primerdesignR tutorial

Emily Humble

7/13/2016

The `primerdesignR` package is designed to make it easy to develop primers for a gene of interest. It will take a gene sequence, create and parse input files to programs used in a typical primer design workflow. It is designed to work for MacOSX and Linux. If you would like help using `primerdesignR`, please email emily.lhumble@gmail.com with any queries.

The minimum requirements are two input files:

1. Genome of interest (.fasta and *blast database*)
2. A query sequence of interest

It is also important that the following programs are installed.

BLAST+ executables: good tutorial for installing executables and running BLAST
bedtools: installation guidelines
primer3: manual including installation guidelines

You can use `devtools` to install `primerdesignR`

```
library(devtools)
install_github("elhumble/primerdesignR")
library(primerdesignR)

# other requirements
library(dplyr)
library(seqinr)
library(stringr)
```

1. Directory stucture.

`primerdesignR` functions work mainly by loading and parsing files in specified directories. Therefore it is important that you stick to the following (or similar) recommended project directory structure:

`data/seqs`: query gene sequence(s) of interest in a closely related species
`data/primers`: example primers used for amplifying gene of interest in a closely related species
`data/genome_db`: directory for genome and formatted blast database
`output/blast_out`: directory for *blast* output
`output/bedtools_out`: directory for *bedtools* output
`output/primers`: directory for final *primer3* output

We will design primers for sequencing the MC1R gene in the Antarctic fur seal genome. You will need to download the full draft fur seal genome from Dryad and format it into a BLAST database using the following command: `makeblastdb -in final.assembly.ArcGaz001_AP3.fasta -dbtype nucl -out genome/furseal`.

2. Retrieving query sequences

It is possible to use a gene sequence or PCR primers to identify the location of the target gene in your species of interest. You should save a fasta file of your gene of interest from at least one closely related species into `data/seqs`. You might also have some primers from a published paper which you can should store in `data/primers`. It is best to save these as a fasta file where the two primers are on one line, separated by Ns (see below).

Example data Two example sequences are included in `primerdesignR`: the MC1R sequence of the dog and PCR primers designed to amplify the MC1R in the Kermode bear.

```
data(c.lupus_mc1r_seq)
data(u.americanus_mc1r_primers)

ls()

c.lupus_mc1r_seq
u.americanus_mc1r_primers
```

Let's save these sequences as fasta files into the `data/seqs` and `data/primers`. It is important to store gene sequences and primer sequences in separate directories.

```
get_fasta <- function(x, outdir){
  name <- getAnnot(c.lupus_mc1r_seq)
  name <- data.frame(paste(">", name, sep = ""))
  fasta <- cbind(name, x[[1]][1])
  fasta <- as.vector(t(fasta))
  fasta <- as.data.frame(fasta)
  write.table(fasta, paste(outdir,substitute(x), ".fasta", sep = ""),
              quote = F, col.names = F, row.names = F)
}

get_fasta(c.lupus_mc1r_seq, "data/seqs/")
get_fasta(u.americanus_mc1r_primers, "data/primers/")
```

Two files, `c.lupus_mc1r_seq.fasta` and `u.americanus_mc1r_primers.fasta` will have been created in `data/seqs` and `data/primers` respectively.

You can also download sequences of interest from GenBank using the function `download_query_seqs`. You must specify the accession numbers (`accessions =`) and the ouput directory and filename (`output =`).

```
download_query_seqs(accessions = c("XM_546772", "AB725456",
                                   "NM_001114534", "JN575070"),
         output = "data/seqs/mc1r_seqs.fasta")
```

A files, `mc1r_seqs.fasta` containing the sequences associated with all of the above accession numbers will have been created in `data/seqs`.

2. Blasting a query sequence against the genome.

This is done using the function `blastseq2genome` and requires the path to a directory containing the query sequence or sequences (`indir =`), the path and name of your genome database (`genomepath =`), the output

directory (`outdir =`) and the full path to the *blastn* executable (`blastpath =`). The function will search for and blast all files ending in .fasta in the specified directory. Let's run this on both the dog MC1R sequence and the sequences downloaded from GenBank.

```
blastseqs2genome(indir = "data/seqs/",
                 genomepath = "data/genome_db/furseal",
                 outdir = "output/blast_out/",
                 blastpath = "~/programs/blastn")
```

In the directory `output/blast_out`, two output files `mc1r_seqs2genome` and `c.lupus_mc1r_seq2genome` will have been created.

To map primer sequences to a genome it is normally necessary to use less stringent blast parameters. Let's see where the set of Kermode bear MC1R primers map to the fur seal genome.

```
blastprimers2genome(indir = "data/primers/",
                 genomepath = "genome_db/furseal",
                 outdir = "output/blast_out/",
                 blastpath = "~/programs/blastn")
```

Let's load in all the blast output files:

```
blastout_files <- paste0("output/blast_out/",
                         list.files(path = "output/blast_out/"))

import_files <- function(file){
       if (!file.size(file) == 0) {
  file <- read.table(file)
  names(file) <- c("Query", "Subject", "PercentIdentity",
                   "QueryLength", "AlignmentLength", "Mismatches",
                   "GapOpening", "QueryStart", "QueryEnd",
                   "SubjectStart", "SubjectEnd", "Evalue", "BitScore")
  return(file)
       }
}

blast_out <- lapply(blastout_files, import_files)
```

3. Extract target location

You will have to determine the target location by looking at the blast output, specifically, columns `Subject`, `QueryStart`, `QueryEnd`,`SubjectStart`, and `SubjectEnd`.

```
head(blast_out)
```

In this case, the MC1R seqeunces are clearly mapping to the same region on **scaffold00610**. The primer sequences resulted in no mappings. From looking at the `SubjectStart`, and `SubjectEnd` coordinates of the first alignment, it looks like the region between base pairs **197544:198489** is what we are interested in.

We will now generate a .bed file to parse to the program `bedtools` *getfasta* which will extract the target region from the fur seal genome as a fasta file allowing an additional 300 base pairs on either side for primer design.

You must specify:
1. The chromosome, scaffold or contig ID (*Subject* in `blast_out`, `chr =`)
2. Start and end coordinates (+ and - 300 bp) making sure that the start is less than the end (`start =` and `end =`). In this case, the query sequence 'gi|62122762|ref|NM_001014282.1|' mapped in reverse so the `SubjectEnd` is actually our `SubjectStart`.
3. Output directory (`outdir =`)
4. Full path to the bedtools executable (`bedtoolspath =`)
5. Path to your genome file (`genomepath =`)

```
extract_target_location(chr = "scaffold00610",
                        start = 197244,
                        end = 198789,
                        outdir = "output/bedtools_out/",
                        bedtoolspath = "/Users/emilyhumble/programs/bedtools",
                        genomepath = "data/genome_db/submission.assembly.fasta")
```

In the directory `output/bedtools_out`, an intermediate file `target_region.bed` and an output file `target_region.fasta` will have been created.

4. Primer design

We can now load the resulting fasta file and prepare it for primer3 by highlighting the target region with square brackets. You may then now choose to run Primer3 through its web interface or through the *command line*. It is important to note that the default parameters for the web interface and the command line version are not the same and therefore ther resulting primers will differ depending on which you use. We have succesfully designed primers using both.

Primer3 web interface

Simply run the following and copy the output sequence into the input box of the web interface.

```
target_web <- read.fasta("output/bedtools_out/target_region.fasta",
                         as.string = T, forceDNAtolower = F) %>%
  lapply(function(x) x[[1]]) %>%
  as.character(unlist(.)) %>%
  insert_str(c("[", "]"), c(300, str_length(.)-300))

target_web
```

Command line

Running the command line version of primer3 can enable more flexibility. The package `primerdesignR` provides functions to prepare input files for running the the program with 3 sets of parameters. `create_primer3_input_A` specifies a few basic input parameters, `create_primer3_input_B` specifies a longer list of parameters that we have used to succesfully design primers for microsatellites and `create_primer3_input_C` specifies the input parameters used in primer3 web version 0.4.0. It is up to you which of these you would like to use. You may also be interested in adding functions to the package for different parameters.

Create the input files by specifying the name of your sequence (`seqID =`), the object in which your target sequence is stored (`target =`), the full path to the primer3 installation directory (`primer3path =`) and the output directory (`outdir =`)

```
target <- read.fasta("output/bedtools_out/target_region.fasta",
                     as.string = T, forceDNAtolower = F) %>%
  lapply(function(x) x[[1]]) %>%
  as.character(unlist(.))

create_primer3_input_A(seqID = "MC1R_A",
                       target = target,
                       primer3path = "/Users/emilyhumble/programs/primer3-2.3.7",
                       outdir = "output/primers/")

create_primer3_input_B(seqID = "MC1R_B",
                       target = target,
                       primer3path = "~/programs/primer3-2.3.7",
                       outdir = "output/primers/")

create_primer3_input_C(seqID = "MC1R_C",
                       target = target,
                       primer3path = "/Users/emilyhumble/programs/primer3-2.3.7",
                       outdir = "output/primers/")
```

Now you can run *primer3* on the resulting files. You must specify the directory containing your input file (`indir =`), the name of your input file (`input =`), the full path to the *primer3_core* executable (`primer3_corepath =`) and the ouput directory (`outdir =`).

```
run_primer3(indir = "output/primers/",
            input = "primer3in_A",
            primer3_corepath = "~/programs/primer3_core",
            outdir = "output/primers/")
```

An output file `primer3in_A_OUT` should have been generated in `output/primers`.
Two additional files: `MC1R_A.for` and `MC1R_B.rev` will also have been generated in the top level working directory. These are a list of all primers that primer3 considered.