Bioinformatics, 34(24), 2018, 4293–4296
doi: 10.1093/bioinformatics/bty548
Advance Access Publication Date: 29 June 2018
Applications Note

OXFORD

Sequence analysis

MrBait: universal identification and design of targeted-enrichment capture probes

Tyler K. Chafin*, Marlis R. Douglas and Michael E. Douglas

Department of Biological Sciences, University of Arkansas, Fayetteville, AR 72701, USA

*To whom correspondence should be addressed.
Associate Editor: John Hancock

Received on April 16, 2018; revised on June 11, 2018; editorial decision on June 27, 2018; accepted on June 28, 2018

Abstract

Motivation: It is a non-trivial task to identify and design capture probes ('baits') for the diverse array of targeted-enrichment methods now available (e.g. ultra-conserved elements, anchored hybrid enrichment, RAD-capture). This often involves parsing large genomic alignments, followed by multiple steps of curating candidate genomic regions to optimize targeted information content (e.g. genetic variation) and to minimize potential probe dimerization and non-target enrichment.
Results: In this context, we developed MrBait, a user-friendly, generalized software pipeline for identification, design and optimization of targeted-enrichment probes across a range of target-capture paradigms. MrBait is an open-source codebase that leverages native parallelization capabilities in Python and mitigates memory usage via a relational-database back-end. Numerous filtering methods allow comprehensive optimization of designed probes, including built-in functionality that employs BLAST, similarity-based clustering and a graph-based algorithm that 'rescues' failed probes.
Availability and implementation: Complete code for MrBait is available on GitHub (https://github.com/tkchafin/mrbait), and is also available with all dependencies via one-line installation using the conda package manager. Online documentation describing installation and runtime instructions can be found at: https://mrbait.readthedocs.io.
Contact: tkchafin@uark.edu
Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

The application of next-generation sequencing methods to non-model organisms has been facilitated by a diverse array of novel 'reduced-representation' methods, whereby a consistent subset of the genome is targeted for sequencing across hundreds or thousands of individuals (Davey *et al.*, 2011). One major trajectory for these methods is to target specific regions for sequencing, by utilizing the hybridization of oligonucleotide probes (or 'baits') to DNA fragments containing complementary sequences, followed by the subsequent separation of these target molecules (Mamanova *et al.*, 2010). Although target-enrichment methods share this general design, numerous derivative methods have been developed and optimized for specific applications. For example, one commonly-applied paradigm is the enrichment of ultra-conserved genomic elements (UCEs), by identifying regions in divergent lineages with extremely low mutation accumulation, with the assay of genetic variation flanking these UCEs as the ultimate goal (e.g. McCormack *et al.*, 2012; Gnirke *et al.*, 2009). Another popular approach is to specifically anchor probes to coding sequences (Bi *et al.*, 2012; Lemmon *et al.*, 2012). Similarly, targeted fragmentation using restriction enzymes (per RADcap, Rapture) is also utilized, followed by a more specific reduction using capture probes (Ali *et al.*, 2016; Hoffberg *et al.*, 2016).

A universal requirement for these methods is that genomic resources be available *a priori*, or at least developed as a pre-requisite to application and from which probe sequences can then be designed. Transparent workflows are not always available (but see Faircloth, 2017 for such a treatment for UCEs), and are thus counter-productive to this endeavor. Some software does exist, but

is often designed for a specific targeted-enrichment approach (Anil *et al.*, 2018; Faircloth, 2017; Johnson *et al.*, 2016). One recently published option (BaitsTools; Campana, 2017) is flexible enough to allow multiple inputs and enrichment schemes, yet does not natively incorporate post-processing steps to optimize bait-specificity. Here, we provide a flexible, user-friendly software, MrBait, that can be generalized to any targeted-enrichment paradigm. MrBait is not only open-source but also employs native Python parallelization. In addition, its memory usage, data management, portability and iterative probe design are efficiently promoted through a relational database back-end using SQLite.

2 Features and user interface

MrBait stores genomic regions or alignments, candidate target regions and candidate probe sequences as an SQLite relational-database with a Python wrapper and command-line interface (CLI). The database can be efficiently parsed then successively re-parsed, so as to allow fast exploration of numerous bait-design and filtering schemes. The general process is as follows:

i. Build a consensus catalog of genomic regions by parsing alignments (as .xmfa, .maf, or .loci output of pyRAD) or genomes (as .fasta, annotated optionally with .vcf or .gff).
ii. Apply a sliding window along each consensus locus to find candidate target regions (depending on user specifications, e.g. indels allowed, frequency of flanking SNPs, etc.).
iii. Target filtering of regions (e.g. by GC content, maximum allowable pairwise identities, BLAST identity to potential contaminant genome) and resolve conflicts (if targets are within specified proximity along a scaffold or chromosome).
iv. Design a prospective bait set from passing target regions based on user-specified schema: tiling, or positional anchoring (e.g. centered or terminal within target region). If baits will be used for more distantly related taxa, polymorphism can be included to mitigate systematic bias in downstream molecular application.
v. Filtering and selection criteria (as in 3) are then applied to baits.
vi. The pipeline can be resumed and any steps iteratively re-visited by providing the SQLite database file (resulting in a significant reduction in runtime for successive runs).

Data are input in a variety of configurations: (i) Whole genomes (.fasta), with optional accompanying structural elements (as .gff) or variant information (.vcf); (ii) multiple-genome alignment using the .maf output of MAFFT (Katoh and Standley, 2013) or the .xmfa format of progressiveMauve (Darling *et al.*, 2010) or (iii) reduced-representation alignments using the .loci format of pyRAD (Eaton, 2014). Numerous filtering criteria are employed natively within MrBait and specified using the CLI, which allows target regions or designed probe sequences to be constrained in a variety of ways: with masking information from programs such as RepeatMasker (Smit *et al.*, 2013), via co-ordinates within a full genome to approximate all or a subset of specific genomic elements, by number of variant sites assayed (e.g. only retaining baits flanking known SNPs), or through other criteria (e.g. GC content, ambiguity or gap content). Targets or probes can be also filtered inclusively by optimizing specificity to a target genome, or exclusively by minimizing hits to a non-target (e.g. contaminant) genome using an internal call to NCBI-BLAST+ with a user-provided genome or database (Altschul *et al.*, 1990). Probe-probe hybridization in downstream molecular application can also be circumvented using built-in clustering in MrBait via the VSEARCH algorithm (Rognes *et al.*, 2016). Clustering results are used to build an undirected graph, with nodes as target regions (or baits) and edges representing pairwise alignments greater than some threshold identity and alignment length (user-provided). MrBait employs a naïve approach to identify the maximal independent set within this graph, optionally weighting nodes according to several user options so as to 'rescue' optimal targets without retaining edges. The motivation behind this approach is to retain a maximal number of baits without duplication. If undesired, this behavior can be easily disabled (or modified) using the CLI.

3 Benchmarking

Runtime and memory-usage were gauged using a ddRAD dataset generated for Whitetail deer (*Odocoileus virginianus*) from Arkansas. Samples ($N = 48$) were digested with PstI and MspI restriction enzymes, size selected between ~375–525 bp and sequenced on an Illumina HiSeq 2500 with paired-end 150 bp reads. Resulting data were assembled in pyRAD with 51 931 loci post-filtering. MrBait then processed these data. Requirements were as follows: A minimum per locus coverage of 25% for individuals; target regions with 1–10 flanking SNPs and baits 60 bp in length tiled across target regions at 1.5X coverage. These yielded 44 808 loci with a conserved region sufficient for bait design, with 27 102 candidate target regions flanking a sufficient number of SNPs. From these, a total of 43 342 baits were output in 392 s across 4 threads on a 2014 iMac desktop. Identical runs with 1, 2 and 3 threads took 1182 s, 591 s and 399 s, respectively, with a greater-than-linear speedup as core number increased. Peak memory usage increased sub-linearly with core count, at 120 Mb for 1 thread and 300 Mb for 4 threads on this dataset. For comparison, BaitsTools (Campana, 2017), with approximately comparable parameter settings, ran in 750 s using the 'SNP-targeting' strategy for pyrad2baits (single-threaded) with no post-processing. The time discrepancy results from the initial setup of the relational database back-end (Step 1), which is the largest overhead for MrBait. Subsequent runs with re-parameterization for target selection, filtering, and bait design ran comparatively quickly. For example, the existing SQLite database was passed to MrBait, with additional filtering on GC content for targets (between 0.3 and 0.7) and a new bait length of 80, in just 12 s (and resulted in 20 023 passed baits). This demonstrates the utility of the database approach in facilitating iterative probe design and exploring parameter values (see Supplementary File S1 for full bait sequences from this final run).

4 Comparison with existing methods

We parsed the existing Whitetail Deer ddRAD dataset so as to compare performance of bait design by MrBait versus BaitsTools, and did so by maintaining maximum consistency in parameter settings between the two programs. We ran the 'pyrad2baits' program in BaitsTools with bait length of 80, a minimum of 20 individuals per locus, 50% overlap between tiled baits and with baits containing gaps or ambiguous (N) characters excluded. These settings were replicated in MrBait, with no additional filtering to make comparison more appropriate. We also filtered the resulting bait sets by eliminating baits containing SNPs. This was accomplished natively within MrBait, and by using custom post-processing scripts for the BaitsTools output. The capacity of MrBait to filter targeted regions by 'informativeness' was not implemented, nor was BLAST-filtering

for specificity. BaitsTools identified 14 276 non-variable bait sequences after manual post-processing in Python (successfully targeting 41.5% of the 20 912 loci with sufficient coverage), whereas MrBait found 12 084 baits, targeting 44% of loci. This demonstrates that both softwares can discover roughly equivalent sets of bait sequences, although in this case BaitsTools output required additional manual filtering while these steps were integrated in MrBait.

To compare accuracy of our bait design, we examined the data for 964 RAD loci from *Wisteria*, curated and assembled from paired-end sequencing data by Hoffberg *et al.* (2016). In parsing these loci, we excluded most of the native filtering methods in MrBait to keep results comparable. MrBait identified 1924 conservative 90-mer baits targeting all 964 loci, compared to the 1928 identified by Hoffberg *et al.*, again indicating that MrBait will produce bait sets comparable to those from other existing methods.

However, users may find the additional utilities included natively in MrBait useful for reducing size of the total bait set, for example to improve specificity of the candidate baits (e.g. to reduce non-target enrichment), to reduce potential for ascertainment bias, or to reduce the overall number of sequences for synthesis (e.g. to meet budgetary requirements). For example, users may desire to remove baits which align to one another, as these can be non-specific to the intended locus (Faircloth, 2017), or remove baits with extreme GC content which may show a phylogenetic bias when applied to broader taxa (Bossert *et al.*, 2017). When applying a GC content filter (GC% >70 or <30), to the *Wisteria* dataset, 475 baits failed, while 25 failed when a conservative duplicate filter was applied (pairwise alignment of >80% identity over >80% of the bait length). Hoffberg *et al.* reported very high matrix occupancy with the designed bait set (99.8% of loci for 90% of samples, with a 4X coverage cutoff), however application of the uncurated bait set at a deeper phylogenetic scale could expose systematic bias associated with GC heterogeneity (e.g. Bossert *et al.*, 2017), or with phylogenetic information content targeted by each bait, depending on the phylogenetic scale and intended method of downstream analysis (Meiklejohn *et al.*, 2016). An additional major consideration is the potential for non-target capture from vastly different sources (e.g. bacterial contaminants), however extensive bioinformatic processing such as via native BLAST filtering in MrBait can significantly mitigate this (Bossert and Danforth, 2018). Users are cautioned to consider any ascertainment biases which may be introduced, particularly when designing bait sets for a different phylogenetic scale than is available (e.g. as reference genomes) for bait design.

5 Conclusion

We provide a customizable and extensible open-source software (MrBait) that facilitates rapid and user-friendly bait development for an array of molecular applications (e.g. ultra-conserved elements, RAD-capture). It simultaneously identifies conservative 'target' regions in user-provided sequence data, designs probes to enrich them and curates the resulting bait set. It also incorporates an array of native filtering strategies to help minimize downstream synthesis of problematic baits (e.g. duplicates), and to maximize specificity of baits to a target genome or desirable elements within them (e.g. known SNPs, or genomic features such as exons). MrBait adopts an SQL relational database back-end to minimize the problem of data files that necessitate high memory loads as well as significant I/O computational time. This allows users to rapidly re-parse the database with multiple different filtering criteria and promotes efficient exploration of parameter space and optimal bait sets for bait specificity and number (which affects synthesis cost). Comparisons with existing methods indicate that MrBait is similar in terms of quantity of targets discovered and runtime efficiency. Documentation and a full description of runtime options can be found at: https://mrbait.readthedocs.io.

Acknowledgements

The authors would like to thank Zach D. Zbinden for contributing to the code base, Pam L. McDill for lab work, Bradley Martin for testing and Arkansas Game and Fish Commission for providing tissues for bait development in Whitetail Deer. They also thank the Editors and two anonymous Reviewers for comments and valuable suggestions improving the software and this manuscript.

Funding

This project was supported by computational resources provided by XSEDE Startup Allocation TG-BIO160058 (MED) and Research Allocation TG-BIO160065 (MRD), and by University of Arkansas Endowments (Bruker Professorship in Life Sciences to MRD and 21st Century Chair in Global Climate Change Biology to MED).

Conflict of Interest: none declared.

References

Ali,O.A. *et al.* (2016) RAD capture (Rapture): flexible and efficient sequence-based genotyping. *Genetics*, **202**, 389–400.

Altschul,S.F. *et al.* (1990) Basic local alignment search tool. *J. Mol. Biol.*, **215**, 403–410.

Anil,A. *et al.* (2018) HiCapTools: a software suite for probe design and proximity detection for targeted chromosome conformation capture applications. *Bioinformatics*, **34**, 675–677.

Bi,K. *et al.* (2012) Transcriptome-based exon capture enables highly cost-effective comparative genomic data collection at moderate evolutionary scales. *BMC Genomics*, **13**, 403.

Bossert,S. *et al.* (2017) The impact of GC bias on phylogenetic accuracy using targeted enrichment phylogenomic data. *Mol. Phylogenet. Evol.*, **111**, 149–157.

Bossert,S. and Danforth,B.N. (2018) On the universality of target-enrichment baits for phylogenomic research. *Methods Ecol. Evol.*, **9**, 1453–1460.

Campana,M.G. (2017) BaitsTools: software for hybridization capture bait design. *Mol. Ecol. Res.*, **18**, 1–6.

Darling,A.E. *et al.* (2010) Progressivemauve: multiple genome alignment with gene gain, loss and rearrangement. *PLoS One*, **5**, e11147.

Davey,J.W. *et al.* (2011) Genome-wide genetic marker discovery and genotyping using next-generation sequencing. *Nat. Rev. Genet.*, **12**, 499–510.

Eaton,D.A.R. (2014) PyRAD: assembly of de novo RADseq loci for phylogenetic analyses. *Bioinformatics*, **30**, 1844–1849.

Faircloth,B.C. (2017) Identifying conserved genomic elements and designing universal bait sets to enrich them. *Methods Ecol. Evol.*, **8**, 1103–1112.

Gnirke,A. *et al.* (2009) Solution hybrid selection with ultra-long oligonucleotides for massively parallel targeted sequencing. *Nat. Biotechnol.*, **27**, 182–189.

Hoffberg,S.L. *et al.* (2016) RADcap: sequence capture of dual-digest RADseq libraries with identifiable duplicates and reduced missing data. *Mol. Ecol. Res.*, **16**, 1264–1278.

Johnson,M.G. *et al.* (2016) HybPiper: extracting coding sequence and introns for phylogenetics from high-throughput sequencing reads using target enrichment. *Appl. Plant Sci.*, **4**, 1600016.

Katoh,K. and Standley,D.M. (2013) MAFFT multiple sequence alignment software version 7: improvements in performance and usability. *Mol. Biol. Evol.*, **30**, 772–780.

Lemmon,A.R. *et al.* (2012) Anchored hybrid enrichment for massively high-throughput phylogenomics. *Syst. Biol.*, **61**, 727–744.

Mamanova,L. *et al.* (2010) Target-enrichment strategies for next-generation sequencing. *Nat. Methods*, **7**, 111–118.

McCormack,J.E. *et al*. (2012) Ultraconserved elements are novel phylogenomic markers that resolve placental mammal phylogeny when combined with species tree analysis. *Genome Res*., **22**, 746–754.

Meiklejohn,K.A. *et al*. (2016) Analysis of a rapid evolutionary radiation using ultraconserved elements: evidence for a bias in some multispecies coalescent methods. *Syst. Biol*., **65**, 612–627.

Rognes,T. *et al*. (2016) VSEARCH: a versatile open source tool for metagenomics. *PeerJ*, **4**, e2409v1.

Smit,A. *et al*. (2013) *RepeatMasker 4.0*. Seattle, WA.