

DRAGEN Decompression v2.5.5 Software Guide

DRAGEN Compression creates a file with an Original Read Archived (.ora) extension. Orad uses the DRAGEN Decompression Software to decompress the .ora files.

Installation Requirements

The following are the minimum requirements for the DRAGEN Decompression Software.

- Hardware
 - 4 GB RAM
 - 2 GB free disk space
- Software

The Orad executable is compatible with the following Linux distributions:

 - CentOS 6 or later
 - Ubuntu 10.10 or later
 - Debian 7 or later
 - Red Hat Enterprise Linux (RHEL) 6 or later
 - Fedora 14 or later

Installing Orad

Use the following steps to install Orad.

1. Untar the archive.

```
tar -xzf orad.2.5.5.tar.gz
```
2. Change to the Orad directory.

```
cd orad_2_5_5
```
3. Move the executable to your preferred location.

```
mv orad ~/bin/orad
```
4. Add Orad to your PATH.

For example:

```
echo 'PATH=$PATH: ~/bin' >> ~/.bashrc
source ~/.bashrc
```
5. Move the oradata folder and its content to the home repository.

```
mv oradata ~
```

If you prefer to store the folder in a different location, you can specify the chosen path in the `ORA_REF_PATH` environment variable:

```
mv oradata ~/otherlocation/
export ORA_REF_PATH=~/otherlocation/oradata/
```

Using Orad

After Orad has been installed, use the following commands to extract the files.

```
orad FILE [args]
```

or

```
orad [args] FILE
```

Available options

```
-t INT
```

```
--threads INT
```

Number of threads. The default value is the maximum allowed by the system, not to exceed 8. You can use this option to set a value larger than 8.

```
-o filename
```

```
--out filename
```

The name of the output file, including the path. The default is the name of the original compressed file.

```
-P path
```

```
--path path
```

The location of the output file, path only. The default file name will be used. If a path is not specified, the file will be created in the same location as the input file.

```
-V
```

```
--verbose
```

Verbose output, includes the execution time and the number of threads used.

```
-h -help
```

```
--help
```

Print help and exit.

```
-v -version
```

```
--version
```

Print software version and exit.

```
--rm
```

Delete the input file after successful execution. By default the input file is not deleted.

```
-f
```

```
--force
```

Overwrite the output file without prompting. By default, if the output file exists, the software exits without overwriting.

```
--debug
```

Print debug information.

```
-C
--check
```

Check the integrity of the specified Orad file. This option decompresses the file in memory, and verifies that the checksum of the decompressed data is identical to the checksum of the original data. It does not save the decompressed file.

```
-c
--stdout
```

Print the decompressed file to stdout. This is useful to pipe the result to another application without writing the decompressed file to disk.

```
-I
--interleave
```

When input was paired reads, the decompression automatically decompresses again to two separate files. Use this option to create a single file with interleaved paired reads.

```
-i
--info
```

Print information about the compressed .ora file, including the software version used to compress the file, total number of sequences in the file, and compression ratio.

```
-r
--repeat-header
```

Write the read header also on the third line of the FASTQ format (the line that starts with +). By default, Orad decompress files with an empty + line.

```
--gz
```

Decompress the .ora file into a gzipped file. This option is activated by default if the original file was gzipped.

```
--raw
```

Decompress the .ora file into an uncompressed FASTQ file. By default Orad will decompress to gzip format if input was gzipped.

Sample commands using Orad

Decompression command:

```
orad myfile.ora
```

Print information summary of an ora file:

```
orad myfile.fastq.ora --info
```

Check the integrity of an ora file:

```
orad myfile.fastq.ora --check
```

Print the first lines in the terminal:

```
orad -c --raw myfile.fastq.ora | head
```

Decompression combined with analysis software

When the FASTQ file is needed for computation, such as mapping with BWA, you can decompress the file and feed it directly to the analysis software without saving it separately. This method reduces read/write to the disk and achieves much better performance.

- If the analysis tool can read from the standard input, such as BWA:

```
orad file.fastq.ora -c --raw | bwa mem humanref.fasta - > resu.sam
```

The `-c` option decompresses to standard output. The result is piped `|` to BWA, which uses the dash option `-` to read from standard input. This also works for paired reads, with the `-p` option of BWA specifying that the input contains interleaved paired reads.

- If the analysis tool cannot read from the standard input, you can either use process substitution or the `mkfifo` mechanism.

- Example with process substitution:

```
bwa mem humanref.fasta <(orad file.fastq.ora -c --raw) > resu.sam
```

For the file name, use the `<()` syntax, containing the command that generates the file to standard output, in this case `ora` with the `-c` option. This system may not work if the software checks the input file name, and also only works if the file is read sequentially one time.

- Example with the `mkfifo` command, used for cutadapt on paired reads:

```
mkfifo r1.fastq
orad -c --raw r1.fastq.ora > r1.fastq &
mkfifo r2.fastq
orad -c --raw r2.fastq.ora > r2.fastq &
python /.local/bin/cutadapt -q 30 -m 30 -a AGATC -A AGATCGG -o
name1.fastq -p
name2.fastq r1.fastq r2.fastq
```

The cutadapt command sees the `r1.fastq` and `r2.fastq` files as if they existed on disk, but the files remain in system memory.