PrInSeS-G

Primer-Initiated Sequence Synthesis for Genomes

Installation

PrInSeS-G requires the following libraries, which must be pre-installed:

`libbam.a`	Interface to BAM files. Included in samtools package. http://samtools.sourceforge.net/. Please edit the file `Makefile` and change the definition of BAMROOT to the directory where the header files and libraries of samtools are installed.
`libpthread.a`	POSIX threads. It's usually found in `/usr/lib`.
`libz.a`	Compression/decompression. It's usually found in `/usr/lib`.

To compile PrInSeS-G, simply type "`make`". This should create the executable "`prinses`".

You will also need perl to run the utility `prinsesConsensus.pl`.

Manual

For a fast way of getting up to speed with PrInSeS-G, please refer to the Tutorial section at the end of this document.

Synopsis

```
prinses prepare alignments.bam > reads.prinses

prinses assemble genome.fa alignment.bam reads.prinses outdir

prinses validate outdir genome.fa alignment_script
```

Description

This application is intended to be used with whole genomes or long genomic sequences. It reads a BAM file containing the reads, as well as their mapping coordinates if aligned to the genome. PrInSeS-G uses heuristics to detect these dips in read depth and attempts to assemble the short region that contains them. It then reports any differences in the assembled region.

For more information, please refer to our publication: Primer-initiated sequence synthesis to detect and assemble structural variants, *Nature Methods* **7**, 485-486, July 2010 (http://www.nature.com/nmeth/journal/v7/n7/full/nmeth.f.308.html)

In the following, “R” represents the length of the longest read.

Commands

Step 1: Reads are converted into a new format:

```
prinses prepare [options] alignments.bam > reads.prinses
```

`alignments.bam`	Output of the alignment program in BAM format. All reads must be present, mapped or not.

Options:

-maxRepeats *n*	Discard reads that map to more than n loci. Default is 2
-minLength *n*	Discard reads shorter than n bases. Default is 20
-maxLength *n*	Discard reads longer than n bases. Default is 50,000
-noscores	Don’t include quality scores in the output file. This reduces the memory requirement for large datasets.

-filter *n* Discard reads whose average quality score is below n. Default is 0, ie no filtering.

Step 2: The main step where local assembly is used to discover variants

```
prinses assemble [options] genome.fa alignments.bam reads.prinses prinsesdir
```

`genome.fa` Fasta file containing the reference genome.

`alignments.bam` Same as in previous step. Used to call SNPs.

`reads.prinses` The output of the previous step.

`prinsesdir` Output directory. All files created will be here

Options:

-minMatchLength *n* The minimum number of bases that each read must overlap with the 3' end of the sequence assembled at any stage for that read to be included in the assembly. Default is 20.

-minDepth *n* The minimum number of reads that must align at each position for assembly to continue. Default is 3.

-maxDepth *n* When *n* reads are found to overlap for each position, the assembly algorithm stops looking for more and calls the base. Default is 10; higher numbers slow down execution, typically without a noticeable improvement in results. A value of 0 results in all reads with the minimum overlap to be used.

-readDepthLow n Read depth below which assembly is always attempted. Default is 0.

-readDepthHigh n Read depth above which assembly is never attempted. Default is 1.5 times the average read depth).

-maxTerminatorDistance *n* Maximum distance of terminator from primer. Default is 10,000.

-maxInsertionLength *n* The maximum additional length allowed for each local assembly. Default is 10000.

-verbose Outputs all attempts to assemble, successful or not, to logfile log.txt.

-viewFile *filename*	When combined with the verbose flag, it outputs information for each read considered to the view file named above. This will create a very large view file unless invoked with a small dataset. The ouput contains the sequences of all reads considered and a tab-separated line per position. Columns are: sequence name : direction of assembly, position, sum of scores of bases considered for this position, read depth, consensus base, list of bases considered with their score contribution (as a percent of the total) and number of occurrences.
-terminatorLength *n*	Sets the length of the terminator chosen for local assembly. Higher numbers reduce the chance for false positives, but increase the chance of false negatives. Default is R-1.
-noscores	Ignore quality scores for reads. Same if used in the prepare step above.
-maxThreads *n*	Run up to n threads in parallel. This speeds up execution significantly on multi-CPU computers. Default: number of CPUs.
-numTerminators *n*	Use n alternative terminators from the reference sequence. Default is 5.
-allowRepeatsInReference	Allows contigs where the corresponding fragment of the reference sequence constains repeats R-1 bp or longer.
-minMappingQuality *n*	When calling SNPs, ignore alignments with mapping quality less than n (default 20)
-progress	Output progress while running

This command produces a number of files in the output directory:

`variants.txt`	This is the most important file, as it contains the variant calls of PrInSeS-G. It does not contain SNPs calls from the aligner. See below for format.

`raw.variants.txt` Unprocessed variants are written as they are found. You can monitor this file while PrInSeS-G is running.

`alignmentConsensus.fa` Modified genome that includes the SNP calls from the BAM file.

`upstream.fa` and `downstream.fa` Modified genome, each file containing variants discovered in the corresponding direction of assembly. These are used in the validation step.

Step 3: The optional validation step. This step involves running the alignment program twice using an executable script passed as an argument. As such it's usually the slowest step and the one using the most disk space. However, it's useful in reducing false positives.

```
prinses validate [options] prinsesdir genome.fa alignments.bam script
```

`prinsesdir`	The PrInSeS-G output directory from the previous step
`genome.fa`	The reference genome
`alignments.bam`	Same as in previous steps
`script`	A script that aligns the reads to a genome. This script must take a genome fasta file as an argument and produce a sorted, indexed bam file. For example: "`script upstream.fa`" must produce `upstream.fa.bam`.

Options:

-nosnps Ignore SNPs in the reported statistics

The main file created is called `variants.verified.txt`. The utility script `prinsesConsensus.pl` can then be used to create a filtered list of variants, thus eliminating many potential false positives.

Step 4: Obtain the final modified genome:

```
cd prinsesdir
prinsesConsensus.pl variants.verified.txt > variants.consensus.txt
prinses apply alignmentConsensus.fa variants.consensus.txt > genome.prinses.fa
```

File `genome.prinses.fa` now contains the genome with all variant calls.

Variant file format

This format applies to all PrInSeS variant files (`variants.txt`, `variants.verified.txt` and others). All lines starting with # are treated as comments. Each variant appears on a separate line in a tab-separated list:

sequence:direction	The sequence/chromosome name followed by the direction of assembly: ‘d’ for downstream, ‘u’ for upstream, ‘b’ for both.
position	The 1-based position withint the sequence/chromosome
number of deleted bases	Number of bases in reference not found in assembled contig
bases inserted	Sequence to be inserted

Examples:

1) SNP: In chromosome 2L position 123456 one base is replaced by a G:

   2L 123456 1 G

2) Insertion: In 2L position 123456 the sequence GA is inserted:

   2L 123456 0 GA

3) Deletion: In 2L position 123456 13 bases are removed:

   2L 123456 13

4) Other: In 2L position 123456 2 bases are replaced by TTTT

   2L 123456 2 TTTT

Tutorial

The following refers to the files included under the tutorial directory in the PrInSeS-G release. To better understand how to deploy PrInSeS-G, take a look at the .sh scripts before running them.

In this tutorial, we use the genome for the phage PhiX-174 genome with one insertion and one deletion introduced. The read alignment program `bwa` as well as `samtools` are used, and they must already be installed in your computer.

If you want to use a different alignment program, modify script `align.sh`. It's important not to change the way it takes the reference genome fasta file as a parameter. The resulting `.bam` file must be sorted, indexed and must contain all the reads, both mapped and unmapped for PrInSeS-G to work.

Please inspect or edit `align.sh` and `prinses.sh`. Then run:

```
./prinses.sh
```

The output files are:

`prinsesdir/variants.txt` contains the variants called by PrInSeS-G (not those called by the alignment program).

`prinsesdir/variants.verified.txt` contains the variants with a note on the improvement of alignment coverage for each one.

`prinsesdir/log.txt` contains a log of all attempts it made to assemble (since we used the "verbose" option). You can see it made 27 attempts to assemble downstream and 28 upstream and found 2 indels in each case. The other attempts succeeded but produced the same sequence as the reference.

`prinsesdir/upstream.fa` and `downstream.fa` are fasta files with a combination of the reference, the aligner's SNPs and PrInSeS's indels. There is one sequence for each direction of assembly for PrInSeS.

`prinsesdir/variants.consensus.txt` contains the final list of variants, filtered after the validation step. Variants that don't result in an improvement of alignment are removed.

`prinsesdir/phix.prinses.fa` contains the modified genome containing all the (filtered) variants called by PrInSeS-G based on both read alignment and local assembly.