Troubleshooting Nextflow

Configuration

Before running Nextflow it is good to know where Nextflow is obtaining its configuration.

Configuration is determined by Nextflow looking for a file named nextflow.config firstly in the current directory, then the workflow script base directory (if it is not the same as the current directory) and finally in $HOME/.nextflow/config. The first config file it finds in this order will be determine the configuration it uses.

For example in the assembly workflow directory the default nextflow.config is

```
trace {
  enabled = true
  file = "pipeline_stats.txt"
  fields = "task_id,\
…
}
docker.runOptions='-u $(id -u):$(id -g)'
process {
    cpus = 1
    memory = "2GB"

    errorStrategy = { task.attempt <= 2 ? "retry" : "ignore" }
    maxRetries = 2
}
```

The trace block determines the name of and fields written to the file that records the pipeline task progress. `docker.runOptions='-u $(id -u):$(id -g)'` ensures that docker runs with permissions of the current user so that it can write out to directories owned by the user who is running the process.

The process block determines the defaults for each process (task) within the workflow. As you can see, by default each process will require 1 CPU and 2GB RAM. This does not mean that bad coding of the process will restrict it to those conditions. For example some software will automatically use as many cpus as are available and so if this is the case it is important that the workflow process script specifies one cpu (often with `-t 1` or `--threads 1`). In addition, if it is likely that a process may take more than the allocated RAM, the process RAM can be increased within the workflow script. For example

```
process ram_hog_software {
  memory '8 GB'
    ……
}
```

For debugging purposes the most important thing is to change the errorStrategy line. As show here it will retry a failed process twice and if it still fails ignore the error and carry on. For debugging purposes delete both of these lines, which will mean that if an error occurs it will fail, display the error and then stop the pipeline.

Debugging

When a task fails an error will be reported in a way similar to that shown below

```
[8a/dc3217] Submitted process > trimming (ERR586796_downsampled)
ERROR ~ Error executing process > 'qc_pre_trimming (ERR586796)'
Caused by:
  Process `qc_pre_trimming (ERR586796)` terminated with an error exit
status (127)

Command executed:
  fastqc ERR586796_1.fastq.gz ERR586796_2.fastq.gz
Command exit status:
  127
Command output:
  (empty)
Command error:
  pyenv: java: command not found
```

The cause of this error is captured in a file called `.command.err` within the work directory for that task.

The work directory is a temporary directory that Nextflow creates as it processes each job. Often there will be one for each sample and task. For example a directory for the task that runs fastqc on a sample, a directory for the task that runs trimmomatic, a directory for the task that runs SPAdes etc. By default Nextflow creates this directory named as work in the directory from where you run Nextflow.

In the case above the qc_pre_trimming step for ERR586796 was run in a directory called work/**8a/dc3217**…. The last part of the directory is not present but pasting this and using the tab key to autocomplete will enable you to look at this file

```
less work/8a/dc3217bdea569fc83568e84c5c3109/.command.err
```

This shows the contents of this file

```
pyenv: java: command not found
The `java' command exists in these Python versions:
  miniconda3-4.3.30
```

So in this case we can see that the error was due to java not being installed. On this occasion the -with-docker argument hadn't been specified so lots of the software required was not available.

You can see the exact command that was executed by examining the `.command.sh` file.

```
less work/8a/dc3217bdea569fc83568e84c5c3109/.command.sh
```

```
#!/bin/bash -ue
mkdir trimmed_fastqs
trimmomatic PE -threads 1 -phred33 ERR586796_downsampled_1.fastq.gz
ERR586796_downsampled_2.fastq.gz trimmed_fastqs/ERR586796_downsampled_1.fastq.gz
/dev/null trimmed_fastqs/ERR586796_downsampled_2.fastq.gz /dev/null
ILLUMINACLIP:adapter_file.fas:2:30:10 SLIDINGWINDOW:4:20 LEADING:25 TRAILING:25
MINLEN:33
```

Ensure that the inputs are present and correct

```
ls work/8a/dc3217bdea569fc83568e84c5c3109/
ERR586796_downsampled_1.fastq.gz ERR586796_downsampled_2.fastq.gz
adapter_file.fas                 trimmed_fastqs
```