FastaRead.jl Documentation

Release 0.3-dev

Carlo Baldassi

Sep 27, 2017

Contents

This module provides the type `FastaReader` which allows to parse FASTA files in Julia. It is designed to be lightweight and fast, and it's inspired to kseq.h. It reads files on the fly, keeping only one entry at a time in memory, and it can read gzip-compressed files.

Here is a quick example:

```
julia> using FastaRead

julia> fr = FastaReader("somefile.fasta")

julia> for (name, seq) in fr
         println("$name : $seq")
       end

julia> close(fr)
```

CHAPTER 1

Installation

You can install FastaRead from Julia's package manager:

```
julia> Pkg.add("FastaRead")
```

CHAPTER 2

Usage

There is one central type provided by the module, `FastaReader`, which takes a file name as argument, and has a parameter to set the output type:

`FastaReader{T}(filename::String)`
creates an object which is able to parse FASTA files. The file can be in plain text format or in gzip-compressed format. The type `T` determines the output type of the sequences: the default is `ASCIIString`, which is the fastest option; another fast output type is `Vector{Uint8}`, which is less readable but has the advantage of being mutable; any other container `T` for which `convert(T, ::Vector{Uint8})` is defined will work, but it will pass through a conversion step and therefore be slower.

The data can be read out by iterating the `FastaReader` object:

```
for (name, seq) in FastaReader("somefile.fasta")
    # do something with name and seq
end
```

As shown, the iterator returns a tuple containing the description (always an `ASCIIString`) and the data (whose type is set when creating the `FastaReader` object (e.g. `FastaReader{Vector{Uint8}}(filename)`).

The `FastaReader` type has a field `num_parsed` which contains the number of entries parsed so far.

Other ways to read out the data are via the `readentry()` and `readall()` functions.

`readentry` (*fr::FastaReader*)
This function can be used to read entries one at a time:

```
fr = FastaReader("somefile.fasta")
name, seq = readentry(fr)
```

See also the `eof()` function.

`readall` (*fr::FastaReader*)
This function extends `Base.readall()`: it parses a whole FASTA file at once, and returns an array of tuples, each one containing the description and the sequence.

rewind (*fr::FastaReader*)
: This function rewinds the reader, so that it can restart the parsing again without closing and re-opening it. It also resets the value of the `num_parsed` field.

eof (*fr::FastaReader*)
: This function extends `Base.eof()` and tests for end-of-file condition; it is useful when using `readentry()`:

```
fr = FastaReader("somefile.fasta")
while !eof(fr)
    name, seq = readentry(fr)
    # do something
end
close(fr)
```

close (*fr::FastaReader*)
: This function extends `Base.close()` and closes the stream associated with the `FastaReader`; the reader must not be used any more after this function is called.

Python Module Index

f

Index