window \pm error `-e` given sliding window `-t`. For instance, it may not make sense to use `-t` 200, `-d` 500 at low coverages BUT if you have at least 10-fold coverage it might since, in principle, you should be able to derive sufficient k-mer pairs within same locus if there's no bias in genome sequencing.

For re-scaffolding white spruce, only 1X coverage was available (since the re-scaffolding used a draft assembly instead of long reads), but even `-t` 200 `-d` 5000 (1st iteration) did merge scaffolds even though, in theory, the `-e` parameter will play an important role limiting linkages outside of the target range `-d` (\pm) `-e` %.

On the data side of things, reducing the coverage (using less long reads), and limiting to only the highest quality reads would help decrease RAM usage.

In v1.5, LINKS builds a Bloom filter that comprises all k-mer of a supplied (`-f`) genome draft and uses it to only hash k-mer pairs from longreads having an equivalent in the Bloom filter. When LINKS runs iteratively, the bloom filter built at the first iteration is re-used thus saving execution time. These are the best tips I can offer at the moment, until we address it further programmatically using even more efficient data structures & code.

Test data

Go to ./test

To reproduce all the assemblies from the manuscript, execute: ./runall.sh

run:

./runme_EcoliK12single.sh

The script will download the baseline E. coli abyss scaffold assembly and full 2D ONT reads (Quick et al 2014) and used the latter to re-scaffold the former, with default parameters (Table 1D in paper).

NEED ~8GB RAM WITH CURRENT PARAMETERS. Increase (-t) to use less RAM.

./runme_EcoliK12iterative.sh

The script will download the baseline E. coli abyss scaffold assembly and full 2D ONT reads (Quick et al 2014) and used the latter to re-scaffold the former, iteratively 30 times increasing the distance between k-mer pairs at each iteration (Table 1F in paper).

NEED ~16GB RAM WITH CURRENT PARAMETERS. Increase (-t) to use less RAM.

./runme_ScerevisiaeW303iterative.sh

This script will download the S. cerevisiae W303 raw ONT long reads and used them iteratively to scaffold a baseline IlluminaMiSeq assembly (both data from <http://schatzlab.cshl.edu/data/nanocorr/>). You will need a computer with at least 132GB RAM. This process was clocked at 6:08:21 (h:mm:ss wall clock) and used 118GB RAM on a Intel(R) Xeon(R) CPU E5-2699 v3 @ 2.30GHz 16 dualcore (but running on a single CPU). (Fig 1 in the main ms, FigS8 in preprint).

NEED <132GB RAM WITH CURRENT PARAMETERS. Increase (-t) to use less RAM.

`./runme_ScerevisiaeS288iterative.sh`

This script will download the *S. cerevisiae* W303 raw ONT long reads and used them iteratively to scaffold a baseline ABySS assembly of Illumina data. You will need a computer with at least 132GB RAM. (Fig 1 in the main ms, FigS8 in preprint).

NEED <132GB RAM WITH CURRENT PARAMETERS. Increase (-t) to use less RAM.

`./runme_StyphiH58iterative.sh`

This script will download the *S. typhi* H58 2D ONT long reads and used them iteratively to scaffold a baseline assembly of Illumina data (both from Ashton, P.M. 2015. Nat. Biotechnol. 33, 296–300). You will need a computer with at least 132GB RAM. (Fig 1 in the main ms, FigS8 in preprint).

Additional info:

The file: LINKSrecipe_pglaucapG29-WS77111.sh is provided to show the re-scaffolding recipe used to produce a re-scaffolded white spruce (*P. glauca*) genome assembly.

Likewise: LINKSrecipe_ecoliRawR7-3.sh is provided to show the process of scaffolding iteratively the *E. coli* assembly (Table 1H in Warren et al. 2015 manuscript). 30 iterations were done for the paper.

Testing the Bloom filters

Insertions:

`cd tools`

`./writeBloom.pl`

Usage: ./writeBloom.pl

-f sequences to scaffold (Multi-FASTA format, required)

-k k-mer value (default -k 15, optional)

-p Bloom filter false positive rate (default -p 0.0001, optional - increase to prevent memory allocation errors)

Queries:

cd tools

./testBloom.pl

Usage: ./testBloom.pl

-f sequences to test (Multi-FASTA format, required)

-k k-mer value (default -k 15, optional)

-r Bloom filter file

How it works

Process: long sequences are supplied as input (-s option, FASTA format) and k-mer pairs are extracted using user-defined k-mer length (-k) and distance between the 5'-end of each pairs (-d) over a sliding window (-t). Unique k-mer pairs at set distance are hashed. FASTA sequences to scaffold are supplied as input (-f), and are shredded to k-mers on both strands, tracking the [contig] sequence of origin, k-mer positions and frequencies of observation.

Algorithm:

FASTA sequences to scaffold are supplied as input (-f), and are shredded to k-mers on both strands, populating a Bloom filter¹⁵ whose number of elements corresponds to a rough approximation of the number of k-mers in the draft genome based on file size. The size of the filter can be adjusted by controlling its false positive rate (-p). Building a Bloom filter is optional (-x), but strongly recommended as it decreases the memory usage and run time. ONT reads are supplied as input (-s option, file-of-

filenames listing FASTA/FASTQ formatted files) and k-mer pairs are extracted using user-defined k-mer length (-k) and distance between the 5'-end of each pairs (-d) over a sliding window (-t). When both k-mers are found in the Bloom filter, unique k-mer pairs at set distance are hashed, tracking the contig or scaffold of origin, k-mer positions and frequencies of observation. LINKS has two main stages: contig pairing, and scaffold layout. Cycling through k-mer pairs, k-mers that are uniquely placed on contigs are identified. Putative contig pairs are formed if k-mer pairs are on different contigs. Contig pairs are only considered if the calculated distances between them satisfy the mean distance provided (-d), while allowing for a deviation (-e). Contig pairs having a valid gap or overlap are allowed to proceed to the scaffolding stage. Contigs in pairs may be ambiguous: a given contig may link to multiple contigs. To mitigate, the number of spanning k-mer pairs (links) between any given contig pair is recorded, along with a mean distance estimate. Once pairing between contigs is complete, the scaffolds are built using contigs in turn until all have been incorporated into a scaffold. Scaffolding is controlled by merging sequences only when a minimum number of links (-l) join two contig pairs, and when links are dominant compared to that of another possible pairing (-a). The predecessor of LINKS is the unpublished scaffolding engine in the widely-used SSAKE assembler¹⁶, and foundation of the SSPACE-LongRead scaffolder⁸. A summary of the scaffold layout is provided (.scaffold) as a text file, and captures the linking information of successful scaffolds. A FASTA file (.scaffold.fa) is generated using that information, placing N-pads to represent the estimated lengths of gaps, and a single "n" in cases of overlaps between contigs. A log summary of k-mer pairing in the assembly is provided (.log) along with a text file describing possible issues in pairing (.pairing_issues), pairing distribution (.pairing_distribution.csv) and compressed Bloom filter (.bloom). The Bloom filter is intended to be re-used (supplied via -r) for iterative LINKS runs.

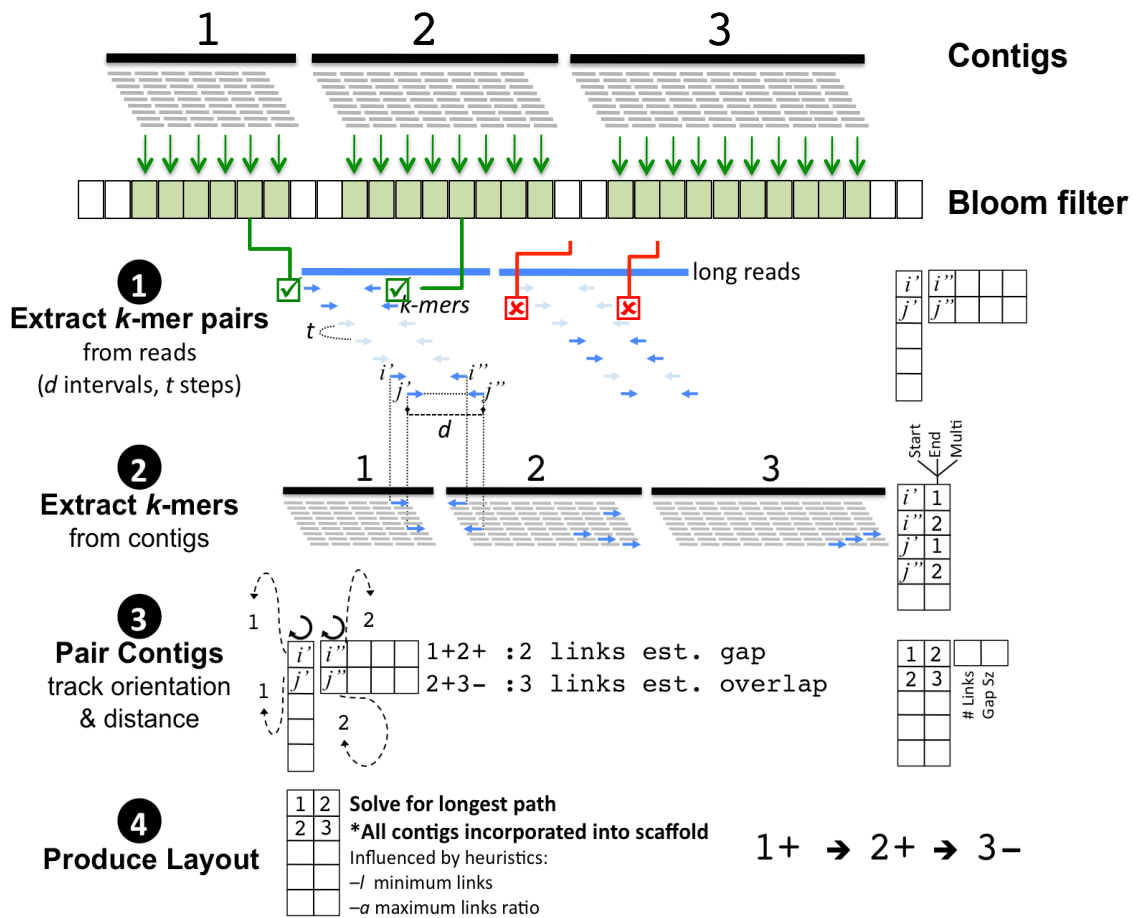


Figure 1. LINKS algorithm. Contigs (three thick black rectangles) are, optionally, shredded into k -mers and those k -mers used to construct a Bloom filter. Long reads (blue rectangles) are processed and k -mer pairs i' and i'' extracted at an interval corresponding to the input distance ($-d$), and window step ($-t$), but stored in memory (step 1, matrix on the right) only if both k -mers of a pair are found in the Bloom filter (**step 1**). Contigs are shredded into k -mers once more (**step 2**) using the same k value, but stored in memory (step 2, matrix on the right) only when its pair, identified in step 1, exists in memory. In **step 3**, contigs are paired when k -mers are not observed in the same sequence. Iterating through the data structure from step 1 and verifying placement (Start, End) and multiplicity (Multi) provides contig linkages, which are stored into memory (step 3, matrix on the right). In **step 4**, the scaffold layout is produced by incorporating all contigs into a scaffold, verifying neighbours and merging only when user-defined parameters support it (\geq / minimum number of links and \leq a maximum link ratio between alternate-to-primary linkage).

Consider the following contig pairs (AB, AC and rAD):

```

      A          B
=====
->          <-
->          <-
->          <-
      ->      <-

```

```

      A          C
=====
->          <-
      ->      <-

```

```

    rA          D          equivalent to rDA, in this order
=====
      ->      <-
      ->      <-
      ->      <-

```

Two parameters control scaffolding (-l and -a). The -l option specifies the minimum number of links (read pairs) a valid contig pair MUST have to be considered. The -a option specifies the maximum ratio between the best two contig pairs for a given seed/contig being extended. For example, contig A shares 4 links with B and 2 links with C, in this orientation. contig rA (reverse) also shares 3 links with D. When it's time to extend contig A (with the options -l and -a set to 2 and 0.7, respectively), both contig pairs AB and AC are considered. Since C (second-best) has 2 links and B (best) has 4 ($2/4 = 0.5$) below the maximum ratio of 0.7, A will be linked with B in the scaffold and C will be kept for another extension. If AC had 3 links the

resulting ratio (0.75), above the user-defined maximum 0.7 would have caused the extension to terminate at A, with both B and C considered for a different scaffold. A maximum links ratio of 1 (not recommended) means that the best two candidate contig pairs have the same number of links -- LINKS will accept the first one since both have a valid gap/overlap. When a scaffold extension is terminated on one side, the scaffold is extended on the "left", by looking for contig pairs that involve the reverse of the seed (in this example, rAD). With AB and AC having 4 and 2 links, respectively and rAD being the only pair on the left, the final scaffolds outputted by LINKS would be:

- 1) rD-A-B
- 2) C

LINKS outputs a .scaffolds file with linkage information between contigs (see "Understanding the .scaffolds csv file" below) Accurate scaffolding depends on many factors. Number and nature of repeats in your target sequence, optimum adjustments of distance (-d), deviation on the distance (-e), kmer sizes (-k), Minimum number of links (-l) and link ratio (-a) and data quality will all affect LINKS's ability to build scaffolds.

NOTE: IT IS ADVISED TO RUN LINKS WITH SMALLER DISTANCES (-d) FIRST, ESPECIALLY WHEN ASSEMBLIES ARE FRAGMENTED.

.log :: text file; Logs execution time / errors / pairing stats

.pairing_distribution.csv :: comma-separated file; 1st column is the calculated distance for each pair (template) with reads that assembled logically within the same contig. 2nd column is the number of pairs at that distance

.pairing_issues :: text file; Lists all pairing issues encountered between contig pairs and illogical/out-of-bounds pairing

.scaffolds :: comma-separated file; see below

.scaffolds.fa :: FASTA file of the new scaffold sequence

.bloom :: Bloom filter created by shredding the -f input into k-mers of size -k

Understanding the .scaffolds csv file

scaffold1,7484,f127Z7068k12a0.58m42_f3090z62k7a0.14m76_f1473z354
column 1: a unique scaffold identifier column 2: the sum of all contig sizes that made it to the scaffold/supercontig column 3: a contig chain representing the layout:

e.g. f127Z7068k12a0.58m42_f3090z62k7a0.14m76_f1473z354

It means: contig f127 (strand=f/+), size (z) 7068 (Z if contig was used as the seed sequence) has 12 links (k), link ratio of 0.58 (a) with a mean gap of 42nt (m) with reverse (r) of contig 3090 (size 62) on the right. if m values are negative, it's just that a possible overlap was calculated using the mean distance supplied by the user and the position of the reads flanking the contig. Negative m values imply that there's a possible overlap between the contigs. But since the pairing distance distribution usually follows a Normal/Gaussian distribution, some distances are expected to be larger than the median size expected/observed. In reality, if the exact size was known between each paired-reads, we wouldn't expect much negative m values unless a break occurred during the contig extension (likely due to base errors/SNPs).

License

LINKS Copyright (c) 2014-2016 Canada's Michael Smith Genome Science Centre. All rights reserved.

LINKS is released under the BC Cancer Agency software license agreement (academic use). Details of the license can be accessed at: http://www.bcgsc.ca/platform/bioinfo/license/bcca_2010

LICENSE AGREEMENT

BC CANCER AGENCY SOFTWARE LICENSE AGREEMENT (ACADEMIC USE)

CAREFULLY READ THE FOLLOWING TERMS AND CONDITIONS. This License Agreement (the "Agreement") is a legal contract between you, your employer, educational institution or organization (collectively, "You") and the British Columbia Cancer Agency ("BCCA") with respect to the license of the software, including all associated documentation (collectively, the "Product").

BCCA is willing to license the Product to You only if You accept the terms and conditions of this Agreement. By clicking on the "I ACCEPT" button, or by copying, downloading, accessing or otherwise using the Product, You automatically agree to be bound by the terms of this Agreement. IF YOU DO NOT WISH TO BE BOUND BY THE TERMS OF THIS AGREEMENT, DO NOT COPY, DOWNLOAD, ACCESS OR OTHERWISE USE THE PRODUCT.

1. **AUTHORITY:** In the event that You are an educational institution or organization, Your representative who is clicking the "I ACCEPT" button, or otherwise copying, downloading, accessing or using the Product hereby, in their personal capacity, represents and warrants that they possess the legal authority to enter into this Agreement on Your behalf and to bind You to the terms of this Agreement.

2. LICENSE TO USE: BCCA hereby grants to You a personal, non-exclusive, non-transferable, limited license to use the Product solely for internal, non-commercial use for non-profit research or educational purposes only on the terms and conditions contained in this Agreement. The Product may be installed at a single site at Your premises only. A copy of the Product installed on a single common machine or cluster of machines may be shared for internal use by Qualified Users only. In order to be a "Qualified User", an individual must be a student, researcher, professor, instructor or staff member of a non-profit educational institution or organization who uses the Product solely for non-profit research or educational purposes.
3. RESTRICTIONS: You acknowledge and agree that You shall not, and shall not authorize any third party to: (a) make copies of the Product, except as provided in Section 2 and except for a single backup copy, and any such copy together with the original must be kept in Your possession or control; (b) modify, adapt, decompile, disassemble, translate into another computer language, create derivative works of, or otherwise reverse engineer the Product, or disclose any trade secrets relating to the Product, except as permitted in Section 5; (c) license, sublicense, distribute, sell, lease, transfer, assign, trade, rent or publish the Product or any part thereof and/or copies thereof, to any third party; (d) use the Product to process any data other than Your own; (e) use the Product or any part thereof for any commercial or for-profit purpose or any other purpose other than as permitted in Section 2; or (f) use, without its express permission, the name of BCCA.
4. INTELLECTUAL PROPERTY RIGHTS: Subject to Section 5 below, all patents, copyrights, trade secrets, service marks, trademarks and other proprietary rights in or related to the Product and any improvements, modifications and enhancements thereof are and

will remain the exclusive property of BCCA or its licensors. You agree that You will not, either during or after the termination of this Agreement, contest or challenge the title to or the intellectual property rights of BCCA or its licensors in the Product or any portion thereof.

5. **OWNERSHIP OF IMPROVEMENTS:** In the event that the Product, in the form provided to You, includes source code (the "Source Code"), You are entitled to make improvements, modifications and enhancements to the Source Code (collectively, "Improvements") which Improvements are to be used by You for non-profit research and educational purposes only and You shall be the owner of those Improvements that You directly make and of all intellectual property rights to such Improvements, subject to the foregoing limits on Your use and distribution of such Improvements. You hereby grant to BCCA a perpetual, non-exclusive, worldwide, fully-paid, irrevocable license to use such Improvements for any purposes whatsoever, and to sublicense such Improvements including the right for third parties to sublicense the same, in perpetuity to the extent such rights are not limited in duration under applicable law, without identifying or seeking Your consent. Notwithstanding the foregoing, You acknowledge that BCCA and its licensors will retain or own all rights in and to any pre-existing code or other technology, content and data that may be incorporated in the Improvements. For greater certainty, this Section applies solely to the Source Code and shall not give You any rights with respect to the object code or any other portion or format of the Product which use, for greater certainty, is limited as set forth in this Agreement including as set out in Section 3(b) above. You acknowledge and agree that you will provide copies of Improvements to BCCA in such format as reasonably requested by BCCA at any time upon the request of BCCA.

6. CONFIDENTIALITY: You acknowledge that the Product is and incorporates confidential and proprietary information developed, acquired by or licensed to BCCA. You will take all reasonable precautions necessary to safeguard the confidentiality of the Product, and will not disclose any information about the Product to any other person without BCCA's prior written consent. You will not allow the removal or defacement of any confidential or proprietary notice placed on the Product. You acknowledge that any breach of this Section 6 will cause irreparable harm to BCCA and its licensors.
7. NO WARRANTIES: THIS PRODUCT IS PROVIDED TO YOU BY BCCA IN ORDER TO ALLOW YOU TO OBTAIN ACCESS TO LEADING ACADEMIC RESEARCH. THE PRODUCT IS PROVIDED TO YOU ON AN "AS IS" BASIS WITHOUT WARRANTY OF ANY KIND. NO WARRANTY, REPRESENTATION OR CONDITION EITHER EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTY OR CONDITION OF MERCHANTABILITY, NON-INFRINGEMENT, PERFORMANCE, DURABILITY OR FITNESS FOR A PARTICULAR PURPOSE OR USE SHALL APPLY. BCCA DOES NOT WARRANT THAT THE PRODUCT WILL OPERATE ON A CONTINUOUS OR TROUBLE FREE BASIS.
8. LIMITATION OF LIABILITY: TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT SHALL THE AGGREGATE LIABILITY OF BCCA TO YOU EXCEED THE AMOUNT YOU HAVE PAID TO ACQUIRE THE PRODUCT ("MAXIMUM AMOUNT") AND WHERE YOU HAVE NOT PAID ANY AMOUNT FOR THE PRODUCT THEN THE MAXIMUM AMOUNT SHALL BE DEEMED TO BE CDN\$100.00. IN NO EVENT SHALL BCCA BE LIABLE FOR ANY INDIRECT, INCIDENTAL, CONSEQUENTIAL, OR SPECIAL DAMAGES, INCLUDING WITHOUT LIMITATION ANY DAMAGES FOR LOST PROFITS OR SAVINGS, REGARDLESS OF

WHETHER THEY HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. EXCEPT TO THE EXTENT THAT THE LAWS OF A COMPETENT JURISDICTION REQUIRE LIABILITIES BEYOND AND DESPITE THESE LIMITATIONS, EXCLUSIONS AND DISCLAIMERS, THESE LIMITATIONS, EXCLUSIONS AND DISCLAIMERS SHALL APPLY WHETHER AN ACTION, CLAIM OR DEMAND ARISES FROM A BREACH OF WARRANTY OR CONDITION, BREACH OF CONTRACT, NEGLIGENCE, STRICT LIABILITY OR ANY OTHER KIND OF CIVIL OR STATUTORY LIABILITY CONNECTED WITH OR ARISING FROM THIS AGREEMENT. YOU AGREE THAT THE FOREGOING DISCLAIMER OF WARRANTIES AND LIMITATION OF LIABILITY ARE FAIR IN LIGHT OF THE NATURE OF THE RIGHTS GRANTED HEREIN AND THE AMOUNT OF FEES PAID BY YOU IN RESPECT OF THE PRODUCT.

9. INDEMNITY: You will indemnify, defend and hold harmless BCCA, its board of directors, staff and agents from and against any and all liability, loss, damage, action, claim or expense (including attorney's fees and costs at trial and appellate levels) in connection with any claim, suit, action, demand or judgement (collectively, "Claim") arising out of, connected with, resulting from, or sustained as a result of Your use of the Product or the downloading of the Product, including without limitation, any Claim relating to infringement of BCCA's intellectual property rights or the intellectual property rights of any third party.
10. SUPPORT AND MAINTENANCE: You acknowledge and agree that, unless and to the extent expressly agreed by BCCA in a separate written document, the Product is provided to You without any support or maintenance from BCCA and, for greater certainty, BCCA shall have no obligation to issue any update or upgrade to any Product.

11. TERM: This Agreement is effective until terminated. You may terminate this Agreement at any time by ceasing use of the Product and destroying or deleting any copies of the Product. This Agreement will terminate immediately without notice from BCCA if You fail to comply with any provision of this Agreement. BCCA may terminate this Agreement at any time upon notice to you where BCCA determines, in its sole discretion, that any continued use of the Product could infringe the rights of any third parties. Upon termination of this Agreement, and in any event upon BCCA delivering You notice of termination, You shall immediately purge all Products from Your computer system(s), return to BCCA all copies of the Product that are in Your possession or control, and cease any further development of any Improvements. On any termination of this Agreement Sections 1, 4, 6, 7, 8, 9, 13 and 14 shall survive such termination.
12. GOVERNMENT END USERS: Where any of the Product is used, duplicated or disclosed by or to the United States government or a government contractor or sub contractor, it is provided with RESTRICTED RIGHTS as defined in Title 48 CFR 52.227-19 and is subject to the following: Title 48 CFR 2.101, 52.227-19, 227.7201 through 227.7202-4, FAR 52.227-14, and FAR 52.227-19(c)(1-2) and (6/87), and where applicable, the customary software license, as described in Title 48 CFR 227-7202 with respect to commercial software and commercial software documentation including DFAR 252.227-7013, DFAR 252,227-7014, DFAR 252.227-7015 and DFAR 252.7018, all as applicable.
13. USE OF THE DOWNLOAD SERVICE: You acknowledge and agree that you will be responsible for all costs, charges and taxes (where applicable) arising out of Your use of the Product and the downloading of the Product. You acknowledge that You are

responsible for supplying any hardware or software necessary to use the Product pursuant to this Agreement.

14. GENERAL PROVISIONS: (a) This Agreement will be governed by the laws of the Province of British Columbia, and the laws of Canada applicable therein, excluding any rules of private international law that lead to the application of the laws of any other jurisdiction. The United Nations Convention on Contracts for the International Sale of Goods (1980) does not apply to this Agreement. The courts of the Province of British Columbia shall have non-exclusive jurisdiction to hear any matter arising in connection with this Agreement. (b) USE OF THE PRODUCT IS PROHIBITED IN ANY JURISDICTION WHICH DOES NOT GIVE EFFECT TO THE TERMS OF THIS AGREEMENT. (c) You agree that no joint venture, partnership, employment, consulting or agency relationship exists between You and BCCA as a result of this Agreement or Your use of the Product. (d) You hereby consent to Your contact information and any other personally identifiable information that You provide to us being disclosed to and maintained and used by us and our business partners for the purposes of (i) managing and developing our respective businesses and operations; (ii) marketing products and services to You and your staff; and (iii) developing new and enhancing existing products. You further agree that we may provide this information to other persons as required to satisfy any legal requirements and to any person that acquires some or all of the assets of BCCA. Where any of the personally identifiable information that You provide to us is in respect of individuals other than Yourself (such as Your staff) then You represent and warrant to use that You have obtained all necessary consents and authorizations from such individuals in order to comply with this provision. Please see the BCCA website for further information regarding personally identifiable information. (e) This Agreement is

the entire Agreement between You and BCCA relating to this subject matter. You will not contest the validity of this Agreement merely because it is in electronic form. No modification of this Agreement will be binding, unless in writing and accepted by an authorized representative of each party. (f) The provisions of this Agreement are severable in that if any provision in the Agreement is determined to be invalid or unenforceable under any controlling body of law, that will not affect the validity or enforceability of the remaining provisions of the Agreement. (g) You agree to print out or download a copy of this Agreement and retain it for Your records. (h) You consent to the use of the English language in this Agreement. (i) You may not assign this Agreement or any of Your rights or obligations hereunder without BCCA's prior written consent. BCCA, at its sole discretion may assign this Agreement without notice to you.