Published online 16 December 2009 *Nucleic Acids Research, 2010, Vol. 38, No. 6* ***1767–1771***
doi:10.1093/nar/gkp1137

SURVEY AND SUMMARY

The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants

Peter J. A. Cock[1,*], Christopher J. Fields[2], Naohisa Goto[3], Michael L. Heuer[4] and Peter M. Rice[5]

[1]Plant Pathology, SCRI, Invergowrie, Dundee DD2 5DA, UK, [2]Institute for Genomic Biology, 1206 W. Gregory Drive, M/C 195, University of Illinois at Urbana-Champaign, IL 61801, USA, [3]Genome Information Research Center, Research Institute for Microbial Diseases, Osaka University, 3-1 Yamadaoka, Suita, Osaka 565-0871, Japan, [4]Harbinger Partners, Inc., 855 Village Center Drive, Suite 356, St. Paul, MN 55127, USA and [5]EMBL Outstation - Hinxton, European Bioinformatics Institute, Wellcome Trust Genome Campus, Hinxton, Cambridge CB10 1SD, UK

Received October 13, 2009; Revised November 13, 2009; Accepted November 17, 2009

ABSTRACT

FASTQ has emerged as a common file format for sharing sequencing read data combining both the sequence and an associated per base quality score, despite lacking any formal definition to date, and existing in at least three incompatible variants. This article defines the FASTQ format, covering the original Sanger standard, the Solexa/Illumina variants and conversion between them, based on publicly available information such as the MAQ documentation and conventions recently agreed by the Open Bioinformatics Foundation projects Biopython, BioPerl, BioRuby, BioJava and EMBOSS. Being an open access publication, it is hoped that this description, with the example files provided as Supplementary Data, will serve in future as a reference for this important file format.

INTRODUCTION

One of the core issues of Bioinformatics is dealing with a profusion of (often poorly defined or ambiguous) file formats. Some *ad hoc* simple human readable formats have over time attained the status of *de facto* standards. A ubiquitous example of this is the 'FASTA sequence file format', originally invented by Bill Pearson as an input format for his FASTA suite of tools (1). Over time, this format has evolved by consensus; however, in the absence of an explicit standard some parsers will fail to cope with very long '>' title lines or very long sequences without line wrapping. There is also no standardization for record identifiers.

In the area of DNA sequencing, the FASTQ file format has emerged as another *de facto* common format for data exchange between tools. It provides a simple extension to the FASTA format: the ability to store a numeric quality score associated with each nucleotide in a sequence. This is a very minimal representation of a sequencing read—nothing about the relative levels of the four nucleotides is captured [e.g. from Sanger capillary sequencing or Solexa/Illumina sequencing (2)] nor did this in any way attempt to deal with flow or colour space data [e.g. Roche 454 (3) or ABI SOLiD (4)].

No doubt because of its simplicity, the FASTQ format has become widely used as a simple interchange file format. Unfortunately, history has repeated itself, and the FASTQ format suffers from the absence of a clear definition (which we hope this manuscript will address), and several incompatible variants.

We discuss the history of the FASTQ format, describing key variants, and conventions adopted by the Open Bioinformatics Foundation (OBF, http://www.open-bio.org) projects Biopython (5), BioPerl (6), BioRuby (http://www.bioruby.org), BioJava (7), and EMBOSS (8) (each represented here by an author) for reading, writing and converting between them. This is intended to provide a public, open access and citable definition of this community consensus of the FASTQ format specification.

*To whom correspondence should be addressed. Tel: +44 1382 562731; Fax: +44 1382 562426; Email: peter.cock@scri.ac.uk

© The Author(s) 2009. Published by Oxford University Press.
This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (http://creativecommons.org/licenses/by-nc/2.5), which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

PHRED SCORES AND THE QUAL FORMAT

The PHRED software reads DNA sequencing trace files, calls bases and assigns a quality value to each base called (9,10). This introduced the PHRED quality score of a base call, defined in terms of the estimated probability of error:

$$Q_{\mathrm{PHRED}} = -10 \times \log_{10}(P_e) \quad \mathbf{1}$$

PHRED also introduced a new file format, known as the QUAL format, after the default file extension, to hold these quality scores. These are FASTA like, holding PHRED scores as space separated plain text integers and supplement a corresponding FASTA file with the associated sequences. For example, here is a single read from the NCBI sequence read archive (SRA, http://www.ncbi.nlm.nih.gov/Traces/sra/sra.cgih) presented as a FASTA entry:

```
>SRR014849.1 EIXKN4201CFU84 length=93
GGGGGGGGGGGGGGGGCTTTTTTTGTTTGGAACCGAAAGG
GTTTTGAATTTCAAACCCTTTTCGGTTTCCAACCTTCCAA
AGCAATGCCAATA
```

and as a QUAL entry holding the PHRED scores:

```
>SRR014849.1 EIXKN4201CFU84 length=93
18 10 5 3 2 1 1 1 1 1 1 1 1 1 1 1 22 37
31 22 16 11 6 1 26 34 30 11 33 26 30 21
33 26 25 36 32 16 36 32 16 36 32 20 6
24 33 25 30 25 2 24 36 32 15 35 31 17
36 32 20 6 25 29 20 30 25 4 32 26 32 23
32 26 30 24 33 26 35 31 14 28 27 30 22
28 24 27 17 32 23 28 28
```

PHRED scores are now a *de facto* standard for representing sequencing read base qualities. For example, the Roche 454 'off instrument' applications allow conversion from a binary Standard Flowgram Format (SFF) file to FASTA and QUAL files. PHRED scores are also used in SAM (Sequence Alignment/Map, http://samtools.sourceforge.net/), Staden Experiment (11), ACE (12), and FASTQ files.

SANGER FASTQ FORMAT

The FASTQ format was invented at the turn of the century at the Wellcome Trust Sanger Institute by Jim Mullikin, gradually disseminated, but never formally documented (Antony V. Cox, Sanger Institute, personal communication 2009). The closest thing to an official description from Sanger can be found on the MAQ/BWA website (13,14), but even this is incomplete.

Full details of the file format, describing the read title, sequence and quality scores are given later. Here, we concentrate on how the quality scores were encoded into a simple string. Early FASTQ files were used for Sanger capillary sequencing, and it was natural to use PHRED quality scores (described above).

Storing PHRED scores as single characters (or bytes) gave a simple but reasonably space efficient encoding. In order that the file be human readable and easily edited, this restricted the choices to the ASCII printable characters 32–126 (decimal), and since ASCII 32 is the space character, Sanger FASTQ files use ASCII 33–126 to encode PHRED qualities from 0 to 93 (i.e. PHRED scores with an ASCII offset of 33).

This gives a very broad range of error probabilities, from 1.0 (a wrong base) through to $10^{-9.3}$ (an extremely accurate read) and so the Sanger FASTQ format is useful both for raw sequencing reads and post-processed assemblies where higher qualities occur.

The OBF projects refer to this, the original or standard FASTQ format, as the Sanger variant, using the format name '`fastq-sanger`' (Table 1).

SOLEXA FASTQ FORMAT

In 2004, Solexa, Inc. introduced their own incompatible (and indistinguishable) version of the FASTQ format (2). Although the FASTQ format only records a single quality score per letter, Solexa also produced other files with quality scores for all four bases, and in order to represent low-quality information more fully an alternative logarithmic mapping was used (15). Solexa quality scores are defined as:

$$Q_{\mathrm{Solexa}} = -10 \times \log_{10}\left(\frac{P_e}{1-P_e}\right) \quad \mathbf{2}$$

Although different sequencing systems estimate their error rates using different methodologies, simply rearranging these two equations and equating the error estimates allows a straightforward mapping between the two. This conversion has gained widespread usage through MAQ (13).

$$Q_{\mathrm{PHRED}} = 10 \times \log_{10}\left(10^{Q_{\mathrm{Solexa}}/10} + 1\right) \quad \mathbf{3}$$

$$Q_{\mathrm{Solexa}} = 10 \times \log_{10}\left(10^{Q_{\mathrm{PHRED}}/10} - 1\right) \quad \mathbf{4}$$

An important consequence of these equations is for high values the two scores are asymptotically equal, and after rounding to the nearest integer scores of ≥ 10 are interchangeable (Figure 1). However, Solexa scores go down to -5 (approximating a random read error probability of 0.75). The Sanger offset of 33 can, therefore, no longer be used. Rather, an offset of 64 was chosen, meaning ASCII 59 to 126 can be used, allowing Solexa scores from -5 to 62 inclusive.

Table 1. The three described FASTQ variants, with columns giving the description, format name used in OBF projects, range of ASCII characters permitted in the quality string (in decimal notation), ASCII encoding offset, type of quality score encoded and the possible range of scores

Description, OBF name	ASCII characters		Quality score	
	Range	Offset	Type	Range
Sanger standard `fastq-sanger`	33–126	33	PHRED	0 to 93
Solexa/early Illumina `fastq-solexa`	59–126	64	Solexa	−5 to 62
Illumina 1.3+ `fastq-illumina`	64–126	64	PHRED	0 to 62

In 2006, Solexa, Inc., was acquired by Illumina, Inc., which continued to use this FASTQ variant. The OBF projects (and others, such as MAQ) refer to this as the Solexa FASTQ variant, format name '`fastq-solexa`' (Table 1).

ILLUMINA 1.3+ FASTQ FORMAT

Although Illumina initially continued to use the Solexa FASTQ variant, from Genome Analyzer Pipeline version 1.3 onwards (16), PHRED quality scores rather than Solexa scores were used. However, rather than adopt the original Sanger format, Illumina introduced a third incompatible FASTQ variant designed to be interchangeable with their earlier 'Solexa FASTQ' files for good quality reads.

The Illumina 1.3+ FASTQ variant encodes PHRED scores with an ASCII offset of 64, and so can hold PHRED scores from 0 to 62 (ASCII 64–126), although currently raw Illumina data quality scores are only expected in the range 0–40.

The OBF projects refer to this variant as the Illumina 1.3+ FASTQ format, under the format name '`fastq-illumina`' (Table 1).

ABI SOLID COLOUR SPACE FASTQ

ABI SOLiD sequencing works in colour space not sequence space (4), leading ABI to introduce Color Space FASTA (CSFASTA) files with matching QUAL files, and also Color Space FASTQ (CSFASTQ) files. These use the digits 0–3 to encode the colour calls (base transitions), but are not considered herein where we focus solely on sequence space FASTQ files.

FASTQ DEFINITION

Here is a Sanger FASTQ read from the NCBI SRA (shown earlier in the FASTA and QUAL formats):

```
@SRR014849.1 EIXKN4201CFU84 length=93
GGGGGGGGGGGGGGGGCTTTTTTTGTTTGGAACCGAAAGG
GTTTTGAATTTCAAACCCTTTTCGGTTTCCAACCTTCCAA
AGCAATGCCAATA
+SRR014849.1 EIXKN4201CFU84 length=93
3+&$#""""""""""""7F@71,'";C?,B;?6B;:EA1EA
1EA5'9B:?:#9EA0D@2EA5':>5?:%A;A8A;?9B;D@
/=<?7=9<2A8==
```

@title and optional description
sequence line(s)
+optional repeat of title line
quality line(s)

There are four line types in the FASTQ format. First a '@' title line which often holds just a record identifier. This is a free format field with no length limit—allowing arbitrary annotation or comments to be included, as in the example above where the NCBI have included an alternative ID and the sequence length. Some sequencing centers encode paired end read information here (alternatively two matched FASTQ files are often used).

Second comes the sequence line(s), which as in the FASTA format can be line wrapped. Also like FASTA format, there is no explicit limitation on the characters expected, but restriction to the IUPAC single letter codes for (ambiguous) DNA or RNA is wise, and upper case is conventional. In some contexts, the use of lower or mixed case or the inclusion of a gap character may make sense. White space such as tabs or spaces is not permitted.

Third, to signal the end of the sequence lines and the start of the quality string, comes the '+' line. Originally this also included a full repeat of the title line text (as shown in the NCBI example above); however, by common usage and the MAQ tool convention, this is optional and the '+' line can contain just this one character, reducing the file size significantly. The OBF tools follow this MAQ convention on output, and omit the optional repeated title text.

Finally, comes quality line(s) which again can be wrapped. As discussed above, these use a subset of the ASCII printable characters (at most ASCII 33–126 inclusive) with a simple offset mapping. Crucially, after concatenation (removing line breaks), the quality string must be equal in length to the sequence string.

It is vital to note that the '@' marker character (ASCII 64) may occur anywhere in the quality string—including at the start of any of the quality lines. This means that any parser must *not* treat a line starting with '@' as indicating the start of the next record, without additionally checking the length of the quality string thus far matches the length of the sequence.

Because of this complication, most tools output FASTQ files *without* line wrapping of the sequence and quality string. This means each read consists of exactly four lines (sometimes very long lines), ideal for a very simple parser to deal with. The OBF tools follow this convention on output, as does the MAQ conversion script. We recommend this for maximum compatibility with (simplistic) parsers.

Because FASTQ files (like FASTA files) are plain text, the new line characters will normally follow the operating system convention. However, as data are shared between machines, any parser should cope with both Unix style new lines (line feed only, ASCII 10) and DOS/Windows style (carriage return and line feed, ASCII 13 then 10).

CONVERTING FASTQ FILES

Conversion from '`fastq-illumina`' to '`fastq-sanger`' will be a common operation, and is very straightforward since both variants use PHRED scores but with different offsets. All that is required is to decrease the quality character codes by 31. The opposite conversion is unlikely to be required, but in this situation the '`fastq-illumina`' format can only hold PHRED scores from 0–62, compared with 0–93 in '`fastq-sanger`'. The OBF projects will all apply 62 as a maximum PHRED score (giving ASCII 126) with a warning message for values outside of this range.

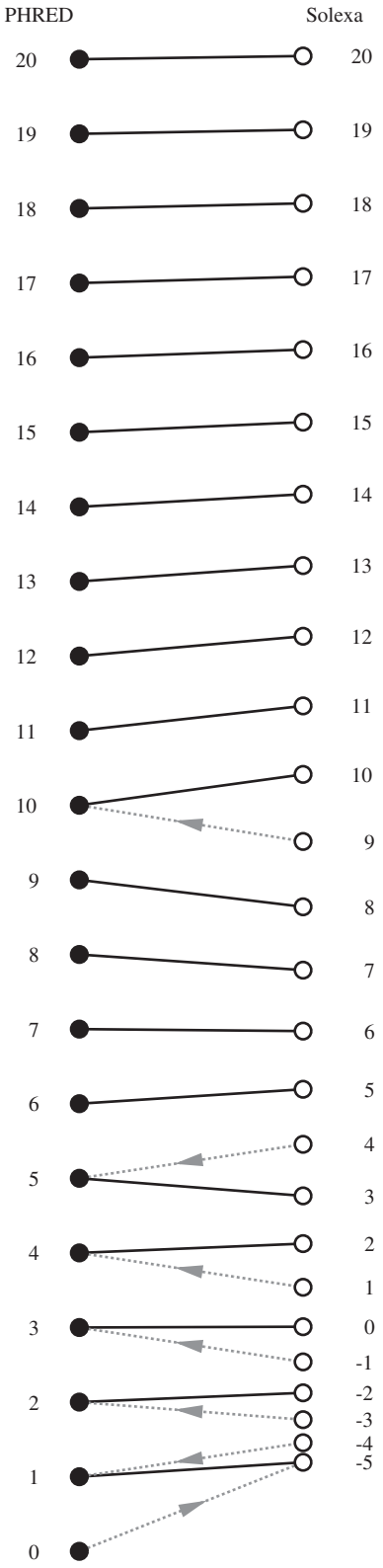

Figure 1. Visual representation of the mapping between PHRED and Solexa quality scores. Vertical layout represents the probability of error on a log scale, therefore the PHRED points are equally spaced (black circles on left), while the Solexa points are not (white circles on right). Solid black lines are reciprocal mappings between scores, and grey arrows are lossy mappings. Near the top of the figure, the black lines are almost horizontal because the two scores are almost equal. The straightforward mapping of higher scores is omitted due to space.

Conversion from '`fastq-solexa`' to '`fastq-sanger`' (or '`fastq-illumina`') requires conversion of Solexa scores to PHRED scores using Equation (3) and rounding to the nearest integer. This mapping is lossy for poor quality reads, for example Solexa scores 9 and 10 both give PHRED score 10 (Figure 1). The reverse conversion uses Equation (4) instead. Taken literally, this maps PHRED 0 to Solexa $-\infty$, but the minimum Solexa score is taken, -5 (corresponding to a random base call). Thus, both PHRED 0 and 1 map to Solexa -5 (Figure 1). A maximum limit of Solexa score 62 applies (giving ASCII 126).

Biopython (version 1.51 or later), BioPerl (version 1.6.1 or later), BioRuby (version 1.4.0 or later), BioJava (verion 1.7.1 or later) and the seqret tool from EMBOSS (version 6.1.0 patch 1 or later) are all able to inter-convert between any of the three FASTQ variants (Table 1).

TEST CASES

Two classes of example files are provided as Supplementary Data. First, a number of invalid files which any parser should reject, including truncated reads, examples where the sequence and quality lengths differ, and invalid ASCII characters in the quality lines. Secondly, a set of valid but challenging FASTQ files together with a standardized version of the same data, plus how that file should be converted to other FASTQ variants. These examples are used in the OBF unit tests.

DISCUSSION

The original Sanger FASTQ format was by no means perfect. The '@' and '+' characters have dual usage as line markers or anywhere within the quality string. Simple indexing of the file looking for lines starting with '@' is therefore not possible.

The lack of ownership of this emerging standard by the Sanger Institute contributed greatly to later confusion, which can mostly be attributed to Solexa/Illumina, who not once but twice have invented their own 'FASTQ' format. With hindsight, we may ask why Solexa used their own scoring system for FASTQ output, given Illumina have since switched to the PHRED convention. Furthermore, as part of this switch for GAPipeline 1.3, Illumina could have adopted the original Sanger format. This would have still caused disruption in the short term, but would have unified the FASTQ format. While the Illumina 1.3+ FASTQ variant is interchangeable with the earlier Solexa FASTQ format for *good* quality reads, as a result of these choices, we now have three incompatible FASTQ variants that cannot be reliably distinguished. A simple measure such as the inclusion of header lines like '`#Solexa FASTQ 1.0`' or '`#Illumina FASTQ 1.3`' would have imposed a trivial overhead on the file size and allowed automatic determination of the file format and thus the quality encoding.

Currently, the onus is on the bioinformatician to determine provenance, which now requires finding out which *version* of the Solexa/Illumina pipeline was used! Even

reading the literature can be confusing, for example Huang *et al.* (17) wrote '*...using Illumina GA processing pipeline V0.2.2.6...MAQ was used to convert Illumina FASTQ to Sanger standard FASTQ format*'. At the time of writing, MAQ does not convert 'fastq-illumina' to 'fastq-sanger' format, so this group could have potentially mis-converted their data. However, as the Illumina pipeline version is given, we can infer they actually started with what we have christened the 'fastq-solexa' FASTQ format, and there is no problem.

Despite this confusion, the Sanger version of the FASTQ format has found the broadest acceptance, supported by many assembly and read mapping tools—for example SSAHA2 (18), MAQ (13), Velvet (19), BWA (14) and BowTie (20). Although some of these tools can convert from the Solexa (and in some cases also the Illumina 1.3+) FASTQ variant, support for the standard Sanger FASTQ files is most common. Therefore, most users will do this conversion very early in their workflows (perhaps using OBF software). We also note that the NCBI SRA makes all its data available as standard Sanger FASTQ files (even if originally from a Solexa/Illumina machine). We hope that this trend will lead to Illumina themselves switching to the original FASTQ convention at a later date, which would eventually relegate this confusion of incompatible variants to a historical concern. A further suggestion is for Roche to extend their SFF tools to produce Sanger FASTQ files in addition to the existing options of FASTA and QUAL files.

In addition to simple conversion between FASTQ variants, other common steps in a sequencing pipeline include quality and adaptor trimming, and contaminant or quality-based filtering. A set of interchangeable tools like the OBF projects, based on a common FASTQ standard, will be of great benefit here.

SUPPLEMENTARY DATA

Supplementary Data are available at NAR Online.

FUNDING

BBSRC (grants BBR/G02264X/1 and BB/D018358/1 to EMBOSS), Scottish Government Rural and Environment Research and Analysis Directorate, UK (to P.J.A.C.), Funding for open access charge: P.M.R.'s EBI group budget.

Conflict of interest statement. None declared.

REFERENCES

1. Pearson,W.R. and Lipman,D.J. (1988) Improved tools for biological sequence comparison. *Proc. Natl Acad. Sci. USA*, **85**, 2444–2448.
2. Bennett,S. (2004) Solexa Ltd. *Pharmacogenomics*, **5**, 433–438.
3. Margulies,M., Egholm,M., Altman,W.E., Attiya,S., Bader,J.S., Bemben,L.A., Berka,J., Braverman,M.S., Chen,J.Y., Chen,Z. *et al.* (2005) Genome sequencing in microfabricated high-density picolitre reactors. *Nature*, **437**, 376–380.
4. Pandey,V., Nutter,R.C. and Prediger,E. (2008) Applied Biosystems SOLiD system: ligation-based sequencing. In Janitz,M. (ed.), *Next Generation Genome Sequencing: Towards Personalized Medicine*. Wiley, pp. 29–41.
5. Cock,P.J.A., Antao,T., Chang,J.T., Chapman,B.A., Cox,C.J., Dalke,A., Friedberg,I., Hamelryck,T., Kauff,F., Wilczynski,B. *et al.* (2009) Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics*, **25**, 1422–1423.
6. Stajich,J.E., Block,D., Boulez,K., Brenner,S.E., Chervitz,S.A., Dagdigian,C., Fuellen,G., Gilbert,J.G., Korf,I., Lapp,H. *et al.* (2002) The BioPerl toolkit: Perl modules for the life sciences. *Genome Res.*, **12**, 1611–1618.
7. Holland,R.C.G., Down,T.A., Pocock,M., Prlic,A., Huen,D., James,K., Foisy,S., Drager,A., Yates,A., Heuer,M. *et al.* (2008) BioJava: an open-source framework for bioinformatics. *Bioinformatics*, **24**, 2096–2097.
8. Rice,P., Longden,I. and Bleasby,A. (2000) EMBOSS: the European Molecular Biology Open Software Suite. *Trends Genet.*, **16**, 276–277.
9. Ewing,B., Hillier,L., Wendl,M.C. and Green,P. (1998) Base-calling of automated sequencer traces using Phred. I. Accuracy Assessment. *Genome Res.*, **8**, 175–185.
10. Ewing,B. and Green,P. (1998) Base-calling of automated sequencer traces using Phred. II. Error probabilities. *Genome Res.*, **8**, 186–194.
11. Bonfield,J.K. and Staden,R. (1996) Experiment files and their application during large-scale sequencing projects. *DNA Seq.*, **6**, 109–117.
12. Gordon,D., Abajian,C. and Green,P. (1998) Consed: a graphical tool for sequence finishing. *Genome Res.*, **8**, 195–202.
13. Li,H., Ruan,J. and Durbin,R. (2008) Mapping short DNA sequencing reads and calling variants using mapping quality scores. *Genome Res.*, **18**, 1851–1858.
14. Li,H. and Durbin,R. (2009) Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics*, **25**, 1754–1760.
15. Bentley,D.R., Balasubramanian,S., Swerdlow,H.P., Smith,G.P., Milton,J., Brown,C.G., Hall,K.P., Evers,D.J., Barnes,C.L., Bignell,H.R. *et al.* (2008) Accurate whole human genome sequencing using reversible terminator chemistry. *Nature*, **456**, 53–59.
16. Illumina, Inc. (2008) Sequencing Analysis Software User Guide for Pipeline version 1.3 and CASAVA version 1.0 Illumina Inc, San Diego, CA, USA.
17. Huang,X., Feng,Q., Qian,Q., Zhao,Q., Wang,L., Wang,A., Guan,J., Fan,D., Weng,Q., Huang,T. *et al.* (2009) High-throughput genotyping by whole-genome resequencing. *Genome Res.*, **19**, 1068–1076.
18. Ning,Z., Cox,A.J. and Mullikin,J.C. (2001) SSAHA: a fast search method for large DNA databases. *Genome Res.*, **11**, 1725–1729.
19. Zerbino,D. and Birney,E. (2008) Velvet: algorithms for de novo short read assembly using de Bruijn graphs. *Genome Res.*, **18**, 821–829.
20. Langmead,B., Trapnell,C., Pop,M. and Salzberg,S. (2009) Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biol.*, **10**, R25.