Video Article

Novel Sequence Discovery by Subtractive Genomics

Kathryn C. Asalone[1], Megan M. Nelson[1], John R. Bracht[1]

[1]Biology Department, American University

Correspondence to: John R. Bracht at jbracht@american.edu

URL: https://www.jove.com/video/58877
DOI: doi:10.3791/58877

Keywords: Genomic subtraction, qPCR, BLAST, Python, Read mapping, *De novo* assembly, Primer design

Date Published: 12/7/2018

Citation: Asalone, K.C., Nelson, M.M., Bracht, J.R. Novel Sequence Discovery by Subtractive Genomics. *J. Vis. Exp.* (), e58877, doi:10.3791/58877 (2018).

Abstract

Subtractive genomics can be used in any research where the goal is to identify the sequence of a gene, protein, or general region that is embedded in a larger genomic context. Subtractive genomics enables a researcher to isolate a target sequence of interest (T) by comprehensive sequencing and subtracting out known genetic elements (reference, R). The method can be used to identify novel sequences such as mitochondria, chloroplasts, viruses, or germline restricted chromosomes, and is particularly useful when T cannot be easily isolated from R. Beginning with the comprehensive genomic data (R + T), the method uses Basic Local Alignment Search Tool (BLAST) against a reference sequence, or sequences, to remove the matching known sequences (R), leaving behind the target (T). For subtraction to work best, R should be a relatively complete draft that is missing T. Since sequences remaining after subtraction are tested through quantitative Polymerase Chain Reaction (qPCR), R does not need to be complete for the method to work. Here we link computational steps with experimental steps into a cycle that can be iterated as needed, sequentially removing multiple reference sequences and refining the search for T. The advantage of subtractive genomics is that a completely novel target sequence can be identified even in cases in which physical purification is difficult, impossible, or expensive. A drawback of the method is finding a suitable reference for subtraction and obtaining T-positive and negative samples for qPCR testing. We describe our implementation of the method in the identification of the first gene from the germline-restricted chromosome of zebra finch. In that case computational filtering involved three references (R), sequentially removed over three cycles: an incomplete genomic assembly, raw genomic data, and transcriptomic data.

Video Link

The video component of this article can be found at https://www.jove.com/video/58877/

Introduction

The purpose of this method is to identify a novel target (T) genomic sequence, either DNA or RNA, from a genomic context, or reference (R) (**Figure 1**). The method is most useful if the target cannot be physically separated, or it would be expensive to do so. Only a few organisms have perfectly finished genomes for subtraction, so a key innovation of our method is the combination of computational and bench methods into a cycle enabling researchers to isolate target sequences when the reference is imperfect, or a draft genome from a non-model organism. At the end of a cycle, qPCR testing is used to determine whether more subtraction is needed. A validated candidate T sequence will show statistically greater detection in known T-positive samples by qPCR.

Incarnations of the method have been implemented in discovery of new bacterial drug targets that do not have host homologs[1,2,3,4] and identification of novel viruses from infected hosts[5,6]. In addition to identification of T, the method can improve R: we recently used the method to identify 936 missing genes from the zebra finch reference genome and a new gene from a germline-only chromosome (T)[7]. Subtractive genomics is particularly valuable when T is likely to be extremely divergent from known sequences, or when the identity of T is broadly undefined, as in the zebra finch germline-restricted chromosome[7].

By not requiring positive identification of T beforehand, a key advantage of subtractive genomics is that it is unbiased. In a recent study, Readhead *et al.* examined the relationship between Alzheimer's disease and viral abundance in four brain regions. For viral identification, Readhead *et al.* created a database of 515 viruses[8], severely limiting the viral agents that their study could identify. Subtractive genomics could have been used to compare the healthy and Alzheimer's genomes in order to isolate possible novel viruses associated with the disease, regardless of their similarity to known infectious agents. While there are 263 known human-targeting viruses, it has been estimated that approximately 1.67 million undiscovered viral species exist, with 631,000-827,000 of them having a potential to infect humans[9].

Isolation of novel viruses is an area in which subtractive genomics is particularly effective, but some studies may not need such a stringent method. For example, studies identifying novel viruses have used unbiased high-throughput sequencing followed by reverse transcription and BLASTx for viral sequences[5] or enriching of viral nucleic acids to extract and reverse transcribe viral sequences[6]. While these studies employed *de novo* sequencing and assembly, subtraction was not used because the target sequences were positively identified through BLAST. If the viruses were completely novel and not related (or distantly related) to other viruses, subtractive genomics would have been a useful technique. The benefit of subtractive genomics is that sequences that are completely new can be obtained. If the organism's genome is known, it can be

subtracted out to leave any viral sequences. For example, in our published study we isolated a novel viral sequence from zebra finch through subtractive genomics, though it was not our original intent[7].

Subtractive genomics has also proved useful in the identification of bacterial vaccine targets, motivated by the dramatic rise in antibiotic resistance[1,2,3,4]. To minimize the risk of autoimmune reaction, researchers narrowed down the potential vaccine targets by subtracting any proteins that have homologs in the human host. One particular study, looking at *Corynebacterium pseudotuberculosis*, performed subtraction of vertebrate host genomes from several bacterial genomes to ensure that possible drug targets would not affect proteins in the hosts leading to side effects[1]. The basic work flow of these studies is to download the bacterial proteome, determine vital proteins, remove redundant proteins, use BLASTp to isolate the essential proteins, and BLASTp against host proteome to remove any proteins with host homologs[1,2,3,4]. In this case, subtractive genomics ensure that the vaccines developed will not have any off-target effects in the host[1,2,3,4].

We used subtractive genomics to identify the first protein-coding gene on a germline-restricted chromosome (GRC) (in this case, T), which is found in germlines but not somatic tissue of both sexes[10]. Before this study, the only genomic information that was known about the GRC was a repetitive region[11]. *De novo* assembly was performed on RNA sequenced from ovary and teste tissues (R+T) from adult zebra finches. The computational elimination of sequences was performed using published somatic (muscle) genome sequence (R_1)[12], its raw (Sanger) read data (R_2), and a somatic (brain) transcriptome (R_3)[13]. The sequential use of three references was driven by the qPCR testing at step 5 of each cycle (**Figure 2A**), showing that additional filtering was required. The discovered α-SNAP gene was confirmed through qPCR from DNA and RNA, and cloning and sequencing. We show in our example that this method is flexible: it is not dependent on matching nucleic acids (DNA vs RNA) and that subtraction can be performed with references (R) that are comprised of assemblies or raw reads.

Protocol

1. *De novo* Assemble Starting Sequence

NOTE: Any Next-Generation Sequence (NGS) data can be used, as long as an assembly can be produced from those data. Suitable input data includes Illumina, PacBio, or Oxford Nanopore reads assembled into a fasta file. For concreteness, this section describes an Illumina-based transcriptomic assembly specific to the zebra finch study we performed[7]; however be aware that the specifics will vary by project. For our example project, raw data were derived from a MiSeq and approximately 10 million paired reads were obtained from each sample.

1. Use Trimmomatic 0.32[14] to remove Illumina adaptors and low-quality bases. On the command line, enter:
   java -jar trimmomatic-0.32.jar PE -phred33 forward.fq.gz reverse.fq.gz -baseout quality_and_adaptor_trimmed ILLUMINACLIP:TruSeq3-PE.fa:2:30:10 LEADING:3 TRAILING:3 SLIDINGWINDOW:4:20 MINLEN:40
2. Use PEAR[15] v. 0.9.6 to create high-quality merged reads from trimmomatic output paired reads, using default parameters. On the command line, enter:
   pear -f <quality_and_adaptor_trimmed_forward_paired_reads.fq> -r <quality_and_adaptor_trimmed_reverse_paired_reads.fq>
3. Use Reptile v. 1.1[16] to error-correct the reads produced through PEAR. Follow the step-by-step protocol described in[17].
4. Use Trinity v. 2.4.0[18] in default mode to assemble the corrected sequences. For strand-specific libraries, use the -SS_lib_type parameter. The output is a fasta file (your_assembly.fasta). On the command line, enter:
   Trinity --seqType fq --SS_lib --max_memory 10G --output Trinity_output --left quality_and_adaptor_trimmed_forward_paired_reads.fq --right quality_and_adaptor_trimmed_reverse_paired_reads.fq --CPU 10
   NOTE: The output will be placed in a new directory, Trinity_output, and the assembly will be named 'Trinity.fasta' which can be renamed as Your_assembly.fasta if desired. See the Trinity website for more details: https://github.com/trinityrnaseq/trinityrnaseq/wiki/Running-Trinity.

2. BLAST the Assembly against the Reference Sequence

NOTE: Use this step when the reference is an assembly or long reads like Sanger; if it is composed of raw Illumina reads, see step 3 below for mapping reads to the query. All BLAST steps were completed with version 2.2.29+ though the commands should work on any recent BLAST version.

1. Make a BLAST database of the reference sequence (nucleotide_reference.fasta) at the command line. Enter into the command line the following:
   makeblastdb -dbtype nucl -in nucleotide_reference.fasta -out nucleotide_reference.db
2. BLAST-match the query assembly (generated in step 1) to the reference database. To obtain an output file, use [**-out BLAST_results.txt**] and to generate tabular output (required for subsequent processing steps with Python scripts), use [**-outfmt 6**]. These options can be combined in any order, so an example complete command is [**blastn -query your_assembly.fasta -db nucleotide_reference.db -out BLAST_results.txt -outfmt 6**]. If an e-value setting is desired, use the -evalue option with an appropriate number, for example [**-evalue 1e-6**]. Be aware however that the subtractive cycle effectively inverts the evalue setting in as described in the discussion.
3. For increased stringency, use protein sequences from the assembly as the BLAST query with translated nucleotide BLAST (tBLASTn), which performs 6-way translation of the (nucleotide) database. This method is recommended for most non-model systems, avoiding the problem of incomplete protein annotations.
   1. Ensure the correct genetic code is selected for the organism being studied, using the -db_gencode option. To obtain protein sequences for the query, run the TransDecoder.LongOrfs command (from TransDecoder package v. 3.0.1) to identify the longest open reading frames from assembled query sequences. The command is [**TransDecoder.LongOrfs -t your_assembly.fasta**]; the output will be placed in directory called 'transcripts.transdecoder_dir' and will contain a file called longest_orfs.pep containing the longest predicted protein sequences from each sequence in your_assembly.fasta.
   2. To use tBLASTn, run the command [**tblastn -query longest_orfs.pep -db nucleotide_reference.db -out BLAST_results.txt -outfmt 6**]. If a high-quality protein reference is available, use protein-protein matching with BLASTp rather than tBLASTn.

3. Make a BLAST database of the protein reference [**makeblastdb -dbtype prot -in protein_reference.fasta -out protein_reference.db**] and then [**blastp -query longest_orfs.pep -db protein_reference.db -out BLAST_results.txt -outfmt 6**]. Make sure to save the results as a file for downstream processing, and use tabular (outfmt 6) to ensure the Python scripts can parse them correctly.

3. Map Reads onto the Assembly

NOTE: This method can be used if the reference dataset consists of raw genomic reads, rather than assembled sequences or Sanger sequences, in which case use BLAST (step 2.1).

1. Using BWA -MEM v. 0.7.12[19] or bowtie2[20], map the downloaded raw reads (raw_reads.fastq) onto the query assembly. The output will be .sam format. Commands are as follows: first index the assembly: [**bwa index your_assembly.fasta**], and then map the reads [**bwa mem your_assembly.fasta raw_reads.fastq >mapped.sam**]. (Note the '>' symbol here is not a greater-than sign; instead it instructs the output to go into the file mapped.sam).

4. Use Python Script to Remove any Matching Sequences

NOTE: Provided scripts work with Python 2.7.

1. Following Step 2, use subtractive Python script by using the command [**./Non-matching_sequences.py your_assembly.fasta BLAST_results.txt**]. Before running the script, ensure that the BLAST output file is in format 6 (tabular). The script will output a file with non-matching sequences in fasta format named your_assembly.fasta_non-matching_sequences_BLAST_results.txt.fasta and also the matching sequences for records, as your_assembly.fasta_matching_sequences_BLAST_results.txt.fasta. The non-matching file will be the most important, as a source of potential T sequences for testing and further cycles of subtractive genomics.
2. Following Step 3, run the Python script removeUnmapped.py to take as input the .sam from step 3.1, and identifies the names of query sequences without any matching reads and saves them to a new text file. Use the command [**./removeUnmapped.py mapped.sam**] and the output will be mapped.sam_contigs_with_no_reads.txt. (The program will generate a slimmed-down sam file with all unmapped reads removed; this file can be ignored for purposes of this protocol but may be useful for other analyses).
3. As the output of the previous step is a list of sequence names in a text file called mapped.sam_contigs_with_no_reads.txt, extract a fasta file with these sequences: [**./getContig.py your_assembly.fasta mapped.sam_contigs_with_no_reads.txt**]. The output will be a file called mapped.sam_contigs_with_no_reads.txt.fasta.

5. Design Primers for the Sequence that Remains

NOTE: At this point there is a fasta file containing candidate T sequences. This section describes qPCR to experimentally test whether they come from T or from previously unknown regions of R. If the subtraction in step 4 removed all sequences, then either the initial assembly failed to include T, or the subtraction may have been too stringent.

1. Use Geneious[21] to determine optimal primer sequences manually.
    1. Highlight a candidate sequence of 21-28 bp for the Forward primer. Avoid runs of 4 or more of any base. Try to target a region with a fairly uniform combination of all basepairs. A single G or C at the 3' end is beneficial, helping to anchor the primer.
    2. Click on the **Statistics** tab on the right-hand side of the screen to view that sequence's estimated melting temperature (Tm) as the candidate region is highlighted. Look to obtain a melting temperature between 55-60 °C, while avoiding repeats and long runs of G/C.
    3. Follow steps 5.1.1. and 5.1.2 to choose a reverse primer, situated 150-250 base pairs 3' of the forward primer. While the primer lengths do not need to match, the predicted Tm should be as close as possible to the Tm of the forward primer. Be sure to reverse complement the sequence (if right-clicking in Geneious while the sequence is highlighted it is a menu option).

2. Use the **Primer Design** function, which is found in the top tool bar in the sequence window.
    1. Click on the **Primer Design** button. Insert the region to amplify under **Target Region**.
    2. Under the **Characteristics** tab, insert desired size, melting temperature (Tm), and %GC (see step 5.1.1.).
    3. Click **OK** to have primers generated. Order the primers through a custom oligo service.

3. Validate primers with control DNA (encoding both T and R) to optimize Tm and extension time. Use regular Taq and gel electrophoresis to see the band size, but optimization can also be performed with qPCR following the methods in step 6.
    1. Make 10X dilutions of both forward and reverse primers so that the primers have a concentration of 10 µM.
    2. Use a PCR mix of 0.5 µL of dNTP, 0.5 µL of forward primer, 0.5 µL of reverse primer, 0.1 µL of Taq polymerase, 2 µL of template, 0.75 µL of magnesium, 2.5 µL of buffer, and 18.15 µL of water so that there is 25 µL per template with a concentration of 5 ng/µL.
    3. Test the primers at different melting temperatures in the PCR program. Usually optimal performance is observed melt temperatures slightly below the predicted Tm of the primers, but not usually above 60 °C. Also test for optimal extension times using this guide: 1 min per 1000 bp (thus, usually 10-30 seconds depending on amplicon length).
    4. Perform end-point gel electrophoresis to confirm that the primers amplify the expected sequence. Run 25 µL of the qPCR product mixed with 5 µL of 6X glycerol dye on a 2% TAE agarose gel at 200 V for 20 min.

6. *qPCR* Validation of the Remaining Sequence

NOTE: This step requires primers validated and PCR conditions established in step 5.

1. Run each template in triplicate with the following mix; 12.5 µL of PowerSYBR Green master mix, 0.5 µL of forward primer with a concentration of 10 µM, 0.5 µL of reverse primer with a concentration of 10 µM, 10.5 µL of water, and 1 µL of template DNA (at a concentration of 2 ng/µL), so that each well contains 25 µL of total volume.
2. Run a qPCR program informed by the validated temperature and extension time from step 4. We designed and validated all primers to be compatible with a two-stage cycle, 95 °C for 10 min initial melt, then 40 cycles of 95 °C for 30 s and 60 °C for 1 min. However, a three-stage (melt-anneal-extend) program may be more optimal for the primers and should be adapted if necessary. We recommend that final denaturing curves be generated at least the first time the primers are employed in qPCR to validate the amplification of a single DNA product.
3. Measure qPCR/SYBR Green signals relative to actin (or any other suitable 'R' control) by Ct. For all cases calculate the average and standard deviation of $2^{-(\text{gene Ct - β-actin Ct})}$.
4. (Optional) Perform end-point gel electrophoresis to confirm correct product size detection by qPCR. Here, run 25 µL of the qPCR product mixed with 5 µL of 6x glycerol dye on a 2% TAE agarose gel at 200 V for 20 min.

7. *Repeat* with a New Reference to Pare Down the Data.

NOTE: If step 6 validated the identified sequences from T, end the cycle here (**Figure 2A**). However, a variety of considerations may motivate a continuation of the cycle, for example if many R sequences remain in the file or if none of the candidate T sequences were validated by qPCR in step 6.

1. Obtain a new reference. This step enables a new iteration of the cycle and may include raw genomic data, raw RNA-seq data, or other assembled datasets. Valuable resources for reference data include the Genome database at the National Center for Biotechnology Information (https://www.ncbi.nlm.nih.gov/genome) which stores assembled genomes accessible through FTP (ftp://ftp.ncbi.nlm.nih.gov/genomes/), and the Gene Expression Omnibus (https://www.ncbi.nlm.nih.gov/geo/) where raw next-generation sequence reads are stored. Genome projects may provide their raw sequence data through other project-associated websites and databases.

Representative Results

After running BLAST, the output file will have a list of sequences from the query that match the database. After Python subtraction, a number of nonmatching sequences will be obtained, and tested by qPCR. The results of this, and next steps, are discussed below.

Negative result. There are two possible negative results that can be seen after BLAST to the reference sequence. There may be no BLAST results, meaning that the total sequence does not have any similar sequences to the reference. This may be an error in selecting the right reference sequence for the sample sequenced. Another possibility is that there are no unique sequences in the starting assembly (everything is subtracted away), therefore no genes are found for the sequence of interest. Check where the reference came from and ensure that it is not the same tissue as the query assembly.

After computational filtering, qPCR may yield a negative result, for examples see **Figure 3A, 3B, C** in which there was no difference in detection across bird tissues. Panels A through C are representative genes from different subtraction cycles, which motivated additional subtractive cycle iterations and the development of the method (**Figure 2A, 2B**).

Positive result. A positive result--the identification of a true target sequence--is confirmed when genomic DNA qPCR shows statistically greater detection in the tissue / sample of interest relative to the reference (**Figure 3D**). The subtractive project in this case started with sequencing the RNA from germline tissue of male and female adult zebra finch, obtaining 10 million read pairs from each sex. For brevity, we will describe the processing of the ovary sequence only, in which 167,929 transcripts were obtained by *de novo* assembly. The subtractive genomics method (BLASTn) was used to eliminate any sequences that matched the published somatic genome[12], which left 5,060 transcripts corresponding to 598 unique proteins, indicating that many of the transcripts were noncoding. The Sanger raw reads used to generate the assembly were then used for the next level of subtraction by tBLASTn, yielding 78 proteins. One final subtraction was performed using RNA-seq raw reads from the auditory lobule[13], which left eight proteins. When these proteins were run through NCBI nr BLAST, six of the proteins were viral, one was a repetitive region in birds, and the last was an α-SNAP that is germline restricted[7] (**Figure 2B**). During this process, 935 somatic genes that were not previously included in the whole genome annotation were identified; several showed uniform qPCR amplification across tissues (**Figure 3A, 3B, 3C**). The α-SNAP gene was validated to be germline restricted using qPCR, because it was depleted in somatic tissue relative to testis DNA where it was present at levels equivalent to actin (**Figure 3D**).

What could go wrong. The main problem that must be overcome when using this method is ensuring that the proper reference sequence is used. The best reference sequence encapsulates, in the broadest sense, the genomic complexity in which the sequence of interest (T) is embedded. This may mean that sequences in different forms; transcriptome, assembly, raw data, or data from multiple studies need to be used as references (**Figure 1**). In the zebra finch study, we developed primers from RNA sequencing data; however, the primers did not always work due to the presence of introns between or within primer binding sites in DNA. We tested each primer set by PCR off genomic DNA from testis DNA, which encodes both the target (T) and the reference (R), making it a suitable positive control. Primer failure at this stage necessitates the design and testing of new primers until a suitable set is identified. Standard pitfalls of PCR-based methods apply: amplification conditions must be optimized, amplification specificity confirmed by testing and/or cloning, and no-template controls must be included in all experiments. For more information on qPCR assays, see[22].

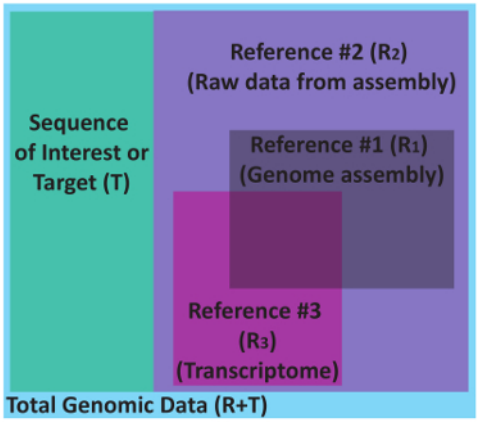

Figure 1. The subtractive approach can iteratively remove multiple references (R) to recover only the target sequence of interest (T) from total genomic data. The reference sequences of individual projects may not overlap in precisely this way and may include datasets not indicated on the figure. Please click here to view a larger version of this figure.

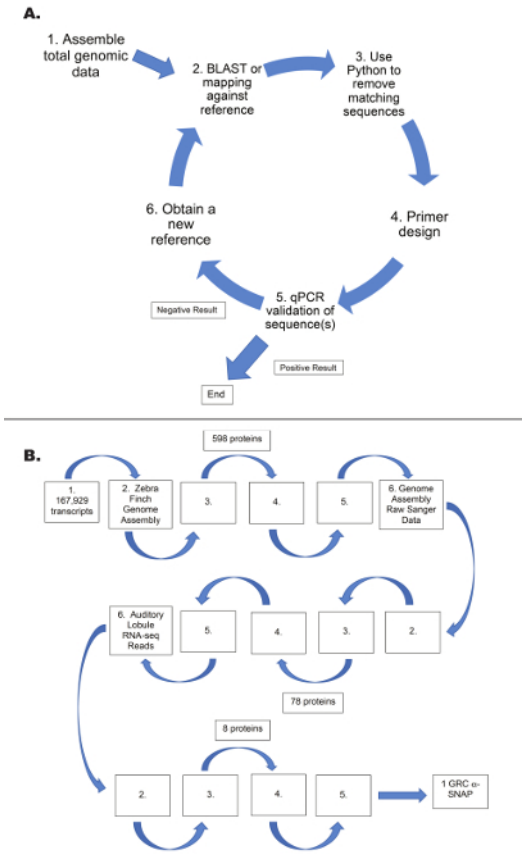

Figure 2. **Visual methods**. (A) Subtractive cycle schematic. The cycle can be iterated as many times as needed, each time utilizing distinct reference sequences, to obtain the best results.(B) Specific example of the subtractive cycle of steps carried out in Biederman *et al.*[7], with steps numbered as in A, and with the number of sequences remaining at each stage shown. Please click here to view a larger version of this figure.

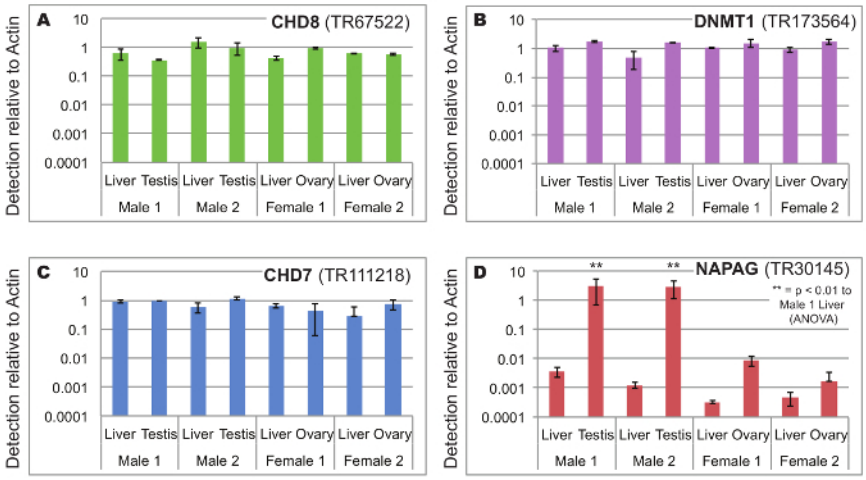

Figure 3. Example data of qPCR results including negative and positive outcomes. (A) Genomic DNA qPCR of CHD8, a negative outcome. (B) Genomic DNA qPCR of DNMT1, a negative outcome. (C) Genomic DNA qPCR of CHD7, a negative outcome. (D) Genomic DNA qPCR of NAPAG, confirming presence specifically in testis samples and depletion from liver and ovary relative to actin, a positive outcome. All panels indicate average +/- standard deviation of three measurements. Please click here to view a larger version of this figure.

Discussion

While subtractive genomics is powerful, it is not a cookie-cutter approach, requiring customization at several key steps, and careful selection of reference sequences and test samples. If the query assembly is of poor quality, filtering steps might only isolate assembly artifacts. Therefore, it is important to thoroughly validate the *de novo* assembly using an appropriate validation protocol to the specific project. For RNA-seq, guidelines are provided on the Trinity website[18] and for DNA, a tool like REAPR[23] can be used. Another critical step when using BLAST is selection of appropriate e-value, which will determine whether the subtraction will be relaxed or stringent. However, an inversion occurs in the method: a more stringent match to reference is actually a less-stringent subtraction, as non-matching sequences are not subtracted. Therefore, a larger (less stringent) e-value should be used in BLAST for a more stringent subtraction. The final essential step of the protocol is reference selection. For greatest efficiency the reference should be as complete as possible; however, it does not need to be perfect because qPCR testing confirms whether remaining sequences are from T or R, and whether more filtering is necessary. During the implementation of the protocol, new references may be used to further narrow down the genes to be validated. We note that sometimes the matching method may change: for the last subtractive step we used the algorithm BWA to map raw reads onto the query sequences, and used custom python scripts to identify query sequences with no matching reads (**Figure 2B**).

Limitations of this method include availability of a reference sequence. For example, Meyer *et al.* evaluated the mitochondrial genome of a new hominin; they used human and Denisovan probes to capture mitochondrial DNA, which was sequenced and mapped to a human reference[24]. In this case, there were no existing nuclear genome reference data that the researchers could have subtracted against to obtain the mitochondrial genome, necessitating the read-mapping alternative strategy[24]. Any extensively diverged regions of the novel mitochondrion relative to the human mitochondrial reference would be lost by read-mapping. Subtractive genomics offers a less-biased approach than read-mapping but is not always applicable depending on the research question, and in this case the low levels of ancient DNA precluded the kind of sequence coverage required for *de novo* assembly (step 1 of subtractive genomics).

Physical purification provides another alternative method to subtractive genomics. Purification of DNA or RNA is often used in sequencing whole chloroplast and mitochondrial genomes because these organellar genomes are much smaller than nuclear genomes[25,26,27,28]. Human and other smaller mitochondrial genomes can be isolated for sequencing through amplification using two primer sets followed by purification[25]. However, subtractive genomics may be helpful for cases in which mitochondrial genomes are unusually large, the primer binding sites are divergent or will not result in the full genome. An example of this is in ciliates, which have large, divergent, linear mitochondrial genomes[29]. Mapping to a reference genome is not a viable option for ciliates due to high divergence across species and lack of homologs even across genuses[30]. By using subtractive genomics, the ciliate mitochondrial genome can be isolated and analyzed while minimizing the potential of missing segments of the genome. Similarly, while a *de novo* assembly approach was used in the Sitka spruce chloroplast genome assembly, gap-closing involved comparative read mapping against the white spruce, potentially introducing bias at these sites[31].

Depending on the project, subtractive genomics may offer time and cost advantages relative to purification or mapping approaches, while offering less bias in the discovery process. In some situations, the target sequence cannot be easily isolated because it is completely unknown, is vital to cell survival (mitochondria), or too large to separate by standard gel electrophoresis. Size-based electrophoretic purification is slow and requires significant starting material (which may be expensive) while optimizing conditions over multiple attempts. Pulse-field gel electrophoresis (PFGE) enables separation of DNA fragments up to 10^7 bp (10 Mb) but takes 2-3 days, large amounts of material, and sometimes specialized equipment that is not commercially available[32]. In Biederman *et al.*, the only sequence that was known from the germline-restricted chromosome was a noncoding repeat[7]. As this chromosome is the largest in the bird, over 100 Mb in length[10], purification would have been impossible; therefore, subtractive genomics was able to do what other methods could not. In the genomic era it is often cheaper and faster to sequence now, and filter by computer later. Enabling the discovery of completely novel sequences, subtractive genomics utilizes a combination of approaches to isolate novel sequences even without a perfect reference sequence.

Disclosures

The authors have nothing to disclose.

Acknowledgements

The authors acknowledge Michelle Biederman, Alyssa Pedersen, and Colin J. Saldanha for their assistance with the zebra finch genomics project at various stages. We also acknowledge Evgeny Bisk for computing cluster system administration and NIH grant 1K22CA184297 (to J.R.B.) and NIH NS 042767 (to C.J.S).

References

1. Barh, D., *et al.* A Novel Comparative Genomics Analysis for Common Drug and Vaccine Targets in Corynebacterium pseudotuberculosis and other CMN Group of Human Pathogens. *Chemical Biology & Drug Design.* **78** (1), 73-84 (2011).
2. Sarangi, A. N., Aggarwal, R., Rahman, Q., & Trivedi, N. Subtractive Genomics Approach for in Silico Identification and Characterization of Novel Drug Targets in Neisseria Meningitides Serogroup B. *Journal of Computer Science & Systems Biology.* **2** (5) (2009).
3. Kaur, N., *et al.* Identification of Druggable Targets for Acinetobacter baumannii *Via.* Subtractive Genomics and Plausible Inhibitors for MurA and MurB. *Applied Biochemistry and Biotechnology.* **171** (2), 417-436 (2013).
4. Rathi, B., Sarangi, A. N., & Trivedi, N. Genome subtraction for novel target definition in Salmonella typhi. *Bioinformation.* **4** (4), 143-150 (2009).
5. Epstein, J. H., *et al.* Identification of GBV-D, a Novel GB-like Flavivirus from Old World Frugivorous Bats (*Pteropus giganteus.*) in Bangladesh. *PLoS Pathogens,.* **6** (7). (2010).
6. Kapoor, A., *et al.* Identification of Rodent Homologs of Hepatitis C Virus and Pegiviruses. *MBio,.* **4** .(2). (2013).
7. Biederman, M. K., *et al.* R. Discovery of the First Germline-Restricted Gene by Subtractive Transcriptomic Analysis in the Zebra Finch, Taeniopygia guttata. *Current Biology.* **28** (10), 1620-1627 (2018).
8. Readhead, B., *et al.* Multiscale Analysis of Independent Alzheimer's Cohorts Finds Disruption of Molecular, Genetic, and Clinical Networks by Human Herpesvirus. *Neuron.* **99**, 1-19 (2018).
9. Carroll, D., *et al.* The global virome project. *Science.*, **359** (6378), 872-874 (2016).
10. Pigozzi, M.I., Solari, A.J. Germ cell restriction and regular transmission of an accessory chromosome that mimics a sex body in the zebra finch. Taeniopygia guttata. *Chromosome Research.* **6**, 105-113 (1998).
11. Itoh, Y., Kampf, K., Pigozzi, M.I., and Arnold, A.P. Molecular cloning and characterization of the germline-restricted chromosome sequence in the zebra finch. *Chromosoma.* **118**, 527-536 (2009).
12. Warren, W.C., *et al.* The genome of a songbird. *Nature.* **464**, 757-762 (2010).
13. Balakrishnan, C.N., Lin, Y.C., London, S.E., and Clayton, D.F. RNAseq transcriptome analysis of male and female zebra finch cell lines. *Genomics.* **100**, 363-369 (2012).
14. Bolger, A.M., Lohse, M., Usadel, B. Trimmomatic: a flexible trimmer for Illumina sequence data. *Bioinformatics.*, **30** (15), 2114-20 (2014).
15. Zhang, J., Kobert, K., Flouri, T., and Stamatakis, A. PEAR: a fast and accurate Illumina Paired-End reAd mergeR. *Bioinformatics.* **30**, 614-620 (2014).
16. Yang, X., Dorman, K.S., and Aluru, S. Reptile: representative tiling for short read error correction. *Bioinformatic.*s. **26**, 2526-2533 (2010).
17. MacManes, M.D., Eisen, M.B. Improving transcriptome assembly through error correction of high-throughput sequence reads. *PeerJ.,.* **1** (113). (2013).
18. Grabherr, M.G., *et al.* Full-length transcriptome assembly from RNA-seq data without a reference genome. *Nature Biotechnology.* **29**, 644-652 (2011).
19. Li, H. *Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM.* arXiv:1303.3997 [q-bio.GN]. (2013).
20. Langmead, B., Salzberg, S. Fast gapped-read alignment with Bowtie 2. *Nature Methods.* **9**. 357-359. (2012).
21. Kearse, M., *et al.* Geneious Basic: an integrated and extendable desktop software platform for the organization and analysis of sequence data. *Bioinformatics.* **28** (12), 1647-1649 (2012).
22. Peirson, S.N., Butler, J.N. Quantitative polymerase chain reaction. *Methods in Molecular Biology,.* **362**, 349-62 (2007).
23. Hunt, M., Kikuchi, T., Sanders, M., Newbold, C., Berriman, M., Otto, T.D. REAPR citation: REAPR: a universal tool for genome assembly evaluation. *Genome Biology.* **14** (5). (2013).
24. Meyer, M., *et al.* A mitochondrial genome sequence of a hominin from Sima de los Huesos. *Nature.* **505** (7483), 403-406 (2013).
25. Gunnarsdóttir, E. D., Li, M., Bauchet, M., Finstermeier, K., & Stoneking, M. High-throughput sequencing of complete human mtDNA genomes from the Philippines. *Genome Research.* **21**.(1), 1-11 (2010).
26. King, J. L., *et al.* High-quality and high-throughput massively parallel sequencing of the human mitochondrial genome using the Illumina MiSeq. *Forensic Science International: Genetics.* **12**., 128-135 (2014).
27. Yao, X., *et al.* The First Complete Chloroplast Genome Sequences in Actinidiaceae: Genome Structure and Comparative Analysis. *Plos One.* **10** (6) (2015).
28. Zhang, Y., *et al.* The Complete Chloroplast Genome Sequences of Five Epimedium Species: Lights into Phylogenetic and Taxonomic Analyses. *Frontiers in Plant Science.* **7**. (2016).
29. Swart, E. C., *et al.* The Oxytricha trifallax Mitochondrial Genome. *Genome Biologyogy and Evolution.* **4** (2), 136-154. (2011).
30. Barth, D., & Berendonk, T. U. The mitochondrial genome sequence of the ciliate Paramecium caudatum reveals a shift in nucleotide composition and codon usage within the genus Paramecium. *BMC Genomics.* **12** (1). (2011).
31. Coombe, L. *et al.* Assembly of the Complete Sitka Spruce Chloroplast Genome Using 10X Genomics' GemCode Sequencing Data. *Plos One.* **11** (9) (2016).
32. Herschleb, J., Ananiev, G., Schwartz, D.C. Pulsed-field gel electrophoresis. *Nature Protocols.*, **2** (3), 677-84 (2007).

