MOLECULAR ECOLOGY
RESOURCES

Molecular Ecology Resources (2011) **11**, 743–748 doi: 10.1111/j.1755-0998.2011.03005.x

PRGMATIC: an efficient pipeline for collating genome-enriched second-generation sequencing data using a 'provisional-reference genome'

SARAH M. HIRD,*† ROBB T. BRUMFIELD*† and BRYAN C. CARSTENS†
**Museum of Natural Science, Louisiana State University, Baton Rouge, LA 70803, USA, †Department of Biological Sciences, Louisiana State University, Baton Rouge, LA 70803, USA*

Abstract

Second-generation sequencing is increasingly being used in combination with genome-enrichment techniques to amplify a large number of loci in many individuals for the purpose of population genetic and phylogeographic analysis. Compiling all the necessary tools to analyse these data is complex and time-consuming. Here, we assemble a set of programs and pipe them together with Perl, enabling research laboratories without a dedicated bioinformatician to utilize second-generation sequencing. User input is a folder of the second-generation sequencing reads sorted by individual (in FASTA format) and pipeline output is a folder of multi-FASTA files that correspond to loci (with 2 alleles called per individual). Additional output includes a summary file of the number of individuals per locus, observed and expected heterozygosity for each locus, distribution of multiple hits and summary statistics (θ, Tajima's D, etc.). This user-friendly, open source pipeline, which requires no a priori reference genome because it constructs its own, allows the user to set various parameters (e.g. minimum coverage) in the dependent programs (CAP3, BWA, SAMTOOLS and VARSCAN) and facilitates evaluation of the nature and quality of data collected prior to analysis in software packages.

Keywords: BWA, CAP3, genomic enrichment, pipeline, Second-generation sequencing

Received 12 November 2010; revision received 27 January 2011; accepted 9 February 2011

Introduction

Many research laboratories are interested in harnessing the sequencing capacity of second-generation sequencing (SGS) platforms, but are hindered by the lack of easily implemented bioinformatics tools for post-sequencing processing. Advances with genome enrichment techniques [or genomic reduction techniques, i.e. CRoPS (Van Orsouw *et al.* 2007), modified AFLP protocols (Gompert *et al.* 2010), RAD tags (Baird *et al.* 2008), molecular inversion probes (Absalan & Ronaghi 2007)], on-array and in-solution hybrid capture methods (reviewed in Mamanova *et al.* 2009) allow researchers to generate data sets that contain loci sampled from across the genome while maximizing overlap of these loci across individuals. The benefits from multi-locus, multi-individual sequence data are obvious—many questions in evolutionary biology (and other biological fields) require such data. Data sets such as this are attractive to researchers doing population or species-level studies as many loci from multiple individuals provides improved inference compared with fewer loci and fewer individuals (Brumfield *et al.* 2008). Enriched genomes are also useful for identifying genomic areas of interest (i.e. under selection, highly variable, conserved enough for interspecific primers, etc.). However, SGS data from genome enrichment techniques pose some specific organizational and analytical problems and have thus far required each researcher to create, de novo, a set of bioinformatics tools to process them. Software that allows any researcher to collate these data into a common format (e.g. FASTA) will facilitate the evaluation of their quality and suitability for further analyses.

Here, we outline a pipeline (PRGMATIC, for *P*rovisional-*R*eference *G*enome auto*matic* pipeline) for the analysis of SGS reads from the enriched genomes of individuals [in this case restriction enzyme-digested, size-selected genomic DNA sequenced on a Roche 454 system; a modified AFLP protocol similar to (Gompert *et al.* 2010) except with individuals as the tagged units, instead of pooled population samples]. PRGMATIC was designed with these major goals in mind: (i) Familiar output format (FASTA). (ii) Friendly to use. (iii) Free. (iv) Relatively fast. PRGMATIC

Correspondence: Sarah M. Hird, Fax: +1 225 578 2597;
E-mail: shird1@tigers.lsu.edu

source code and documentation are available at http://www.lsu.edu/faculty/carstens/archives/PRGmatic.zip or https://sites.google.com/site/sarahhird/project-code/prgmatic.

A full plate of 454 pyrosequencing using a typical genome reduction method may consist of hundreds of thousands of reads from thousands of loci across tens of individuals (e.g. Gompert *et al.* 2010). To evaluate data such as these, the pipeline constructs a 'provisional-reference genome' (PRG) from the loci targeted through genome enrichment. First, reads within individuals are clustered at a high level of stringency (99% identity) into 'alleles' (Fig. 1). Second, the 'alleles' are clustered across all individuals at a lower per cent identity into 'loci'. Third, a consensus sequence is created from each 'locus', and all consensus sequences are concatenated to form the PRG. All original reads are then aligned to the PRG and individual genotypes can be called based on the cumulative reads that 'stick' to each locus. The PRG is particularly useful because a plethora of analytic tools have been written for researchers who have a reference genome, yet relatively few researchers have access to a fully annotated 'real' reference genome. This pipeline creates what can be used as a real reference genome but only contains information from reads contained within a sample. The nine discrete steps of PRGmatic are outlined in the following paragraphs.

PRGmatic

Step 1: Preprocess data

PRGmatic requires that individuals are the sequence-tagged units (i.e. one sequence tag for each individual). The data that comes directly from the sequencer needs to be separated by tags and quality controlled to remove low-quality sequences. One option for preprocessing the data is the Ribosomal Database Project (Cole *et al.* 2009) Web site which has a 'Pyrosequencing Pipeline' that will process raw data, apply quality filters and return tag-separated files.

Step 2: Cluster reads within each individual

PRGmatic begins by clustering and aligning reads within each individual at a high per cent identity using the program CAP3 (Huang & Madan 1999). This step collects all reads that are almost identical (i.e. separated by very few SNPs/errors) into a single contig to form a putative allele. The default per cent identity for collating alleles is 99%, but the user can modify this value.

Step 3: Call alleles

Contigs within an individual that are above a given coverage (default: 5×) are designated high-confidence alleles. A consensus sequence is called for each high-confidence allele. Higher minimum coverage results in fewer alleles and fewer loci, although the user may have higher confidence in their alleles. Lower minimum coverage can result in many loci that correspond to very small clusters within a single individual (and are thus unusable for downstream analysis).

Step 4: Cluster alleles across individuals into loci

A second cluster/alignment step (again using CAP3), at a lower 'locus-level' per cent identity (default: 90%) col-

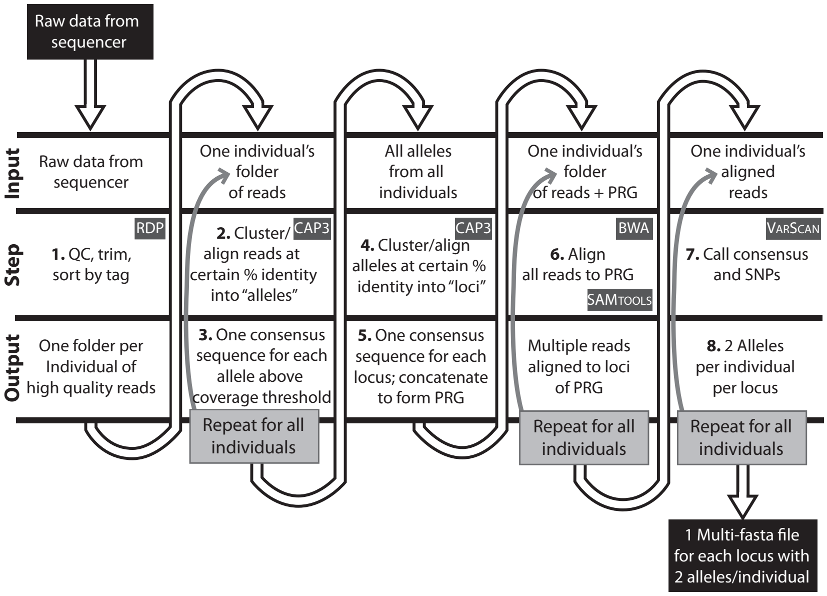
Raw data from sequencer
Input
Step
Output
Raw data from sequencer
RDP
1. QC, trim, sort by tag
One folder per Individual of high quality reads
One individual's folder of reads
2. Cluster/ align reads at certain % identity into "alleles"
CAP3
3. One consensus sequence for each allele above coverage threshold
Repeat for all individuals
All alleles from all individuals
CAP3
4. Cluster/align alleles at certain % identity into "loci"
5. One consensus sequence for each locus; concatenate to form PRG
One individual's folder of reads + PRG
BWA
6. Align all reads to PRG
SAMtools
Multiple reads aligned to loci of PRG
Repeat for all individuals
One individual's aligned reads
VarScan
7. Call consensus and SNPs
8. 2 Alleles per individual per locus
Repeat for all individuals
1 Multi-fasta file for each locus with 2 alleles/individual

Fig. 1 Schematic flowchart of PRGmatic. Dark grey boxes indicate dependent software used to accomplish each step (descriptions in Table 1). Bolded numbers refer to the numbered steps in the manuscript; PRG, provisional-reference genome; SNP, single nucleotide polymorphism.

lates all the high-confidence alleles into contigs treated as putative loci. This collects all variants of a locus together and ensures only one sequence per locus is contained in the PRG.

Step 5: Construct provisional-reference genome

PRGMATIC uses the loci to write the PRG, where each locus is annotated as if it were a chromosome in a full reference genome. The construction of the PRG allows the utilization of software designed for projects with a reference genome; these tools allow alignments to occur very quickly by eliminating pairwise comparisons across all reads and instead simply checking for similarity to the reference.

Step 6: Align all reads to the provisional-reference genome

The program BWA (Li & Durbin 2009) is called to format the PRG and align all the reads to it. As there is some variation across individuals in the number of reads, as well as the quality of reads, this enables loci that are sequenced above some threshold in certain individuals to be detected and genotyped in potentially all individuals. Each locus can then be interpreted on an individual basis to detect SNPs and errors and compute a consensus sequence.

Step 7: Call SNPs within individuals

SAMTOOLS (Li *et al.* 2009) is used to sort alignments generated by BWA and outputs a summary of each individual's reads in 'pileup' format. A custom Perl script converts the pileup format to a table of counts for all the reads at each position in the PRG. This table is then used to infer consensus sequence and SNPs. The program VARSCAN (Koboldt *et al.* 2009) is used to find insertions and deletions between the sequence reads and the PRG. These are incorporated in the next step. The user is prompted for the values they would like to use for minimum consensus coverage, SNP coverage and SNP per cent composition (what percentage of total reads the SNP comprises).

Step 8: Write alleles

A custom Perl script evaluates whether an individual's set of reads at a locus meet the minimum coverage cut-off value and writes each acceptable locus as a FASTA file with two alleles per individual. To write an allele, a Perl script evaluates the bases at each position. The base with the highest per cent composition is the consensus sequence and written as the first allele. If there is a second base at a position that exceeds the SNP cut-off value and composition per cent, the SNP is incorporated on the second allele, which is otherwise identical to the first. The cut-off values ensure that the SNP is both absolutely and relatively supported [if 4× coverage and 20% read composition is required, a SNP with 5× coverage would be accepted if there were 10× total coverage at that base (50%), but not if there were 100× coverage (5%)]. The unaligned loci files are output to a separate folder ('calledAlleles') in the PRGMATIC folder. The program MUSCLE (Edgar 2004) may be used to align the called alleles; MUSCLE is a fast multiple sequence aligner that is not required for the functioning of PRGMATIC but provides aligned FASTA files, an input format necessary for many downstream sequence analyses.

This method of allele calling is heuristic, as it is entirely based on the reads in the sample and their relative composition to the sample as a whole; a more sophisticated method for allele calling, such as using maximum likelihood (Hohenlohe *et al.* 2010) that uses statistical models for allele inference, will be incorporated into future versions of PRGMATIC. Currently, statistical tests may be conducted on the data used to generate our allele calls, as they are output as separate files. However, with our current methodology, it is recommended that some loci are confirmed by the user by inspecting the raw data that generated the alleles (discussed further elsewhere).

Step 9: Compute summary statistics

The final step of the pipeline computes several summary files. The first is a table of how many and which individuals have been called for each locus. The second is a table of the number of individuals, number of heterozygotes, number of alleles, and the observed and expected heterozygosity for each locus. A third file contains information about multiple hits: if an individual has more than two bases for a single position, the individual, position and locus are recorded. Finally, if the COMPUTE portion of the ANALYSIS package (Thornton 2003) is available on the local machine, a suite of summary statistics is computed for each locus (Watterson's θ, Tajima's D, etc.).

Four dependent programs

CAP3 (Huang & Madan 1999) is a fast sequence assembly program that incorporates base quality and automatically clips the low-quality ends of reads (Table 1). The clipping is controlled by several parameters that can be set by the user. CAP3 was chosen for the pipeline because it is quick, Unix-based, easy to use and free. It also automatically computes the consensus sequence for any contig it builds and outputs files in .ace format, which allow the contigs to be viewed in visualization programs like TABLET (Milne *et al.* 2009) or CONSED (Gordon *et al.* 1998).

Table 1 Synopsis of software used by PRGMATIC

Program	Use	Internal dependencies	Required?	Citation
BWA	Quickly align reads to PRG	None	Yes	Li & Durbin (2009)
CAP3	Cluster and alignment of reads	None	Yes	Huang & Madan (1999)
COMPUTE	Calculate summary statistics of loci	Libsequence	No	Thornton (2003)
MUSCLE	Multiple sequence alignment	None	No	Edgar (2004)
SAMTOOLS	Format/index reads aligned to PRG; pileup format	None	Yes	Li *et al.* (2009)
TABLET	Visualization of SGS data	None	No	Milne *et al.* (2009)
VARSCAN	Call consensus and SNPs from pileup format	None	Yes	Koboldt *et al.* (2009)

By constructing a reference genome from the data, we are able to utilize several programs that have been designed for genome assembly. After clustering and assembling the reads into loci, and combining the loci into a PRG, BWA (for Burrows–Wheeler Aligner, Li & Durbin 2009) aligns all the reads within an individual to the reference. The advantages of BWA include speed and accuracy with reads greater than 200bp. After indexing the reference genome, BWA sequentially finds the starting position for each read in the reference genome. It then generates alignments and outputs in SAM format.

SAM-formatted alignments are read by SAMTOOLS (Li *et al.* 2009) and converted to a binary version of the SAM format for quicker analysis; SAMTOOLS sorts the data according to their position in the reference genome and creates an index for both the reference sequence and the aligned reads to optimize speed. The reads are then compiled into pileup format, in which each line represents a reference base containing information on the number of reads, number of each base, read qualities, etc.

The pileup file is used to generate the consensus sequence and SNPs and is used as input for VARSCAN (Koboldt *et al.* 2009), which identifies insertions and deletions (indels).

To use PRGMATIC

PRGMATIC was developed for MacOSX. The user needs an 'inputFASTA' folder of their data separated by tag (i.e. one FASTA file for each individual). All data must first be quality controlled to the level desired by the user and in multi-FASTA format. Once all the dependent programs have been unpacked and the executables of the dependent programs are placed in the PRGMATIC folder (accomplished using the included Setup script), the user enters a single command and is prompted for five parameters. If desired, the user can set additional parameters by opening the PRGMATIC script (written in Perl) and manually adding the appropriate flags to the script. This should not be necessary for the majority of cases, as we have prompted the user for the most influential parameters.

Upon completion of the script, the user may view all the contigs and the PRG with the aligned reads from each individual. This is useful for examining the quality of the data underlying the output, understanding how the programs work and discarding paralogous loci.

PRGMATIC was designed with speed intended as one of its primary strengths. We have tested the pipeline on data sets generated by several researchers. The fastest runtime was <20 min in which 780 loci were generated from a 20-individual data set containing a total of 157 000 high quality reads. The longest runtime was 14 h in which 545 loci were generated from an 80-individual data set containing a total of 404 000 high-quality reads (all preliminary data generated on a 2.66GHz Intel Xeon processor with 16 GB memory and used the default settings).

Proof of concept

Two simulations and confirmation of four loci with Sanger sequencing provide proof of concept for PRGMATIC. The first included simulating 454 data using 100 empirical loci from a beta tester as a template with the program FLOWSIM (Balzer *et al.* 2010). We simulated 10 000 reads then quality controlled them by removing reads <100 bp and any reads containing an 'N'. This resulted in 6825 high-quality reads that we then used as input for PRGMATIC. After <3 min of run time, 589 alleles (average coverage 10×) were identified; the 236 alleles with ≥5× coverage were subsequently clustered into 101 loci, forming the PRG. All the original reads were then aligned to the PRG (average coverage 60×; range of coverage 7–83 reads). We assembled the 101 PRGMATIC loci with the 100 empirical loci using GENEIOUS (Drummond *et al.* 2011). For 99 loci, each empirical locus aligned to one PRGMATIC locus without a single SNP to differentiate the two. There was one instance of two loci being called by PRGMATIC from a single empirical locus—the first PRGMATIC locus was identical to the empirical locus and the second was shorter than the empirical locus by 35 bp and contained two SNPs (one adjacent to a 2-bp homopolymer and the second within a 4-bp homopolymer). In other words, PRGmatic recovered 100% of the loci (and a 99% 1:1 correspondence between empirical loci and estimated loci) but also called a single 'incorrect' locus which differed from the 'correct' locus by 37 bp. This error may be because of the fact that the simulated data was not identical to genome-enriched SGS data in that we did not ensure the correct forward and reverse primers were on each read.

The second simulation contained five individuals with five loci each: one monomorphic locus, one polymorphic at a single site locus where each individual is a homozygote, one locus with heterozygotes, one locus with a four base pair indel and a fifth 'locus' with three unique alleles within 90% similarity of each other to simulate a paralogous locus (see Table 2). Data mimicked genome enriched data by containing primer sequences and individual sequence tags like each empirical read would have; then 10 000 reads per individual were simulated with FLOWSIM and edited by hand to remove all sequences in the reverse direction (which are not found in empirical datasets), data was sorted by tag and primer sequences were removed. Data was also quality controlled for sequences that were too short (<100 bp) or contained Ns. This resulted in an average of 2177 high-quality reads per individual (range 2117–2245). The pipeline was run with the default settings—which took <20 min to complete. Six loci were called from 263 alleles—the simulated locus with three alleles was split into two loci, whereas all the other loci were correctly identified. Additionally, all genotypes were correctly called with one exception: the four base pair indel, where heterozygotes for the indel were called with one allele containing three of the bases and the second allele containing the remaining base (see Table 2).

Finally, a beta user has empirically verified four loci using Sanger sequencing (J. Maley, personal communication). Primers were designed based on loci called by PRGMATIC and Sanger sequenced on an ABI 3100; all were found to be single copy and the variation (1–4 SNPs/locus) identified by the pipeline were verified.

Recommendations

Perhaps the biggest consideration with the use of genomic reduction data is the identification of multi-copy genes. If a gene duplication event occurred recently, such that the two paralogs share a per cent identity above the threshold for clustering alleles into loci, all the reads generated from the two different places in the genome will align to the same locus in the PRG. Grouping multiple loci into a single locus is problematic for SNP calling and may distort downstream analyses. It should, however, skew certain summary statistics in predictable ways. If paralogous loci have acquired high frequency or fixed differences, this should dramatically skew the apparent heterozygosity within populations. For this reason, a table containing observed and expected heterozygosity, calculated on a haplotypic basis, is included. If paralogs are grouped as one locus, the observed heterozygosity should be high relative to expected heterozygosity. The user would want to visually inspect their data as well as the summary statistics for loci that look biologically

Table 2 Five individual, five loci simulation conditions and results. Up to three alleles (X.1, X.2, X.3) are shown for each locus, the results contain the six loci called by PRGmatic. The total length of each locus is given in parentheses

	Monomorphic (302)		Polymorphic (288)		Heterozygotes (331)		Indels (334/338)		Paralogous (323)			
Actual	1.1	1.2	2.1	2.2	3.1	3.2	4.1	4.2	5.1	5.2	5.3	
ind1	A	A	AGT	AGT	CA	CA	CACA	—	AACGC	AACGT	CTGTC	
ind2	A	A	GGC	GGC	AT	CA	CACA	—	AACGC	AACGT	CTGTC	
ind3	A	A	GGT	GGT	AT	CT	CACA	—	AACGC	AACGT	CTGTC	
ind4	A	A	ATT	ATT	AA	AT	—	—	AACGC	AACGT	CTGTC	
ind5	A	A	ATT	ATT	AA	AT	CACA	CACA	AACGC	AACGT	CTGTC	
Results	Monomorphic (302)		Polymorphic (288)		Heterozygotes (331)		Indels (335/337)		Paralogous (323)			
	1.1	1.2	2.1	2.2	3.1	3.2	4.1	4.2	5.1	5.2	6.1	6.2
ind1	*	*	*	*	*	*	CAC	A	*	*	*	*
ind2	*	*	*	*	*	*	CAC	A	*	*	*	*
ind3	*	*	*	*	*	*	CAC	A	*	*	*	*
ind4	*	*	*	*	*	*	*	*	*	*	*	*
ind5	*	*	*	*	*	*	*	*	*	*	*	*

*Estimated genotype was identical to the actual genotype in the column above.

suspect. Actually viewing the reads that were used to call each locus should increase user confidence that a locus is homologous across individuals and across reads within a single individual. A supplementary guide to detecting common errors, complete with screen shots of various errors and real data, is included in the PRGMATIC distribution, in order to facilitate understanding of PRGMATIC output (Supporting Information).

Acknowledgements

The authors thank their funding source, NSF grant DEB-0956069. We thank M. Koopman for thoughtful discussions regarding data analysis and preliminary data sets and D. Ence for Perl scripting help. We thank N. Reid for discussion on paralogous loci and N. Reid, Y. E. Tsai, T. Pelletier, A. Zellmer and J. McVay for comments on the manuscript. J. McCormack and J. Maley provided preliminary data sets, and J. McCormack, J. Maley and A. Zellmer were helpful test users.

References

Absalan F, Ronaghi M (2007) Molecular inversion probe assay. *Methods in Molecular Biology*, **396**, 16.

Baird N, Etter P, Atwood T *et al.* (2008) Rapid SNP discovery and genetic mapping using sequenced RAD markers. *PLoS One*, **3**, 3376.

Balzer S, Malde K, Lanzen A, Sharma A, Jonassen I (2010) Characteristics of 454 pyrosequencing data—enabling realistic simulation with flowsim. *Bioinformatics*, **26**, 6.

Brumfield RT, Liu L, Lum DE, Edwards SV (2008) Comparison of species tree methods for reconstructing the phylogeny of bearded manakins (Aves: Pipridae, Manacus) from multilocus sequence data. *Systematic Biology*, **57**, 13.

Cole JR, Wang Q, Cardenas E *et al.* (2009) The Ribosomal Database Project: improved alignments and new tools for rRNA analysis. *Nucleic Acids Research*, **37**, 5.

Drummond A, Ashton B, Buxton S *et al.* (2011) *Geneious v5.4*, Available from http://www.geneious.com/.

Edgar RC (2004) MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Research*, **32**, 6.

Gompert Z, Forister ML, Fordyce JA, Nice CC, Williamson RJ, Buerkle CA (2010) Bayesian analysis of molecular variance in pyrosequencing quantifies population genetic structure across the genome of *Lycaeides* butterflies. *Molecular Ecology*, **19**, 19.

Gordon D, Abajian C, Green P (1998) Consed: a graphical tool for sequence finishing. *Genome research*, **8**, 8.

Hohenlohe PA, Bassham S, Etter PD, Stiffler N, Johnson EA, Cresko WA (2010) Population genomics of parallel adaptation in threespine stickleback using sequenced RAD tags. *PLoS Genetics*, **6**, 23.

Huang X, Madan A (1999) CAP3: a DNA sequence assembly program. *Genome Research*, **9**, 10.

Koboldt D, Chen K, Wylie T *et al.* (2009) VarScan: variant detection in massively parallel sequencing of individual and pooled samples. *Bioinformatics*, **25**, 3.

Li H, Durbin R (2009) Fast and accurate short read alignment with Burrows–Wheeler transform. *Bioinformatics*, **25**, 7.

Li H, Handsaker B, Wysoker A *et al.* (2009) The Sequence Alignment/Map format and SAMtools. *Bioinformatics*, **25**, 2.

Mamanova L, Coffey A, Scott C *et al.* (2009) Target-enrichment strategies for next-generation sequencing. *Nature Methods*, **7**, 111–118.

Milne I, Bayer M, Cardle L *et al.* (2009) Tablet—next generation sequence assembly visualization. *Bioinformatics*, **26**, 2.

Thornton K (2003) libsequence: a C++ class library for evolutionary genetic analysis. *Bioinformatics*, **19**, 3.

Van Orsouw N, Hogers R, Janssen A *et al.* (2007) Complexity reduction of polymorphic sequences (CRoPS): a novel approach for large-scale polymorphism discovery in complex genomes. *PLoS One*, **2**, 1172.

Supporting Information

Additional supporting information may be found in the online version of this article.

File S1 README file for PRGMATIC.

File S2 Guide to Common Errors with PRGMATIC.

Please note: Wiley-Blackwell are not responsible for the content or functionality of any supporting information supplied by the authors. Any queries (other than missing material) should be directed to the corresponding author for the article.