Published online 8 February 2011

Nucleic Acids Research, 2011, Vol. 39, No. 9 ***e57***
doi:10.1093/nar/gkq1268

RATT: Rapid Annotation Transfer Tool

Thomas D. Otto[1,*], Gary P. Dillon[1], Wim S. Degrave[2] and Matthew Berriman[1]

[1]Parasite Genomics, Wellcome Trust Sanger Institute, Wellcome Trust Genome Campus, Cambridge, CB10 1SA, UK and [2]Laboratório de Genômica Funcional e Bioinformática, Instituto Oswaldo Cruz, Fiocruz, Rio de Janeiro, Brazil

Received August 31, 2010; Revised November 19, 2010; Accepted November 23, 2010

ABSTRACT

Second-generation sequencing technologies have made large-scale sequencing projects commonplace. However, making use of these datasets often requires gene function to be ascribed genome wide. Although tool development has kept pace with the changes in sequence production, for tasks such as mapping, *de novo* assembly or visualization, genome annotation remains a challenge. We have developed a method to rapidly provide accurate annotation for new genomes using previously annotated genomes as a reference. The method, implemented in a tool called RATT (Rapid Annotation Transfer Tool), transfers annotations from a high-quality reference to a new genome on the basis of conserved synteny. We demonstrate that a *Mycobacterium tuberculosis* genome or a single 2.5 Mb chromosome from a malaria parasite can be annotated in less than five minutes with only modest computational resources. RATT is available at http://ratt.sourceforge.net.

INTRODUCTION

Second generation technologies are drastically reducing the cost of DNA sequencing, while dramatically increasing throughput (1), a trend that is likely to continue (2). Major advances have been made in data processing, particularly with the development of numerous algorithms for assembly (3,4) and alignment of short reads against a reference sequence (5,6), known as mapping. However, interpretation of new genome data frequently requires annotation—whereby functions are ascribed to genes and other regions of biological interest—but this process is expensive, often placing a disproportionate demand on human and computational resources compared with data production.

For most genome projects, large numbers of genes are annotated based on *ab initio* predictions using gene finding software, such as GLIMMER (7) or TwinScan (8), often trained using existing transcript sequence data. The newly predicted coding sequences are annotated based on sequence similarity searches against protein or domain databases. Functions are ascribed at levels of granularity that reflect the strength of sequence-similarity based evidence and are recorded as free-text descriptions or by using controlled vocabulary terms chosen from an ontology such as the Gene Ontology (9). In addition, non-coding features like tRNAs and promoters are identified using other tools (10).

Automated annotation tools or pipelines do exist, e.g. Ensembl, (11) GARSA (12) or SABIA (13), but their installation and operation is complicated by their dependence on third party software packages, server clusters and bioinformatics expertise. These annotation pipelines are often beyond the resources of a small lab and too labour- or machine-intensive to perform regularly. Alternatively, annotation servers exist. For instance, the RAST Server (14) or the integrated microbial genome system (15) but these are currently restricted to prokaryotes.

Faced with diminishing annotation resources available for each new genome, yet a need for more rapid annotation of new sequences, we have developed a simple method of annotation called RATT (Rapid Annotation of Transfer Tool). The program obviates the need for *de novo* annotation and uses conservation of synteny to transfer annotation from a previously well-annotated reference. The transfer can be used between any closely related sequences, either to transfer annotations between successive versions of a draft genome, or to annotate new strains or species.

In RATT, positional data—based on conserved synteny and similarity between a reference and query—are used to infer orthology, and hence function, more accurately. This method is much faster than performing sequence similarity searches to map each feature, without reference to their genomic context. Furthermore, as genomic features, such as genes, differ in their underlying sequence between strains, we refine all genes features in a correction step, to take into account changes to start/stop codons, length or the presence of internal stop codons. As an aid to the

*To whom correspondence should be addressed. Tel: +44 1223 494864; Fax: +44 (0) 1223 494919; Email: tdo@sanger.ac.uk; thomasd.otto@googlemail.com

annotator, a detailed mapping report is produced and all changes and remaining errors can be checked using visualization tools such as Artemis (16).

Here we evaluate the performance of RATT when applied to three different mapping scenarios using datasets of *Mycobacterium tuberculosis*, *Plasmodium berghei* and *Plasmodium chabaudi*. The tool is available for download, with further information from http://ratt.sourceforge.net.

MATERIALS AND METHODS

Algorithm design

RATT is programmed in 'bash' and 'PERL' and its design is illustrated in Figure 1 and Supplementary Figure S1. First, two sequences are compared using 'nucmer' from the MUMmer package (17) to define sequence regions that share synteny. Those regions are filtered using configurable parameters depending on the type of annotation mapping that is being attempted. Preset parameters are provided for transfers between assembly versions, strains or species (see Supplementary Table S1). To be included, the minimum nucleotide sequence identity between synteny blocks must be 40%. Synteny information is stored as a base range in the query and its associated base range in the reference. However, this information alone is inadequate to map the annotation because insertions or deletions (indels) change the relative distance between mapped synteny blocks. The coordinates are therefore sequentially adjusted across a synteny block by calling indels using 'show-snp' from the MUMmer package. Accurately calling indels within repetitive regions presents a particular challenge. Therefore, RATT recalibrates the adjusted coordinates using single nucleotide polymorphisms (SNPs, also called using 'show-snp') as unambiguous anchor points within synteny blocks. In transfers between very closely related sequences (e.g. successive assembly versions), SNPs may occur with insufficient frequency to perform this coordinate adjustment. In such cases, RATT modifies the query by inserting a *faux* SNP every 300 bp to aid in the recalibrating step. The final sequence and transferred annotations remain unaffected.

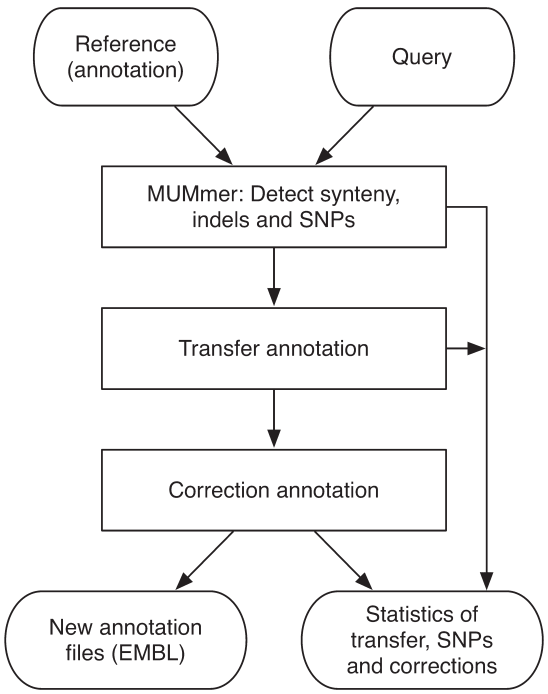

Figure 1. Workflow of RATT.

Once the coordinates within synteny blocks have been defined, RATT proceeds to the annotation-mapping step, whereby each feature within a reference EMBL file is associated with new coordinates on the query (Supplementary Figure S1B). A feature is not mapped (and is put in the non-transferred bin file), if it bridges a synteny break and if its coordinate boundaries match different chromosomes, different DNA strands, or if the new mapped distance of its coordinates has increased by more than 20 kb. If a short sequence from the beginning, middle or the end of a feature can be placed within a synteny region, mapping is attempted (see Supplementary Figure S1B). In addition, if the exons of a single gene model map to different gene regions, the model is split and identified in the output file. The bin is an EMBL-format file that can be loaded onto the reference sequence for analysis (see Figure 2, brown colour track). Further outputs include statistics about transferred features or the amount of synteny conserved between the reference and query, as well as Artemis-readable files showing SNPs, indels and regions that lack synteny between the compared sequences, see the example on the sourceforge site.

Although two sequences may be related, differences can occur, such as a change in the start or stop codons of a protein-coding sequence. Therefore, we implemented a correction algorithm in RATT (see Supplementary Figure S1C). Figure 3 shows examples of the correction step. First, the start codon is checked. If it is not present, the upstream sequence is searched for a new start codon (Figure 3A). If a stop codon is found, the first start codon downstream is used. In the absence of any start codon, an error is recorded in the results file. If the sequence between exons has no stop codon and a length divisible by three bases but the splice acceptor or donor sequences are wrong, then the intron is eliminated. Likewise, frameshifts previously introduced into the reference to maintain conceptual translations (for instance, in apparent pseudogenes) will also be removed from coding sequences in the query. RATT will also detect, and attempt to fix, incorrect splice sites. As splice sites are difficult to annotate correctly, RATT only tries to correct a gene model that has one wrong splice site. If one incorrect splice site is detected, the closest alternative splice donor or acceptor is found that, when used, generates no frame shifts. Next, RATT searches for genes or exons with internal stop codons, further than 150 bp from the 3′-end. If the introduction of a frameshift would generate a model without internal stop codons, the model is corrected (Figure 3C). Stop codons are corrected last: if a model has less than five internal stops in its last

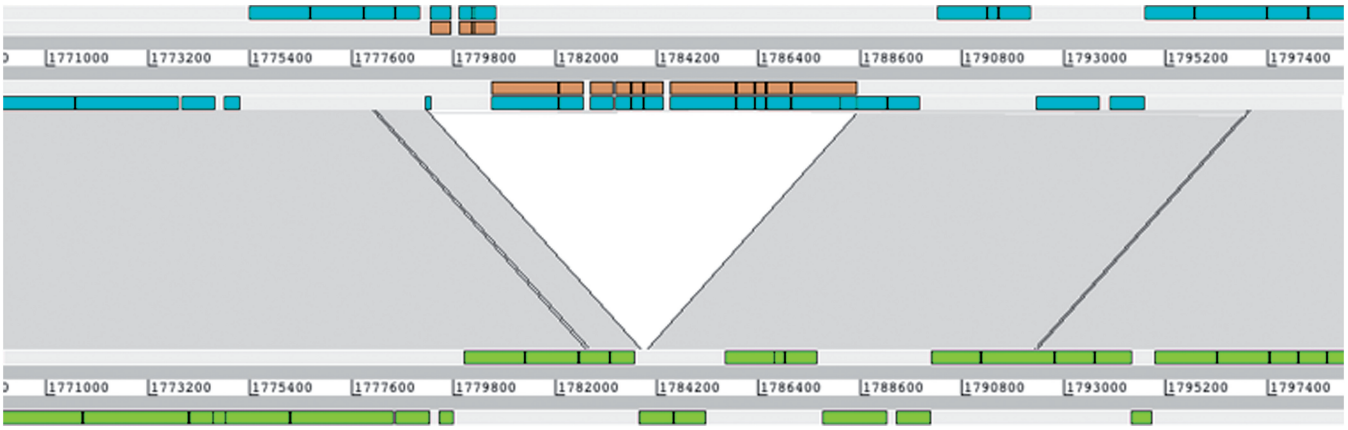

Figure 2. Transfer of annotation from the *M. tuberculosis* strain H37Rv onto the strain F11 sequence, over a deletion. The genomes of H37Rv (upper) and F11 (lower) are shown using the Artemis Comparison Tool (ACT). The source H37Rv annotation (light blue) is directly mapped onto F11 by RATT (green) except for those features corresponding to a region that is unique to the source strain that cannot be transferred and are written to a separate output file (brown).

exon, the model is shortened to the first stop codon (Figure 3B). If the model has no stop codon it is extended downstream until a stop codon is found.

Different criteria can be specified depending on the translation that an organism uses (e.g. such bacterial TTG and GTG start codons) or whether unsual splice sites are used. RATT is programmed in PERL and was tested in UNIX/LINUX environments. The output can be loaded into Artemis/Act. The list and explanation of all the output files can be found at the sourceforge site.

Datasets and comparison

To evaluate RATT, we assessed its performance using manually annotated genomes. The *M. tuberculosis* strain H37Rv (NCBI:AL123456) was used to annotate the genome of strain F11 using the 'Strain' comparison option. Results were compared with the existing annotation of F11 (NCBI:CP000717). In addition, the annotation of *P. chabaudi* was mapped to the *P. berghei* version 9 genome using the 'Species' comparison option. The *P. chabaudi*/*P. berghei* dataset can be downloaded from http://ratt.sourceforge.net/Chab_berg.zip. The files relating to the transfer of annotation between *P. berghei* assemblies can be found at http://ratt.sourceforge.net/Berg_berg.zip. The transfer was performed using the 'Assembly.Repetitive' parameters, and the results are included in the latest GeneDB version (http://www.genedb.org).

Although, direct benchmarking was not possible because RATT presents a new strategy, we ran Glimmer3—a popular *ab initio* gene predictor—on the tuberculosis dataset as a comparator. Particular attention was given to the number of CDSs transferred or predicted, and whether their boundaries coincided with curated models. After running RATT on each of the three datasets, the transferred annotations were manually checked in Artemis and ACT.

RESULTS

Initial designs of RATT used BLAST and later FASTA to transfer the annotation by comparing the annotation features of the reference against the query. Although the results were reasonable (data not shown), errors were generated when determining feature borders, small features or gene families (18). A significant improvement was gained when we first defined the synteny between both sequences and then identified SNPs and indels. Here, we present the performance of RATT in three case studies of transfer between sequences with different levels of similarity. Where relevant we also present a comparison with *ab initio* predictions using GLIMMER.

Transfer between strains

First, we applied RATT to *M. tuberculosis*, and used strain H37Rv (AL123456l) to annotate the strain F11 (CP000717) sequence. The existing annotation for F11 (http://www.broadinstitute.org/annotation/genome/mycobacterium_tuberculosis_spp/GeneFinding_f11.html) was used to evaluate RATT performance. The transfer itself took <2 min, and the correction of the transfer a further 2 min, on an Intel(R) Xeon 3.00 GHz processor, using around 2 Gb of memory. Synteny is conserved throughout the majority of the sequence: 1.07% of the reference has no synteny with the query, and 1.45% of the query has no synteny with the reference.

Of 9557 strain H37Rv features (Figure 2), only 191 could not be transferred (44 CDSs, see Table 1) and of the unmapped coding sequences, 20 encoded transposases with multiple insertion sites and 11 resided in deleted regions (Figure 2). The remaining models were within regions of limited synteny (8) or had divergent sequences (5). The 9366 successfully assigned features included 85 corrected CDSs and three models with unresolved corrections. Tables 1 and 2 summarize the results. Therefore, the majority of gene models do not need to be corrected, in total only 47 CDSs (~1.2%) required manual intervention. RATT was also able to identify 99 gene models

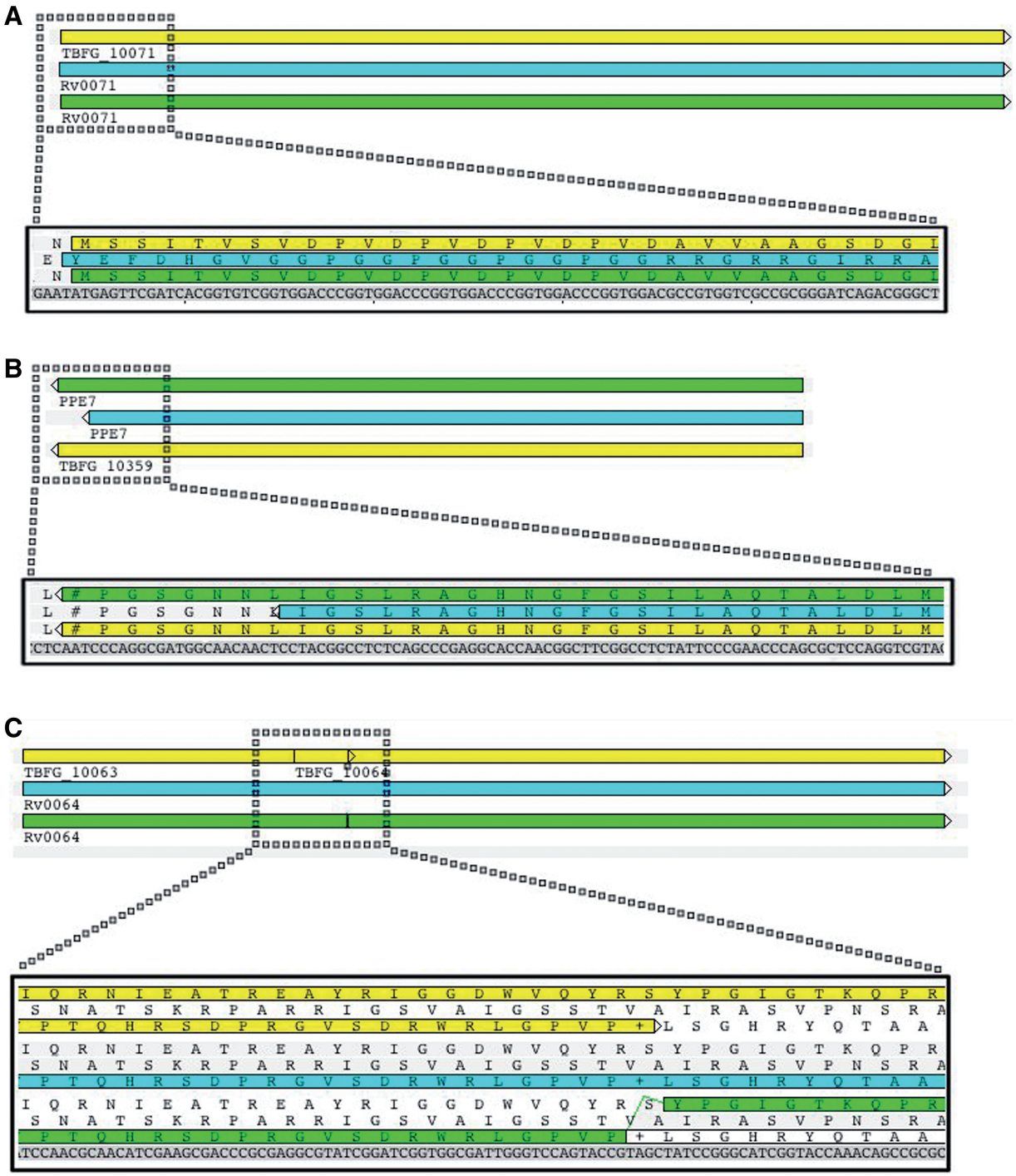

Figure 3. RATT corrections of transferred annotations. Annotation from H37Rv were transferred onto the F11 sequence (pale blue), corrected (green) and then compared with the existing strain F11 annotation in EMBL (yellow). (**A** and **B**) The correction of start and stop codons, respectively. In a more complex mapping situation (**C**), where all three reading frames are shown for clarity, RATT maps a large single coding sequence (CDS) from H37Rv to a locus within F11 that includes several in-frame stop codons. By inserting a frameshift (i.e. to indicate a pseudogene) the conceptual translation is preserved. This contrasts with two overlapping genes predicted as part of the F11 genome project.

Table 1. Comparison of three annotation transfers by RATT

Transfer	Features transferred/total	CDS transferred/total	Partial transfers/CDS	Corrected	Manual correction needed
M. tuberculosis	9406/9557	3955/3999	44	113	3
P. chab : *P. berg*	626/686	626/686	78	139	21
P. berg v9:v10	4970/5105	4889/4902	157/135	7	3

present in strain H37Rv but not annotated in the publicly available F11 sequence (see Supplementary Data). Further, RATT suggested a frameshift in the transfer of gene Rv0064 and flagged it to be revisited, in contrast to two gene models at that locus within the existing F11 annotation (Figure 3C). After manual inspection, including sequence comparisons with related species, we concluded that RATT's suggestion was more likely to be correct.

In addition to annotation transfer we performed *de novo* gene prediction using Glimmer with the *long-orfs* setting (7). CDS prediction is relatively fast, taking <2 min on the same machine. However, a predicted CDS must subsequently be annotated using BLAST, which takes much longer. Glimmer has a slight tendency to over-predict coding sequences: 284 extra CDS were discovered. Moreover, while there were considerable overlaps between Glimmer predictions and *bona fide* gene models, only 64% of the predictions overlapped precisely with those models. In contrast, RATT managed >88% overlap (Table 3).

Transfer between species

We also compared RATT's ability to transfer annotation between two closely related eukaryotic species: *P. chabaudi* chromosome 14 (2.5 Mb, the longest chromosome) was used as a reference for the *P. berghei* chromosome 14. The transfer and correction was complete in less than a minute. While synteny is mostly conserved along the chromosomes, 10.57% of the *P. chabaudi* reference had no synteny with the query, and 7.64% of the query had no synteny with the reference.

The annotation transfer was comprehensive: of 686 CDSs only 60 could not be transferred (171 exons). Of those CDS features that did not transfer, the majority (43) were either not present or their orthologues were highly diverged in *P. berghei*. Furthermore, 40 of the 43 models were present in the subtelomeres—highly plastic regions in which synteny between *Plasmodium* spp. are not conserved. The 17 remaining models were partial transfers; conserved regions were assigned while the remaining divergent exons were not. The 626 successfully assigned models included 160 corrected CDSs and 21 models with unresolved corrections. Where possible, as a guide to the annotator, RATT will suggest corrected exon boundaries. Of 17 'fixed' splice sites, 9 were correct and a further 4 were within 15 base pairs of the actual splice site; the remaining four were conservative underestimates of substantial exon extensions in *P. berghei* relative to *P. chabaudi*. Tables 1 and 2 summarize the results.

The complete annotation of the *P. berghei* genome can be found on GeneDB (http://www.genedb.org/).

Table 2. Corrections by RATT in three annotation transfers

Transfer	Wrong *start* codons (corrected/total)	Wrong *stop* codons (corrected/total)	Number of frameshifts (corrected/total)	Wrong splice sites (corrected/total)
M. tuberculosis	44/44	62/62	40/43	–
P. chab:*P. berg*	37/40	88/97	61/70	9/27
P. berg v9:v10	0/0	4/4	0/3	1/1

Table 3. Comparison of predicted CDS annotations with original strain F11 annotations

Annotation method	Predicted CDSs	Start matches	End matches	Exact matches
Glimmer	4234	2525	3838	2522
RATT	3955	3505	3821	3495

Predicted annotations by RATT (by transfer from the H37Rv strain annotation) are compared with the existing 3950 CDS annotations in the public version of strain F11.

Transfer between assemblies

Re-sequencing and assembly of *P. berghei* improved and, consequently, changed the underlying sequence of the *P. berghei* genome. The new genome build corrected 257 single base errors, 76 short indels by using iCORN (19), and deleted a 54k subtelomeric region (previously inserted as an assembly error). We applied RATT using the 'Assembly.Repetitive' option to the old and new *P. berghei* genomes. It took ~20 min to run. In the transfer, 13 gene models were not mapped due to the deletion (Table 1). The remaining untransferred features were gap tags, which are not mapped, as gaps do not have synteny. As the *P. berghei* genome is undergoing finishing, 121 gene models spanning gaps and 47 pseudogenes were flagged to prevent correction. All but seven gene models mapped perfectly, and the latter had frameshifts that could not be corrected automatically. These frameshifts were due to small indel corrections in homopolymer regions. The other corrections were accurate.

DISCUSSION

As the number of new genome projects is increasing dramatically as sequencing costs become lower, better annotation tools are needed. Here we present RATT as the first stand-alone solution for direct transfer of annotation between different versions of sequence assemblies or between sequences of related strains and species.

Our strategy to transfer the annotation with the help of conserved synteny is new. The alternative is a complete *ab initio* gene prediction and *de novo* annotation, which is prone to under- or over-prediction, as illustrated by the GLIMMER example. When transferring between assemblies, mapping transfer is highly accurate. Even repetitive genes, such as the PPE gene family in the *M. tuberculosis* strain-to-strain transfer, could be mapped unambiguously as RATT transfers genes based on unique flanking sequences, an advantage over similarity searching alone. In addition, our comparative transfer method can improve upon manual annotation added to genomes in isolation. For example, an undetected frameshift, representing a putative pseudogene was identified in the public *M. tuberculosis* strain F11 annotation

(Figure 3C), as well as 99 additional putative coding sequences. Most of these coding sequences encode short proteins within unknown functions, which probably accounts for their absence in the F11 EMBL file. However, their conservation between numerous strains adds weight to the possibility that they are real.

Alternative strategies

To evaluate the performance in terms of quality and speed of RATT, we attempted to compare it with other annotation transfer tools. The only published example, GATU, is a Java applet intended to map annotation between viral genomes. However, this tool failed to run on the large datasets in our study. The absence of an alternative, readily available annotation transfer solutions may reflect the nature of many previous large genome-sequencing projects. In particular, projects that have employed large clones to walk across a genome generate islands of perfect sequence that do not require breaking up and re-annotating on reassembly. In contrast, increasingly prevalent high-throughput shotgun-sequencing projects from a range of technologies undergo significant rearrangement as assemblies improve from draft to finished genome and present a real challenge for updating annotation.

While, RATT cannot identify genes in regions where no synteny exists, a subsequent *ab initio* annotation could be restricted to a few regions/models, resulting in a massively reduced workload for the annotator. For example, in the annotation transfer between Plasmodium species, putative coding sequences for variable surface proteins could not be mapped between the reference and query. These genes are present in highly plastic subtelomeric regions where such variation will inevitably require manual annotation, regardless of the methods deployed. Indeed, there are two major advantages to using synteny-based annotation transfer. First manual annotation can be concentrated on novel or changed genes and regions with indels. Second, gene-order information is preserved, which obviates the need to reconstruct orthologue links between strains or species during subsequent comparative genomics analyses, such as the calculation of dN/dS ratios or phylogenetic tree construction.

Using RATT in a pipeline

Although RATT is a stand-alone tool it has not been developed in isolation, and can work as part of an established annotation pipeline. Genome builds are frequently corrected by gap closing/contig reorienting (3), or SNP calling and consensus correction (19). RATT is intended to operate downstream of these tools, quickly and unambiguously transferring annotation between genome builds. RATT is also aimed at facilitating the annotation of genomes from new species/strains from resequencing projects produced using tools such as ABACAS (20). In the context of annotating bacterial genomes, RATT a multiple transfer option is also provided (see Supplementary Table S1) to transfer annotation from multiple genomes on to one query. This is designed to identify and annotate regions where large interspecies transfers or plasmid integrations have occurred. The MUMmer package can attribute the most similar regions of a reference sequence to the query subsequence (if two references are identical, one is randomly picked). Therefore, RATT transfers onto the query the most similar bits of the diverse references.

In conclusion, we present RATT for transferring annotation rapidly and efficiently between similar genomes. Where an annotated reference exists, RATT surpasses available automated annotation methods. Its relative simplicity means it can be used by laboratories that lack extensive bioinformatics expertise, as well as by high-throughput sequencing centres. Not only does RATT transfer annotation where synteny has been preserved, it also highlights areas where rearrangements have occurred or sequences are highly divergent. This allows the annotator to focus on variable regions without their effort being diverted to already well-annotated and conserved regions. We are using RATT successfully on many genome projects at the Wellcome Trust Sanger Institute.

SUPPLEMENTARY DATA

Supplementary Data are available at NAR Online.

ACKNOWLEDGEMENTS

We would like to thank Ulrike Böhme for the annotation of *P. berghei* and testing the program. We thank Adam Reid and Jason Tsai for comments and reviewing the article.

FUNDING

European Union 7th framework EVIMalaR (to T.D.O.); Wellcome Trust (grant WT 085775/Z/08/Z to G.D., M.B.). Funding for open access charge: Wellcome Trust Sanger Institute.

Conflict of interest statement. None declared.

REFERENCES

1. Fox,S., Filichkin,S. and Mockler,T.C. (2009) Applications of ultra-high-throughput sequencing. *Methods Mol. Biol.*, **553**, 79–108.
2. Clarke,J., Wu,H.C., Jayasinghe,L., Patel,A., Reid,S. and Bayley,H. (2009) Continuous base identification for single-molecule nanopore DNA sequencing. *Nat. Nanotechnol.*, **4**, 265–270.
3. Tsai,I.J., Otto,T.D. and Berriman,M. (2010) Improving draft assemblies by iterative mapping and assembly of short reads to eliminate gaps. *Genome Biol.*, **11**, R41.
4. Zerbino,D.R. and Birney,E. (2008) Velvet: algorithms for de novo short read assembly using de Bruijn graphs. *Genome Res.*, **18**, 821–829.
5. Li,H., Ruan,J. and Durbin,R. (2008) Mapping short DNA sequencing reads and calling variants using mapping quality scores. *Genome Res.*, **18**, 1851–1858.
6. Li,H. and Durbin,R. (2009) Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics*, **25**, 1754–1760.

7. Delcher,A.L., Bratke,K.A., Powers,E.C. and Salzberg,S.L. (2007) Identifying bacterial genes and endosymbiont DNA with Glimmer. *Bioinformatics*, **23**, 673–679.
8. Korf,I., Flicek,P., Duan,D. and Brent,M.R. (2001) Integrating genomic homology into gene structure prediction. *Bioinformatics*, **17(Suppl. 1)**, S140–S148.
9. Ashburner,M., Ball,C.A., Blake,J.A., Botstein,D., Butler,H., Cherry,J.M., Davis,A.P., Dolinski,K., Dwight,S.S., Eppig,J.T. *et al.* (2000) Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat. Genet.*, **25**, 25–29.
10. Stein,L. (2001) Genome annotation: from sequence to biology. *Nat. Rev. Genet.*, **2**, 493–503.
11. Hubbard,T.J., Aken,B.L., Ayling,S., Ballester,B., Beal,K., Bragin,E., Brent,S., Chen,Y., Clapham,P., Clarke,L. *et al.* (2009) Ensembl 2009. *Nucleic Acids Res.*, **37**, D690–D697.
12. Davila,A.M., Lorenzini,D.M., Mendes,P.N., Satake,T.S., Sousa,G.R., Campos,L.M., Mazzoni,C.J., Wagner,G., Pires,P.F., Grisard,E.C. *et al.* (2005) GARSA: genomic analysis resources for sequence annotation. *Bioinformatics*, **21**, 4302–4303.
13. Almeida,L.G., Paixao,R., Souza,R.C., Costa,G.C., Barrientos,F.J., Santos,M.T., Almeida,D.F. and Vasconcelos,A.T. (2004) A System for Automated Bacterial (genome) Integrated Annotation–SABIA. *Bioinformatics*, **20**, 2832–2833.
14. Aziz,R.K., Bartels,D., Best,A.A., DeJongh,M., Disz,T., Edwards,R.A., Formsma,K., Gerdes,S., Glass,E.M., Kubal,M. *et al.* (2008) The RAST Server: rapid annotations using subsystems technology. *BMC Genomics*, **9**, 75.
15. Markowitz,V.M., Chen,I.M., Palaniappan,K., Chu,K., Szeto,E., Grechkin,Y., Ratner,A., Anderson,I., Lykidis,A., Mavromatis,K. *et al.* (2010) The integrated microbial genomes system: an expanding comparative analysis resource. *Nucleic Acids Res.*, **38**, D382–D390.
16. Carver,T., Berriman,M., Tivey,A., Patel,C., Bohme,U., Barrell,B.G., Parkhill,J. and Rajandream,M.A. (2008) Artemis and ACT: viewing, annotating and comparing sequences stored in a relational database. *Bioinformatics*, **24**, 2672–2676.
17. Kurtz,S., Phillippy,A., Delcher,A.L., Smoot,M., Shumway,M., Antonescu,C. and Salzberg,S.L. (2004) Versatile and open software for comparing large genomes. *Genome Biol.*, **5**, R12.
18. Otto,T.D. (2008) Métodos para montagem e anotação de genomas (PhD thesis), Fundação Oswaldo Cruz, Rio de Janeiro.
19. Otto,T.D., Sanders,M., Berriman,M. and Newbold,C. (2010) Iterative correction of reference nucleotides (iCORN) using second generation sequencing technology. *Bioinformatics*, **26**, 1704–1707.
20. Assefa,S., Keane,T.M., Otto,T.D., Newbold,C. and Berriman,M. (2009) ABACAS: algorithm-based automatic contiguation of assembled sequences. *Bioinformatics*, **25**, 1968–1969.