What is Ultraplex?

Ultraplex is an all-in-one software package for processing and demultiplexing fastq files. It performs the following processing steps:

1. Remove poor quality bases
2. Remove sequencing adaptors (eg Illumina universal sequencing adaptor)
3. Move unique molecular identifiers (UMIs) to the read header
4. Detect 5’ and (optionally) 3’ barcodes for (combinatorial) demultiplexing
5. Write out files for each barcode (or barcode combination)

Ultraplex was designed with speed and ease of use in mind. It is capable of demultiplexing an entire HiSeq lane, consisting of ~400 million reads, in just 20 minutes.

Should I use Ultraplex?

Ultraplex is primarily designed for the demultiplexing of sequencing data generated using in-house library preparation protocols, with custom adaptors (for example iCLIP libraries). If instead you are using a commercial library prep kit (eg Illumina Truseq or Lexogen Quantseq) then in all likelihood the sequencing facility will already have demultiplexed the files for you, so Ultraplex is probably not for you. Furthermore, if you are doing single cell RNA-seq or using, for example, Oxford Nanopore long-read sequencing, there are other softwares specifically designed for this purpose.

How do I use Ultraplex?

Required inputs

Ultraplex requires two inputs: a gzip-compressed fastq file, and a comma-separated file (csv) containing all the barcodes you have used. There are also multiple optional inputs, which are detailed below.

The simplest usage of Ultraplex is as follows:

```
ultraplex -i your_fastq_file.fastq.gz -b your_barcode_csv.csv
```

Fastq file:

The fastq file should be a 4-line per read fastq. It should be in gzipped format. If your fastq is uncompressed, or compressed in a different format, convert it to gzipped format or Ultraplex will not work.

Barcode csv:

For the barcode csv, the first column contains a list of all the 5’ barcodes, whereas the second column (optional) contains a list of semi-colon separated 3’ barcodes which are linked to each 5’ barcode. For example, the following csv has three 5’ barcodes, the second of which is linked to two 3’ barcodes:

```
NNNATGNN,
NNNCCGNN,ATG;TCA
NNNCACNN,
```

There are certain constraints on the barcode sequences that can be used. It is required that the length and position of the non-N nucleotides in the barcodes be consistent for all 5’ barcodes, and all 3’ barcodes if used. 5’ barcodes do not have to be consistent with 3’ barcodes. Barcodes can have different numbers of Ns, provided that the non-N characters are all consistent.

N characters are used to denote positions which contain randomers. These are of no use for demultiplexing; instead, then allow the removal of PCR duplicates further downstream (for example by UMI-Tools). The bases detected in positions corresponding to the “N”s in the barcode are removed from the read and placed in the read header, after “rbc:” (which stands for “random barcode”).

For example, the 5’ barcodes...

```
NNN ATG NN
NNN CCG NNN
NNN ATG
```

...are all consistent because the non-N characters are all in positions 4-6 relative to the 5’ end of the read. However…

```
NN ATG NN
NNN ATGC NN
```

... are not consistent with the first three barcodes (or each other) because, relative to the 5’ end the barcodes are in positions 3-5 and 4-7 respectively.

The rules governing 3’ barcodes are the same, except that the positions are defined relative to the 3’ end of the read. For example...

```
NN ATG NNN
 N CCG NNN
```

...are consistent (positions -6 to -4), but...

```
NN ATG NN
```

...is not (positions (-5 to -3).

We have noticed that some programs save csvs in a format that is incompatible with Ultraplex. If unexpected errors emerge, try saving the csv as a plain comma-separated file, avoiding any operating system-specific encoding.

Optional inputs

Output directory (-d, --directory):

This allows you to specify an output directory to which the temporary and final files, and the log file, are saved. If the directory does not already exist, Ultraplex will create it automatically.

5’ mismatches (-m5, --fiveprimemismatches):

This option allows the user to specify how many mismatches are permitted when detecting which 5’ barcode a read contains. If set to zero, then the 5’ barcode must match the expected barcode perfectly. By default, this value is set to one mismatch.

3’ mismatches (-m3, --threeprimemismatches):

This option allows the user to specify how many mismatches are permitted when detecting which 3’ barcode a read contains. If set to zero, then the 3’ barcode must match the expected barcode perfectly. By default, this value is set to zero mismatches.

Minimum quality score (-q, --phredquality):

The minimum quality score for 3’ end trimming (this uses the Cutadapt functionality). By default this is set to 30 (0.1% error rate). However, in some circumstances (especially when 3’ adaptors are used) it may be desired to reduce the stringency of this.

Threads (-t, --threads):

The number of threads used for multithreaded operation. Larger values result in a less-than-additive increase in speed, until the limits of your machine are reached. By default this is set to 4, however a small speed increase may be seen by setting this to 8 or 16.

Adapter sequence (-a, --adapter):

The 3’ adapter sequence to be removed. By default this is the Illumina universal sequencing adapter "AGATCGGAAGAGCGGTTCAG".

Output file prefix (-o, --outputprefix):

A prefix that is added to the output files to help identify which sequencing run they are derived from. By default this is “demux”.

Ultra mode and sbatch compression (-u, --ultra; -sb, --sbatchcompression):

These are two optional running modes that can increase the speed of the program. By default, Ultraplex writes compressed temporary files. Ultra mode instead writes uncompressed temporary files, then compresses at the end. Depending on the system, this may slightly increase performance, at the expense of requiring large amounts of free storage space.

Larger performance gains from Ultra mode are seen when it is used in conjunction with sbatch compression mode. This mode is only compatible with high performance computing clusters which have SLURM job management. After temporary files are concatenated, then are compressed using sbatch commands, thus enabling the compression workload to be spread across multiple nodes of a supercomputing cluster. Using Ultra mode and sbatch compression mode together may double overall performance speed. Sbatch compression mode is only relevant when using ultra mode.

Minimum length (-l, --min_length):

This is the minimum length of a read, before trimming and demultiplexing. By default this is set to 22. Note that if large barcodes are used, or tandem barcodes, it may be necessary to set this to an even larger value to avoid zero-length reads.

Ignore free space warning (-ig, --ignore_space_warning):

When Ultraplex is run, it first estimates how much free space is needed for the job to complete, and then checks whether there is enough space. If there is not, it will not run. This option overrides this behaviour, allowing Ultraplex to run even if it does not think there is enough free storage space.

Minimum 5’ quality score (-q5, --phred_quality_5_prime):

The minimum quality score used for 5’ end trimming. By default this is set to zero. It is strongly recommended that this is retained at 0, as 5’ end trimming will result in incorrect detection of 5’ barcodes.