Installation and use of the `OneCellPipe` pipeline

1CellBio Support

Version 1.0, October 2018

Table of Contents

Company . . . 1
Summary . . . 1
Supported platforms . . . 1
System requirements and recommendations . . . 1
Quickstart . . . 2
Description . . . 2
Pipeline execution overview . . . 2
Data processing details . . . 3
Included reference genomes . . . 3
Running the pipeline . . . 4
  Command line parameters . . . 4
  Parallelization . . . 6
  Custom `indrops` configuration . . . 6
Example data and results . . . 12
Using the container for additional analyses . . . 13
Software packages and licenses . . . 14
License . . . 15
Disclaimer . . . 15

Company

Our flagship inDrop™ System, a high-resolution, single-cell transcriptomics platform, delivers greater experimental control, more rare actionable information and lower overall cost per result compared to all other existing platforms.

Research laboratories around the world are now adopting our platform for a wide range of single-cell applications from tumor profiling to stem cells to embryo development to the identification and validation of new drug targets. Founded by leading scientists at Harvard University, we are based in Cambridge, MA, and we support our growing number of customers through a team of international sales and field application scientists.

Summary

Described here is a software wrapper (a.k.a., `OneCellPipe` or `OCP`) which controls the management and execution of the indrops software pipeline for processing single-cell sequencing data generated using 1CellBio's inDrop™ sequencing technology. The software leverages the NextFlow workflow management software to control the processing steps in a validated and consistent Singularity environment.

Supported platforms

We support execution of `OCP` in two configurations.

1. One a single server (or cloud instance) running Ubuntu 16.04.
2. On a high-performance computing cluster, with job scheduling manged by SLURM

It may be possible, however, to install and execute `OCP` on systems with different configurations or batch scheduling software. However, 1CellBio will only provide support for the above configurations.

System requirements and recommendations

While individual experiments may warrant different amounts of resources, we recommend that the servers which process the sequencing data have at least eight modern CPUs and at least 16 GB of RAM. In addition to size of the sequencing files, an additional 300—500GB of data storage will also be required to accommodate filtered sequencing files, alignment files, and other temporary files utilized by `OCP`.

- The OneCellPipe software should be run on a Linux-based operating system with `bash` and `curl` installed. We built the pipeline on and recommend Ubuntu 16.04.
- You need to install Nextflow as the workflow engine Singularity as the container software. Our install script may be able to do this for you.
- The pipeline can process data produced by the OneCellBio inDrop™ technology. We only support **Version 2** of the library protocol at this time. The files may be organized in several parts but in

a single library (e.g., A5_S1_L001_R1_001.fastq.gz, A5_S1_L001_R2_001.fastq.gz, A5_S1_L002_R1_001.fastq.gz, A5_S1_L002_R2_001.fastq.gz) or as a single part in a single library (e.g., R1.fastq.gz and R2.fastq.gz).

Quickstart

After a server meeting the platform and resource requirements is setup, the OCP pipeline can be installed with a single command.

```
export ONECELLPIPE='1.21'; wget -qO- https://goo.gl/PHa1UG | bash
```

After the installation of all required components and software is complete, the pipeline can be started using the included sample data.

```
cd ~/onecellpipe
nextflow onecellpipe.nf --dir sampledata
```

Description

The inDrop technology greatly simplifies the massively parallel sequencing of single cells. It is a high-throughput droplet-microfluidic approach for barcoding the RNA from thousands of individual cells for the analysis by next-generation sequencing and is described in the following key publications:

- Klein, et al. 2015, Cell 161, 1187-1201 “Droplet Barcoding For Single-Cell Transcriptomics Applied To Embryonic Stem Cells”
- Baron, et al. 2016, Cell Syst. 3(4):346-360. “A Single-Cell Transcriptomic Map of the Human and Mouse Pancreas Reveals Inter- and Intra-cell Population Structure.”
- Derr, et al. 2016, Genome Research 26(10):1397-1410. “End Sequence Analysis Toolkit (ESAT) expands the extractable information from single-cell RNA-seq data.”

1CellBio is commercializing this technology and is providing this software pipeline, OCP as a service for the biomedical research community.

Pipeline execution overview

1. As a first step the 1CellBio pipeline will run a quick parameter and system check.
2. If the Singularity image containing all additional software can not be found in the bin folder it will be downloaded.
3. Next, two quality control steps perform FastQC and a barcode counting step against the fastq files of single-cell data you provided. The results will be formatted as web page generated by MultiQC.

4. The indrops processing itself consists of five consecutive steps described below.
5. Finally a second version of the output matrix is created, showing genes (rows) counts of barcode (columns) data.
6. The indrops filtering results as well as a barcode abundance diagram are presented on a web page. All other data from the indrops analysis can be found within a folder with the name of the library.

Data processing details

To simplify dependency management as well as provide a consistent and tested environment, most of the software for the analysis of inDrop™ data is packaged inside a container (e.g, Singularity). The first time the pipeline is run on a machine, the container image will be downloaded. This will occur based on configuration.

1. If run locally the file onecellpipe-*.simg will be downloaded during the first run.
2. If run on an HPC system, the images are downloaded on demand. The timing will be the same as for the previous option. This option is required to efficiently run on a cluster: Every node in the cluster will need a copy of the image - or it can (and should be) be shared using the `cacheDir` setting.

After downloading and launching the container image, the indrops software takes over and performs the following steps:

1. Filtering of the input reads
2. Identification of abundant barcodes
3. Sorting of the reads according to their assigned barcode of origin
4. Filtering and quantification of the barcode expression and generation of BAM files (i.e., alignment)
5. Aggregation of the quantification results and generation of the expression matrix

Included reference genomes

The following reference genomes are provided. These are soft-masked Bowtie indices

- Human: Homo sapiens, GRCh38, Ensembl build 91
- Mouse: Mus musculus, GRCm38, Ensembl build 91

Setting the species value

When executing the pipeline, you should explicitly specify this paramter with the appropriate value, either:

`--speces human` or `--species mouse`

Running the pipeline

Command line parameters

You can run the entire raw data processing with the following command after navigating to the `onecellpipe` folder

```
nextflow onecellpipe.nf --dir {path/to/your/fastq-files}
```

Required arguments:

`--dir {path}`	Path to the directory with the fastq files you want to process. The result data will be written here as well unless you specify --out. Avoid spaces and special characters.

Optional arguments:

`--worker {number N}`	Use N different parallel processing jobs, default: 1. This will affect the filter and sorting processes. It is recommeded to keep this number of 1 to process a single library, especially if you see strange failures or error messages.
`--worker2 {number N}`	Use N different parallel processing jobs for the quantification step, default: 10.
`--config {file.yaml}`	Use a complete configuration file as expected by the indrops library. Use this option if your set up differs from the default the system is expecting, default: NULL (auto-generate the config file with standard settings).
`--out {dir}`	Write the output to a specific directory. Default: `.` (Use the same directory the fastq files are in).
`--min_reads {number N}`	Indrops quantify parameter: Ignore barcodes with less than the specified number of reads, default: 100.
`--min_counts {number N}`	Indrops quantify parameter: Ignore output for barcodes with less than the specified number of UMIFM counts (This would speed up downstream processing.), default: 0
`--samplesheet {path to SampleSheet.csv}`	If you provide the sample sheet for the given run, QC results can list sample names instead of barcode sequences. The required format of the file can be seen in sampledata/SampleSheet.csv, Default: `-` (no sample sheet).
`-email {your@email.com}`	Send a notification email (hosting and receiving mail server must support it) when the pipeline has ended, default: NULL (do not send an email).
`--qc {0,1}`	Activate the QC steps for fastq files, default: 0 (don't run QC).
`--species {human, mouse}`	Use the reference genome of either 'human' (default) or 'mouse'.
`--name {name}`	Provide a specific name for the project. This is only used within indrops. Avoid spaces and special characters. Default: - (use library name read from fastq files).
`--help`	Only print usage instructions.
`--transpose {0,1}`	Create a second count matrix with barcodes vs. genes at the end. (Default = 0; do not transpose).
`--timezone {name}`	Set a specific timezone for the timestamp at the top of the log file. Default "EST".
`--bam {0,1}`	Create bam files with the processed data, default: 0.

QC data can be very useful

- We recommend always running with `--qc 1` to generate the `FastQC` and `MultiQC` output

Don't forget about the species parameter

- Don't forget to make sure that `--species` is set appropriately. For example, if your sequencing data is from mouse, and you don't specify `--species mouse` when running `OCP`, your reads will be mapped against the default species (i.e., human), which is not likely something that you want.

Additional Nextflow parameters:

Note that these are with used with a single `-`.

`-with-report [file name]`	Report the execution status, the launch command, overall execution time and some other workflow metadata (Details)
`-with-timeline [file name]`	Nextflow can render an HTML timeline for all processes executed (Details)
`-with-trace`	Create trace.txt in the current directory with information about each process (Details)

Parallelization

To enable or increase parallel processing and therefore increase the analysis speed (and when using a HPC), increase the number of "worker" jobs on the command line, e.g. 100 for the quantification step and 5 for other steps with the OneCellPipe parameters:

```
--worker 5 --worker2 100
```

Please be aware that the indrops software sometimes may fail at the next step of the pipeline if the number of workers is too high.

Custom `indrops` configuration

If your file set up different from the default expected by the pipeline, you can still use the `OCP` by using the `--config` parameter, providing it with your prepared configuration file. The file has to be in the expected format, as in the examples below.

Single part and single library

Files

```
R1.fastq.gz
R2.fastq.gz
```

Command line

```
nextflow onecellpipe.nf --config
/data/onecellpipe/data_results/indrop_config_to_use.yaml --out
/data/onecellpipe/data_results
```

indrop_config_to_use.yaml

```
# project and library settings
project_name : "libA5"
project_dir : "/data/onecellpipe/data_results"
sequencing_runs :
  - name : "libA5"
    version : "v2"
    dir : "/data/onecellpipe/data"
    fastq_path : "{read}.fastq.gz"
    library_name: "libA5"
# standard indrops config
# part 1: general software paths within the container, do not change
paths :
  bowtie_index : '/home/onecellbio/ref/Homo_sapiens.GRCh38.91.annotated'
  bowtie_dir : '/home/onecellbio/bowtie'
  rsem_dir : '/home/onecellbio/RSEM/bin'
  python_dir : '/home/onecellbio/pyndrops/bin'
  indrops_dir : '/home/onecellbio/indrops'
  java_dir : '/usr/bin'
  samtools_dir : '/home/onecellbio/samtools-1.3.1'
# part 2: analysis parameters
parameters :
  umi_quantification_arguments:
    m : 10 #Ignore reads with more than M alignments, after filtering on distance from
transcript end.
    u : 1 #Ignore counts from UMI that should be split among more than U genes.
    d : 600 #Maximal distance from transcript end, NOT INCLUDING THE POLYA TAIL
    split-ambigs : False  #If umi is assigned to m genes, add 1/m to each genes count
(instead of 1)
    min_non_polyA : 15  #Require reads to align to this much non-polyA sequence. (Set
to 0 to disable filtering on this parameter.)
  output_arguments :
    output_unaligned_reads_to_other_fastq : False
    filter_alignments_to_softmasked_regions : False
  bowtie_arguments :
    m : 200
    n : 1
    l : 15
    e : 80
  trimmomatic_arguments :
    LEADING : "28"
    SLIDINGWINDOW : "4:20"
    MINLEN : "30"
    argument_order : ['LEADING', 'SLIDINGWINDOW', 'MINLEN']
  low_complexity_filter_arguments :
    max_low_complexity_fraction : 0.50
```

Single part, multiple libraries

Another more complicated example for a single part but multiple libraries is following.

Files

```
A5_S12_L001_R1_001.fastq.gz
A5_S12_L001_R2_001.fastq.gz
A5_S12_L002_R1_001.fastq.gz
A5_S12_L002_R2_001.fastq.gz
A6_S1_L001_R1_001.fastq.gz
A6_S1_L001_R2_001.fastq.gz
A6_S1_L002_R1_001.fastq.gz
A6_S1_L002_R2_001.fastq.gz
```

Command line

```
nextflow onecellpipe.nf  --config
/data/onecellpipe/data_results/indrop_config_to_use.yaml --out
/data/onecellpipe/data_results_2
```

indrop_config_to_use.yaml

```
# project and library settings
project_name : "libA5"
project_dir : "/data/onecellpipe/data_results_2"
sequencing_runs :
  - name : "libA5"
    version : "v2"
    dir : "/data/onecellpipe/more_data"
    fastq_path : "{read}.fastq.gz"
    split_affixes : ["L001", "L002"]
    libraries :
      - {library_name: "A5", library_prefix: "A5_S12"}
      - {library_name: "A6", library_prefix: "A6_S1"}
# standard indrops config
# part 1: general software paths within the container, do not change
paths :
  bowtie_index : '/home/onecellbio/ref/Homo_sapiens.GRCh38.91.annotated'
  bowtie_dir : '/home/onecellbio/bowtie'
  rsem_dir : '/home/onecellbio/RSEM/bin'
  python_dir : '/home/onecellbio/pyndrops/bin'
  indrops_dir : '/home/onecellbio/indrops'
  java_dir : '/usr/bin'
  samtools_dir : '/home/onecellbio/samtools-1.3.1'
# part 2: analysis parameters
parameters :
  umi_quantification_arguments:
    m : 10 #Ignore reads with more than M alignments, after filtering on distance from
transcript end.
    u : 1 #Ignore counts from UMI that should be split among more than U genes.
    d : 600 #Maximal distance from transcript end, NOT INCLUDING THE POLYA TAIL
    split-ambigs : False  #If umi is assigned to m genes, add 1/m to each genes count
(instead of 1)
    min_non_polyA : 15  #Require reads to align to this much non-polyA sequence. (Set
to 0 to disable filtering on this parameter.)
  output_arguments :
    output_unaligned_reads_to_other_fastq : False
    filter_alignments_to_softmasked_regions : False
  bowtie_arguments :
    m : 200
    n : 1
    l : 15
    e : 80
  trimmomatic_arguments :
    LEADING : "28"
    SLIDINGWINDOW : "4:20"
    MINLEN : "30"
    argument_order : ['LEADING', 'SLIDINGWINDOW', 'MINLEN']
  low_complexity_filter_arguments :
    max_low_complexity_fraction : 0.50
```

Multiple runs, multiple libraries

Even more complicated: An example for multiple runs (for the same samples in different directories) and multiple libraries is following.

Files

```
Run1/A5_S12_L001_R1_001.fastq.gz
Run1/A5_S12_L001_R2_001.fastq.gz
Run1/A5_S12_L002_R1_001.fastq.gz
Run1/A5_S12_L002_R2_001.fastq.gz
Run1/A6_S1_L001_R1_001.fastq.gz
Run1/A6_S1_L001_R2_001.fastq.gz
Run1/A6_S1_L002_R1_001.fastq.gz
Run1/A6_S1_L002_R2_001.fastq.gz

Run2/A5_S12_L001_R1_001.fastq.gz
Run2/A5_S12_L001_R2_001.fastq.gz
Run2/A5_S12_L002_R1_001.fastq.gz
Run2/A5_S12_L002_R2_001.fastq.gz
Run2/A6_S1_L001_R1_001.fastq.gz
Run2/A6_S1_L001_R2_001.fastq.gz
Run2/A6_S1_L002_R1_001.fastq.gz
Run2/A6_S1_L002_R2_001.fastq.gz
```

Command line

```
nextflow onecellpipe.nf  --config
/data/onecellpipe/data_results/indrop_config_to_use.yaml
```

indrop_config_to_use.yaml

```
# project and library settings
project_name : "libA5"
project_dir : "/data/onecellpipe/data_results_2"
sequencing_runs :
  - name : "Run1"
    version : "v2"
    dir : "/data/onecellpipe/more_data/Run1"
    fastq_path : "{read}.fastq.gz"
    split_affixes : ["L001", "L002"]
    libraries :
      - {library_name: "A5", library_prefix: "A5_S12"}
      - {library_name: "A6", library_prefix: "A6_S1"}
  - name : "Run2"
    version : "v2"
    dir : "/data/onecellpipe/more_data/Run1"
    fastq_path : "{read}.fastq.gz"
    split_affixes : ["L001", "L002"]
    libraries :
```

```
      - {library_name: "A5", library_prefix: "A5_S12"}
      - {library_name: "A6", library_prefix: "A6_S1"}
# standard indrops config
# part 1: general software paths within the container, do not change
paths :
  bowtie_index : '/home/onecellbio/ref/Homo_sapiens.GRCh38.91.annotated'
  bowtie_dir : '/home/onecellbio/bowtie'
  rsem_dir : '/home/onecellbio/RSEM/bin'
  python_dir : '/home/onecellbio/pyndrops/bin'
  indrops_dir : '/home/onecellbio/indrops'
  java_dir : '/usr/bin'
  samtools_dir : '/home/onecellbio/samtools-1.3.1'
# part 2: analysis parameters
parameters :
  umi_quantification_arguments:
    m : 10 #Ignore reads with more than M alignments, after filtering on distance from
transcript end.
    u : 1 #Ignore counts from UMI that should be split among more than U genes.
    d : 600 #Maximal distance from transcript end, NOT INCLUDING THE POLYA TAIL
    split-ambigs : False  #If umi is assigned to m genes, add 1/m to each genes count
(instead of 1)
    min_non_polyA : 15  #Require reads to align to this much non-polyA sequence. (Set
to 0 to disable filtering on this parameter.)
  output_arguments :
    output_unaligned_reads_to_other_fastq : False
    filter_alignments_to_softmasked_regions : False
  bowtie_arguments :
    m : 200
    n : 1
    l : 15
    e : 80
  trimmomatic_arguments :
    LEADING : "28"
    SLIDINGWINDOW : "4:20"
    MINLEN : "30"
    argument_order : ['LEADING', 'SLIDINGWINDOW', 'MINLEN']
  low_complexity_filter_arguments :
    max_low_complexity_fraction : 0.50
```

Example data and results

The download contains a set of example fastq files you can use to test your set up. It is small enough to run on a standard desktop or laptop computer. As it is not a full data set, the results will not allow a meaningful interpretation with data analysis tools, but will show you the type of output. Within the `onecellpipe` folder you can run the samples using the command:

```
nextflow onecellpipe.nf --dir sampledata ①
```

① By default, this will use `--species human`, not run quality control steps, and generate a default configuration file for use with `indrops`.

Depending on the computer system you are using, the pipeline will go through all steps in about 30 minutes. Progress will be printed on the command line as well as in the `sampledata/indrop_log.txt` file. The results will be written into the sampledata folder, the directory should look like the following after the pipeline finished successfully:

Data	Description

`A5_S12_L001_R1_001.fastq.gz`
`A5_S12_L001_R2_001.fastq.gz`
`A5_S12_L002_R1_001.fastq.gz`
`A5_S12_L002_R2_001.fastq.gz`
	Input files with the correct naming scheme

`index.html`	Web page generated with result data and quality control information
`indrop_log.txt`	The pipeline log file
`qc`	Directory with quality control data (optional, specified with `--qc 1`)
`barcode-test`	Directory with files from barcode sequence counting
`fastqc`	Directory with FastQC results
`multiqc`	Directory with MultiQC results
`resources`	Files used during the pipeline run
`A5.counts.2.tsv`	Barcode count matrix (barcodes as headers)
`A5`	Indrops result data
`abundant_barcodes.pickle`	Temporary data storage
`filtered_parts`	Directory with filtered fastq files
`A5.bam` `libA5.bam.bai`	Bam file & index from processed data (optional)
`A5.barcode_abundance.png` `A5.barcode_abundance_by_barcode.png`	Diagrams showing the barcode distribution
`A5.counts.tsv.gz`	Barcode count matrix (genes as headers) (optional)
`A5.filtering_stats.csv`	Table with filtering results
`A5.ignored_barcodes.txt`	Any barcodes that were filtered out
`A5.quant_metrics.tsv.gz`	Table with quantification results
`quant_dir`	Directory with quantification results

The main output from running this analysis can also be seen in the sampleresults folder provided.

Using the container for additional analyses

The onecellpipe software image has all the software installed to run the indrops pipeline and the

other analysis parts described here within an Ubuntu Linux system. If you are familiar with this environment, you can also manually create a container from the image, log in and run these or other analysis steps on the command line. Using Singularity with the image file already available this would be:

```
singularity shell /path/to/onecellpipe-25-2.img ①
```

① By default, the image is located in `/tmp`

Software packages and licenses

- The Nextflow framework used as a wrapper is released under the GNU GPLv3 license, available at https://github.com/nextflow-io/nextflow and published as P. Di Tommaso, et al. “Nextflow enables reproducible computational workflows.” Nature Biotechnology 35, 316–319 (2017) doi:10.1038/nbt.3820
- Singularity is a container system available at http://singularity.lbl.gov and released with a BSD 3-clause. It is used to package all third-party software except NextFlow.
- Docker open source is distributed under the Apache 2.0 license and available from https://store.docker.com/search?type=edition&offering=community. It is used as an alternative container system to package all third-party software except NextFlow.
- The Singularity images provided through this pipeline contain a small Ubuntu-based system called phusion/baseimage version 0.9.22. It is released under the MIT license and available at https://github.com/phusion/baseimage-docker. Within this image the following software packages are installed.
- The indrops software (version 4.2) is open source software and available at https://github.com/indrops/indrops. It is used as the main processing and analysis method.
- Indrops contains the Trimmomatic software available here: https://github.com/timflutre/trimmomatic with a GPL license.
- Miniconda2 (version 4.3.27-Linux-x86_64) is used as a package and environment management system. More details can be found at https://conda.io/docs/ and the license terms are described at https://conda.io/docs/license.html.
- Samtools (version 1.3.1) is used by indrops and is available here: https://github.com/samtools/samtools with a MIT/Expat license.
- Bowtie (version 1.2.2.2) is used by indrops and is available here: http://bowtie-bio.sourceforge.net with an Artistic license.
- RSEM (version 1.3.0) is used by indrops and is available here: https://github.com/deweylab/RSEM with a GPL3 license.
- FastQC (version 0.11.5) is used for QCing the fastq files. The software is available at https://www.bioinformatics.babraham.ac.uk/projects/fastqc with a GPL3 licence.
- MultiQC (version 1.3) is used to generate a QC report. It is available as open source at http://multiqc.info.
- Ensembl data (release 91) is used as genome index files. Software and data disclaimer can be

found here: http://www.ensembl.org/info/about/legal.

License

The software is released under the GNU GPLv3 License, a copy of which is included in the bin directory.

Disclaimer

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)

HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.