# Getting Started DECIPHERing

Erik S. Wright

December 20, 2024

## Contents



# 1 About DECIPHER

DECIPHER is a software that can be used for deciphering and managing biological sequences efficiently in the R programming language. The program features tools falling into seven categories:

- Sequence databases: import, maintain, view, export, and interact with a massive number of sequences.
- Homology finding: rapidly query sequences for homologous hits among a set of target sequences or genomes. Cluster into groups of related sequences.
- Sequence alignment: accurately align thousands of DNA, RNA, or amino acid sequences. Quickly find and align the syntenic regions of multiple genomes.
- Oligo design: test oligos in silico, or create new primer and probe sequences optimized for a variety of objectives.
- Manipulate sequences: trim low quality regions, correct frameshifts, reorient nucleotides, determine consensus, or digest with restriction enzymes.

- Analyze sequences: find chimeras, classify into a taxonomy of organisms or functions, detect repeats, infer recombination, predict secondary structure, create phylogenetic trees, and reconstruct ancestral states.
- Gene finding: predict coding and non-coding genes in a genome, extract them from the genome, and export them to a file.

DECIPHER is available under the terms of the GNU Public License version 3.

# 2 Design Philosophy

DECIPHER is designed for large-scale comparison of nucleotide and protein sequences. The package is built upon a foundation of sequence databases to curate large volumes of biological sequences and is dependent on the Biostrings package for low-level sequence storage. The guiding inspiration behind DECIPHER is to empower users with powerful tools that help to convert biological sequences into results. To this end, the package strives to be well-documented, well-maintained, multi-functional, state-of-the-art, scalable, and fast.

## 2.1 Curators Protect the Originals

One of the core principles of DECIPHER is the idea of the non-destructive workflow. This concept revolves around the view that the original sequence information should never be altered. Essentially, the sequence information in the database is thought of as a backup of the original sequence file and no function is able to directly alter the sequence data. All of the workflows simply *add* information to the database, which can be used to analyze, organize, and maintain the sequences. When it comes time to export all or part of the sequences they are preserved in their original state without alteration.

Interaction with DECIPHER represents a different paradigm than typical bioinformatics pipelines. Data in R is typically moved between functions without writing files. This avoids the clutter associated with most bioinformatics pipelines. Using R packages, including DECIPHER, it is possible to go from input data to output results in a single series of commands, often without ever needing to leave R or write intermediate results to files. This keeps the overall workflow tidier than traditional pipelines and supports repeatability and reproducibility.

## 2.2 Don't Reinvent the Wheel

DECIPHER makes use of the Biostrings package that is a core part of the Bioconductor suite. This package contains numerous functions for common operations such as searching, manipulating, and reverse complementing sequences. Furthermore, DECIPHER makes use of the Biostrings interface for handling sequence data so that sequences are stored in `XStringSet` objects. These objects are compatible with many useful packages in the Bioconductor suite.

A wide variety of user objectives necessitates that DECIPHER be extensible to customized projects. R provides a simple way to place the power of thousands of packages at your fingertips. Likewise, R enables direct access to the speed and efficiency of the programming language C while maintaining the utility of a scripting language. Therefore, minimal code is required to solve complex new problems. Best of all, the R statistical programming language is open source, and maintains a thriving user community so that direct collaboration with other R users is available on several Internet forums.

## 2.3 That Which is the Most Difficult, Make Fastest

A core objective of DECIPHER is to make massive tasks feasible in minimal time. To this end, many of the most time consuming functions are parallelized to make use of multiple processors. For example, the function `DistanceMatrix` gets almost an additional 1-fold speed boost for each processor core. A modern processor with $N$ cores can see a factor of close to $N$-fold speed improvement. Similar speedups can be achieved in many other DECIPHER functions by setting the *processors* argument. This is all made possible through the use of OpenMP in C-level code.

Other time consuming tasks are handled efficiently. The function `FindChimeras` can uncover sequence chimeras by searching through a reference database of over a million sequences for thousands of 30-mer fragments in a number of minutes. This incredible feat is accomplished by using the *PDict* class provided by Biostrings. Similarly, the `SearchDB` function can obtain the one-in-a-million sequences that match a targeted query in a matter of seconds. Such high-speed functions enable the user to find solutions to problems that previously would have been extremely difficult or nearly impossible to solve using antiquated methods.

### 2.4 Stay Organized

It is no longer necessary to store related data in several different files. DECIPHER is enabled by DBI, which is an R interface to a variety of databases. DECIPHER creates an organized collection of sequences and their associated information known as a sequence database. By default, *SQLite* databases are used if the RSQLite package is installed. *SQLite* databases are flat files, meaning they can be handled just like any other file. There is no setup required since *SQLite* does not require a server, unlike many other database engines. These attributes of *SQLite* databases make storing, backing-up, and sharing sequence databases relatively straightforward. However, DECIPHER supports other *SQL* databases, such as *MariaDB* via RMariaDB, if concurrency, scalability, or performance are critical.

Separate projects can be stored in distinct tables in the same sequence database. Each new table is structured to include every sequence's description, identifier, and a unique key (called *row_names*) all in one place. The sequences are referenced by their *row_names* or *identifier* throughout most functions in the package. Using *row_names*, new information created with DECIPHER functions can be added as additional database columns to their respective sequences' rows in the database table. To prevent the database from seeming like a black box there is a function named `BrowseDB` that facilitates viewing of the database contents in a web browser. A similar function is available to view sequences called `BrowseSeqs`.

The amount of DNA sequence information available is currently increasing at a phenomenal rate. DECIPHER stores individual sequences using a custom compression format, called *nbit*, so that the database file takes up much less drive space than a standard text file of sequences. The compressed sequences are stored in a hidden table that is linked to the main information table that the user interacts with regularly. For example, by default sequence information is stored in the table "Seqs", and the associated sequences are stored in the table "_Seqs". Storing the sequences in a separate table greatly improves access speed when there is a large amount of sequence information. Separating projects into distinct tables further increases query speed over that of storing every project in a single table.

## 3 Installation

### 3.1 Typical Installation (recommended)

1. Install the latest version of R from http://www.r-project.org/, because the version of DECIPHER that will be installed from Bioconductor is dependent on the version of R.

2. Install the pre-built version of DECIPHER in R by entering:

```
> if (!requireNamespace("BiocManager", quietly=TRUE))
+     install.packages("BiocManager")
> BiocManager::install("DECIPHER")
```

Note that this automatic installation method does not enable multi-threading on macOS, where use of multiple processors requires manual installation.

### 3.2 Manual Installation

#### 3.2.1 All platforms

1. Install the latest R version from http://www.r-project.org/.

2. Install Biostrings in R by entering:

```
> if (!requireNamespace("BiocManager", quietly=TRUE))
+     install.packages("BiocManager")
> BiocManager::install("Biostrings")
```

3. Optionally, install the suggested package RSQLite in R by entering:

```
> install.packages("RSQLite")
```

4. Download DECIPHER from http://DECIPHER.codes or https://doi.org/doi:10.18129/B9.bioc.DECIPHER

#### 3.2.2 macOS

1. First install Command Line Tools from Apple, which contains compliers that are required to build packages on the Mac. Then, in R run:

```
> install.packages("<<path to macOS DECIPHER.tgz>>", repos=NULL)
```

For parallelization on macOS, extra steps are required to http://mac.r-project.org/openmp/enable OpenMP before install. In summary, run "clang -v" and then download the corresponding LLVM tar.gz file. Run the sudo tar command as shown in the OpenMP instructions, and then add these two lines to your ∼/.R/MAKEVARS file:
CPPFLAGS += -Xclang -fopenmp
LDFLAGS += -lomp
Then DECIPHER can be built and installed the usual way via the command line, as shown below for linux-alikes.

#### 3.2.3 Linux

In a shell enter:

```
R CMD build --no-build-vignettes "<<path to DECIPHER source>>"
R CMD INSTALL "<<path to newly built DECIPHER.tar.gz>>"
```

#### 3.2.4 Windows

Two options are available: the first is simplest, but requires the pre-built binary (DECIPHER.zip).

1. First Option:

```
> install.packages("<<path to Windows DECIPHER.zip>>", repos=NULL)
```

2. Second Option (more difficult):

   (a) Install Rtools from http://cran.r-project.org/bin/windows/Rtools/. Be sure to check the box that says edit PATH during installation.

   (b) Open a MS-DOS command prompt by clicking Start -> All Programs -> Accessories -> Command Prompt.

   (c) In the command prompt enter:

```
R CMD build --no-build-vignettes "<<path to DECIPHER source>>"
R CMD INSTALL "<<path to newly built DECIPHER.zip>>"
```

# 4 Getting help

To get started we need to load the DECIPHER package, which automatically loads several other required packages:

```
> library(DECIPHER)
```

Help for any function can be accessed through a command such as:

```
> ? DECIPHER
```

Once DECIPHER is installed, a list of package vignettes (tutorials) can be found via:

```
> browseVignettes("DECIPHER")
```

# 5 Functionality

The DECIPHER package contains many different functions. Figure 1 shows the relationships among functions based on how often they co-occur in the same R script. Functions listed in the same color appear together in one of the package vignettes. Table 1 shows the timeline of publications associated with the flagship functions within DECIPHER.

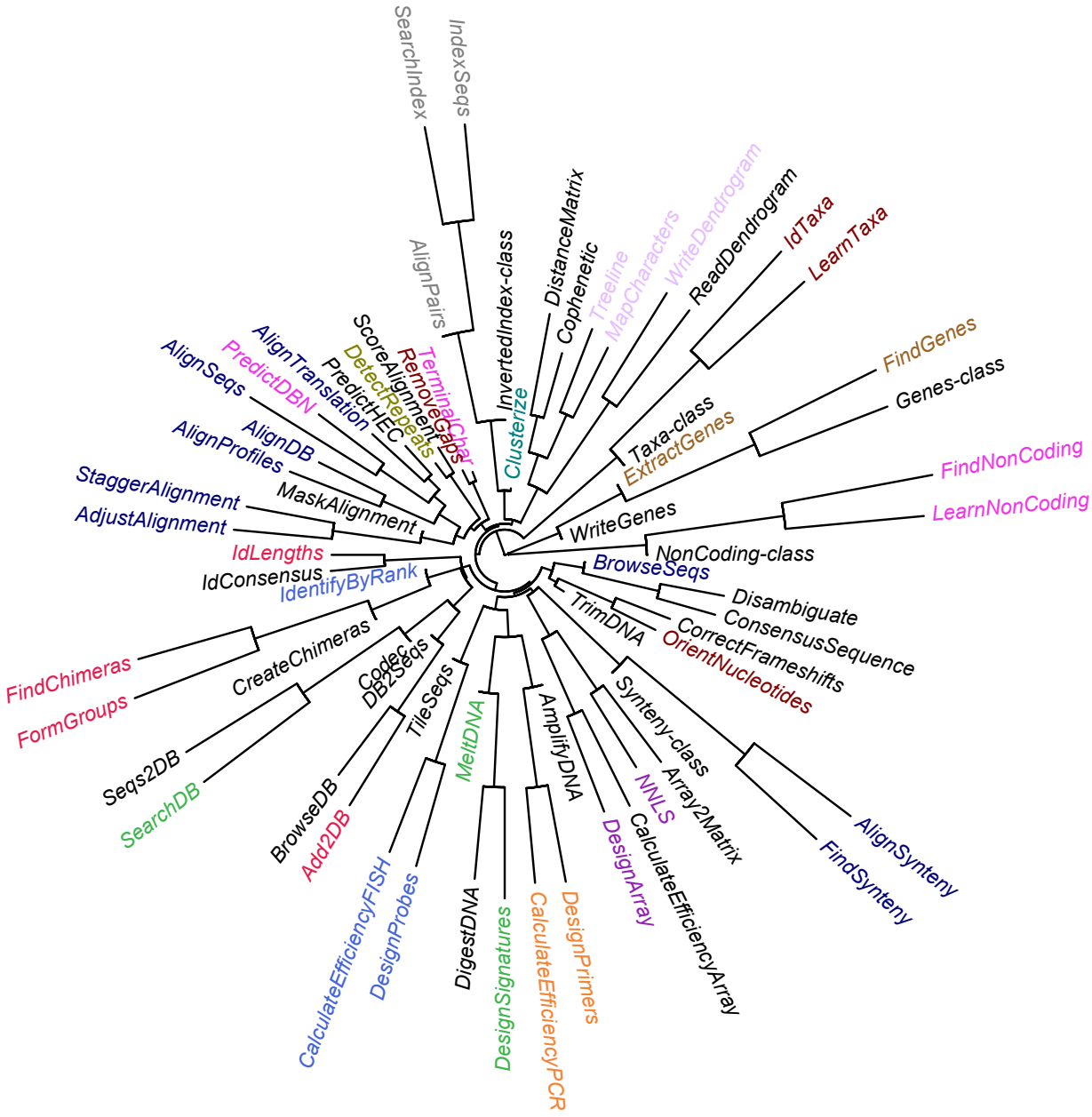


Figure 1: Depiction of relationships among all functions in the package. Those near the center of tree are less related to any other function in particular, while the functions further from the center along the same branch are more related to each other.

Table 1: Evolution of the DECIPHER package for R

| | |
|---|---|
| **2011** | DECIPHER first released in Bioconductor. |
| **2012** | Chimera finding published [4] (`FindChimeras`, `FormGroups`, `FindChimeras`). |
| **2012** | ProbeMelt published [15] (`CalculateEfficiencyArray`). |
| **2013** | Array design function published [5] (`DesignArray`). |
| **2014** | Primer design functions published [6] (`DesignPrimers`, `CalculateEfficiencyPCR`, `AmplifyDNA`). |
| **2014** | FISH probe design functions published [7] (`DesignProbes`, `CalculateEfficiencyFISH`). |
| **2014** | Array analysis functions published [3] (`Array2Matrix`, `NNLS`). |
| **2015** | Protein alignment functions published [8] (`AlignSeqs`, `AlignProfiles`, `AlignTranslation`, `PredictHEC`). |
| **2016** | Sequence databases published [9] (`SearchDB`, `Codec`, `Add2DB`, `Seqs2DB`, `DB2Seqs`, `BrowseDB`). |
| **2016** | General purpose primer design published [10] (`DesignSignatures`, `MeltDNA`, `DigestDNA`). |
| **2018** | IDTAXA for nucleotides published [2] (`IdTaxa`, `LearnTaxa`). |
| **2020** | Nucleotide alignment functions published [11] (`AlignSeqs`, `PredictDBN`). |
| **2021** | IDTAXA for proteins published [1] (`IdTaxa`, `LearnTaxa`). |
| **2022** | `FindNonCoding` published [12] (`FindNonCoding`, `LearnNonCoding`). |
| **2024** | `Clusterize` published [13] (`Clusterize`). |
| **2024** | Search and mapping functions published [14] (`IndexSeqs`, `SearchIndex`, `AlignPairs`). |

# References


[1] NP Cooley & ES Wright Accurate annotation of protein coding sequences with IDTAXA *NAR Genomics and Bioinformatics*, 3(3), lqab080, 2021.

[2] A Murali, A Bhargava, & ES Wright IDTAXA: a novel approach for accurate taxonomic classification of microbiome sequences *Microbiome*, 6, 1-14, 2018.

[3] DR Noguera, ES Wright, P Camejo, & LS Yilmaz Mathematical tools to optimize the design of oligonucleotide probes and primers *Applied Microbiology and Biotechnology*, 98, 9595-9608, 2014.

[4] ES Wright, LS Yilmaz, & DR Noguera DECIPHER, a search-based approach to chimera identification for 16S rRNA sequences *Applied and Environmental Microbiology*, 78(3), 717-725, 2012.

[5] ES Wright, JM Strait, LS Yilmaz, GW Harrington, & DR Noguera Identification of Bacterial and Archaeal Communities From Source to Tap *Water Research Foundation*, 1-76.

[6] ES Wright, LS Yilmaz, S Ram, JM Gasser, GW Harrington, & DR Noguera DECIPHER, a search-based approach to chimera identification for 16S rRNA sequences *Environmental Microbiology*, 16(5), 1354-1365, 2014.

[7] ES Wright, LS Yilmaz, AM Corcoran, HE Ökten, & DR Noguera Automated Design of Probes for rRNA-Targeted Fluorescence In Situ Hybridization Reveals the Advantages of Using Dual Probes for Accurate Identification *Applied and Environmental Microbiology*, 80(16), 5124-5133, 2014.

[8] ES Wright DECIPHER: harnessing local sequence context to improve protein multiple sequence alignment *BMC Bioinformatics*, 16(322), 1-14, 2015.

[9] ES Wright Using DECIPHER v2.0 to Analyze Big Biological Sequence Data in R *The R Journal*, 8(1), 352-359, 2016.

[10] ES Wright & KH Vetsigian DesignSignatures: a tool for designing primers that yields amplicons with distinct signatures *Bioinformatics*, 32(10), 1565-1567, 2016.

[11] ES Wright RNAconTest: comparing tools for noncoding RNA multiple sequence alignment based on structural consistency *RNA*, 26(5), 531-540, 2020.

[12] ES Wright FindNonCoding: rapid and simple detection of non-coding RNAs in genomes *Bioinformatics*, 38(3), 841-843, 2022.

[13] ES Wright Accurately clustering biological sequences in linear time by relatedness sorting *Nature Communications*, 15, 1-13, 2024.

[14] ES Wright Fast and Flexible Search for Homologous Biological Sequences with DECIPHER v3 *The R Journal*, *in Press*.

[15] LS Yilmaz, A Loy, ES Wright, M Wagner, & DR Noguera Modeling formamide denaturation of probe-target hybrids for improved microarray probe design in microbial diagnostics *PLOS ONE*, 7(8), e43862.