Gigsaw – physical simulation of next-generation sequencing for education and outreach

David Michael Alan Martin*
College of Life Sciences,University of Dundee, Dundee, United Kingdom

* corresponding author (d.m.a.martin@dundee.ac.uk)

Received 2 May 2012; **Accepted** 21 June 2012; **Published** 15 October 2012
Competing Interests: none

Abstract

Modern sequencing methodologies produce more data in one run than a human being can read in a lifetime. Understanding how such vast quantities of information can be marshalled, assembled and interpreted is a challenging task for students and experienced researchers; it is even more challenging to have to explain this to lay audiences. Abstract representations, such as graphs or algorithms, or practical exercises with 'black-box' software, are limited in cultivating understanding. Gigsaw provides a physical model of next-generation sequencing data that can be readily manipulated, and different algorithms/experiments investigated at the bench top level. It is flexible in application and inexpensive to produce for public-understanding-of-science exercises or undergraduate/postgraduate training.

Availability: **a Web server implementation of the Gigsaw software is freely available at** http://www.compbio.dundee.ac.uk/gigsaw/ **and provides the Gigsaw output as PDF aligned for double-sided printing. Source code is available upon request under an open-source license.**

Introduction

Next generation sequencing (NGS) has brought about a paradigm shift in the prosecution of molecular biology research. Modern instruments can produce tens of millions of short DNA reads per day (Metzker, 2010). The conceptual challenge of understanding how to proceed from these tens of millions of sequence reads to a biological interpretation is considerable (Flicek and Birney, 2010). Sequence-assembly algorithms are complex concepts that can be hard for a student to grasp when presented in the traditional form of lectures, and even as practical exercises, where the algorithms are obscured by the quantity of data and 'black-box' software. It can take considerable effort to understand sequence assembly from textbooks or journal articles, an approach that is often beyond many undergraduates, and not suitable for educating the lay public, even though the basic idea of matching sequences is very simple.

Faced with the increasing interest of the public in biological research, and the inclusion of NGS approaches in undergraduate curricula, a new approach was needed to provide an elementary first step in understanding. Two inspirations led to the development of Gigsaw. The first is the 'table of learning' that is attributed to Glasser (Table 1), although the origins remain obscure (Smart and Paulsen, 2011).

Table 1. Quote attributed to William Glasser though the provenance is uncertain.

We learn:
10% of what we read;
20% of what we hear;
30% of what we both see and hear;
50% of what we discussed with others;
80% of what we experienced personally;
95% of what we teach to someone else.

If Glasser's paradigm holds, then providing students with a physical exercise should enable improved understanding over abstract learning by reading or lectures. The second inspiration was an introductory comment by Pevzner and colleagues (Pevzner *et al.*, 2001):

> *"Children like puzzles, and they usually assemble them by trying all possible pairs of pieces and putting together pieces that match. Biologists assemble genomes in a surprisingly similar way, the major difference being that the number of pieces is larger."*

We have developed Gigsaw as a 'Genome jigsaw generator'. It can produce pieces in PDF format that can be readily printed, laminated and used in the classroom, or on the street, for representation of almost any experiment that can be performed with NGS. The physical nature of the model, with the reverse complement appropriately printed on the converse, provides a tangible representation of the abstract concepts behind sequence alignment and assembly. Sequence searching and data-mining become literal concepts that the students can get their hands on. Several example applications are available on the Gigsaw website. Gigsaw has been used in scenarios from bioinformatics conferences with experienced researchers, to public-understanding-of-science (PUS) events with children of all ages who 'get' the concept of building an assembly by matching all the colours very rapidly, often even before they can read.

Implementation

Gigsaw is implemented in Perl as a dual-purpose Common Gateway Interface or command line application. The interface allows almost every aspect of the model to be configured to suit the experiment under consideration. A full list of configurable parameters is given in Table 2. Read length is fixed at 21 bases for single-end simulation. Paired-end simulation has paired reads of

Table 2. Configurable parameters for Gigsaw.

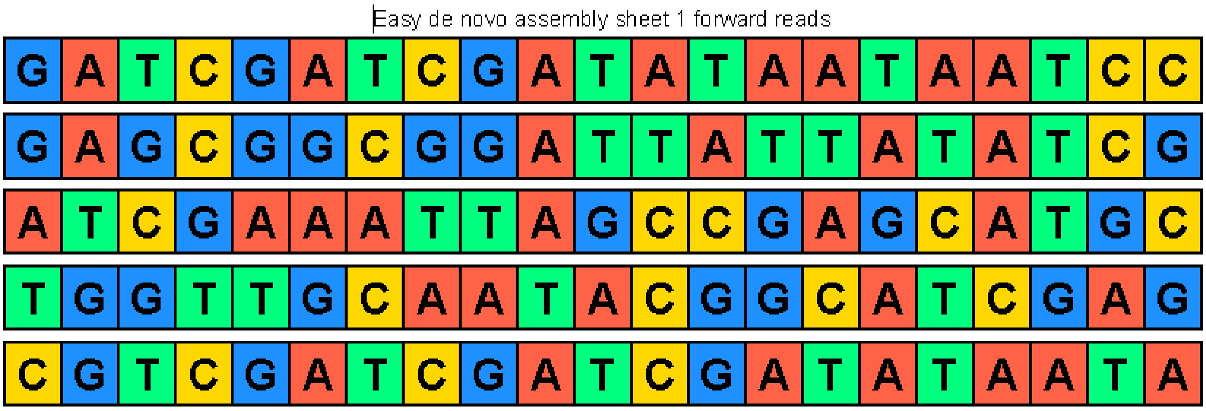
Parameter	Options	Description
Name	Free text	A title for the Gigsaw output
Sequence	1-1,000 characters from the set A,C,T or G	The source sequence from which to derive a Gigsaw.
Number of reads	Positive integer. The output formats 20 reads per A4 page.	All single reads are length 21bp or 10+x+10 for paired end reads
Error rate	Positive integer or 0	The error rate per 1,000 bases. 0 for perfect reads.
SNPs	[0-9]+:[ACTG]+[, [0-9]+:[ACTG]+[, …]]	Specify as position: bases with the number of bases proportional to their prevalence. EG C:T at position 20 in a 3:1 ratio would be 20:CCCT. Separate SNP definitions with commas and/or spaces
Paired-end gap size	0 or positive integer	0 for single-end reads. For paired ends, the actual gap is +- 5%
Sequence colour	X11 or hexadecimal (#FFFFFF) colour	The colour for the read font so multiple experiments can be separated.
Print reference sequence	Boolean	Print a reference sequence ruler from the source sequence.

10 bases each, separated by a distance sampled from a Gaussian distribution with a configurable mean and a standard deviation of 5% of the specified insert size.

Reads are generated as follows: the source sequence is read, and the start point for the read selected at random from a new copy that has been edited according to both the random-error rate selected and any single nucleotide polymorphisms (SNPs) defined. Errors are modelled by a process that randomly samples the genome sequence twice. The first sampling selects, at random, a single nucleotide to replace from the source sequence padded to a length of 1,000 bases, and the second sample randomly selects from the source sequence a replacement nucleotide. This process is repeated until the desired error rate (errors per 1,000 bases) is reached. Not every iteration will induce an error in the source sequence copy, and substitution rates will reflect the base composition of the source sequence. The real error rate is therefore below the requested error rate. Orientations for reads are then assigned randomly.

The desired number of reads (or read pairs) are then printed, 20 to a sheet of A4 paper, with the forward and reverse complement aligned on subsequent pages (Figure 1a and 1b). Upon printing, these can be preferentially laminated for durability, and separated with a guillotine or scissors ready for use.

The source sequence can, if desired, also be generated as a double-sided PDF. This is produced with ruler markings and a one-base pair (bp) overlap at the end of each 21bp segment, allowing the fragments to be joined together into the complete sequence (Figure 2).



Easy de novo assembly sheet 1 forward reads

G A T C G A T C G A T A T A A T A A T C C

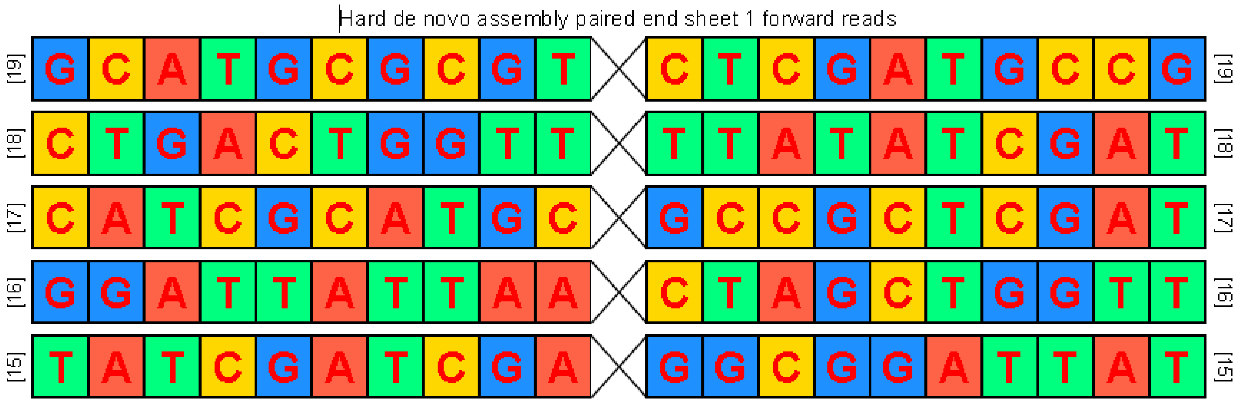
G A G C G G C G G A T T A T T A T A T C G

A T C G A A A T T A G C C G A G C A T G C

T G G T T G C A A T A C G G C A T C G A G

C G T C G A T C G A T C G A T A T A A T A

Figure 1a. Panel A: a set of single-end reads. Each time the application is run, a new set of reads is created.



Hard de novo assembly paired end sheet 1 forward reads

[19] G C A T G C G C G T — C T C G A T G C C G [19]

[18] C T G A C T G G T T — T T A T A T C G A T [18]

[17] C A T C G C A T G C — G C C G C T C G A T [17]

[16] G G A T T A T T A A — C T A G C T G G T T [16]

[15] T A T C G A T C G A — G G C G G A T T A T [15]

Figure 1b. Panel B: a set of paired-end reads. Each pair is labelled with a unique number. The two central arrows should be pointing towards each other when aligned and be approximately the gap distance apart.

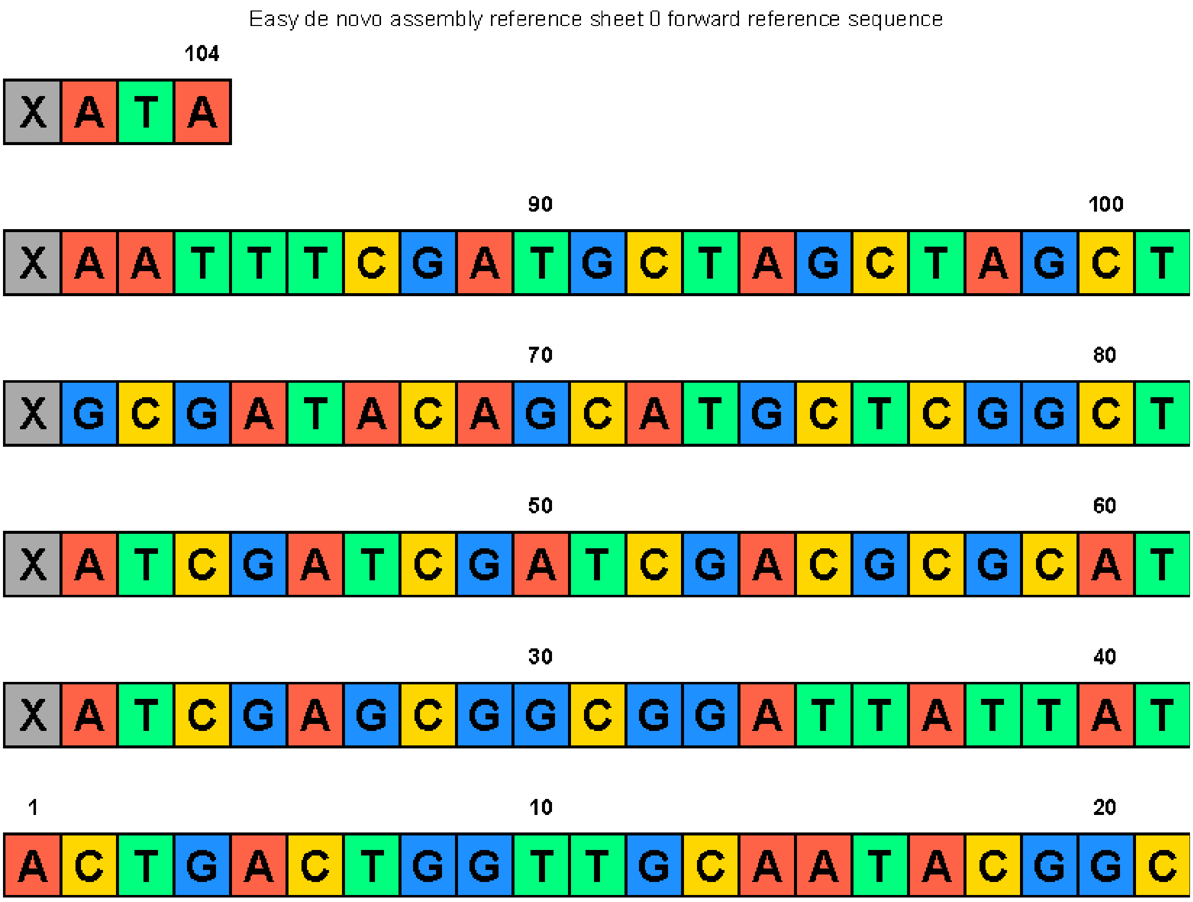

Easy de novo assembly reference sheet 0 forward reference sequence
104
X A T A
90
100
X A A T T T C G A T G C T A G C T A G C T
70
80
X G C G A T A C A G C A T G C T C G G C T
50
60
X A T C G A T C G A T C G A C G C G C A T
30
40
X A T C G A G C G G C G G A T T A T T A T
1
10
20
A C T G A C T G G T T G C A A T A C G G C

Figure 2. Reference sequence ruler for a short sequence. The grey cross indicates where the previous section should be attached.

Applications

De novo sequencing

Gigsaw is configured to produce reads only with no reference sequence. For PUS exercises, a very low error rate (1% or lower), with a total source sequence length of 50-60 bases, and around 40 reads, works well. For undergraduate exercises, a longer source sequence (up to 150bp) is used, and students are then encouraged to perform database searches with their assembled sequence to try to identify the gene. Care must be taken to avoid repeat regions of longer than about 15 bp, unless the exercise is to illustrate the problem of repeats. Students are asked to consider the evenness of coverage across their sequence assembly as a quality-control measure.

De novo sequencing with repeats

Gigsaw is configured to produce paired-end reads with an insert size that will span the repeats in question. It is also possible to combine single- and paired-end reads by running the Gigsaw application twice, once to generate single-end, and once to generate paired-end reads.

Mutation detection

Gigsaw is run once with the wild-type source sequence to generate a reference genome only. It is then run again with the mutated source sequence to produce reads. Students align the reads to the reference, and identify the mutated residues. A mixture of synonymous and non-synonymous mutations allows mapping onto a disease-related protein of interest, and develops understanding of how sequencing can identify this. Care should be taken to ensure that the students recognise the potential for sequencing errors, so a relatively high error rate (5%) will reinforce the need to see multiple reads with the same mutation. It is probably best to choose smaller proteins, or a single domain with an upper bound of about 200bp to maintain interest, while still providing sufficient intellectual stimulation.

Single Nucleotide Polymorphism detection

SNP detection can be performed with a single run of Gigsaw. SNPs are configured by specifying the base position and then the SNP ratio: e.g., for a 3:1 ratio of C to T at position 25, this would be specified as 25:CCCT. Multiple SNPs can be specified. Additional learning points here for the students are the statistical significance of SNP calling depending on the coverage depth and error rate.

RNAseq– intron/exon identification

A reference genome sequence is produced from the source DNA sequence. The RNA sequence is then used to produce a sufficient quantity of reads. It is best to not make the intron too long – anything longer than about 20bp is unnecessary. Students should note that the reads that bridge the splice sites should match both sides.

Quantitative RNAseq

This will require a larger group to get anything reasonably meaningful for analysis. A long genome read approaching the upper limit of Gigsaw (1,000bp) is produced. For each individual 'gene' (probably of 80-100bp) a separate Gigsaw run is required, and the appropriate number of reads generated. These reads are then mixed, and a sample of an appropriate size taken for alignment. Learning points here can include discussion of the detectable dynamic range with respect to read number, and how to deal with multiple matches for a read.

Conclusion

Gigsaw provides an educational tool that is adaptable, durable (if laminated) and extremely cost effective for teaching DNA-sequencing applications. It can be applied in situations from advanced training courses down to public engagement exercises by adjustment of the scale and complexity of the problems presented. Indeed, our experience, though statistically unsound, is that pre-schoolers often perform better than bioinformatics professors at simple *de novo* sequence assembly!

Acknowledgements

The author would like to acknowledge his colleagues and members of the Scottish Next Generation Bioinformatics User Group for helpful suggestions, and Dr. Tom Walsh for expert systems support.

References

1. Flicek P, Birney E (2009) Sense from sequence reads: methods for alignment and assembly. *Nat. Methods* **6**, S6-S12. doi: 10.1038/nmeth.1376
2. Metzker ML (2010) Sequencing technologies - the next generation. *Nat Rev Genet.* **11**, 31-46. doi: 10.1038/nrg2626
3. Pevzner PA, Tang H, Waterman MS (2001) An Eulerian path approach to DNA fragment assembly. *Proc Natl Acad Sci USA* **98**, 9748-9753. doi: 10.1073/pnas.171285098
4. Smart JC and Paulsen MB (2011) Higher Education: Handbook of Theory and Research. In: Smart, John C.; Paulsen, Michael B. (Eds.). Springer ISBN: 9400707010, Vol. 26, pp. 323.