A complete bacterial genome assembled *de novo* using only nanopore sequencing data

Nicholas J Loman[1], Joshua Quick[1] & Jared T Simpson[2,3]

We have assembled *de novo* the *Escherichia coli* K-12 MG1655 chromosome in a single 4.6-Mb contig using only nanopore data. Our method has three stages: (i) overlaps are detected between reads and then corrected by a multiple-alignment process; (ii) corrected reads are assembled using the Celera Assembler; and (iii) the assembly is polished using a probabilistic model of the signal-level data. The assembly reconstructs gene order and has 99.5% nucleotide identity.

The MinION (Oxford Nanopore Technologies) is a portable, single-molecule genome sequencing instrument no larger than a typical smartphone. As this instrument directly senses native, individual DNA fragments without the need for amplification, it is able to sequence extremely long fragments (>10 kb) of DNA without a reduction in sequence quality[1].

The availability of very long reads is important when assembling genomes because they span repetitive elements and anchor repeat copies within uniquely occurring parts of the genome. Many bacterial genomes can be assembled into single contigs if reads longer than 7 kb are available, as these reads span the conserved rRNA operon, which is typically the longest repeat sequence in a bacterial genome[2].

The accuracy of the sequence reads is another potentially limiting step for genome assembly; at launch, data from the Pacific Biosciences sequencing instrument were hard to use for *de novo* assembly owing to the lack of bioinformatics tools designed for long reads with a high error rate. Several algorithmic improvements have led to this platform becoming widely adopted for genome assembly; Koren *et al.* demonstrated that hybrid techniques could generate contiguous assemblies[3]. This method uses accurate short-read data and/or Pacific Biosciences circular consensus reads to correct error-prone long reads sufficiently for assembly[4]. Nonhybrid *de novo* assemblies of only Pacific Biosciences' Single Molecule, Real-Time read data in absence of short-read data remained an open problem until Chin *et al.* developed the HGAP (hierarchical genome-assembly process) assembler[5]. HGAP typically uses a subset of the longest reads in the sequencing data set as the input for the assembly process. This subset of long reads is corrected using the entire data set, and the corrected set of reads is assembled with the Celera Assembler. The final assembly is 'polished' to high accuracy using a signal-level consensus algorithm. This assembly strategy has revolutionized genome assembly of small genomes and is now being applied to large genomes[6,7].

Nanopore sequencing data have clear similarities to Pacific Biosciences data, with reads sufficiently long to be of great use in *de novo* assembly[7,8]. Recently, a hybrid approach that used Illumina short reads to correct nanopore reads before assembly was shown to give good results for a bacterial and a yeast genome[9]. However, nonhybrid assemblies have not yet been described. Recent versions of nanopore chemistry (R7.3) coupled with the latest base caller (Metrichor versions 1.9 and later) permit read-level accuracies of 78–85% (refs. 1,8). Although this is slightly lower than accuracies achieved by the latest version of Pacific Biosciences chemistry[6], we hypothesized that a similar assembly approach in which nanopore reads are self-corrected may result in highly contiguous assemblies.

Overlaps can be detected more easily with higher-accuracy data. Therefore, for this study we exclusively used high-quality 'passing filter' two-direction (2D) reads. DNA strands that have been read in both directions by the MinION and combined during base-calling are of higher quality than the individual template and complement strands[8]. All 2D reads from four separate MinION runs using R7.3 chemistry were combined. Each run was made with a freshly prepared sequencing library using library protocol SQK-MAP-003 (first run) or SQK-MAP-004 (three further runs). In total, 22,270 2D reads were used comprising 133.6 Mb of read data, representing ~29× theoretical coverage of the 4.6-Mb *E. coli* K-12 MG1655 reference genome (**Supplementary Table 1**).

The FASTA sequences for reads were extracted using Poretools[10]. Potential overlaps between the reads were detected using the DALIGNER software[11]. Each read and its overlapped reads were used as input to the partial-order alignment (POA) software[12]. POA uses a directed acyclic graph to compute a multiple alignment, which we use to determine a consensus sequence. The use of partial-order graphs permits a more sensitive reconstruction of consensus sequences in the presence of large numbers of insertions or deletions (indels). This step is analogous to the precorrection step of the HGAP pipeline, which also uses partial-order graphs (implemented in the pbdagcon software) to compute a consensus sequence. Our assembly pipeline runs this correction process multiple times, using the corrected reads as the new input. Error correction increased the mean percent identity from 80.5% to 95.9% after the first iteration and to 97.7% on the second iteration (**Supplementary Figs. 1** and **2**). Further rounds of correction resulted in a reduction in available reads without a substantial improvement in accuracy.

[1]Institute of Microbiology and Infection, University of Birmingham, Birmingham, UK. [2]Ontario Institute for Cancer Research, Toronto, Ontario, Canada. [3]Department of Computer Science, University of Toronto, Toronto, Ontario, Canada. Correspondence should be addressed to J.T.S. (jared.simpson@oicr.on.ca).

RECEIVED 11 MARCH; ACCEPTED 22 MAY; PUBLISHED ONLINE 15 JUNE 2015; DOI:10.1038/NMETH.3444

The reads resulting from two rounds of correction were used as input to version 8.2 of the Celera Assembler[13] (for settings see **Supplementary Fig. 3**). The critical parameter for tuning is the overlap minimum identity, which should be double the read error rate, in this case set to 4% (ovlErrorRate = 0.04). This resulted in a highly contiguous assembly with three contigs, the largest being 4.6 Mb long and covering the entire *E. coli* chromosome (**Fig. 1**). The other two contigs, with lengths 11 kb and 8 kb, were present in the long contig and so we did not include them in further analysis. Short leftover contigs are expected from overlap assembly.

The initial draft assembly had 3,726 mismatches (80 per 100 kb) and 42,752 indels of ≥1 base (921 errors per 100 kb), as determined by the QUAST validation tool when compared to the *E. coli* K-12 MG1655 reference genome (NC_000913.3)[14].

The electric current signal emitted by the MinION contains more information than the base-called reads. We implemented an algorithm that uses the electric current signal to compute an improved consensus sequence for the assembly. Our consensus algorithm iteratively proposes and evaluates modifications to the assembly. The evaluation step uses a hidden Markov model to calculate the probability of observing a sequence of current signals given an arbitrary nucleotide sequence. Full details of this algorithm are provided in the Online Methods and **Supplementary Note**. After the assembly was polished using this algorithm, the base-level accuracy improved to 99.5%, comprising 1,202 mismatches (26 per 100 kb) and 17,241 indels of ≥1 base (371 errors per 100 kb).

We explored the sequence context of errors in our assemblies before and after polishing by comparing the 5-mer counts in our assemblies

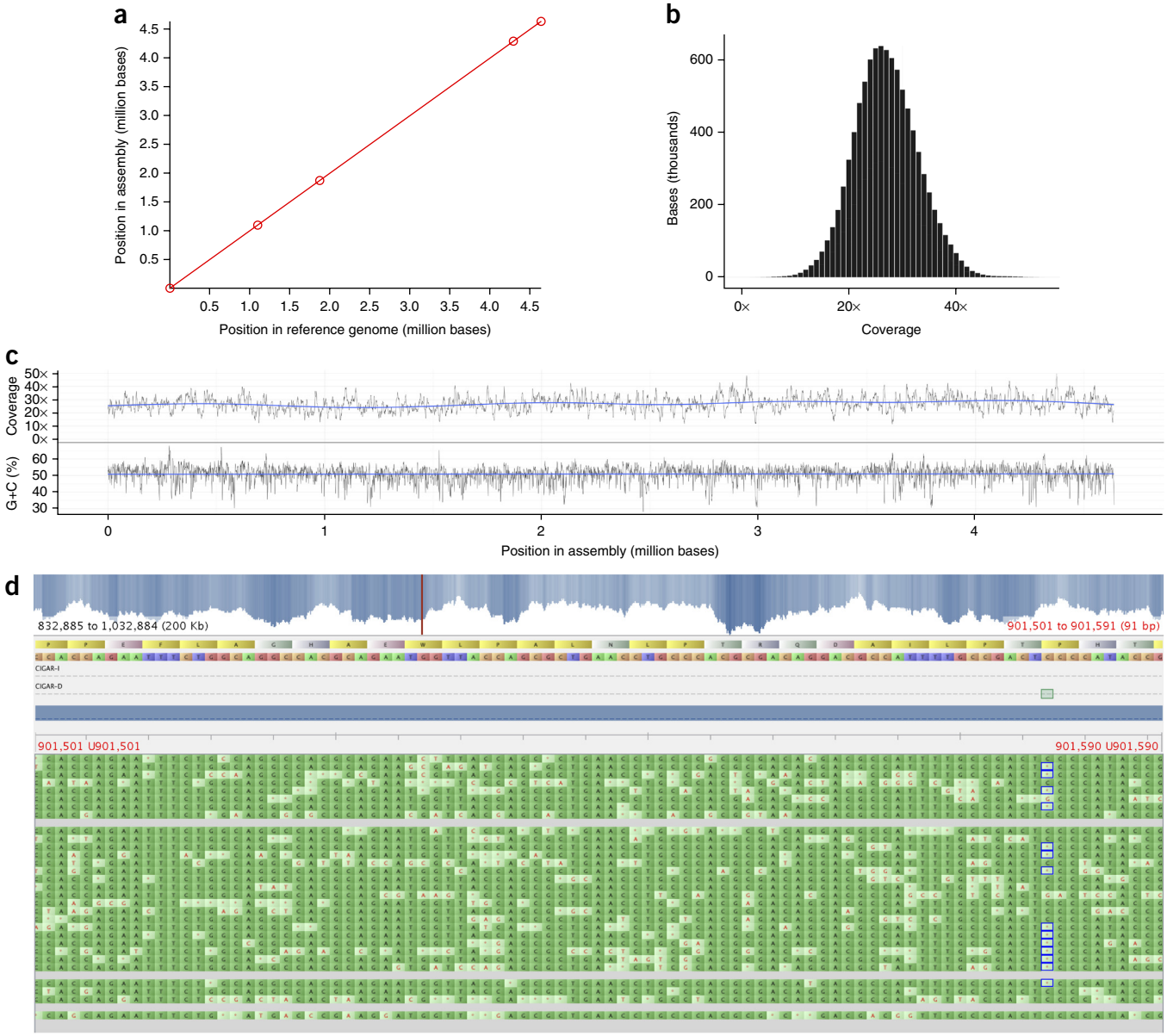

Figure 1 | Single-contig assembly of *E. coli* K-12 MG1655. (**a**) Dot plot comparing our assembly with the reference genome. (**b**) Histogram of read coverage when nanopore reads are aligned against the assembled genome. (**c**) Read coverage and G+C composition across the length of the assembled genome. (**d**) View in the genome-assembly tool Tablet[16] of the read coverage (top blue panel) for a randomly chosen section of the assembly genome, and the underlying reads used to construct the assembly.

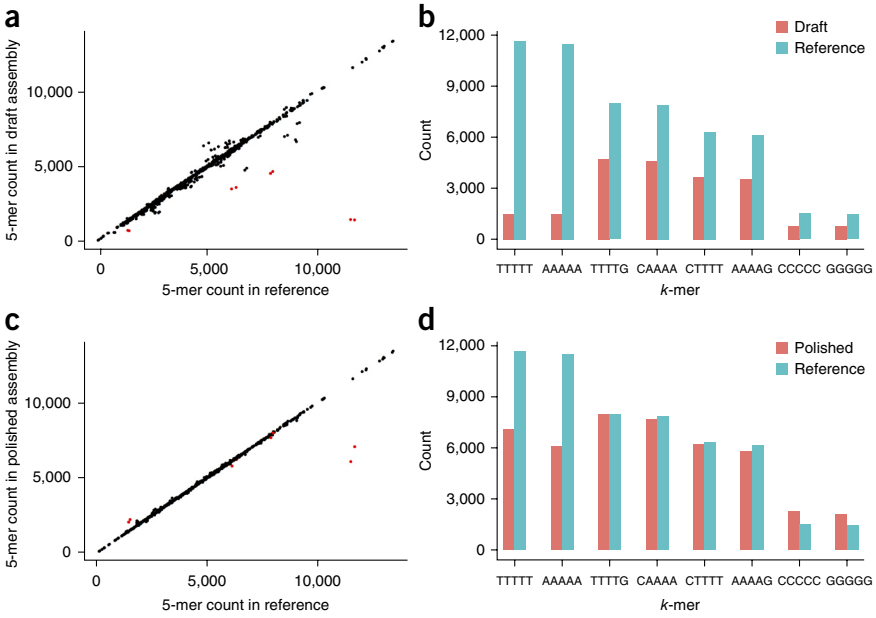

Figure 2 | Comparing 5-mer counts of the assembly and the reference genome before and after signal-level polishing. (**a**,**c**) Correlation between 5-mer counts in the reference (*x* axis) and an assembly (*y* axis) before (**a**) and after (**c**) signal-level polishing. Red dots in **a**,**c** denote 5-mers with ≥50% more occurrences in the reference genome than the unpolished assembly. (**b**,**d**) The counts for these 5-mers are compared to the reference for the unpolished assembly (**b**) and the polished assembly (**d**).

to those in the reference genome. Our initial draft assembly displayed substantial under-representation of 5-mers consisting of a single base (AAAAA, CCCCC, GGGGG or TTTTT) or containing a 4-base stretch of a single base (**Fig. 2a**,**b**). This error mode is expected as the MinION relies on a change in electric current to detect base-to-base transitions, which may not occur or may not be detectable as locally repetitive sequence transits the pore. After we polished the assembly, the representation of homopolymer tracts improved and the 5-mer counts between the reference genome and the polished assembly were well correlated (Pearson's $r = 0.99$; **Fig. 2c**), although AAAAA and TTTTT remained difficult to correctly identify (**Fig. 2d**).

QUAST reported two misassemblies (**Supplementary Table 2**) with respect to the reference sequence. On visual inspection of a whole-genome alignment generated by the Mauve software[15], we determined only one large (>500-base) region of difference between our assembly and the reference. This represented an insertion of a transposase-encoding region into our assembly. The sequence, approximately 750 bases long, was present nine times in the reference genome. Insertion events involving transposons are common in bacterial genomes and can affect wild-type strains commonly used in laboratory experiments, but such insertions are difficult to detect using draft assemblies from short-read sequencing technologies.

We conclude that long-read data from the Oxford Nanopore MinION can be used to assemble complete bacterial genomes to give an accurate reconstruction of gene order and orientation, without the need to use data from other sequencing technologies. Using signal-level algorithms, we were able to improve the per-base accuracy of our assembly further. However, homopolymer tracts remain error prone. The errors remaining in our assembly may complicate subsequent analysis—for example, gene prediction—and improvements are therefore needed. We expect that improved probabilistic models of the data, possibly incorporating information about fragment translocation duration, may result in higher accuracy. Additionally, we have not evaluated accuracy for sequencing coverage above 30×. The read error-correction software, Nanocorrect, is available at https://github.com/jts/nanocorrect/. The signal-level consensus software, Nanopolish, is available at https://github.com/jts/nanopolish/. Both programs are freely available under the open-source MIT license.

METHODS

Methods and any associated references are available in the online version of the paper.

Accession codes. European Nucleotide Archive: ERX708228, ERX708229, ERX708230, ERX708231.

Note: Any Supplementary Information and Source Data files are available in the online version of the paper.

ACKNOWLEDGMENTS

Data analysis was performed on the Medical Research Council Cloud Infrastructure for Microbial Bioinformatics (CLIMB) cyberinfrastructure. N.J.L. is funded by a Medical Research Council Special Training Fellowship in Biomedical Informatics. J.Q. is funded by the UK National Institute for Health Research (NIHR) Surgical Reconstruction and Microbiology Research Centre. J.T.S. is supported by the Ontario Institute for Cancer Research through funding provided by the Government of Ontario. We thank the staff of Oxford Nanopore for technical help and advice during the MinION Access Programme. We are grateful to the EU COST action ES1103, whose funding allowed us to attend a hackathon that kick-started the work presented here. We thank L. Parts for comments on the manuscript and H. Eno for help with proofreading.

AUTHOR CONTRIBUTIONS

N.J.L. and J.T.S. conceived the project. N.J.L., J.Q. and J.T.S. implemented the Nanocorrect pipeline. J.T.S. conceived and implemented the Nanopolish pipeline. J.Q. generated the nanopore *E. coli* sequence data. N.J.L. and J.T.S. performed *de novo* assembly and analyzed the results. N.J.L. and J.T.S. wrote the manuscript. All authors approved the final manuscript.

COMPETING FINANCIAL INTERESTS

The authors declare competing financial interests: details are available in the online version of the paper.

1. Jain, M. *et al. Nat. Methods* **12**, 351–356 (2015).
2. Koren, S. *et al. Genome Biol.* **14**, R101 (2013).
3. Koren, S. *et al. Nat. Biotechnol.* **30**, 693–700 (2012).
4. Rasko, D.A. *et al. N. Engl. J. Med.* **365**, 709–717 (2011).
5. Chin, C.-S. *et al. Nat. Methods* **10**, 563–569 (2013).
6. Kim, K.E. *et al. Sci. Data* **1**, 140045 (2014).
7. Koren, S. & Phillippy, A.M. *Curr. Opin. Microbiol.* **23**, 110–120 (2015).
8. Quick, J., Quinlan, A.R. & Loman, N.J. *Gigascience* **3**, 22 (2014).
9. Goodwin, S. *et al.* Preprint at *bioRxiv* doi:10.1101/013490 (2015).
10. Loman, N.J. & Quinlan, A.R. *Bioinformatics* **30**, 3399–3401 (2014).
11. Myers, G. in *Int. Workshop Algorithms Bioinformatics* (eds. Brown, D. & Morgenstern, B.) 52–67 (Springer, 2014).
12. Lee, C., Grasso, C. & Sharlow, M.F. *Bioinformatics* **18**, 452–464 (2002).
13. Myers, E.W. *et al. Science* **287**, 2196–2204 (2000).
14. Gurevich, A., Saveliev, V., Vyahhi, N. & Tesler, G. *Bioinformatics* **29**, 1072–1075 (2013).
15. Darling, A.E., Mau, B. & Perna, N.T. *PLoS ONE* **5**, e11147 (2010).
16. Milne, I. *et al. Brief. Bioinform.* **14**, 193–202 (2013).

ONLINE METHODS

Assembly pipeline and software. The complete pipeline used to generate the assembly, including downloading the input data and required software, is provided as a Makefile on GitHub at https://github.com/jts/nanopore-paper-analysis/blob/master/full-pipeline.make. Additional scripts used to analyze the assembly are provided in the same repository. An IPython notebook documenting the analysis workflow is also provided.

The error-correction pipeline is implemented in Nanocorrect. The signal-level consensus algorithm is implemented in Nanopolish. Both programs are available on GitHub at https://github.com/jts/nanocorrect/ and https://github.com/jts/nanopolish/.

Read preprocessing. The DALIGNER program was designed for Pacific Biosciences data and expects the input reads to be provided in FASTA format with metadata encoded in the read name. To allow DALIGNER to be run on nanopore data, we wrote the program nanocorrect-preprocess.pl to convert the FASTA output of Poretools into the format expected by DALIGNER.

Error correction. We ran DALIGNER on the input reads using default parameters and the procedure described in the documentation to compute read-read overlaps. For each input read, "*A*," we queried the DALIGNER overlap database to extract the set of reads overlapping *A*. We trimmed each read that overlapped *A* to the coordinates matching *A* (as reported by DALIGNER) and reverse-complemented the sequence if necessary to make the read match the strand of *A*. This set of strand-matched subsequences was written to a FASTA file along with the sequence of *A*. This was the input to POA, which was run with parameters -hb -clustal using the BLOSUM80 sequence similarity matrix.

The consensus sequence was parsed from the clustal-formatted multiple alignment. We first computed the read depth at each column of the multiple alignment. We then trimmed the multiple alignment by excluding any leading and trailing columns that had depth less than 3. This trimming step is designed to remove uncorrected or unreliable regions of the multiple alignment, which may affect the quality of the downstream assembly. We reported the first consensus record calculated by POA (labeled "CONSENS0") over the trimmed region as the consensus sequence for the read.

Genome assembly. After two rounds of error correction, we assembled the reads using v8.2 of the Celera Assembler with the parameters file described in **Supplementary Figure 3**. The position in the large contig corresponding to the *E. coli* origin of replication was determined through a BLAST homology search for the first 1,000 bases of the reference genome, and the contig was circularized using the minimus2 package in AMOS[17] by the introduction of an artificial contig break. Minimus2 relies on the nucmer component of MUMmer[18] to perform the initial overlap alignments. We refer to this initial assembly, before signal-level consensus calling, as the draft assembly.

Computing the consensus sequence using signal data. We improved the accuracy of the assembly by calling a new consensus sequence using the raw electric signals emitted by the MinION. Our consensus procedure starts with the initial draft assembly, "*G*," and modifies it by introducing substitutions, insertions, deletions and substrings sampled from the 2D base-called reads. We use a probabilistic model of the nanopore sequencing process to evaluate whether the modifications to *G* improve the probability of observing the electric signal data for the collection of MinION reads. This modification and evaluation process is performed iteratively until no more improvements to the assembly can be made. *A* complete description of this algorithm, including the probabilistic model and our method to propose modifications to *G*, can be found in the **Supplementary Note**. This algorithm is implemented in the Nanopolish software package. We refer to the assembly after signal-level consensus calling as the polished assembly.

Assembly analysis. For the analysis of nanopore sequence read accuracy, reads were mapped to the *E. coli* K-12 MG1655 reference genome using BWA-MEM 0.7.10-r960-dirty[19]. The resulting BAM files were processed with the expand-cigar.py script, and mismatches, insertions and deletions were counted using the count-errors.py script from A. Quinlan's nanopore-scripts package (https://github.com/arq5x/nanopore-scripts).

Dot-plot analysis. The circularized contig was analyzed through the nucmer component of MUMmer[18], and the .delta file was visualized using mummerplot.

Coverage analysis. The sorted, aligned BAM file of raw reads against the assembly was analyzed using the genomeCoverageBed module of BEDTools[20]. Per-base coverage was averaged over 1,000 base segments and plotted in R using ggplot2.

G+C analysis. The G+C plot was constructed from the assembly with a Python script using Biopython averaging G+C content over 1,000 base segments[21].

QUAST analysis. The circularized contig, both before and after polishing was analyzed by QUAST 2.3 (ref. 14), which was run with default parameters and supplying the NC_000913.3 reference genome as the comparator.

Mauve analysis. The circularized contig was analyzed by running the progressiveMauve package on the command line with option --collinear and visualized in the Mauve Genome Viewer[15].

5-mer analysis. To explore over- and underrepresentation of 5-mers, we wrote a program to count the number of occurrences of the 1,024 different 5-mers in a set of sequences. This program counts 5-mers on the forward and reverse strands independently. The analysis used the counts for the forward strand of the reference and the reverse strand of the largest contig of our assemblies. We plotted the results using R and the ggplot2 library.

17. Treangen, T.J., Sommer, D.D., Angly, F.E., Koren, S. & Pop, M. *Curr. Protoc. Bioinformatics* **33**, 11.8 (2011).
18. Delcher, A.L., Phillippy, A., Carlton, J. & Salzberg, S.L. *Nucleic Acids Res.* **30**, 2478–2483 (2002).
19. Li, H. Preprint at http://arxiv.org/abs/1303.3997 (2013).
20. Quinlan, A.R. & Hall, I.M. *Bioinformatics* **26**, 841–842 (2010).
21. Cock, P.J.A. *et al. Bioinformatics* **25**, 1422–1423 (2009).

doi:10.1038/nmeth.3444