Nanopore 16S Software Testing

Author: Sarah Burns

Date: 18 Dec 2019

1. Unit testing

1.1. Main scripts in project

Unit tests were written for the Python scripts contained within the 16S-Pipeline/scripts directory.

A total of 50 tests were ran and passed. There were six ResourceWarning alerts, which when researched are an issue with Python's unittest module, and indeed did not appear when running the functions normally.

```
..................................../home/nanopore/miniconda2/envs/general/lib/python3.6/subproc
ess.py:786: ResourceWarning: subprocess 13276 is still running
  ResourceWarning, source=self)
/home/nanopore/PycharmProjects/16S-Pipeline/tests/test_parse_taxdump.py:41: ResourceWarning: unc
losed file <_io.BufferedReader name=4>
  taxid = self.parser.get_tree_id(accession, accession_file)
./home/nanopore/miniconda2/envs/general/lib/python3.6/subprocess.py:786: ResourceWarning: subpro
cess 13279 is still running
  ResourceWarning, source=self)
/home/nanopore/PycharmProjects/16S-Pipeline/tests/test_parse_taxdump.py:47: ResourceWarning: unc
losed file <_io.BufferedReader name=4>
  taxid = self.parser.get_tree_id('NR_12345', 'accession.txt')
./home/nanopore/miniconda2/envs/general/lib/python3.6/subprocess.py:786: ResourceWarning: subpro
cess 13280 is still running
  ResourceWarning, source=self)
/home/nanopore/PycharmProjects/16S-Pipeline/scripts/parse_taxdump_files.py:95: ResourceWarning:
unclosed file <_io.BufferedReader name=5>
  tree_id = self.get_tree_id(seq_id, accession_file)
............
----------------------------------------------------------------------
Ran 50 tests in 0.171s

OK
```

The "coverage" tool was used to calculate the percentage of code covered by these tests.

```
Name                                                                                           Stmts
Miss  Cover
----------------------------------------------------------------------------------------------------
--------------
/home/nanopore/PycharmProjects/16S-Pipeline/config.py                                            25
0   100%
/home/nanopore/PycharmProjects/16S-Pipeline/docker/centrifuge_parser/parse_centrifuge.py        149
4    97%
/home/nanopore/PycharmProjects/16S-Pipeline/docker/seqmatch_parser/parse_seqmatch.py             46
4    91%
/home/nanopore/PycharmProjects/16S-Pipeline/scripts/__init__.py                                   0
0   100%
/home/nanopore/PycharmProjects/16S-Pipeline/scripts/dnanexus.py                                  27
8    70%
/home/nanopore/PycharmProjects/16S-Pipeline/scripts/gridion_watch.py                            190
88    54%
/home/nanopore/PycharmProjects/16S-Pipeline/scripts/hash.py                                       6
0   100%
/home/nanopore/PycharmProjects/16S-Pipeline/scripts/parse_taxdump_files.py                       90
29    68%
/home/nanopore/PycharmProjects/16S-Pipeline/scripts/rename_fasta_multiheader.py                  25
4    84%
test_create_pdf.py                                                                               50
1    98%
test_dnanexus.py                                                                                 21
1    95%
test_gridion_watch/test_gridion_watch.py                                                         66
1    98%
test_hash.py                                                                                     15
1    93%
test_parse_centrifuge.py                                                                        112
1    99%
test_parse_seqmatch.py                                                                           42
1    98%
test_parse_taxdump.py                                                                            58
1    98%
test_rename_fasta_multiheader.py                                                                 29
1    97%
/home/nanopore/PycharmProjects/16S-Pipeline/webapp/create_pdf.py                                 58
5    91%
----------------------------------------------------------------------------------------------------
--------------
TOTAL                                                                                          1009
150    85%
```

The majority of code not tested was code contained within the main function (which is tested elsewhere) or the script arguments.

1.2. Django unit tests for web app

Unit tests were written for the models, views and tags within the webapp using Django test module.

A total of 18 tests were ran and passed.

```
(general) nanopore@Nanopore-Office:~/PycharmProjects/16S-Pipeline/webapp$ coverage run manage.py
test results_app/tests/
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
..................
----------------------------------------------------------------------
Ran 18 tests in 0.343s

OK
Destroying test database for alias 'default'...
```

As above, “coverage” was used and produced the below report.

```
(general) nanopore@Nanopore-Office:~/PycharmProjects/16S-Pipeline/webapp$ coverage report
Name                                                        Stmts   Miss  Cover
---------------------------------------------------------------------------
/home/nanopore/PycharmProjects/16S-Pipeline/config.py          25      0   100%
create_pdf.py                                                  58     44    24%
manage.py                                                      13      6    54%
results_app/__init__.py                                         0      0   100%
results_app/admin.py                                            8      0   100%
results_app/migrations/0001_initial.py                          7      0   100%
results_app/migrations/0002_auto_20191128_0828.py               5      0   100%
results_app/migrations/__init__.py                              0      0   100%
results_app/models.py                                          69      0   100%
results_app/templatetags/results_tags.py                       21      0   100%
results_app/tests/test_models.py                               34      0   100%
results_app/tests/test_tags.py                                 43      0   100%
results_app/tests/test_views.py                                42      0   100%
results_app/urls.py                                             3      0   100%
results_app/views.py                                          102     55    46%
webapp/__init__.py                                              0      0   100%
webapp/settings.py                                             20      0   100%
webapp/urls.py                                                  4      0   100%
---------------------------------------------------------------------------
TOTAL                                                         454    105    77%
```

The views not fully tested in this way were functions which required nanoplot images etc. so testing for this is covered within workflow and user testing. The pdf script is tested in the above section.

2. User testing

Test Plan

A survey was created to capture user evaluation of the web app and emailed to five individuals (both scientific and technical) who frequently use the web app. There were five questions:

1. Are you able to? (Y/N)
    - Register a new account
    - Login to the site
    - Sort the results by one of the columns

- Filter the results by keyword
- Select a result
- Find the sample information
- Find the run quality information
- Find the FASTQ statistics for each time interval available
- Find the classification results for each time interval available
- Open the full NanoPlot report for each time interval available
- Download the results for each time interval available
- Return to the main page
- Logout

2. If you answered "No" to any of the above questions, please provide details.
3. In your opinion, does the app meet its requirements? If not, please provide details.
4. How easy do you find the app to use? (1=impossible to understand; 10=really obvious)
5. Are there any features you would like to be added in the future?

Success criteria

- Users are able to complete essential tasks
- The app meets requirements
- Ease of use is above 5 for all users

Results

Responses: 2

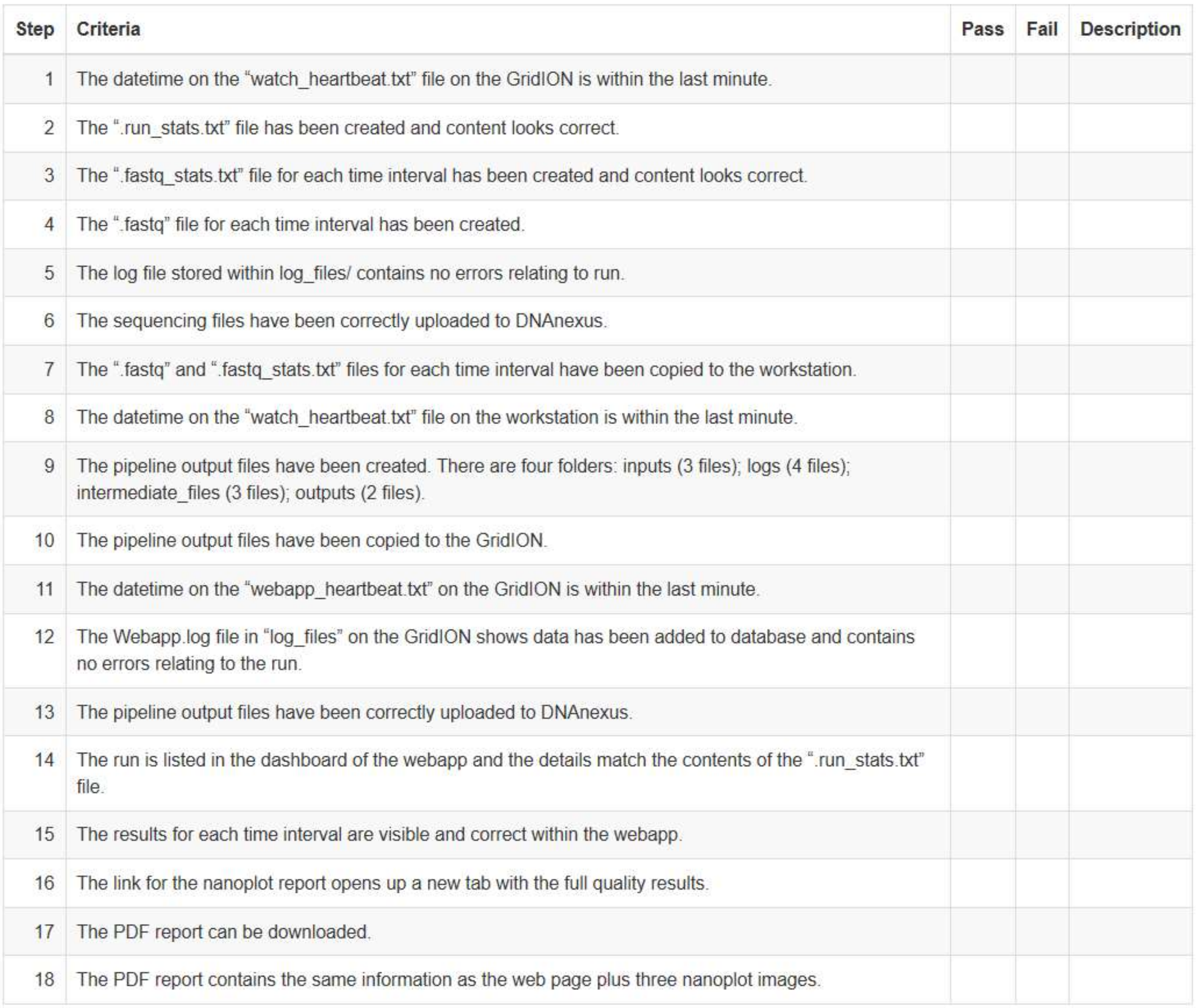
1. Two "No"'s from same person
2. One person did not know how to sort or filter results but commented that is not needed. All other tasks could be completed.
3. "Absolutely" and "Yes".
4. Only request was for a change of font.

User testing deemed successful, although would be useful to obtain more user results.

3. Workflow testing

3.1. Testing procedure

A dummy run was retrospectively put through the workflow (on 10.161.19.213) and tested against the below checklist. Any errors found were corrected before re-running the dummy run until all checks passed. Following this, a full validation run was sequenced allowing the workflow to be checked live (on 10.161.19.249).



Step	Criteria	Pass	Fail	Description
1	The datetime on the “watch_heartbeat.txt” file on the GridION is within the last minute.			
2	The “.run_stats.txt” file has been created and content looks correct.			
3	The “.fastq_stats.txt” file for each time interval has been created and content looks correct.			
4	The “.fastq” file for each time interval has been created.			
5	The log file stored within log_files/ contains no errors relating to run.			
6	The sequencing files have been correctly uploaded to DNAnexus.			
7	The “.fastq” and “.fastq_stats.txt” files for each time interval have been copied to the workstation.			
8	The datetime on the “watch_heartbeat.txt” file on the workstation is within the last minute.			
9	The pipeline output files have been created. There are four folders: inputs (3 files); logs (4 files); intermediate_files (3 files); outputs (2 files).			
10	The pipeline output files have been copied to the GridION.			
11	The datetime on the “webapp_heartbeat.txt” on the GridION is within the last minute.			
12	The Webapp.log file in “log_files” on the GridION shows data has been added to database and contains no errors relating to the run.			
13	The pipeline output files have been correctly uploaded to DNAnexus.			
14	The run is listed in the dashboard of the webapp and the details match the contents of the “.run_stats.txt” file.			
15	The results for each time interval are visible and correct within the webapp.			
16	The link for the nanoplot report opens up a new tab with the full quality results.			
17	The PDF report can be downloaded.			
18	The PDF report contains the same information as the web page plus three nanoplot images.			

Table 1: Workflow testing criteria

3.2. Testing results

Workflow testing identified several issues across both the dummy and validation runs. There was an additional issue identified with the first live run which has also been added to the table.

Run	Step	Issue	Cause	Fix
Dummy1	2	Incorrect run name in ".run_stats.txt" file.	This occurs when pulling out quality information from the "Flongle_Stats_Quality.ods" file. To overcome typing issues, I was converting each run to alphabetical order but this would make FGD116 and FGD161 the same.	Removed function and replaced with a sub of "FDG" for "FGD" as this is the most common typing issue.
Dummy1	6	FAST5 files not uploaded to DNAnexus.	I had hard-coded the DNAnexus folder name in the gridion_watch.py script so it was uploading files but to the wrong location.	Removed hard-coding and replaced with the correct config parameter.
Dummy2	5	DNAnexus upload error for sequencing log_files directory.	The log files are saved to a folder called "log_files" and compressed before uploading. This wasn't deleted prior to repeating the run which created a nested folder and caused the error.	Shouldn't affect future runs as files are only uploaded once – no action required.
Dummy3	2	Incorrect start time in ".run_stats.txt" file.	Regex incorrect when parsing fast5 file; greedy so was not matching to first time.	Added a "?" into the regex to force the regex to be non-greedy.
Dummy3	2	Sample details missing from ".run_stats.txt" file.	Script to populate file only runs once and this is before the data gets inputted into the spreadsheet (doesn't get populated until after sequencing starts).	Changed GridION watch script to repeatedly update and overwrite file.
Val	7	Files not copied to workstation.	Communication between new GridION and workstation not established.	Performed an "ssh" on each machine to the other to establish the connection and set each up as a known host.
Val	12	Sample details missing from ".run_stats.txt" file.	Quality information was not completed in spreadsheet.	Added a workaround to still allow parsing even if information is not present.
Val	12	ValueError: time data does not match format.	When calculating sequencing start time, had previously changed the format of the timestamp going into the file, but didn't change the reading of it.	Changed the reading of the timestamp to the correct format.
Live	7	Mean read length of zero.	None of the 91 reads could be found in the fastq files at 1 hr, suspect this is to do with basecaller holding reads in memory and not saving to file until reached 100 reads.	Set reads per file to 1.

Table 2: Workflow testing errors

Additional general issues observed while performing the testing:

- GridION watch script continues to create ".run_stats.txt" file even after analysis is complete. This overwriting is needed during the analysis as the quality information does not get completed until after sequencing starts, but this code was moved to only run if the analysis is not complete.
- There was a database locked error in the webapp log files. This is error is caused by multiple instances trying to save to the database at any one time which is not possible with a SQLite database. So I added a file indicator to the script to ensure only one instance of the parsing script is run at once.