Code listings and tables accompanying the Journal of Open Source Software paper

Crosbie, N.D. (2025) grepq: A Rust application that quickly filters FASTQ files by matching sequences to a set of regular expressions

Code listing 1

```
# For each matched pattern in a search of no more than
# 20000 matches of a gzip-compressed FASTQ file, print
# the pattern and the number of matches to a JSON file
# called matches.json, and include the top three most
# frequent variants of each pattern, and their respective
# counts
grepq --read-gzip 16S-no-iupac.json SRX26365298.fastq.gz \
 tune -n 20000 -c --names --json-matches --variants 3
```

Output (abridged) written to matches.json:

```
{
    "regexSet": {
        "regex": [
            {
                "regexCount": 2,
                "regexName": "Primer contig 06a",
                "regexString": "[AG]AAT[AT]G[AG]CGGGG",
                "variants": [
                    {
                        "count": 1,
                        "variant": "GAATTGGCGGGG",
                        "variantName": "06a-v3"
                    },
                    {
                        "count": 1,
                        "variant": "GAATTGACGGGG",
                        "variantName": "06a-v1"
                    }
                ]
            },
            // matches for other regular expressions...
    ],
    "regexSetName": "conserved 16S rRNA regions"
  }
}
```

Code listing 2

```
# For each matched pattern in a search of no more than
# 20000 matches of a gzip-compressed FASTQ file, print
# the pattern and the number of matches to a JSON file
# called matches.json, and include all variants of each
# pattern, and their respective counts. Note that the
# --variants argument is not given when --all is specified.
grepq --read-gzip 16S-no-iupac.json SRX26365298.fastq.gz \
 tune -n 20000 -c --names --json-matches --all
```

Tables

Table 1: Wall times and speedup of various tools for filtering FASTQ records against a set of regular expressions. Test FASTQ file: SRX26365298.fastq (uncompressed) was 874MB in size, and contained 869,034 records. *grepq* v1.4.1, *fqgrep* v.1.02, *ripgrep* v14.1.1, *seqkit grep* v.2.9.0, *grep* 2.6.0-FreeBSD, *awk* v. 20200816, and *gawk* v.5.3.1. *fqgrep* and *seqkit grep* were run with default settings, *ripgrep* was run with **-B 1 -A 2 --colors 'match:none' --no-line-number**, and *grep* was run with **-B 1 -A 2 --color=never**. *awk* and *gawk* scripts were also configured to output matching records in FASTQ format. The pattern file contained 30 regular expression representing the 12-mers (and their reverse compliment) from Table 3 of Martinez-Porchas et al. (2017). The wall times, given in seconds, are the mean of 10 runs, and S.D. is the standard deviation of the wall times, also given in seconds.

tool	mean wall time (s)	S.D. wall time (s)	speedup (× grep)	speedup (× ripgrep)	speedup (× awk)
grepq	0.19	0.01	1796.76	18.62	863.52
fqgrep	0.34	0.01	1017.61	10.55	489.07
ripgrep	3.57	0.01	96.49	1.00	46.37
seqkit grep	2.89	0.01	119.33	1.24	57.35
grep	344.26	0.55	1.00	0.01	0.48
awk	165.45	1.59	2.08	0.02	1.00
gawk	287.66	1.68	1.20	0.01	0.58

Table 2: Wall times and speedup of various tools for filtering gzip-compressed FASTQ records against a set of regular expressions. Test FASTQ file: SRX26365298.fastq.gz was 266MB in size, and contained 869,034 records. Test conditions and tool versions as above, but *grepq* was run with the **--read-gzip** option, *fqgrep* with the **-Z** option, and *ripgrep* with the **-z** option. SRX26365298.fastq was gzip-compressed using the *gzip* v.448.0.3 command (Apple Inc. 2019) using default (level 6) settings. The pattern file contained 30 regular expression representing the 12-mers (and their reverse compliment) from Table 3 of Martinez-Porchas et al. (2017). The wall times, given in seconds, are the mean of 10 runs, and S.D. is the standard deviation of the wall times, also given in seconds.

tool	mean wall time (s)	S.D. wall time (s)	speedup (× ripgrep)
grepq	1.703	0.002	2.10
fqgrep	1.834	0.005	1.95
ripgrep	3.584	0.013	1.00

Table 3: Wall times and speedup of various tools for filtering FASTQ records against a set of regular expressions. Test FASTQ file: SRX22685872.fastq was 104GB in size, and contained 139,700,067 records. Test conditions and tool versions as described in the footnote to Table 1. Note that when *grepq* was run on the gzip-compressed file, a memory resident time for the *grepq* process of 116M as reported by the *top* command (Apple Inc. 2023). *fastq-dump* v3.1.1 (Sherry et al. 2012) was used to download SRX22685872 as a gzip compressed file from the NCBI SRA. The pattern file contained 30 regular expression representing the 12-mers (and their reverse compliment) from Table 3 of Martinez-Porchas et al. (2017). The wall times, given in seconds, are the mean of 10 runs, and S.D. is the standard deviation of the wall times, also given in seconds.

tool	mean wall time (s)	S.D. wall time (s)	speedup (× ripgrep)
uncompressed			
grepq	26.972	0.244	4.41
fqgrep	50.525	0.501	2.36
ripgrep	119.047	1.227	1.00
gzip-compressed			
grepq	149.172	1.054	0.98
fqgrep	169.537	0.934	0.86

tool	mean wall time (s)	S.D. wall time (s)	speedup (× ripgrep)
ripgrep	144.333	0.243	1.00

References

Apple Inc. 2019. *The Gzip Command*. https://ss64.com/osx/gzip.html.

———. 2023. *The Top Command*. https://ss64.com/osx/top.html.

Martinez-Porchas, Marcel, Enrique Villalpando-Canchola, Luis Enrique Ortiz Suarez, and Francisco Vargas-Albores. 2017. “How Conserved Are the Conserved 16S-rRNA Regions?” *PeerJ* 5: e3036. https://doi.org/10.7717/peerj.3036.

Sherry, Stephen, Chunlin Xiao, Kenneth Durbrow, Michael Kimelman, Kurt Rodarmer, Martin Shumway, and Eugene Yaschenko. 2012. “NCBI Sra Toolkit Technology for Next Generation Sequence Data.” In *Plant and Animal Genome XX Conference (January 14-18, 2012). Plant and Animal Genome*.