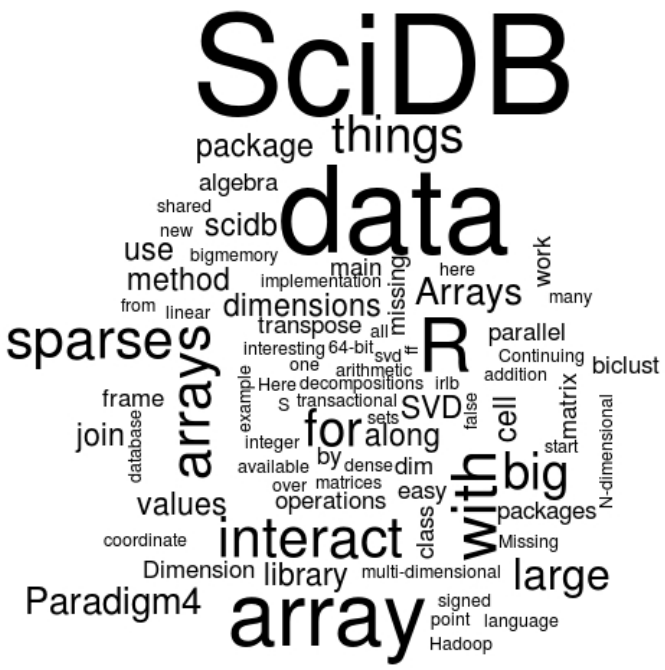
paradigm4

data-driven discovery



Bryan Lewis, Paradigm4
blewis@paradigm4.com

Agenda

1. Brief Introduction to SciDB & SciDB-R
2. Demos

SciDB

Developed by Paradigm4

Open-source high-performance database

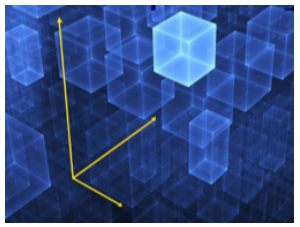
Data organized in multi-dimensional sparse arrays

Distributed storage, parallel processing

Excels at parallel linear algebra

ACID, data replication, versioned data

About Paradigm4



Paradigm4 develops & supports SciDB

CTO is database researcher Mike Stonebraker

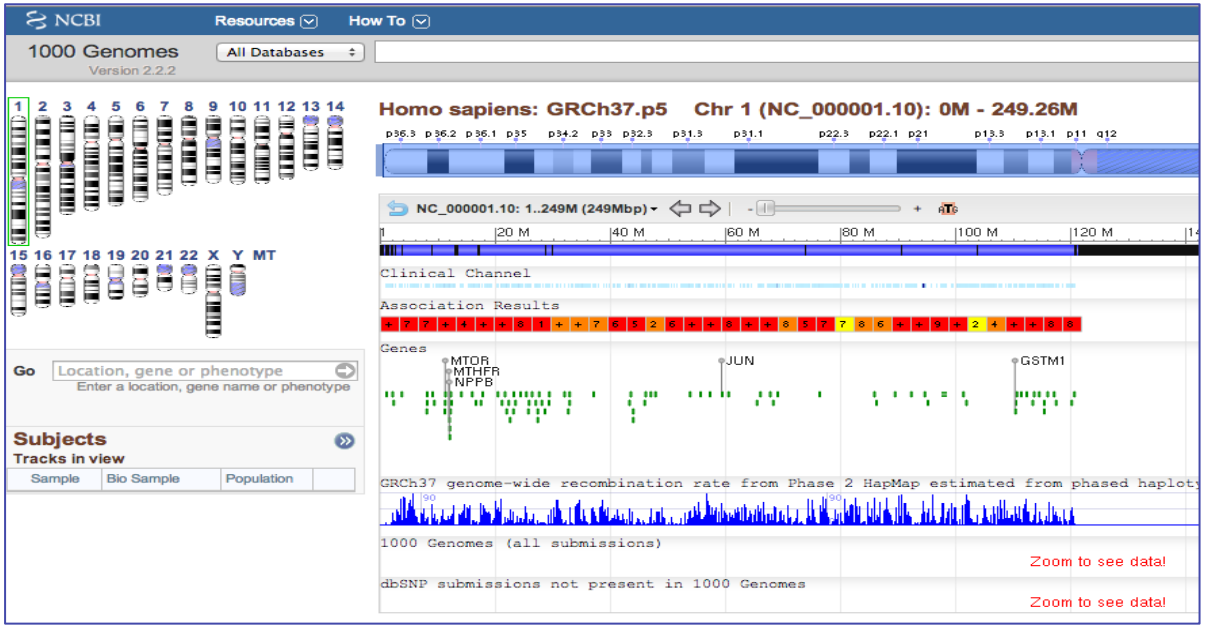
- Force behind many major advances in commercial database products (Postgres, Ingres, Vertica, et al)

Community edition: fully scalable, unrestricted

Enterprise edition

- More functionality
- Fault tolerance and system management tools

The NCBI 1K Genome Browser Runs on SciDB



http://www.ncbi.nlm.nih.gov/variation/tools/1000genomes/

http://www.ncbi.nlm.nih.gov/variation/tools/get-rm/browse

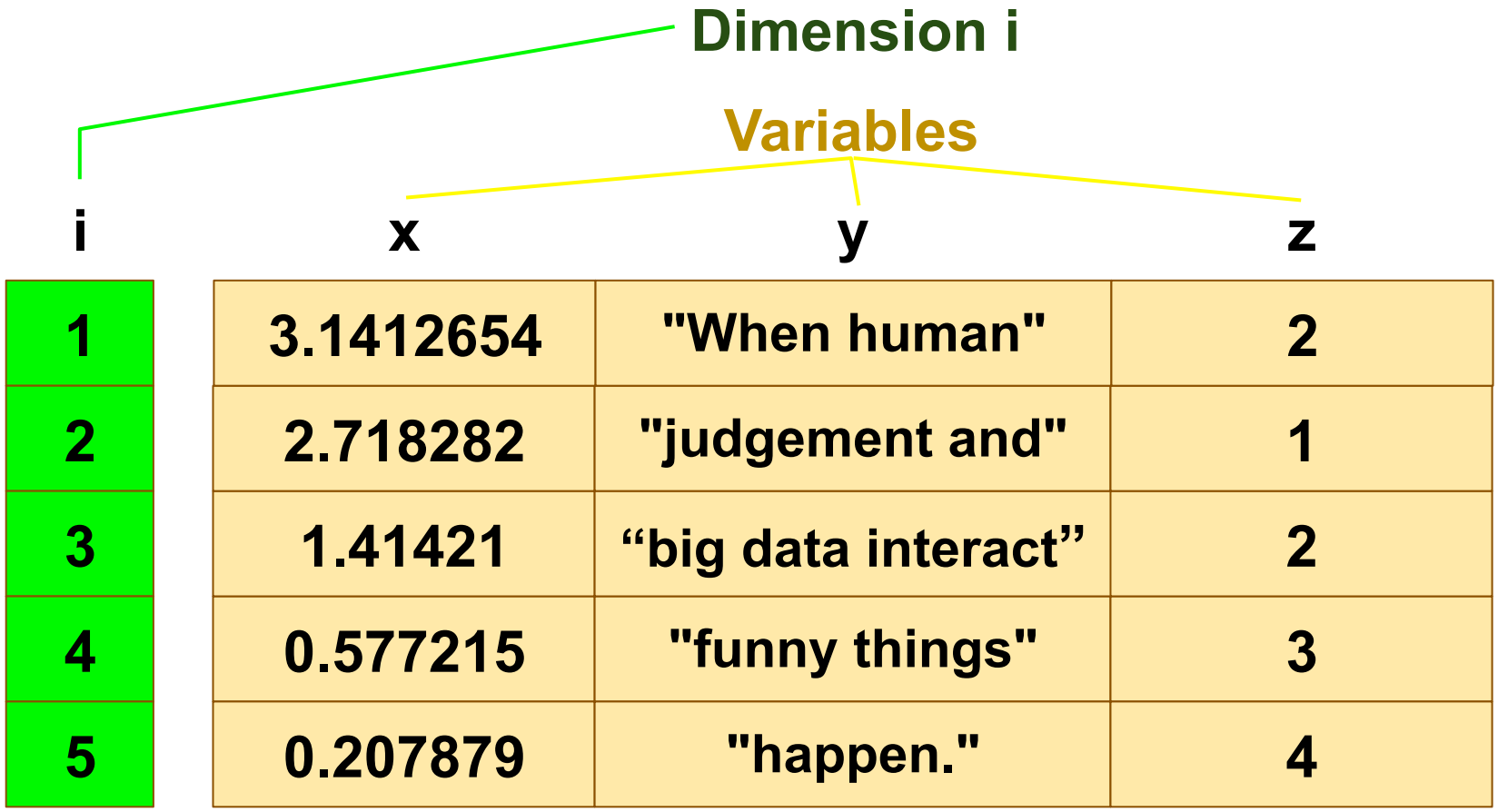
SciDB Arrays

Each cell in a SciDB array consists of a fixed number of typed values

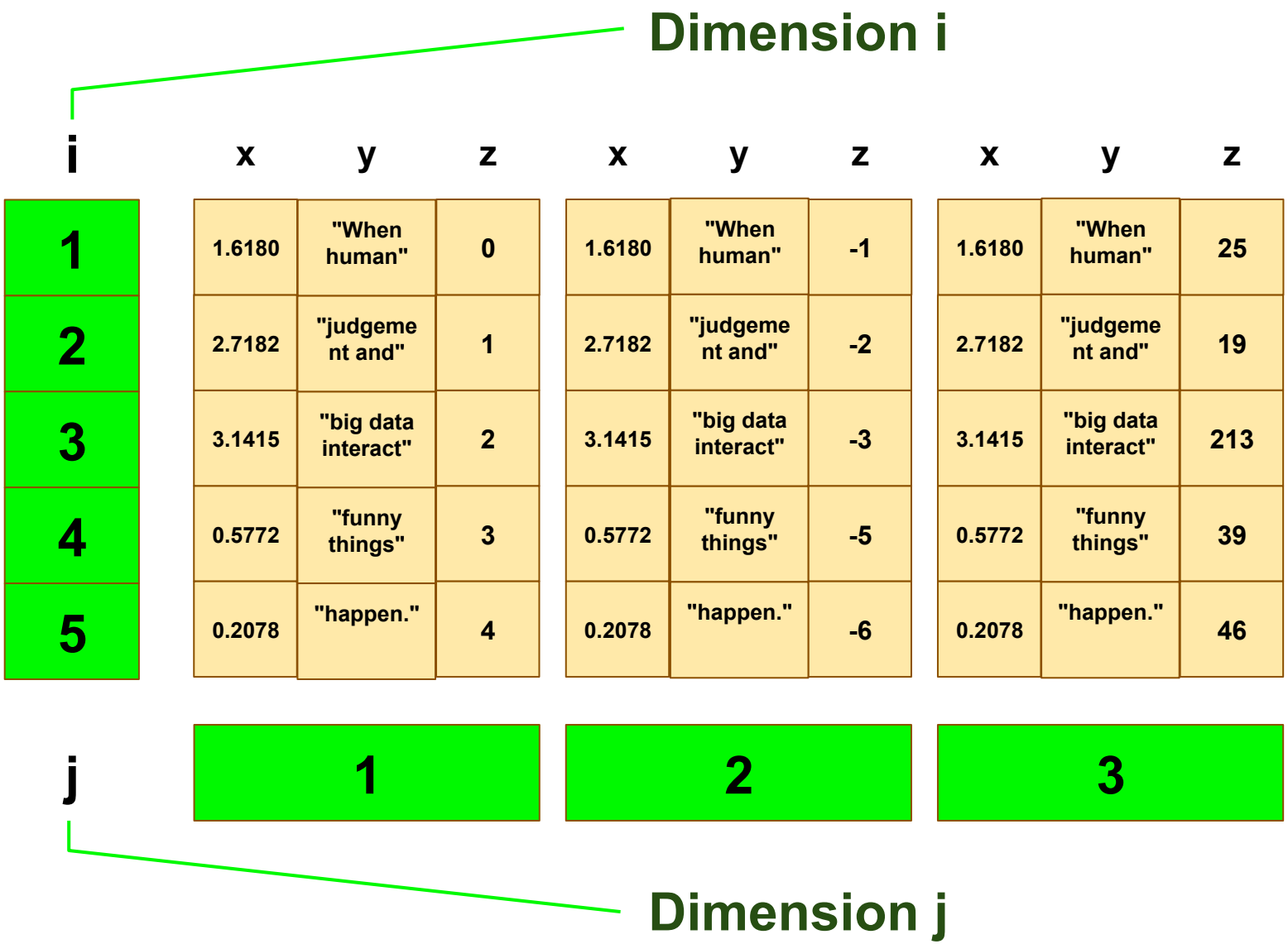
Here is an example cell:

x	y	z
3.141593	"When human"	2

Cells are ordered along coordinate axes. A 1-D array looks like an R data frame.



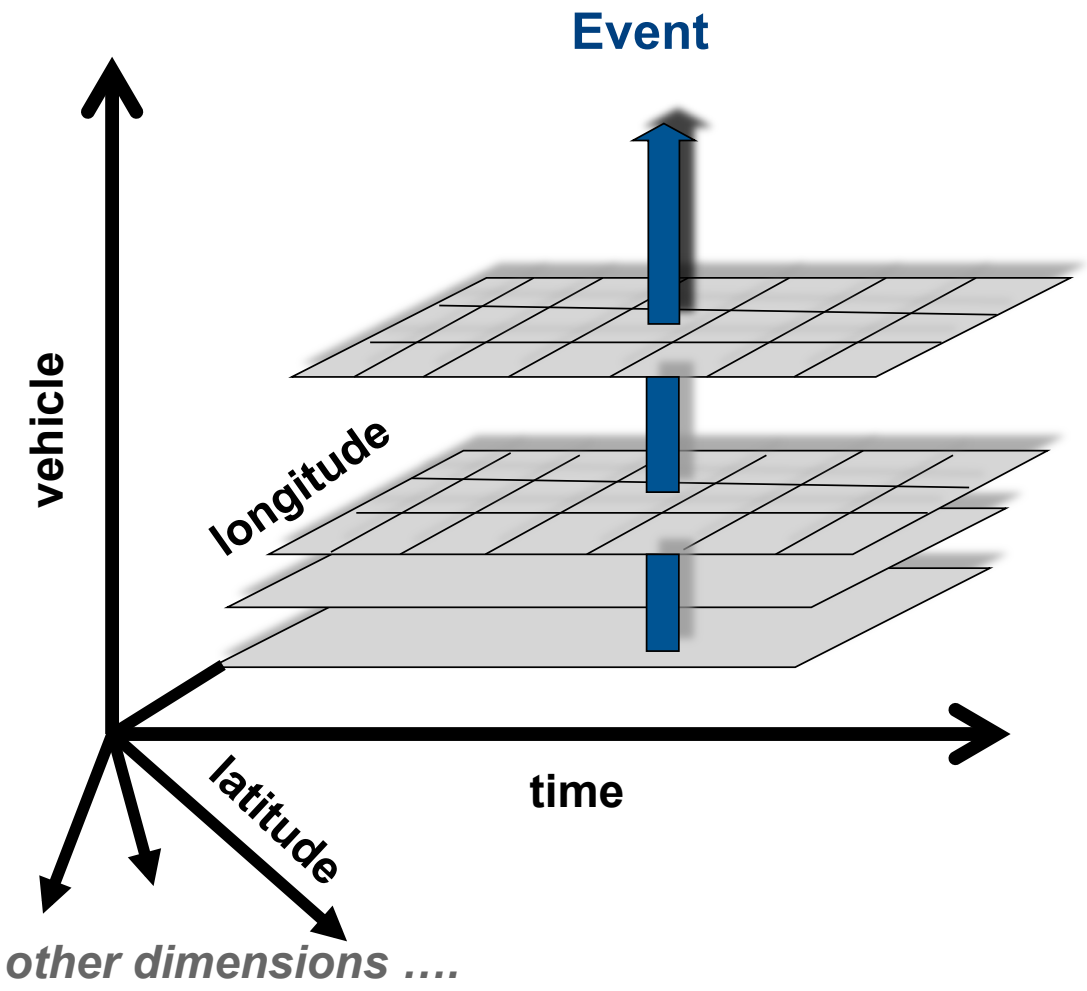
SciDB arrays can be multi-dimensional



Arrays can be sparse and values may be explicitly marked missing in several ways.

i	x	y	z
1	NA	"When human"	0
2			
3	Missing(1)	"big data interact"	Missing(7)
4	0.577215	"funny things"	3
5	0.207879	"happen."	4

SciDB Arrays



Arrays are chunked …. with optional overlap

Chunk 1

0.02	0.01	0.01	0.02
0.01	0.01	0.5	0.02
0.01	0.02	0.01	0.01
0.02	0.01	0.02	0.02

Chunk 2

0.02	0.01	0.01	0.02
0.01	0.01	0.5	0.02
0.01	0.02	0.01	0.01
0.02	0.01	0.02	0.02

Chunk 3

0.02	0.01	0.01	0.02
0.01	0.01	0.5	0.02
0.01	0.02	0.01	0.01
0.02	0.01	0.02	0.02

Chunk 4

0.02	0.01	0.01	0.02
0.01	0.01	0.5	0.02
0.01	0.02	0.01	0.01
0.02	0.01	0.02	0.02

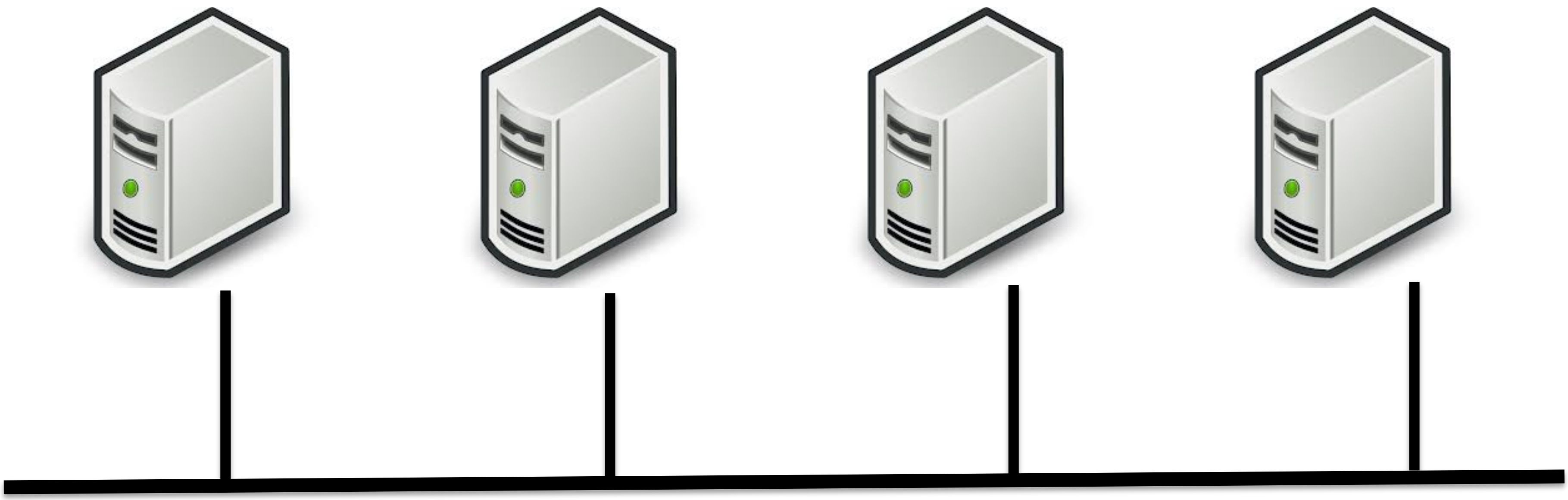
Data are distributed across SciDB instances

0.02	0.01	0.01	0.02
0.01	0.01	0.5	0.02
0.01	0.02	0.01	0.01
0.02	0.01	0.02	0.02

0.02	0.01	0.01	0.02
0.01	0.01	0.5	0.02
0.01	0.02	0.01	0.01
0.02	0.01	0.02	0.02

0.02	0.01	0.01	0.02
0.01	0.01	0.5	0.02
0.01	0.02	0.01	0.01
0.02	0.01	0.02	0.02

0.02	0.01	0.01	0.02
0.01	0.01	0.5	0.02
0.01	0.02	0.01	0.01
0.02	0.01	0.02	0.02



SciDB Arrays

Arrays can be **joined**
along dimensions or subsets of dimensions

Values can be **aggregated**
along dimensions and optionally over windows

Functions can be **applied** over values in arrays

Arrays can be sparse

Linear algebra operations, matrix decompositions, and other interesting operations are defined for matrices and vectors

The SciDB package for R

Defines two main ways to interact with SciDB:

1. **Iterable data frame interface using SciDB query language**

2. **N-dimensional sparse/dense array class for R backed by SciDB arrays**

Taxonomy of out-of-core packages

List/Dataframe-like

RObjectTables, g.data, filehash, ff, DBI (RPgSQL, RMySQL, ROracle, ...), Vertica/R, Netezza/R, rredis, **scidb**, RBerkeley, RCassandra, LaF, lazy.frames

Hadoop

rmr, HadoopStreaming, RHIPE

Array-like

ff, bigmemory, pbdR, **scidb,** Netezza

Other

rdsm, exciting work forthcoming from Simon, flexmem

```
library("scidb")
scidbconnect(host="localhost")

# An example reference to a SciDB matrix:
A <- scidb("A")
dim(A)
[1] 50000 50000
```

```
# Subarrays return new SciDB array objects:
```

```
A[c(0,49000,171), 5:8]
```

```
Reference to a 3x4 SciDB array
```

```
# Use [] to materialize data to R:
```

```
A[c(0,49000,171), 5:8][]
```

```
           [,1]       [,2]       [,3]       [,4]
[1,]  0.9820799 -0.4563357 -1.2947495 -0.8085465
[2,] -1.5090126  0.1547963 -0.2435732 -0.1836875
[3,]  1.3296710 -1.5006536 -0.5980172  0.3752186
```

```
# Arithmetic composed with subsetting:


X <- A %*% A[,1:5]

dim(X)

[1] 50000     5
```

```
# Mixed SciDB and R object arithmetic


Z <- A[c(0,49000,171), 5:7]
(0.5*(Z + t(Z)) %*% rnorm(3)[, drop=FALSE]

          [,1]
[1,]  3.707263
[2,] -2.833560
[3,]  3.518370
```

SciDB

Developed by Paradigm4

The SciDB array class facilitates exploration and analysis of large data in a familiar language.

Big Data Analytics with SciDB-R

All of the work of parallelism, data distribution, and transactional integrity is handled by SciDB.

It is sometimes possible to use SciDB arrays in R packages with little (or sometimes even no) modification.

```
library("biclust")
library("s4vd")
data(lung)
  A <- lungx <- biclust(A, method=BCssvd, K=1)
# Now with SciDB arrays:
library("ssvdp4")
X <- as.scidb(A)x1 <- biclust(X, method=BCssvd, K=1)
# Compare the results (identical up to machine epsilon):
sqrt( x@info$res[[1]]$u - x1@info$res[[1]]$u) )
              [,1]
  [1,] 5.202109e-16
```

Let's see exactly what we had to do to use the s4vd and biclust packages…

Change:

t (A) %*% x

to:

t (crossprod (x, A))

The trick here is to avoid a transpose of the large array...
Optionally, use irlb package instead of P4 svd.

Principal Components

```
S <- svd(A, nu=3, nv=3)
dim(S)
[1]     4 50000 50000

# Result is a 3-D array containing U,
  S (sparse), and V
```

How I see this package evolving ….

More data frame (1-D array) integration

- Natural R syntax aggregate and apply
- and join, tabulation, tapplys and plyR support

Continuing addition of core matrix decompositions as they become available

Continuing addition of new modeling methods as they become available

Improved hybrid R/SciDB algorithm efficiency

SciDB

Developed by Paradigm4

SciDB-R makes Big Data Analytics easy to use from R

Questions?

Tell us about your application

- info@paradigm4.com

Try our Quick Start at scidb.org/forum

- Read the SciDB-R QuickStart and SciDB-R docs
- Download a VM or EC2 AMI