Say Hello to ALTREP

Jiefei Wang(jwang96@buffalo.edu)
Roswell Park Comprehensive Cancer Center
Buffalo, New York, USA

Outline

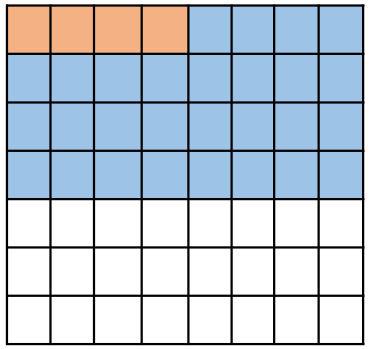
1. Review of the concept of the ALTREP
2. Challenges with the ALTREP
3. Introduction of the Travel package
4. Travel examples
5. Acknowledgements
6. Q&A

The memory layout of R's vector

A vector =

Header:	Payload:
• Type • Length • Attributes	**All** values of the vector

Memory



1. The header and payload are not separable

```
x <- c(1,2,3)
y <- x
x <- as.matrix(x)
```

2. The payload must be the entire data of the vector

```
x <- 1:100
```

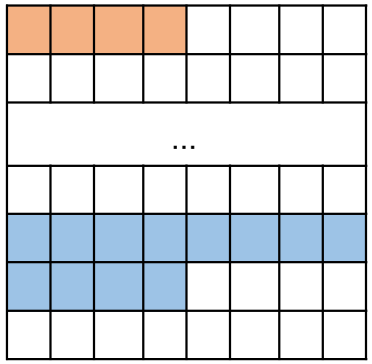
The memory layout of R's vector

An ALTREP vector =

Header:
• Type • Length • Attributes

Payload:
Any data structure

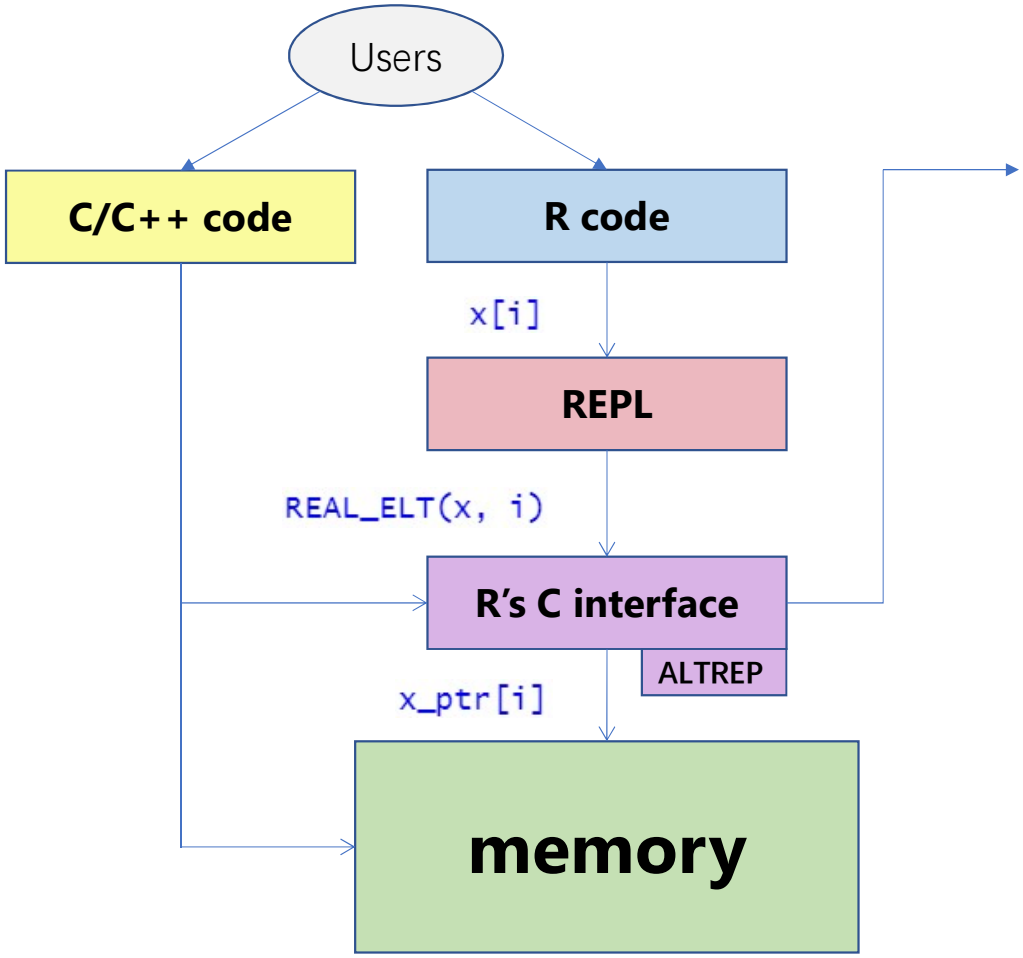
Memory



1. Header and payload can be linked and unlinked
2. The payload can be anything

ALTREP is designed with the hope that it does not break the existing user’s code

Program diagram



Accessors for the numeric vector

- REAL_ELT
- REAL_GET_REGION
- DATAPTR or REAL

Low-hanging fruit

1. Attributes wrapper
2. Compact sequence
   - seq(1, 10)
3. Sharing data across multiple R processes
   - SharedObject
4. On-disk data representation
   - vroom

Challenges with the ALTREP

With great flexibility comes great complexity

- Some R's C code is compatible with ALTREP, some are not (Even for the base R)
- Compatibility does not grant efficiency

Example

```r
mySum <- function(x){
    result <- 0
    for(i in seq_along(x)){
        result <- result + x[i]
    }
    result
}
```

REAL_ELT()

```c
double mySum1(SEXP x)
{
    double s = 0;
    R_xlen_t len = XLENGTH(x);
    for (R_xlen_t i = 0; i < len; i++)
    {
        s += REAL_ELT(x, i);
    }
    return s;
}
```

Performance: ★☆☆
Conciseness: ★★★
Readability: ★★★

REAL_GET_REGION()

```c
#include "R_ext/Itermacros.h"
double mySum2(SEXP x)
{
    double s = 0.0;
    ITERATE_BY_REGION(x, ptr, ind, nbatch, double, REAL,
                      {
                          for (int i = 0; i < nbatch; i++)
                          {
                              s = s + ptr[i];
                          }
                      });
    return s;
}
```

Performance: ★★☆
Conciseness: ★★☆
Readability: ★☆☆

DATAPTR()
REAL()

```c
double mySum3(SEXP x)
{
    double s = 0.0;
    double* ptr = (double*)DATAPTR(x);
    R_xlen_t len = XLENGTH(x);
    for (R_xlen_t i = 0; i < len; i++)
    {
        s += ptr[i];
    }
    return s;
}
```

Performance: ★★★
Conciseness: ★★★
Readability: ★★★

Performance

```
> x <- runif(128*1024*1024)
> class(x)
[1] "numeric"
> head(x)
[1] 0.9882310 0.2723803 0.2368246 0.7830110 0.6787153 0.6755918
> format(object.size(x),units = "GB")
[1] "1 Gb"
> .Internal(inspect(x))
@0x00007ff280d60010 14 REALSXP g0c7 [REF(1)] (len=134217728, tl=0)
 0.988231,0.27238,0.236825,0.783011,0.678715,...
```

REAL_ELT()

```
> system.time(mySum1(x))
   user  system elapsed
   2.36    0.00    2.55
```

Performance: ★☆☆
Conciseness: ★★★
Readability: ★★★

REAL_GET_REGION()

```
> system.time(mySum2(x))
   user  system elapsed
   0.89    0.00    0.92
```

Performance: ★★☆
Conciseness: ★★☆
Readability: ★☆☆

DATAPTR()
REAL()

```
> system.time(mySum3(x))
   user  system elapsed
   0.85    0.00    0.84
```

Performance: ★★★
Conciseness: ★★★
Readability: ★★★

Performance

```
> x <- as.numeric(seq(2*1024*1024*1024))
> class(x)
[1] "numeric"
> head(x)
[1] 1 2 3 4 5 6
> format(object.size(x),units = "GB")
[1] "16 Gb"
> .Internal(inspect(x))
@0x00000228e9aa5620 14 REALSXP g0c0 [REF(65535)]
 1 : -2147483648 (compact)
```

REAL_ELT()

```
> system.time(mySum1(x))
   user  system elapsed
 42.269   0.000  42.321
```

REAL_GET_REGION()

```
> system.time(mySum2(x))
   user  system elapsed
  3.867   0.000   3.879
```

DATAPTR()
REAL()

```
> system.time(mySum3(x))
Error: cannot allocate vector of size 16.0 Gb
Timing stopped at: 0.092 0 0.091
```

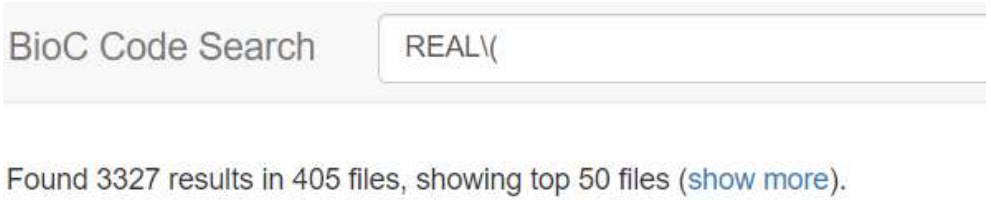
- Some R's C code is compatible with ALTREP, some are not
- Compatibility does not mean efficiency

Function frequency

REAL_ELT()

REAL_GET_REGION()

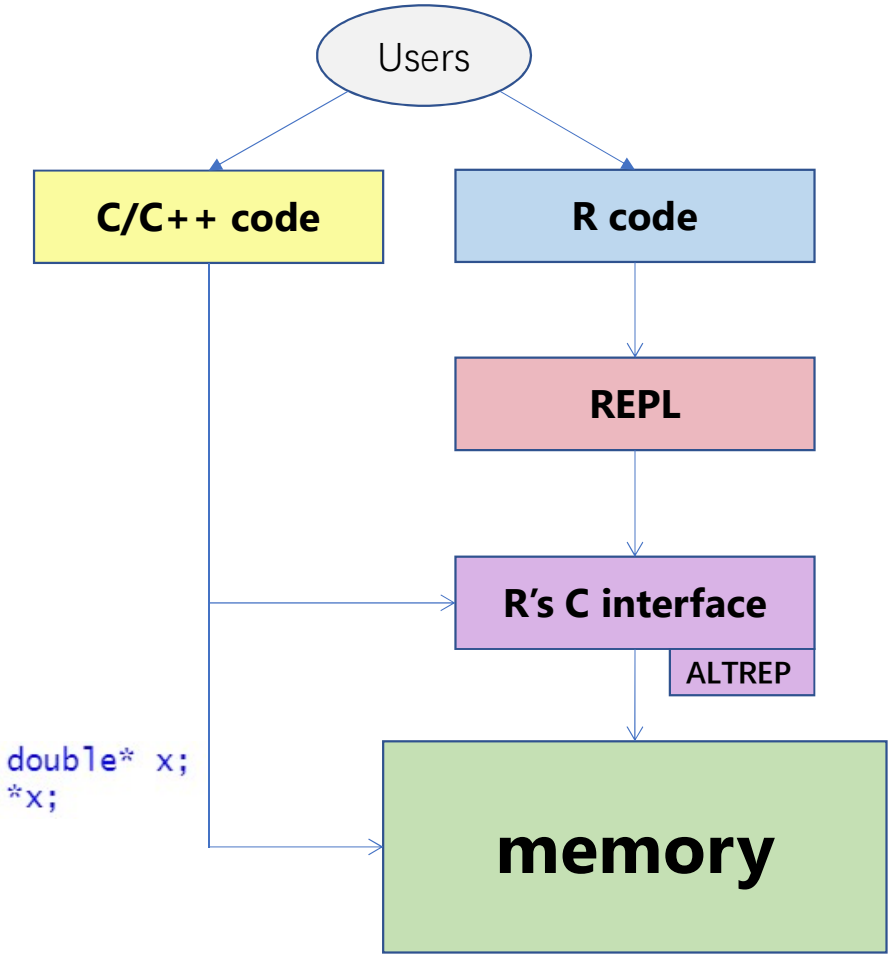
DATAPTR()
REAL()



Travel Package

- A package for creating a virtual pointer that the ALTREP object needs.

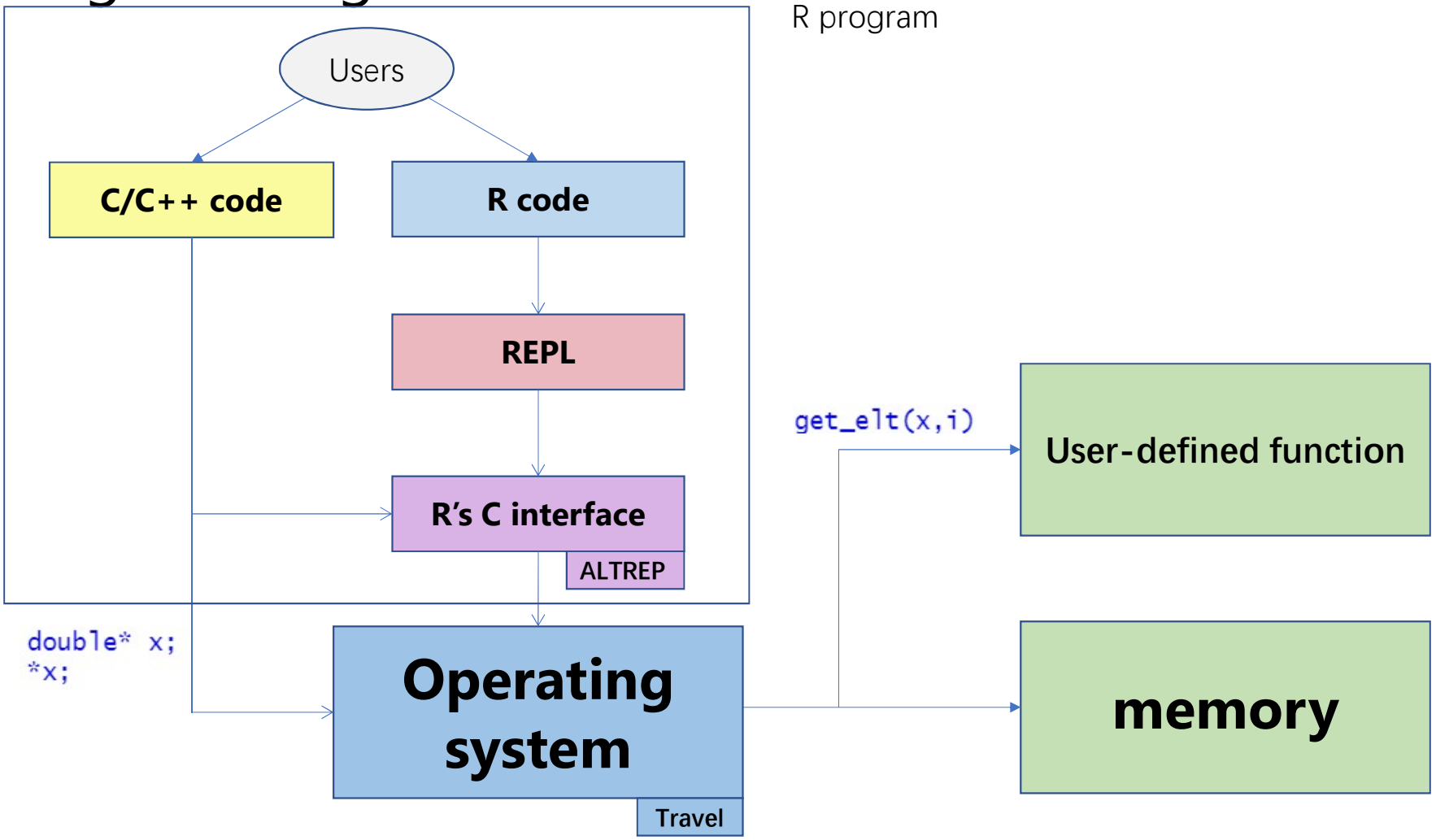
Program diagram



All problems in computer science can be solved by another level of indirection

-David Wheeler

Program diagram



Travel Programming model

API: `SEXP Travel_make_altrep(Travel_altrep_info altrep_info);`

Data structure:

```
struct Travel_altrep_info
{
  Travel_altrep_operations operations;
  int type = 0;
  uint64_t length = 0;
  void *private_data = nullptr;
  SEXP protected_data = R_NilValue;
};
```



```
struct Travel_altrep_operations
{
  Travel_get_region get_region = NULL;
  Travel_read_blocks read_blocks = NULL;
  Travel_set_region set_region = NULL;
  Travel_get_private_size get_private_size = NULL;
  Travel_extract_subset extract_subset = NULL;
  Travel_duplicate duplicate = NULL;
  Travel_coerce coerce = NULL;
  Travel_serialize serialize = R_NilValue;
  Travel_unserialize unserialize = R_NilValue;
  Travel_inspect_private inspect_private = NULL;
};
```

Travel example

```
#include "Travel.h"
size_t arithmetic_sequence_region(const Travel_altrep_info *altrep_info, void *buffer,
                                  size_t offset, size_t length)
{
    for (size_t i = 0; i < length; i++)
    {
        ((double *)buffer)[i] = offset + i + 1;
    }
    return length;
}

// [[Rcpp::export]]
SEXP Travel_compact_seq(size_t n)
{
    Travel_altrep_info altrep_info;
    altrep_info.length = n;
    altrep_info.type = REALSXP;
    altrep_info.operations.get_region = arithmetic_sequence_region;
    SEXP x = Travel_make_altrep(altrep_info);
    return x;
}
```

Performance

Candidate 1

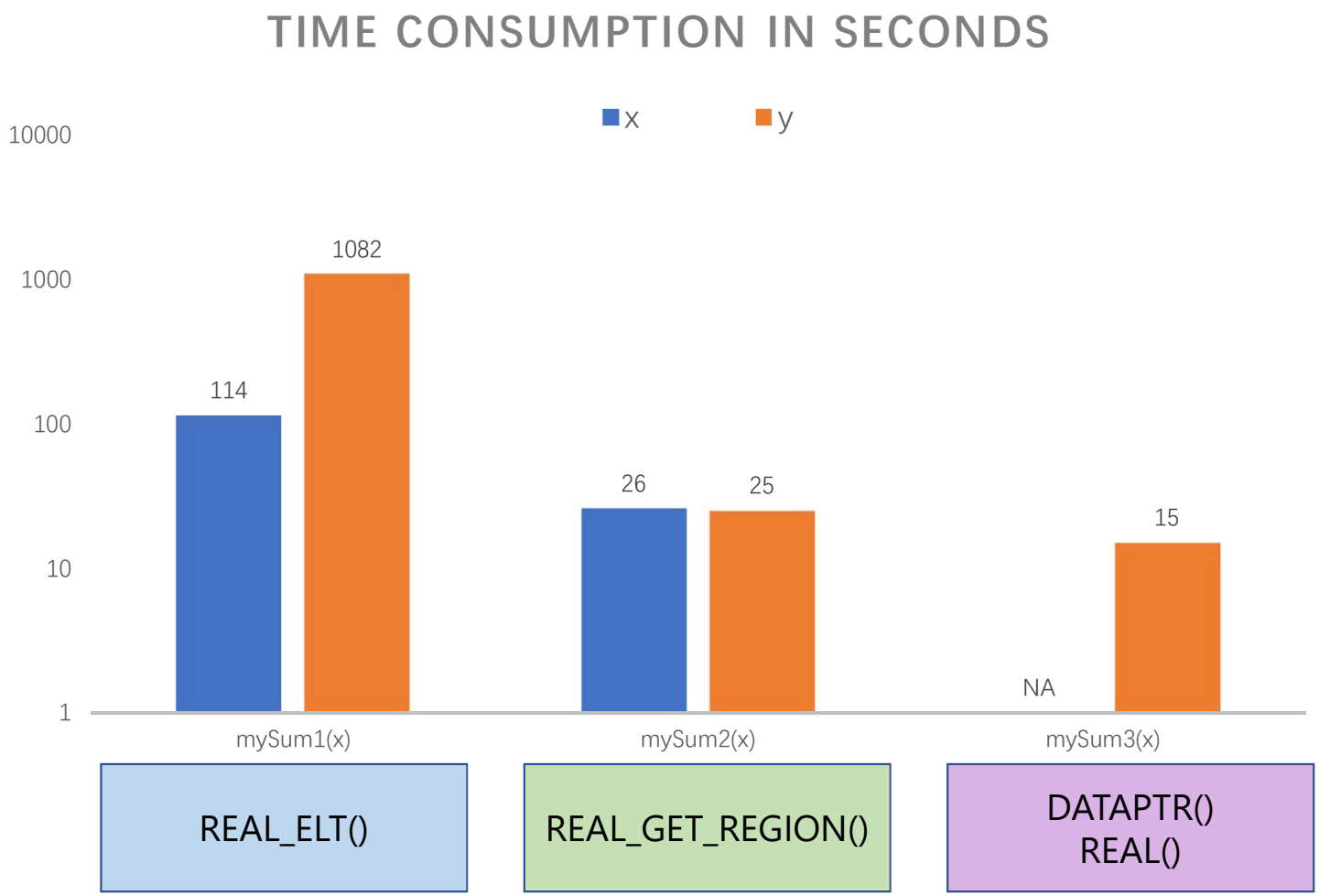
```
> x <- as.numeric(seq(2*1024*1024*1024))
> class(x)
[1] "numeric"
> head(x)
[1] 1 2 3 4 5 6
> format(object.size(x),units = "GB")
[1] "16 Gb"
> .Internal(inspect(x))
@0x00000228e9aa5620 14 REALSXP g0c0 [REF(65535)]
 1 : -2147483648 (compact)
```

Candidate 2

```
> y <- Travel:::Travel_compact_seq(2*1024*1024*1024)
> class(y)
[1] "numeric"
> head(y)
[1] 1 2 3 4 5 6
> format(object.size(y),units = "GB")
[1] "16 Gb"
> .Internal(inspect(y))
@0x00000228eb55ef88 14 REALSXP g0c0 [REF(2)] File type: real,
 size: 17179869184, length:2147483648, cache num: 0
```

Performance

```
> x <- as.numeric(seq(2*1024*1024*1024))
> y <- Travel:::Travel_compact_seq(2*1024*1024*1024)
```



Application: HighFive package

Read the HDF5 file object to R

Data file

```
> library(rhdf5)
> h5ls("myhdf5file.h5")
  group name       otype   dclass          dim
0     /    A H5I_DATASET  INTEGER 1024 x 8192
1     /   df H5I_DATASET COMPOUND       524288
```

Read the vector

```
> A1 <- HDF5Array::HDF5Array("myhdf5file.h5","A")
> A2 <- HighFive::h5Dataset("myhdf5file.h5","A")
> A1[1:4,1:4]
<4 x 4> matrix of class DelayedMatrix and type "integer":
     [,1] [,2] [,3] [,4]
[1,]   57   49   44   59
[2,]   42   48   51   43
[3,]   58   48   56   45
[4,]   49   47   52   46
> A2[1:4,1:4]
     [,1] [,2] [,3] [,4]
[1,]   57   49   44   59
[2,]   42   48   51   43
[3,]   58   48   56   45
[4,]   49   47   52   46
```

```
> class(A1)
[1] "HDF5Matrix"
attr(,"package")
[1] "HDF5Array"
> class(A2)
[1] "matrix" "array"
```

```
> sum(A1)
[1] 419419638
> sum(A2)
[1] 419419638
```

Available out-of-box

```
> mySum3(A1)
Error in mySum3(A1) : LENGTH or similar
 applied to S4 object
> mySum3(A2)
[1] 419419638
```

Application: HighFive package

Read the HDF5 file object to R

Data file

```
> library(rhdf5)
> h5ls("myhdf5file.h5")
  group name       otype   dclass          dim
0     /    A H5I_DATASET  INTEGER 1024 x 8192
1     /   df H5I_DATASET COMPOUND       524288
```

Read the compound data

```
> df <- h5Dataset("myhdf5file.h5","df")
> class(df)
[1] "data.frame"
> df[1:4,]
  V1 V2 V3 V4 V5 V6 V7 V8 V9 V10 V11 V12 V13 V14 V15 V16
1 57 47 50 48 46 45 60 52 59  51  43  45  40  44  53  43
2 42 47 51 49 52 52 55 52 42  45  58  54  53  57  58  40
3 58 52 53 55 54 46 57 41 47  45  57  54  46  45  51  47
4 49 42 52 51 50 53 55 46 48  45  49  52  52  49  40  52
```

```
> colMeans(df)
      V1       V2       V3       V4       V5       V6       V7       V8       V9
49.98900 49.99100 49.99709 50.00236 49.99357 49.99293 49.99828 49.99765 50.00935
     V10      V11      V12      V13      V14      V15      V16
50.00299 49.99555 50.01747 49.99670 49.99813 49.99133 50.00608
```

Acknowledgements

- Martin Morgan
- Luke Tierney, Gabriel Becker and many others in r-devel
- Mike Smith and Bioc Devel Forum

Q&A