Published online 7 March 2010 *Nucleic Acids Research, 2010, Vol. 38, No. 8* ***2617–2623***
doi:10.1093/nar/gkq093

CLONEQC: lightweight sequence verification for synthetic biology

Pablo A. Lee[1], Jessica S. Dymond[2], Lisa Z. Scheifele[3], Sarah M. Richardson[2,4], Katrina J. Foelber[2], Jef D. Boeke[2,5] and Joel S. Bader[2,6,*]

[1]Department of Computer Science, Johns Hopkins University, 3400 N. Charles St., Baltimore, MD 21215, [2]High Throughput Biology Center, Johns Hopkins University School of Medicine, 733 N. Broadway, Baltimore, MD 21205, [3]Department of Biology, Loyola University, 4501 N. Charles St., Baltimore, MD 21210, [4]McKusick-Nathans Institute of Genetic Medicine, Johns Hopkins University School of Medicine, 733 N. Broadway, Baltimore, MD 21205, [5]Department of Biology and [6]Department of Biomedical Engineering, Johns Hopkins University, 3400 N. Charles St., Baltimore, MD 21215, USA

Received November 30, 2009; Revised February 15, 2010; Accepted February 16, 2010

ABSTRACT

Synthetic biology projects aim to produce physical DNA that matches a designed target sequence. Chemically synthesized oligomers are generally used as the starting point for building larger and larger sequences. Due to the error rate of chemical synthesis, these oligomers can have many differences from the target sequence. As oligomers are joined together to make larger and larger synthetic intermediates, it becomes essential to perform quality control to eliminate intermediates with errors and retain only those DNA molecules that are error free with respect to the target. This step is often performed by transforming bacteria with synthetic DNA and sequencing colonies until a clone with a perfect sequence is identified. Here we present CLONEQC, a lightweight software pipeline available as a free web server and as source code that performs quality control on sequenced clones. Input to the server is a list of desired sequences and forward and reverse reads for each clone. The server generates summary statistics (error rates and success rates target-by-target) and a detailed report of perfect clones. This software will be useful to laboratories conducting in-house DNA synthesis and is available at http://cloneqc.thruhere.net/ and as Berkeley Software Distribution (BSD) licensed source.

INTRODUCTION

Quality control is an essential component of any high-throughput operation. Quality control in DNA sequencing is familiar through established protocols that quantify errors in DNA sequencing reads. These sequencing error rates correspond to differences between the observed sequence inferred from sequencing reads and the true physical DNA sequence.

Synthetic biology introduces an analogous quality checkpoint: assessing the difference between the designed or desired target DNA sequence, and the physical sequence that was actually made. The physical sequence must in turn be observed from experimental sequencing data. We distinguish between synthesis errors (differences between the target and the physical sequence) and sequencing errors (differences between the physical sequence and the observed sequence), and focus entirely on the problem of synthesis errors.

Previously we described a production line for synthetic DNA instituted as an undergraduate course (1). Using software developed in-house (2,3), chromosome-length sequences of desired DNA are hierarchically decomposed into 700–800 nt building blocks that can be synthesized by performing PCR on overlapping 60-nt oligomers that tile across the region and can be routinely ordered from DNA vendors. Due to the intrinsic error rate of chemical synthesis, many of these oligos have base deletions, substitutions or less frequently insertions relative to the target sequence.

While various biochemical techniques can help to select for error-free DNA molecules (4–7), a key step in many DNA synthesis protocols is to clone a population of PCR products into bacteria by ligation into plasmid vector and transformation. Bacterial colonies can then be sequenced to identify a perfect clone. The 700–800 nt length was selected with sequence verification in mind because sequencing reads in both directions provide sufficiently accurate sequence information over the length of the physical construct.

*To whom correspondence should be addressed. Tel: +1 410 516 7417; Fax: +1 410 516 6240; Email: joel.bader@jhu.edu

We have observed error rates of 0.15–0.5% in chemically synthesized oligomers, with typical values close to 0.3–0.4%. This per-nucleotide error rate leads to a roughly Poisson-distributed error count with mean from 1 to 5 errors per building block, with additional errors potentially arising during PCR. The probability of a construct with no errors is approximately the negative exponential of the mean error count. In practice, often 10–20 clones must be sequenced before identifying one that matches the target exactly. In some cases differences between the physical sequence and the target sequence may be acceptable, for example, synonymous substitutions in protein-coding regions. Nevertheless, clone quality control by sequence verification is an important component of synthetic biology workflow.

Sequence verification is ideal for automation. It is tedious and error prone for human experts. Furthermore, synthesis of several target sequences often proceeds in parallel, and humans require workflow tracking to link a bacterial clone to the target sequence it is supposed to match. Automation of this step is feasible: given a sequencing read and a database of desired target sequences, the target sequence matching the sequencing read can be selected by BLAST (8), multiple sequencing reads for a single clone can then be aligned to the target using an algorithm such as ClustalW (9) and the alignment output can be automatically parsed to provide an automated assessment of clone quality.

Synthetic biology projects often use and generate DNA constructs at a hierarchy of sizes, from 60-nt oligomers to gene-sized sequences to entire chromosomes. While sequence verification at the bottom of the hierarchy is relatively easy, sequence verification for chromosome-sized DNA molecules at the top of the hierarchy requires resources similar to a genome sequencing project. Ideally, it would be attractive to establish a unified verification pipeline for all sequences regardless of length. In practice, however, synthesis errors that trace back to errors in the source oligomers are the dominant contributors early in the pipeline; far fewer errors arise during biological replication of DNA. A reasonable strategy, therefore, is to have two major checkpoints: one at the stage that chemically synthesized DNA is first put into a biological host, and second for the output of an entire synthesis project. Lightweight methods are required for the initial checkpoint. Lightweight here means that there is little need for workflow tracking or a supporting database. Instead, a stripped-down software pipeline can process sequencing reads serially, clone-by-clone, and is amenable to implementation as a web application. For the final checkpoint, a heavy-duty pipeline that permits genome-scale assembly of ultra-high-throughput sequencing reads is instead required.

The CloneQC software described here provides a public resource for lightweight sequence verification. The software pipeline is built from standard components, adapted for the purpose of validating relatively short synthetic DNA sequences easily checked in a single pair of reads from forward and reverse sequencing primers. The pipeline is in fact adapted to a workflow that generates one forward and reverse read for each clone, but could be readily modified for a single read or multiple reads. While the overall goal of identifying differences between an observed and a reference sequence is similar to other goals in genome biology, such as identifying naturally occurring sequence variation, the synthetic biology application benefits from a dedicated software solution. The software has been designed to satisfy two classes of users. A public server permits users to upload target and observed sequences, processes the data and reports back the results. The turnaround time for this operation is seconds per clone, eliminating sequence verification as a bottleneck. Furthermore, archives comprising data for upwards of 1600 clones can be uploaded and processed at once, providing convenient sequence verification for projects of small to medium scope. Alternatively, full source code under the Berkeley Software Distribution (BSD) license is provided for users who wish to incorporate this lightweight method as a component in their in-house pipelines.

MATERIALS AND METHODS

Quality control engine

The CloneQC engine uses BioPerl 1.4 utilities (10), the Staden package (v. 1.8.12 or earlier for compatibility) for I/O of sequencing reads (11), BLAST (NCBI v. 2.2.17) for fast matching of sequencing reads to target sequences (8) and ClustalW for aligning the reads to the target sequences (9).

Sequence reads are provided either as ab1 format (`.ab1` extension) or as pairs of plain text files containing base calls and PHRED-style quality scores (`.seq` and `.qual` extensions). Sequence reads may be trimmed, but this is not strictly necessary as read sequence that extends past the desired target sequence is ignored. If trimmed files are provided, the extensions `.trimmed.seq` and `.trimmed.qual` are recognized.

The algorithm expects a single pair of reads, forward and reverse, for each clone. Paired reads are associated using the naming convention `cloneid_F.xxx` for the forward read and `cloneid_R.xxx` for the reverse read, where cloneid is a unique identifier that identifies the physical clone that supplied the DNA for sequencing and `.xxx` is a recognized file extension. Forward and reverse here describes the experimentally known sequencing primers used to amplify the synthetic sequence. As cloning protocols often do not select for a particular orientation of the insert relative to a cloning vector, the orientation relative to the target sequence requires a comparison of the observed to the target sequence.

Target sequences are provided as one or more `.fasta` format files with one record per target. These records are parsed and used to generate a local `BLAST` database. The observed sequence reads and their reverse complements are then compared with the sequence database to identify the correct match. Although inconclusive matches are possible, these virtually always result from truncated or absent inserts and are flagged as synthesis failures. Sequencing reads with low-quality sequence

are flagged as sequencing failures, rather than synthesis failures. This distinction is important because the physical clones may house perfect sequence, and re-queuing the clones for sequencing may be desirable.

The paired reads for a single clone are then checked for matching to the same target. Furthermore, parity requires that exactly one of the two reads must be reverse complemented to match the target. Various causes can lead to the violation of these constraints: improper naming, a sample swap or contamination. Because the matching of sequencing read to clone is not trustworthy, it is not possible to provide a quality assessment. For matching failures, the overall QC value is therefore set to 'NA' for 'not applicable'.

Paired reads that match a consistent target are then entered into a three-way multiple sequence alignment with the target. Reads are trimmed to match the ends of the target, and then the final 15-nt of each read are also trimmed to eliminate possible low-quality sequence. We found that trimming low-quality sequence at the end of a read was important for generating accurate alignments. Prior to trimming, low-quality sequence in this region often caused poor alignments. The 15-nt trim works well in practice; improvements based on trimming to a quality-score threshold may yield further improvements. We compared ClustalW (9), a classic alignment algorithm, and MUSCLE (12), a more recent algorithm that uses *k*-mer matches to speed the alignment. We found that MUSCLE was approximately twice as fast, but provided a much higher error rate in alignments (results not shown). Consequently, we selected ClustalW to provide alignments.

The resulting alignments are then parsed, together with the quality-score information for the base calls on the forward and reverse reads. Each column of the alignment is then checked in turn. If both reads match the target, the column is recorded as a match. If only one read is available and it matches with high-sequence quality (PHRED 25 and above), the column is also scored as a match. If both reads differ from the target, the position is assumed to have a synthesis error and is scored as a mismatch. If the two reads disagree at a position and one matches, the quality scores are investigated. If the matching read is high quality (again PHRED 25 and above) and the mismatch is low quality (PHRED 20 and below), the disagreement is assumed to be due to a sequencing error and the column is scored as a match. Alternatively, if the read with the mismatch is high quality and the match is low quality, the column is scored as a mismatch. If neither of these conditions holds, the column is marked as a check.

The summary assessment for a clone is then calculated based on the number of matches, mismatches and checks across the alignment. If all columns are matches, the assessment is PASS. If at least one column is a mismatch, the assessment is FAIL. If no column is a mismatch, but at least one column is a check, the assessment is CHECK.

Finally, the failing clones are assessed for 'fixable' errors. A clone is fixable if it has at most six mismatches, and all occur within the final 20 bp of either end. These synthesis errors can be corrected by reamplification with primers that have the target nucleotide sequence, and such clones are assessed as FIXABLE.

Server configuration and requirements

The command-line CloneQC tool is also available as a web application written in the Ruby on Rails framework compatible with all major web servers. The web application accepts sequencing data uploads as archives (`.zip` or `.tar` format), including compressed archives (`.tar.gz` and `.tgz` format). Any `.fasta` files included in the archive are used as a source of target sequences. The target sequences may also be provided in an additional `.fasta` file. The public server permits uploads of up to 25 MB, selected to match the attachment limit on `gmail`. This limit is sufficient for `.ab1` files for 60 clones or uncompressed `.seq` and `.qual` files for 1600 clones. Uploaded sequencing data and results are permitted to reside on the public server for 1 week.

User interface design

Users are prompted to upload sequencing data and target sequences. Sequence data sets often come in batches corresponding to 96- or 384-well plates of clones. Processing may take several seconds per clone, corresponding to several minutes of waiting time. The interface provides a user with a link to check back for results and options to receive summary reports by email when a job completes. Access to results has two levels of authentication: a random key that is part of the link and a possible password selected by the user. This architecture avoids the need for users to create a password-protected login account and permits easy sharing of results job-by-job.

RESULTS

Processing overview

A flowchart of the processing steps is provided (Figure 1, see 'Materials and Methods' section for technical details). An entire archive of sequencing reads can be processed at once, with a maximum archive size of 25 MB. Uploaded with the archive is a FASTA format file with one record per synthetic target. The synthetic targets need not be arranged in any particular order, nor must each synthetic target be represented by a clone in the archive.

Sequencing reads are matched to synthetic targets using BLAST, and a quality control step is performed to ensure that each clone is represented by a pair of one forward and reverse read. This check may be easily modified for workflows that generate a different number of reads for each clone. Violations of these workflow constraints indicate that clone sequencing data may be untrustworthy, and the corresponding clones are marked as NA.

Following a successful match, sequencing data are processed clone-by-clone by aligning the sequencing reads for each clone to the target sequence. Clones with no synthesis errors are marked as PASS. Clones with at least one error are marked as FAIL. At some positions, one sequencing read may match with the target sequence while the other does not. If one read is high quality and the other low

quality, the high-quality read overrides the low-quality read to yield a PASS or FAIL decision. Otherwise, the position is marked for checking. Clones with at least one position marked for checking but with no other mismatched are marked as CHECK.

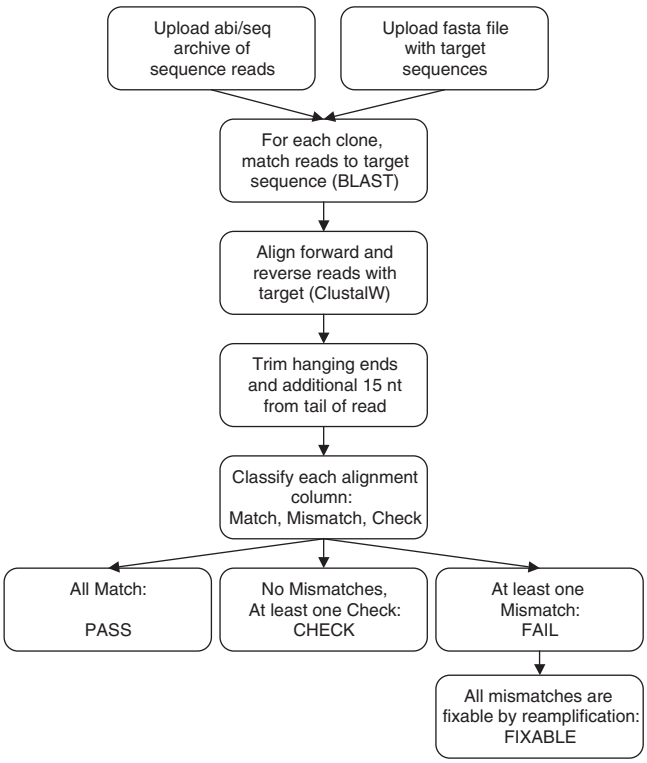

Figure 1. Flowchart of the CloneQC sequence validation pipeline. See 'Materials and Methods' section for details.

Finally, clones marked as FAIL with at most six synthesis errors, all within 20 bp of the ends, are classified as FIXABLE. These errors may be corrected by reamplification with primers that correct the errors.

Automated results

When CloneQC is run as a web application, it provides a key-protected URL for monitoring the process of job and, when completed, for viewing summary results (Figure 2). For each target represented by at least one sequenced clone, a summary table lists the number of clones in the PASS, CHECK and FAIL categories. Next, target-by-target, the identifiers for clones marked as PASS and CHECK are provided. These clones are presumably of greatest interest to the user.

In addition to the on-line results, the software generates a detailed report with one record per clone (Table 1). This detailed report may be downloaded from the web application, and it is e-mailed if an email address is provided. The detailed report indicates the positions of each synthesis error, the difference between the physical sequence and the target sequence and the base call quality scores. The same information is provided for each position where a discrepancy between sequencing reads leads to a CHECK assessment. The position information is useful for directing a traceviewer to visualize the locations of discrepancies in sequencing reads.

Comparison to expert analysis

To validate the software pipeline, quality control assessments generated automatically were compared with assessments provided by a human expert (Table 2). Instead of a CHECK category, the Human equivalent is UNCLEAR, meaning that the sequencing reads are

STATISTICS

Target	# Passes	# Checks	# Fixables	# Fails	Total
9L.3_32.Y3.07	1	0	0	17	18
9L.3_32.Y3.08	1	0	0	3	4
9L.3_32.Y3.09	1	0	0	1	2
9L.3_32.X3.02	1	1	0	15	17
9L.3_32.Y3.01	0	1	0	20	21
9L.3_23.C1.09	0	0	19	1	20
9L.3_32.Y2.06	2	2	0	8	12

9L.3_32.Y3.07

Passing Clones	Check Clones	Fixable Clones
BAG2008F_3_17_F12		

9L.3_32.Y3.08

Passing Clones	Check Clones	Fixable Clones
BAG2008F_3_17_E07		

Figure 2. Summary results for CloneQC run as a web application. 'STATISTICS' provides summary statistics for each synthetic target matched by at least one clone. Following are summary tables for each target sequence (only the first two shown), giving the identities of clones that contain perfect physical DNA for the target (Passing Clones), have discrepancies between reads but may have perfect physical DNA (Check Clones) or have errors that are fixable by reamplification (Fixable Clones).

Table 1. Column descriptions for detailed results spreadsheet with one record per clone

key1	Clone unique identifier (primary key).
bb_id	Target sequence unique identifier, taken from the best match among target sequences provided in the fasta file of targets.
length	Length of the target sequence.
overallqc	Overall QC for the clone: PASS, FAIL, CHECK, FIXABLE or NA if Reverse Complement QC or Match QC failed.
mutnqc	Mutation QC: PASS if no mutations, FAIL if 1 or more mutations (some of which may be fixable), or NA if Reverse Complement QC or Match QC failed.
revcomqc	Reverse complement QC: PASS if exactly one read must be reverse complemented; FAIL otherwise. This QC is specific for a workflow in which each clone has a forward and reverse read; it can be modified for workflows that provide different numbers of reads per clone.
matchqc	Match QC for all reads matching the same target sequence: PASS if all reads match the same target; FAIL otherwise.
pctid	Percent identify of the reads to the target sequence, taken as the number of matches in the three-way sequence alignment of the target with the two reads relative to the target length.
PF	Percent identity for the matching region of the forward read and the target.
PR	Percent identity for the matching region of the reverse read and the target.
LF	Length of the forward read.
LR	Length of the reverse read.
read1	Filename of the forward read. For the workflow described, this file is `<key1>_F.<ext>` where `<ext>` is an acceptable extension for read files, either ab1 or qual. If the read is reverse complemented to match the target, the reported name is `<key1>_F.<ext>_revcom.`
read2	Filename of the reverse read, either `<key1>_R.<ext>` or `<key1>_R.<ext>_revcom` if reverse complemented.
read_extra	Filenames of extra reads provided for the clone.
n_ins	Number of insertion synthesis errors (multi-base insertion count as 1 error).
n_del	Number of deletion synthesis errors (multi-base deletion count as 1 error).
n_sub	Number of substitution synthesis errors (multi-base substitutions count as 1 error).
n_chk	Number of regions to check for possible errors where individual reads disagree, with one matching the target sequence (multi-base regions count as 1 error).
n_tot	Sum of n_ins, n_del, n_sub, n_chk.
mutnstr	Space-delimited list of the errors. Each list item has the form `<type>:<pos>:<targetseq>:<readseq>`. The `<type>` is the type of error: `ins`, `del`, `sub`, or `chk`. The `<pos>` is the starting position in nucleotides in the target sequence, and the `<targetseq>` is the target sequence at the position. For an error of a single type that extends over multiple nucleotides (e.g. a multi-base insertion or deletion), the sequence over the length of the error is provided, using the '-' character to represent a deleted base. The `<readseq>` provides base calls from the sequencing reads. If the reads agree, their consensus base call is provided for each position. For a chk discrepancy, the reads differ, and bases and quality scores are provided for each read. If the `chk` region is a single base, the `<readseq>` is `[b1(q1)b2(q2)]` where `b1` and `b2` are the base calls from the forward and reverse reads and `q1` and `q2` are the corresponding quality scores. For a chk that extends over multiple bases, these records are concatenated, one for each position.

Table 2. Comparison of human and computer quality control assessments for sequences from 133 different clones

CloneQC	Human			
	PASS	FIXABLE	FAIL	UNCLEAR
PASS	21	0	0	0
CHECK	7	0	1	0
FIXABLE	0	4	0	0
FAIL	0	0	97	3

unclear and the status of the clone remains indeterminate. The accuracy of CloneQC is very high. It provides definitive assessments of PASS, FAIL or FIXABLE for 94% of the clones, punting with a CHECK on only 8 of the 133 cases. For the 125 clones with definitive assessments, the accuracy is at least 98%. The only difference is that three clones marked as FAIL by CloneQC were judged to be UNCLEAR by the expert, indicating that the sequencing reads were of too poor quality to make a final judgment.

Of the eight clones marked as CHECK, the expert assessed that seven should pass and only one should fail. These calls required visualization of the sequence traces, which are not yet used by the CloneQC algorithm.

DISCUSSION AND CONCLUSION

CloneQC has now been in use in a synthetic biology production setting for over a year. Prior to introducing this software component, sequence validation was an error-prone bottleneck requiring human intervention. The CloneQC pipeline provides faster, better assessment of synthetic DNA sequence quality. It is amenable to introduction in a production facility or, through a public web server, available for small-scale research and education applications.

The strategy of the automated assessment is to separate clone sequences into four categories: 'PASS', for clones with no synthesis error; 'FAIL', for clones with at least one synthesis error; 'CHECK', for clones whose sequencing reads have discrepancies that cannot be resolved automatically and require attention from a human expert; and 'NA' for clones whose sequencing data are not trustworthy, preventing a quality assessment. This algorithm provides a vast speedup in the processing time by weeding out the clones with evident synthesis errors and identifying clones that should be resubmitted for sequencing.

There are several directions for improving this framework. First, as with any machine learning task, we aim to reduce the number of clones that fall into the CHECK

category. The current automated procedure trims the end of each read, then uses a combination of PHRED score evaluation on both strands. This procedure, however, discards valuable information. Usually there is still good information in the trimmed sequence. The problems are not typically wrong or questionable base calls (which PHRED is good at flagging), but merged of peaks at the tail end of the sequencing read. This is particularly true in homopolymer runs where it becomes difficult to resolve the exact number of identical bases. Homopolymer runs require special attention because they are a common site of base deletion (or more rarely, insertion) in gene synthesis. An improved CloneQC algorithm could trim or mask regions based on unreliable or uninterpretable peak morphologies. For CHECKs that remain, the algorithm could trace files (such as `.ab1` format) to pregenerate views of the traces at the relevant locations. Even better, the algorithm could make direct use of sequence traces, as is already done by mutation detection software.

A related problem that would also benefit from pregenerated trace views is the possibility of a non-clonal mixed population, which in the case of a single substitution would result in a low-quality nucleotide at a single position in both reads. If the mixture is dominated by the correct clone, the position would be scored as a PASS. Enhancing CloneQC to provide a series of trace views zoomed to any questionable low-quality regions would permit fast user assessments of possible mixed populations when low-quality nucleotides have correlated positions in the paired reads.

Second, it will be useful to distinguish between the severity of different synthesis errors. We already do this for clones having only a few errors close to the ends, which are assessed as FIXABLE by reamplification with error-correcting primers. Clones with errors that lead to synonymous substitutions in protein-coding regions may similarly be acceptable. We have not yet implemented this option, primarily because it would require a more heavyweight solution augmenting the sequence data with annotations of protein-coding regions and reading frames.

Third, the need for BLAST matching of synthesized to target sequences could be circumvented by workflow tracking. Even in this case, BLAST could provide a quick quality control for sample tracking errors. Matching based on sequence similarity might introduce problems when multiple target regions have similar or even identical DNA sequences, which can arise for many reasons: reuse of genes or promoters, design of combinatorial libraries or design of chromosomes with duplicated genes. Matching based on BLAST could consistently score one target higher, and thus might always match the synthetic DNA to only one of multiple identical regions. With improved workflow, duplicated synthetic targets could be checked and the desired DNA would be synthesized only once. When combinatorial libraries or similar but not identical sequences are targets, a synthesis error for one target could actually produce the correct DNA for a different target. In this case, matching based on sequence similarity would remain useful.

Finally, as previously noted, the main needs for sequence quality control are at the entry of a chemically synthesized DNA molecule into a biologically replicating system, and at the exit of a completed large-scale synthesis project. The lightweight system described here is suitable for the entry stage; it is far faster than generating the sequence data. On the other hand, this system is unlikely to scale as is to the exit-stage analysis of a completed synthesis project. Verifying that the final physical output of a synthetic biology project matches the designed target sequence, and identifying each synthesis error, remains a challenge that can benefit from similar dedicated tools.

An important difference between sequence verification for individual clones and for a finished synthetic construct is the potential of ultra-high-throughput sequencing technologies, which provide massive numbers of short reads. Massive short reads provide superior performance for chromosome-length sequences, and next-generation sequencing is natural for this purpose. Next-generation sequencing may be difficult to adapt for sequence verification at the level of individual clones, however, because thousands of clones must be multiplexed to take advantage of the sequencing capacity of a single run. Multiplexing is possible with next-generation sequencing, but current tag-labeling kits for Illumina are limited to 12-fold and kits for 454 are limited to 141-fold. Furthermore, stockpiling thousands of clones prior to sequence verification also imposes a cost. It remains likely, therefore, that Sanger sequencing will continue to have a niche in sequence verification of initial synthetic constructs for the foreseeable future.

AVAILABILITY

A public server is available at http://cloneqc.thruhere.net/. BSD-licenced source code is available as supplementary material and for download from www.baderzone.org and http://baderlab.bme.jhu.edu/baderlab/index.php/Servers.

ACKNOWLEDGEMENTS

We acknowledge helpful discussions with Srinivasan Chandrasegaran and Jean Peccoud.

FUNDING

National Science Foundation (grant numbers MCB-0718846 to J.D.B., MCB-0546446 to J.S.B.); a Department of Energy Computational Sciences Graduate Research Fellowship (to S.M.R.); a Microsoft External Research Award (to J.S.B.). Funding for open access charge: Microsoft External Research Award.

Conflict of interest statement. None declared.

REFERENCES

1. Dymond,J.S., Scheifele,L.Z., Richardson,S., Lee,P., Chandrasegaran,S., Bader,J.S. and Boeke,J.D. (2009) Teaching synthetic biology, bioinformatics and engineering to undergraduates: the interdisciplinary Build-a-Genome course. *Genetics*, **181**, 13–21.

2. Richardson,S.M., Wheelan,S.J., Yarrington,R.M. and Boeke,J.D. (2006) GeneDesign: rapid, automated design of multikilobase synthetic genes. *Genome Res.*, **16**, 550–556.
3. Richardson,S.M., Nunley,P.W., Yarrington,R.M., Boeke,J.D. and Bader,J.S. (2010) Genedesign 3.0: An updated synthetic biology toolkit. *Nucleic Acids Res.,* (in press).
4. Carr,P.A., Park,J.S., Lee,Y.-J., Yu,T., Zhang,S. and Jacobson,J.M. (2004) Protein-mediated error correction for de novo DNA synthesis. *Nucleic Acids Res.*, **32**, e162.
5. Tian,J., Gong,H., Sheng,N., Zhou,X., Gulari,E., Gao,X. and Church,G. (2004) Accurate multiplex gene synthesis from programmable DNA microchips. *Nature*, **432**, 1050–1054.
6. Binkowski,B.F., Richmond,K.E., Kaysen,J., Sussman,M.R. and Belshaw,P.J. (2005) Correcting errors in synthetic DNA through consensus shuffling. *Nucleic Acids Res.*, **33**, e55.
7. Czar,M.J., Anderson,J.C., Bader,J.S. and Peccoud,J. (2009) Gene synthesis demystified. *Trends Biotechnol.*, **27**, 63–72.
8. Altschul,S.F., Gish,W., Miller,W., Myers,E.W. and Lipman,D.J. (1990) Basic local alignment search tool. *J. Mol. Biol.*, **215**, 403–410.
9. Thompson,J., Higgins,D. and Gibson,T. (1994) CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res.*, **22**, 4673–4680.
10. Stajich,J.E., Block,D., Boulez,K., Brenner,S.E., Chervitz,S.A., Dagdigian,C., Fuellen,G., Gilbert,J.G.R., Korf,I., Lapp,H. *et al.* (2002) The BioPerl toolkit: Perl modules for the life sciences. *Genome Res*, **12**, 1611–1618.
11. Staden,R., Beal,K.F. and Bonfield,J.K. (2000) The Staden package, 1998. *Methods Mol. Biol.*, **132**, 115–130.
12. Edgar,R.C. (2004) MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Res.*, **32**, 1792–1797.