SPIRAL GENETICS

SpEC
User Instructions

How to Install and Run Spiral Encrypted Compression (SpEC)

SpEC Version 1.3.1

November 2015

Introduction

The Spiral Encrypted Compression (SpEC) format uses advanced compression techniques to reduce the storage size required for alignment data. The resulting files are typically much smaller than files using gzip or bzip2 compression. SpEC uses multiple processing threads to take advantage of all available CPUs, resulting in runtimes comparable to bzip2.

SPEC is a lossless compression format by default. SpEC files can easily be converted back into BAM files containing records identical to the original BAM file.

SpEC also provides optional lossy compression techniques, including read quality binning (using Illumina's quality binning algorithm) and the ability to discard unused BAM tags. For applications that do not require byte-for-byte isomorphism with the original BAM data, this can result in significantly reduced storage and processing time, without sacrificing accuracy in analysis.

Installation

Extract the archive using GNU tar:

```
$ tar jvxf spec-1.3.1.tar.bz2
```

This creates a spec-1.3.1/ directory with the following files:

- **spec**: the SpEC engine
- **bam2spec**, **spec2bam**, and **fasta2ref**: symlinks to spec
- **README.txt**: these user instructions.

SpEC itself requires no special privileges. You may simply add the installation directory to your PATH:

```
$ export PATH=$PATH:$PWD/spec-1.3.1/
```

If you have superuser rights, a system-wide directory such as /usr/local/bin/ may be appropriate.

```
$ sudo cp -v spec-1.3.1/* /usr/local/bin/
```

If you are not sure where to copy the files, ask your system administrator.

Getting Started

Convert BAM to SpEC (SpEC Encode)

To create a file in SpEC format, you will need a BAM file and a reference:

```
$ bam2spec --in original_file.bam --ref organism.fasta --out my_file.spec
```

The resulting SpEC file should be smaller than the original BAM.

Convert SpEC to BAM (SpEC Decode)

To convert a SpEC file back to BAM, use the freely available SpEC decoder:

```
$ spec2bam --in my_file.spec --ref organism.fasta --out my_file.bam
```

When run with the default options, SpEC Encode produces a BAM file that contains the same data as the original BAM. Running 'samtools view' will produce identical results for my_file.bam and original_file.bam. In most cases the BAM files will also be identical byte-for-byte, but due to minor variations in the internal gzip compression used by BAM, this is not guaranteed.

Reference Files

SpEC requires a reference file for the alignments stored in the BAM. The reference significantly improves the compression by replacing well-aligned sequences with reference coordinates. For the reference compression to work properly, the reference contig IDs must match the BAM contig IDs. During compression, if a BAM-to-reference mismatch is found, the compression is stopped and an error message is issued:

```
$ bam2spec --in original_file.bam --ref wrong.fasta --out my_file.spec
Error: Contig "X" was not found in the reference. Please use the correct
reference or add "--no-match-reference" to override
```

If the reference provided has contig IDs that do not match, supply the correct reference instead. But if only a few of the reads have contig IDs that are not in the reference, use the --no-match-reference flag to override the error message and compress the file anyway.

Note: The command requires that the --ref parameter be specified even if the --no-match-reference option is used.

✶ **Important: Using --no-match-reference and the wrong reference will degrade compression.**

A file can be successfully compressed with a reference with contig IDs that are different from the BAM file, using --no-match-reference, but the compression will be much less effective. As an example, a human BAM file was compressed to 53% of its original size with a reference with matching contig IDs, but only 86% of its original size using a reference with mismatching IDs.

If the number of unmapped or otherwise poorly compressible reads is unusually high, bam2spec will issue a warning:

```
Warning: 127211/4137138 (3.07%) reads used fallback encoding.
Warning: 572346/4137138 (13.83%) reads are unmapped.
To maximize compression, align your bam or use a better reference.
```

This is an indication that a significant number of reads cannot be mapped to the reference. In this case, a less efficient fallback compression method is used. For best compression performance, use aligned BAM files and a reference that most accurately reflects your read data.

Note: To see the unmapped and fallback statistics even if they are not unusually high, use the --stats option.

Achieving Better Compression without an Exactly Matching Reference File

The best possible compression performance is achieved by using a reference that matches your dataset as closely as possible. In some circumstances, the reference that was used with a particular BAM file (e.g. homo sapiens GRCh37) may not be readily available, but another (e.g. HG19) may be "close enough" to provide acceptable compression.

In this case, the sequence data is largely equivalent but the contig IDs do not match. For example, GRCh37 uses the convention "1", "2", "X", "MT", etc. while the equivalent in HG19 would be "chr1", "chr2", "chrX", "chrM".

You can specify a file that performs automatic contig ID remapping with the --remap option. This provides much better compression performance than --no-match-reference when most of the reads would otherwise match the reference:

```
$ bam2spec --in original_file.bam --ref HG19.ref --out my_file.spec
Error: Contig "chrM" was not found in the reference. Please use the correct
reference or add "--no-match-reference" to override

$ bam2spec --in original_file.bam --ref HG19.ref --out my_file.spec --remap
GRCh37_to_HG19.txt
```

You can use the same remapping file when converting back to BAM format, if desired:

```
$ spec2bam --in my_file.spec --ref HG19.ref --out my_file.bam --remap
GRCh37_to_HG19.txt
```

The format of the remapping file is tab delimited and consists of two columns. The left column specifies the contig IDs used in the BAM file, and the right column contains contig IDs used in the reference. **Note:** Lines beginning with # are ignored.

Our example GRCh37_to_HG19.txt would look like this:

```
# bam     ref
  chr1    1
  chr2    2
  chrX    X
  chrM    MT
  ...
```

Creating a REF File from a FASTA File

Although reference input in FASTA format is fully supported, the time taken to create a SpEC file can be reduced by compressing the reference to the REF format before processing. This may make sense if that reference will be used multiple times. Normally, a FASTA reference is compressed before the run and then discarded afterwards. But if multiple runs are likely, compressing the reference in advance and then reusing that compressed reference for all subsequent runs will increase efficiency. You can compress a FASTA to REF format using the fasta2ref command:

```
$ fasta2ref organism.fasta organism.ref
```

You can then use the REF file in place of the FASTA and the compression will run faster:

```
$ bam2spec --in original_file.bam --ref organism.ref --out my_file.spec
```

The resulting compressed reference can be used to create all SpEC files that use the same reference.

Keep in mind that the reference file used for encoding **must** also be used for decoding. If you do not have access to the identical reference used for encoding, the SpEC file cannot be decoded.

You can embed the reference in a SpEC file while encoding by using the --embed-ref option:

```
$ bam2spec --in original_file.bam --ref organism.ref --out my_file.spec --
embed-ref
```

This ensures that the reference is always available, but will increase the size of the SpEC file. For maximum storage savings, a single REF file should be shared among all SpEC files of the same species.

Advanced Compression Options

You can further reduce storage space and processing time by using lossy compression techniques. In many cases, these techniques will produce almost identical analysis results while saving substantial storage space.

Read Quality Binning

Illumina recommends a method for reducing the storage space required for read quality data. This method, called quality binning, can reduce storage space "without sacrificing accuracy, or standard analysis and variant calling performance".

http://www.illumina.com/documents/products/whitepapers/whitepaper_datacompression.pdf

To enable Illumina-style quality binning, use the --bin-qualities switch:

```
$ bam2spec --bin-qualities --in original_file.bam --ref organism.ref --out
my_file.spec
```

Discarding Unused Tags from Alignment Records

Alignment records in BAM format may contain any number of optional attribute fields, or tags, represented as two-character strings. Removing unwanted tags from alignment records can result in substantial storage savings. You can discard any extraneous tags using the --discard-tags option:

```
$ bam2spec --in original_file.bam --ref organism.ref --out my_file.spec --
discard-tags X0:X1:XN
```

If you want to keep just a few tags, use --keep-tags instead:

```
$ bam2spec --in original_file.bam --ref organism.ref --out my_file.spec --
keep-tags BQ:NM:MD
```

To discard all tags, use either of the following:

- --keep-tags none
- --discard-tags all

```
$ bam2spec --in original_file.bam --ref organism.ref --out my_file.spec --
keep-tags none
```

```
$ bam2spec --in original_file.bam --ref organism.ref --out my_file.spec --
discard-tags all
```

Note: These options apply to tags in the alignment section of the file, not to those fields in the header section, also referred to as tags.

Encryption

SpEC files can be encrypted using a strong symmetric key cipher (AES-GCM). Encryption is turned off by default.

To create an encrypted SpEC file, specify an encryption key with the --key parameter:

```
$ bam2spec --key "my secret key" --in original_file.bam --ref organism.ref --
out my_file.spec
```

To decrypt the file and recreate the original BAM, use the same key you just specified:

```
$ spec2bam --key "my secret key" --in my_file.spec --ref organism.ref --out
my_file.bam
```

If the spec2bam key does not match the key used in the bam2spec step, the decryption will fail:

```
$ spec2bam --key "something else" --in my_file.spec --ref organism.ref --out
my_file.bam
Error: Unable to decrypt. Check your --key and try again.
```

> ✶ **Warning: Back up your encryption key.**
>
> SpEC uses a strong symmetric key cipher (AES-GCM), and no backup of the key is made. Take care to keep track of your encryption keys. If a key is lost or forgotten, the SpEC file cannot be decrypted.

You can also specify the encryption key using the SPEC_KEY environment variable:

```
$ SPEC_KEY="my secret key" bam2spec --in original_file.bam --ref organism.ref
--out my_file.spec
$ export SPEC_KEY="my secret key"
$ spec2bam --in my_file.spec --ref organism.ref --out my_file.bam
```

This method is preferred in shared environments, where untrusted users may be able to see the key specified on the command line by using a ps command.

Miscellaneous Options

Overwriting Existing Files

For safety reasons, bam2spec and spec2bam refuse to overwrite existing output files. To force overwriting an existing file, use -f or --force:

```
$ bam2spec --in original_file.bam --ref organism.ref --out
already_exists.spec
 Refusing to overwrite 'already_exists.spec'. Use -f to override.

$ bam2spec --in original_file.bam --ref organism.ref --out
already_exists.spec –force
```

Suppressing Warnings

Non-fatal warnings are written to standard error, and can be suppressed with the --quiet switch.

```
$ bam2spec --in original_file.bam --ref organism.ref --out my_file.spec --
quiet
```

Threading

By default, bam2spec will automatically determine the available CPUs and use an appropriate number of threads for parallel processing. In most cases this shouldn't need to be changed.

You can specify a different number of threads with the --threads switch:

```
$ bam2spec --threads 2 --in original_file.bam --ref organism.ref --out
my_file.spec
```

Optimal settings vary from system to system, but a good rule of thumb for otherwise unloaded systems is one thread for each physical CPU core.

Writing a .bai File

Some analysis tools (such as GATK) require index files for proper operation. Running bam2spec with the --write-bai option will create an index file in addition to the SpEC file:

```
$ bam2spec --in original_file.bam --ref organism.ref --out my_file.spec --
write-bai
```

This will create my_file.spec and my_file.spec.bai. This file is needed only when using GATK integration with SpEC Stream.

Compression Statistics

To dump the entropy of all BAM fields and tags while encoding, use the --stats switch:

```
$ bam2spec --in original_file.bam --ref organism.ref --out my_file.spec –
stats
0/4137138 (0.00%) reads used fallback encoding.
428/4137138 (0.01%) reads are unmapped.

Entropy statistics:
  AM: 2.62138
  MD: 9.04054
  NM: 2.93508
  RG: 2.04124
  SM: 2.52501
  X0: 2.25389
  X1: 2.28602
  XA: 2.34792
  XC: 0.600558
  XG: 2.08
  XM: 3.01958
  XN: 0.000198977
  XO: 2.05743
  XT: 2.31434
  _C: 4.13838
  _F: 4.24792
  _M: 2.65521
  _N: 42.6706
  _P: 2.88652
  _Q: 240.888
  _R: 0.00560563
  _S: 207.937
  _T: 1.06865
  _p: 8.97873
  _r: 0.0614455
 Total: 551.661

$ du -h my_file.spec
  1.1G my_file.spec
```

Optional BAM tags consist of two alphanumeric characters, and mandatory fields start with the underscore character _. Larger numbers indicate more entropy, and therefore better potential for compression savings if the tag can be discarded.

The _S stat indicates the sequence (typically high entropy), and the _Q stat indicates the read quality scores. Running with --bin-qualities on the same (non-binned) BAM file results in a much lower _Q stat, and a smaller file:

```
$ bam2spec --in original_file.bam --ref organism.ref --out my_file.spec --
stats --bin-qualities
0/4137138 (0.00%) reads used fallback encoding.
428/4137138 (0.01%) reads are unmapped.

Entropy statistics:
  AM: 2.62138
  MD: 9.04054
  NM: 2.93508
  RG: 2.04124
  SM: 2.52501
  X0: 2.25389
  X1: 2.28602
  XA: 2.34792
  XC: 0.600558
  XG: 2.08
  XM: 3.01958
  XN: 0.000198977
  XO: 2.05743
  XT: 2.31434
  _C: 4.13838
  _F: 4.24792
  _M: 2.65521
  _N: 42.6706
  _P: 2.88652
  _Q: 110.153
  _R: 0.00560563
  _S: 207.937
  _T: 1.06865
  _p: 8.97873
  _r: 0.0614455
  Total: 420.927

 $ du -h my_file.spec
  787M my_file.spec
```

Even more compression can be achieved by discarding all optional tags:

```
$ bam2spec --in original_file.bam --ref organism.ref --out my_file.spec --
stats --bin-qualities --discard-tags all
0/4137138 (0.00%) reads used fallback encoding.
428/4137138 (0.01%) reads are unmapped.

Entropy statistics:
  _C: 4.13838
  _F: 4.24792
  _M: 2.65521
  _N: 42.6706
  _P: 2.88652
  _Q: 110.153
  _R: 0.00560563
  _S: 207.937
  _T: 1.06865
  _p: 8.97873
  _r: 0.0614455
 Total: 384.803

 $ du -h my_file.spec
  719M my_file.spec
```

Writing Uncompressed BAM Files

By default, spec2bam produces BAM files using gzip compression. You can write uncompressed BAM files with the --uncompressed switch:

```
$ spec2bam --uncompressed --in my_file.spec --ref organism.ref --out big.bam
```

Extracting a Region

You can extract a region of interest from the SpEC file to output in BAM format by using the --region option. A region can be specified in the same format used by the samtools view command: RNAME[:STARTPOS][-ENDPOS]

```
$ spec2bam --in my_file.spec --ref organism.ref --out my_file.bam --region
chr22

$ spec2bam --in my_file.spec --ref organism.ref --out my_file.bam --region
chr1:10000-50000
```

Note: SpEC currently does not allow extracting multiple regions.

Command Reference

Run any command with the -h (or --help) switch for a command reference.

bam2spec

Converts a BAM file to Spiral SpEC format.

Usage:

```
bam2spec --in [bamfile] --out [specfile] --ref [reference] [OPTIONS]
```

Options:

`-h [ --help ]`	Display this help message.
`--in`	Input BAM file to compress.
`--out`	Output SpEC file to write to.
`--ref`	Input reference file.
`--bin-qualities`	Bin quality values (smaller but lossy).
`--discard-tags`	Discard this colon-separated list of BAM tags. Default = none.
`--embed-ref`	Embed the reference in the SpEC file.
`-f [ --force ]`	Overwrite existing output SpEC file.
`--keep-tags`	Keep only this colon-separated list of BAM tags. Default = all.
`--key`	Encryption key.
`--no-match-reference`	Force compression without regard to a matching reference file. BAM file contigs not in the reference will not cause an error.
`-q [ --quiet ]`	Suppress warnings.
`--remap file.txt`	Remap the contig IDs of an unavailable reference file to those of a reference that is available, with an argument (file.txt) that specifies the name of the text file containing the correspondences.
`--stats`	Dump entropy statistics on completion.
`--threads`	Number of concurrent threads to use. Default = auto.
`--write-bai`	Output a BAI file as well.

spec2bam

Converts a Spiral SpEC file to BAM format.

Usage:

```
spec2bam --in [specfile] --out [bamfile] --ref [reference] [OPTIONS]
```

Options:

-h [--help]	Display this help message.
--in	Input SpEC file to decompress.
--out	Output BAM file to write to.
--ref	Input reference file. Notes: The command requires that the --ref parameter be specified even if the SpEC file was originally generated using the --no-match-reference option. It is optional, however, for embedded references.
-f [--force]	Overwrite existing output BAM file.
--key	Decryption key.
-q [--quiet]	Suppress warnings.
--region	Region of the SpEC file to decompress.
--remap file.txt	Remap the contig IDs of an unavailable reference file to those of a reference that is available, with an argument (file.txt) that specifies the name of the text file containing the correspondences.
--threads	Number of concurrent threads to use. Default = auto.
--uncompressed	Output uncompressed BAM. Default = compressed.

fasta2ref

Converts a .fasta file to a .ref reference.

Usage:

```
fasta2ref <fasta_file> <ref_file>
```

Options:

-h [--help]	Display this help message.
--fasta-file	Input reference FASTA file.
--ref-file	Output compressed reference file.