

**SciDB:**  
Massively Parallel Array Data Storage,  
Processing and Analysis

Paul G. Brown

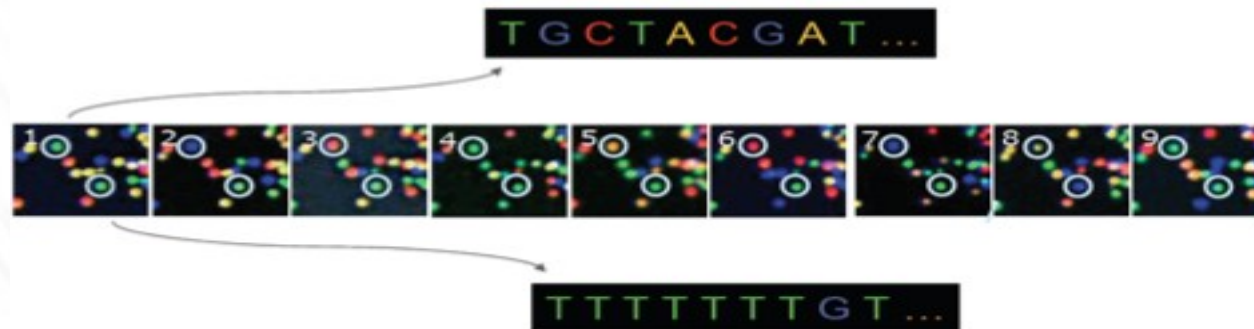
```

@PHUSCA-W21172_91027:1:1:0:1521#0/1
NAGCCCTGCCCCTTCTGAGAGTCCCTTGTTAAGCAA
+PHUSCA-W21172_91027:1:1:0:1521#0/1
DNWWWVWUWWWWVWUWUWUWWWWVUTSTSB
@PHUSCA-W21172_91027:1:1:0:1784#0/1
NACCAGGACTATTTGCACTCTTTGGGGAAGGCTCGA
+PHUSCA-W21172_91027:1:1:0:1784#0/1
DOWYUVVUYUYYYWWWYYYWUTVYRPWUWYWWT

```

**X 85 Million**

**X 10 per day**



@PHUSCA-W21172\_91027 :

1 : 1 : 0 : 1939

#0 / 1

**Sequencing Machine Identifier**

1. Flow Cell Lane
2. Tile Number
3. X Index on Tile
4. Y Index on Tile

**Something complex and biological.**

```
@PHUSCA-W21172_91027:1:1:0:1521#0/1
NAGCCCTGCCCCTTCTGAGAGTCCCTTGTTAAGCAA
+PHUSCA-W21172_91027:1:1:0:1521#0/1
DNWWWVVUWWWWWVVUWUWUWWWWWVUTSTSB
@PHUSCA-W21172_91027:1:1:0:1784#0/1
NACCAGGACTATTTGCACTCTTTGGGGAAGGCTCGA
+PHUSCA-W21172_91027:1:1:0:1784#0/1
DOWYUVVUYUYYYYWWWYYYWUTVYRPWUWYWWT
```

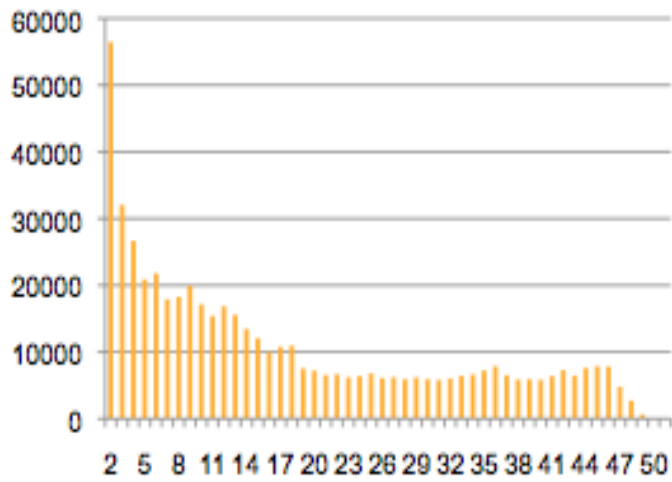
 **'Short Reads' have fixed length, 50 offsets in this case.**

```
...
NAGCCCTGCCCCTTCTGAGAGTCCCTTGTTAAGCAA
NACCAGGACTATTTGCACTCTTTGGGGNAGGCTCGA
NACAATCTGAGTNTGGTGATCCTGGTACCCCAGACC
```

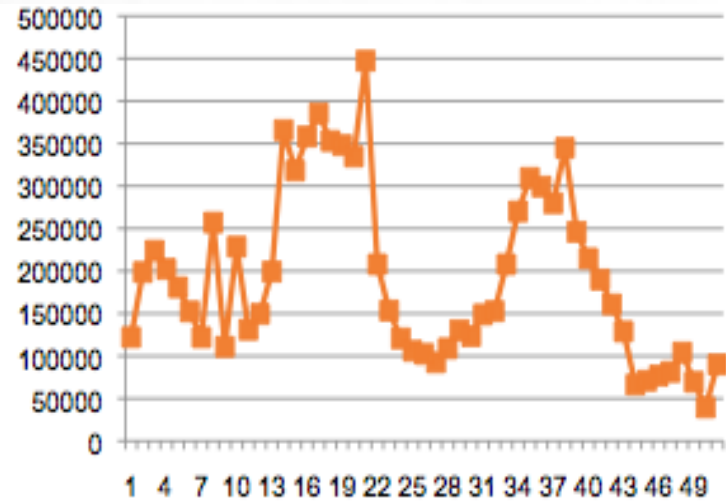
- ... A - Adenine
- G - Guanine
- T - Thymine (Uracil)
- C - Cytosine
- N - "ERROR"

*Quality control—how many 'N' values appear, and where—is an important process management question.*

# Quality Control Metrics



Sequences with a "N"



Position of the "N"

# Why another DBMS?

```
CREATE TABLE Short_Reads (  
  Machine_ID          CHAR(25) NOT NULL,  
  Flow_Cell_Lane     INTEGER  NOT NULL,  
  Tile_Number        INTEGER  NOT NULL,  
  X                   INTEGER  NOT NULL,  
  Y                   INTEGER  NOT NULL,  
  Offset              INTEGER  NOT NULL,  
  Base_Pair           CHAR(1)  NOT NULL,  
  Supp                FLOAT  
  PRIMARY KEY ( Machine_ID,  
                Flow_Cell_Lane_Number,  
                Tile_Number, X, Y, Offset )  
);
```

- Explicit storage of key information multiplies data storage requirements.
- Order of key (or index) specification biases performance.
- Many problems where 'previous' and 'next' have semantic heft.

# Introducing SciDB

```
CREATE ARRAY Reads
  ( Read::CHAR(1), Supp::FLOAT )
  [ Seq_ID:0,*, Offset:0,50 ];
```

G <sup>0.5</sup>	C <sup>0.3</sup>	A <sup>0.1</sup>	A <sup>0.9</sup>	C <sup>0.2</sup>	A <sup>0.9</sup>	T <sup>0.1</sup>	.	.	.	.	T <sup>0.2</sup>	T <sup>0.1</sup>	A <sup>0.6</sup>	A <sup>0.5</sup>
C <sup>0.2</sup>	T <sup>0.9</sup>	T <sup>0.7</sup>	A <sup>0.8</sup>	C <sup>0.5</sup>	N <sup>?</sup>	G <sup>0.5</sup>	.	.	.	.	C <sup>0.8</sup>	C <sup>0.3</sup>	A <sup>0.5</sup>	T <sup>0.2</sup>
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
T <sup>0.3</sup>	A <sup>0.6</sup>	C <sup>0.4</sup>	N <sup>?</sup>	G <sup>0.8</sup>	C <sup>0.5</sup>	C <sup>0.7</sup>	.	.	.	.	G <sup>0.3</sup>	C <sup>0.9</sup>	A <sup>0.7</sup>	A <sup>0.9</sup>

```
( AGGREGATE (
  COUNT(), R.Offset,
  FILTER ( R.Read = 'N', READS AS R )
));
```

# Architecture Overview 1

G <sup>0.5</sup>	C <sup>0.3</sup>	A <sup>0.1</sup>	A <sup>0.9</sup>	C <sup>0.2</sup>	A <sup>0.9</sup>	T <sup>0.1</sup>	.	.	.	.	T <sup>0.2</sup>	T <sup>0.1</sup>	A <sup>0.6</sup>	A <sup>0.5</sup>
C <sup>0.2</sup>	T <sup>0.9</sup>	T <sup>0.7</sup>	A <sup>0.8</sup>	C <sup>0.5</sup>	N <sup>?</sup>	G <sup>0.5</sup>	.	.	.	.	C <sup>0.8</sup>	C <sup>0.3</sup>	A <sup>0.5</sup>	T <sup>0.2</sup>
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
T <sup>0.3</sup>	A <sup>0.6</sup>	C <sup>0.4</sup>	N <sup>?</sup>	G <sup>0.8</sup>	C <sup>0.5</sup>	C <sup>0.7</sup>	.	.	.	.	G <sup>0.3</sup>	C <sup>0.9</sup>	A <sup>0.7</sup>	A <sup>0.9</sup>

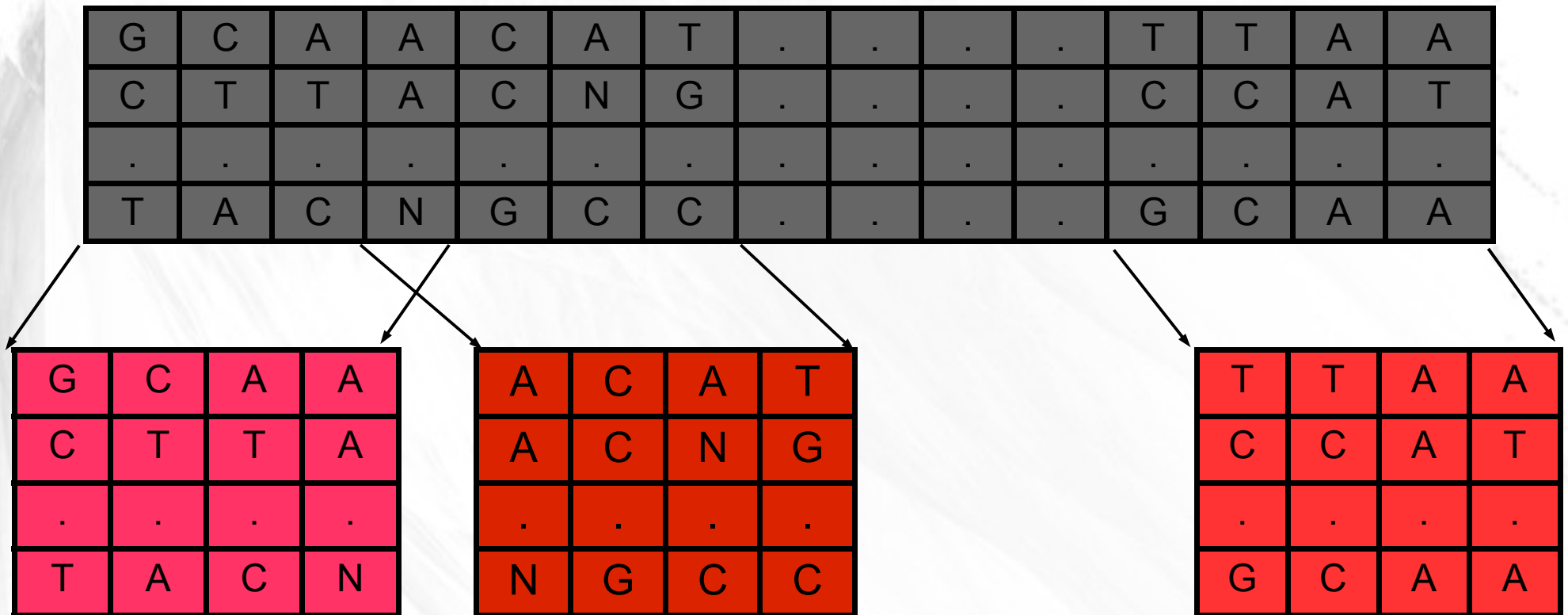
## 1. Vertical Partitioning into multiple single-attribute arrays.

G	C	A	A	C	A	T	.	.	.	.	T	T	A	A
C	T	T	A	C	N	G	.	.	.	.	C	C	A	T
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
T	A	C	N	G	C	C	.	.	.	.	G	C	A	A

0.5	0.3	0.1	0.9	.	.	.	0.6	0.5
0.2	0.9	0.7	0.8	.	.	.	0.5	0.2
.	.	.	.	.	.	.	.	.
0.3	0.6	0.4	?	.	.	.	0.7	0.9



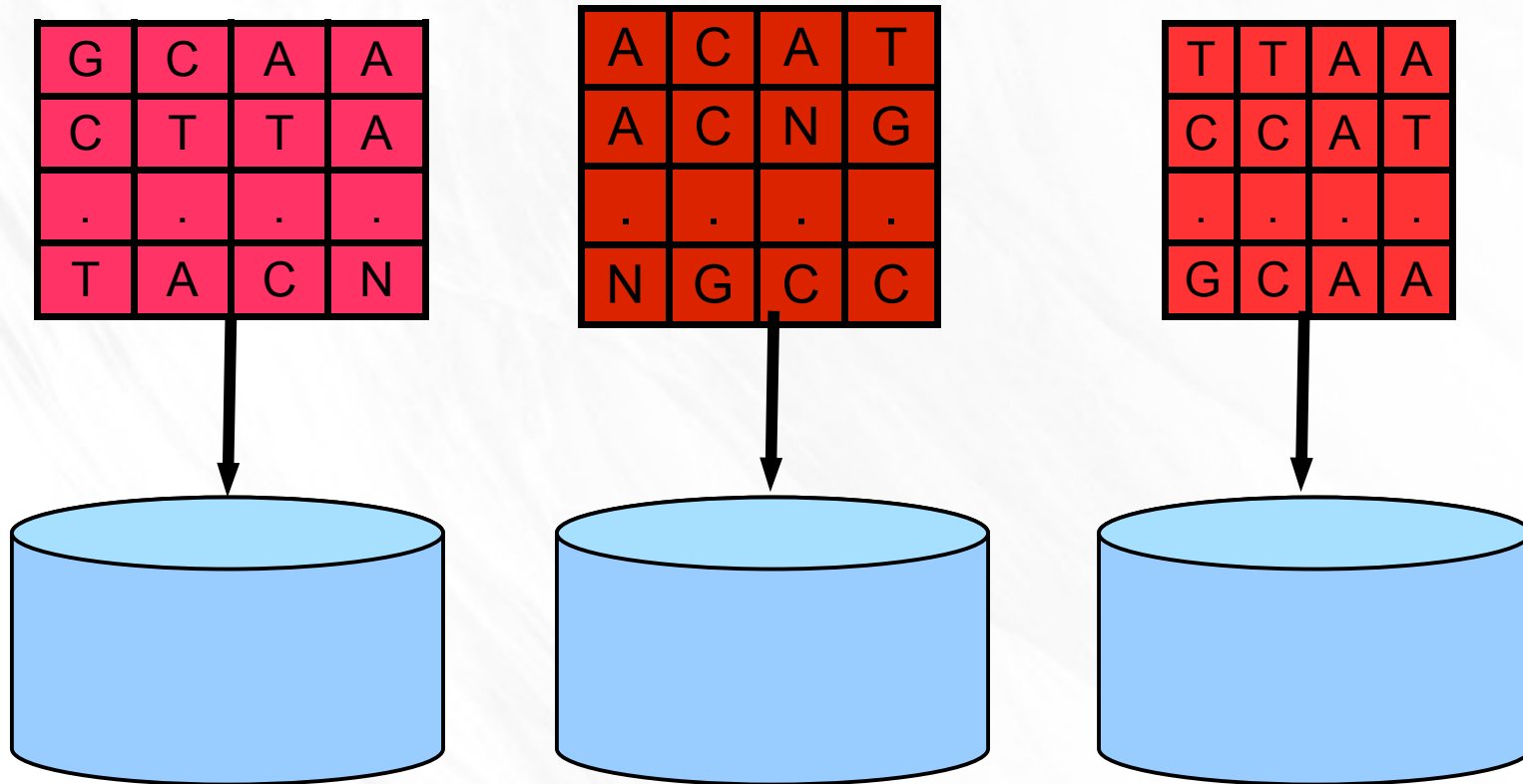
# Architecture Overview 2



**2. Divide the single-value array into overlapping partitions, or *chunks*.**



# Architecture Overview 3.



**3. Assign *chunks* to physical nodes in a massively parallel architecture.**

# Architecture 4: Query Processing

- **Query expressions parsed into physical plans.**
  - Minimal heuristics for coarse plan.
  - Run time re-organization of plan based on progress.
- **Parallelizable segments of plans distributed.**
  - Governing query manager coordinates segments.
- **Local segment execution.**
  - 'Chunk at a time' processing.
  - Extensible type system and Array-Value Operators.

# Project Status

- **Pick-up team of developers.**
  - Science project developers.
  - Experienced DBMS server types.
  - SciDB is an Open-Source (GPL-3) project.
- **Crash-or-Crash-Through development.**
  - Demonstration system at VLDB-09.
  - Currently developing V1, due Q2 2010.
- **Help!?**
  - Fun research questions to answer.
  - Lots of work to do.
  - <http://www.scidb.org/>