Towards an Optimized Illumina Microarray Data Analysis Pipeline

Pan Du, Simon Lin

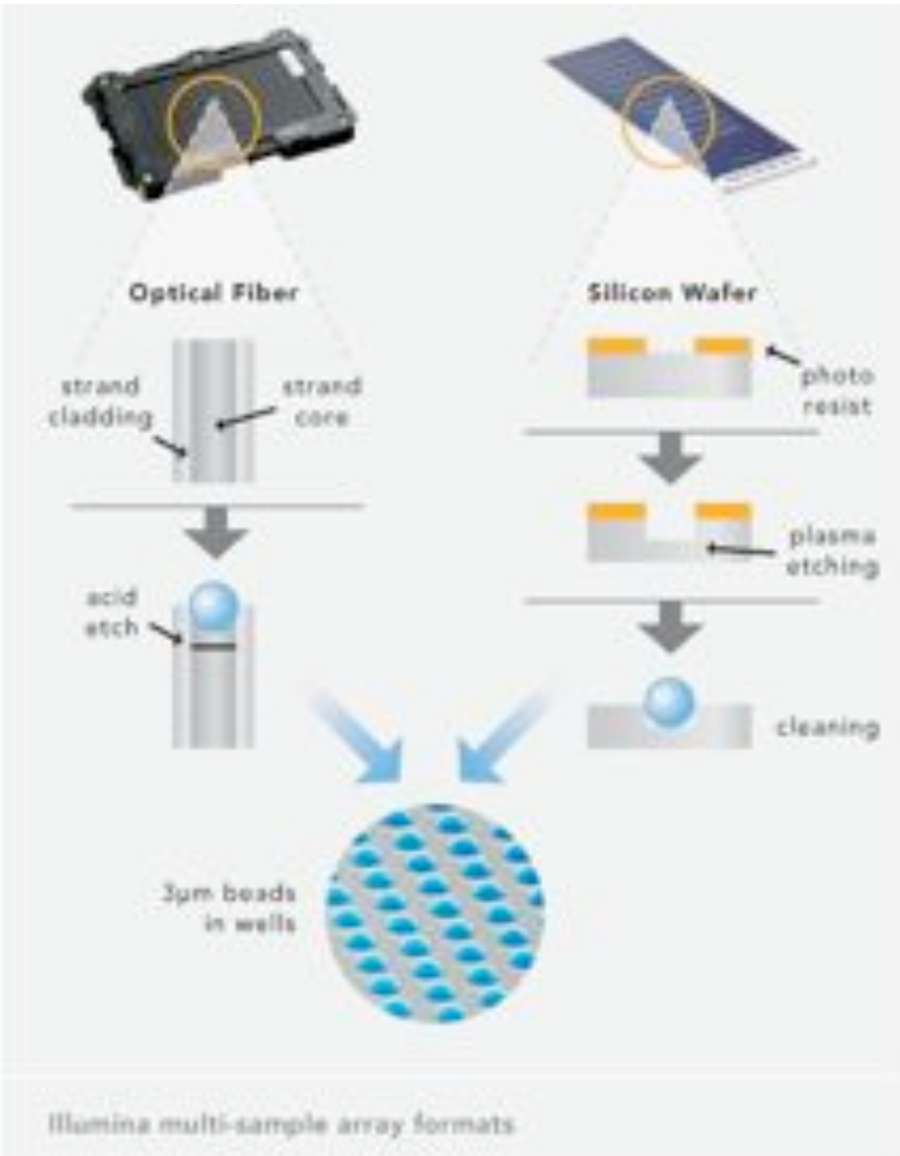
Robert H. Lurie Comprehensive Cancer Center,
Northwestern University

August 06, 2007

Outline

- Introduction of Illumina Beadarray technology
- Lumi package and its unique features
  - VST (variance stabilizing transform)
  - RSN (robust spline normalization)
  - nuID annotation packages
- Analysis pipeline
- Example R code

Illumina BeadArray Technology



Two formats: 96-sample Array Matrix and the multi-sample BeadChip formats

Uniform pits are etched into the surface of each substrate to a depth of approximately 3 microns prior to assembly.

Each type of bead has about 30 technique replicates

Beads are then randomly assembled and held in these microwells by Van der Waals forces and hydrostatic interactions with the walls of the well.

(Previous) Concerns

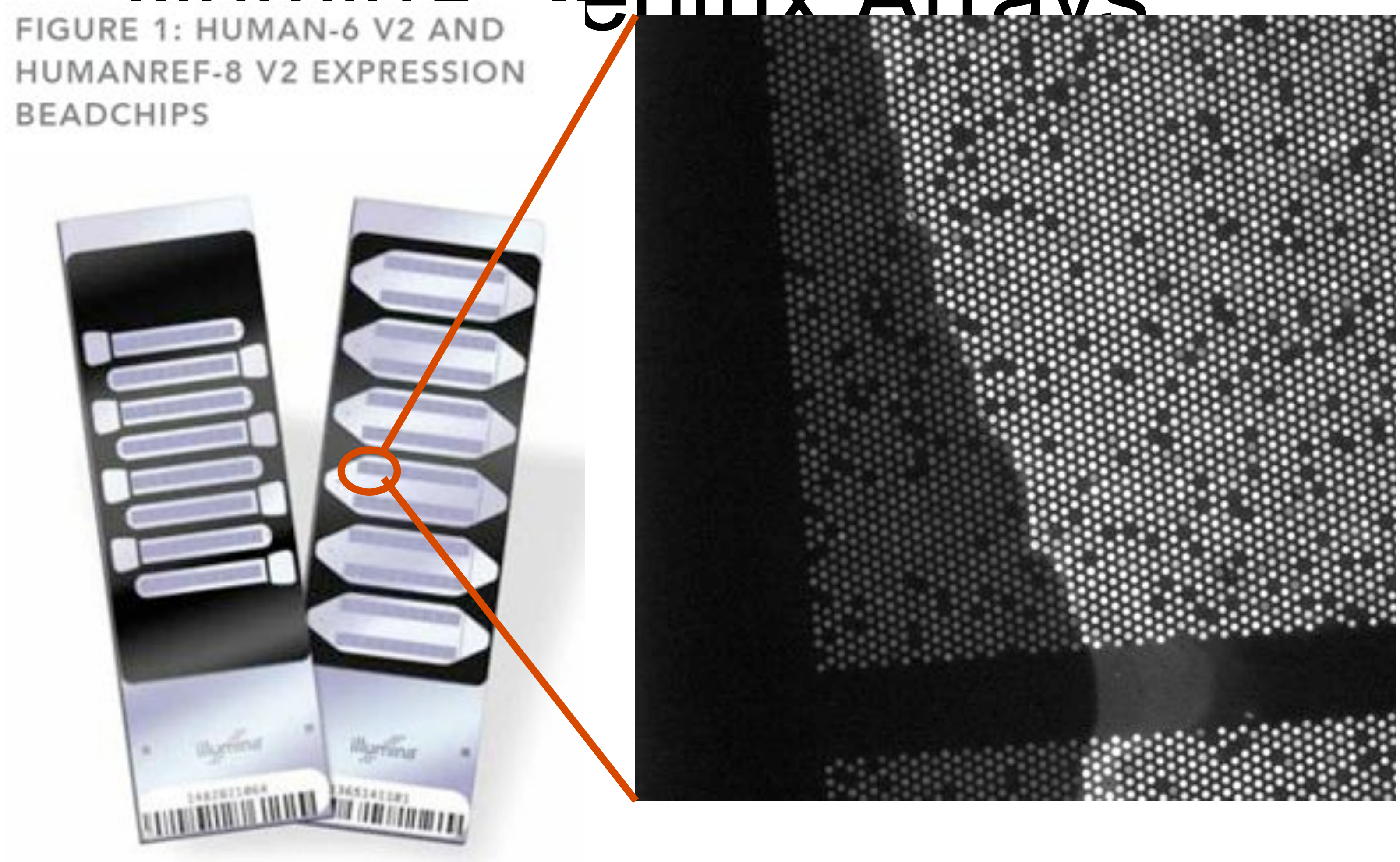
Challenges	Illumina Solutions
• Uneven distribution of BG (air bubble and washing) • Contamination of debris • Scratches on the surface	• Larger number of beads • Random distribution of beads
Spot morphology and uniformity	Coated beads instead of printing
Array manufacturing defect	Tested in the decoding process
Failure in labeling of mRNA	Labeling control on array
Scanning conditions	¿ Still a concern ?

(Previous) Concerns (continued)

Challenges	Illumina Solutions
Probe Specificity	50-mer design
Normalization issues	6 to 12 arrays on the same slide

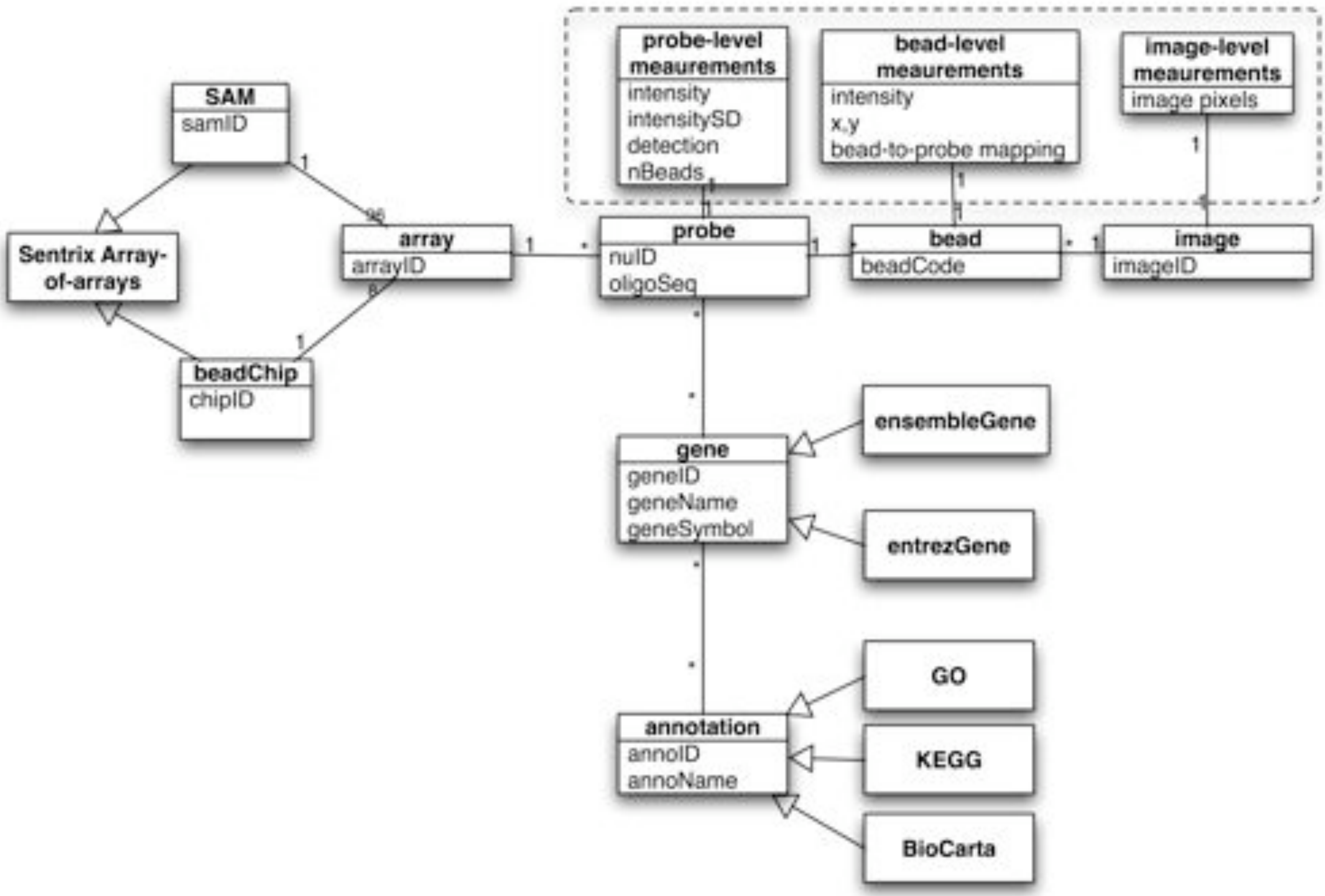
¿

Illumina Sentrix Arrays



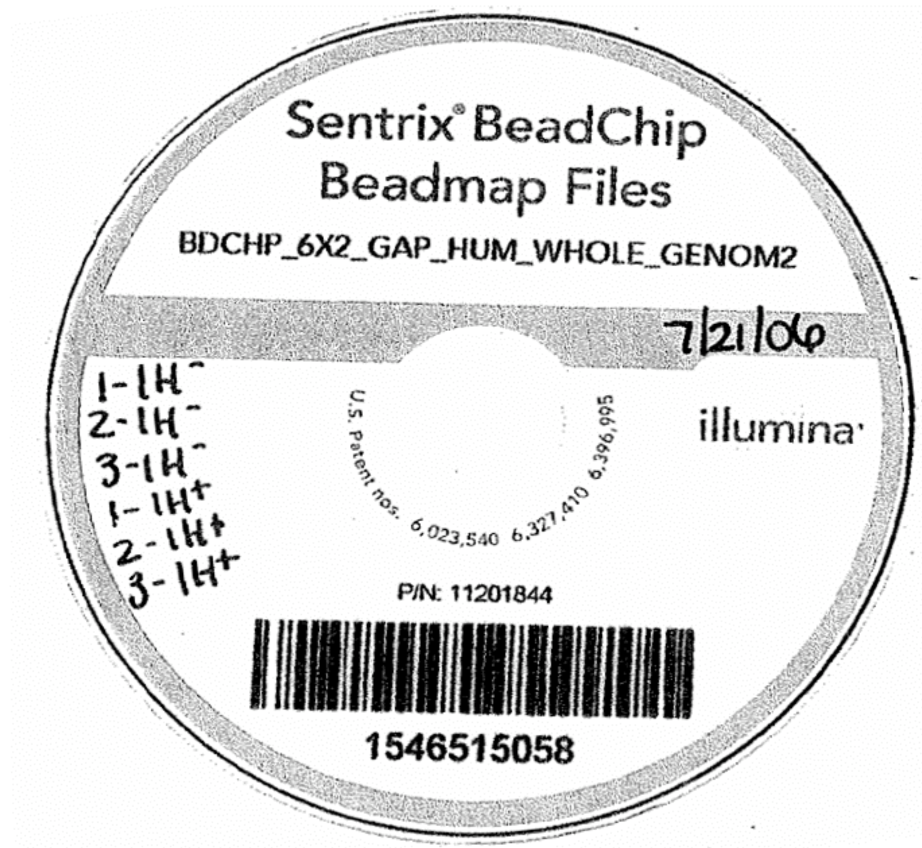
- Slide: 2 cm x 7cm
- Bead: 3 um

Illumina Terminology (UML diagram)



Each array is different

FIGURE 1: HUMAN-6 V2 AND HUMANREF-8 V2 EXPRESSION BEADCHIPS



Unique Features of Illumina Microarray

- Multiple samples on the same chip. Reduce the batch effects and cost per sample.
- Randomness
  - Randomly arranged beads, each of which carries copies of a gene-specific probe.
- Redundancy
  - For each type of beads, there are about 30 randomly positioned replicates.
- The preprocessing and quality control of Illumina microarray should be different from other types of microarrays.

Other Bioconductor Packages for Illumina Microarray

- *beadarray* is mainly designed for quality control and low-level analysis of BeadArrays.
- *BeadExplorer* is aimed to provide data exploration and quanilty control by leveraging the output of *Illumina BeadStudio* and existing algorithms in the *affy* package;
- Does not take the advantage of larger number of technical replicates available on the Illumina microarray in the preprocessing.

Design Objectives of *lumi* Package

- To provide algorithms uniquely designed for Illumina
- To best utilize the existing functionalities by following the class infrastructure and identifier management framework in Bioconductor

Functionalities of *lumi* Package

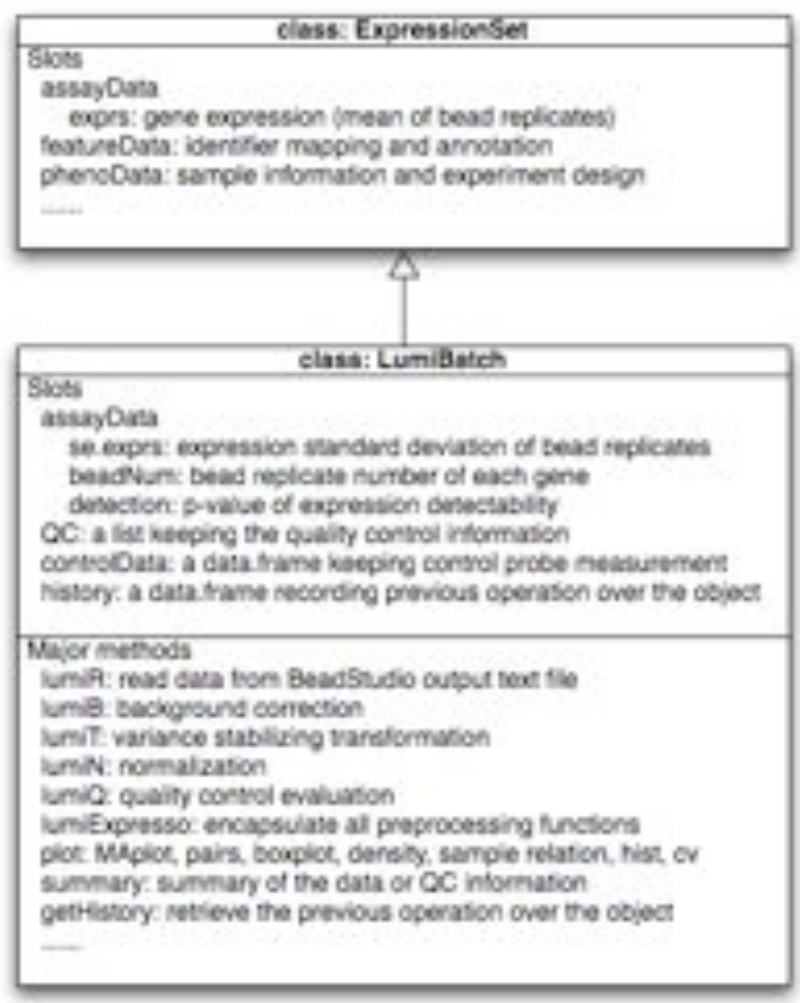
- Range from data import, quality control, preprocessing to gene annotation
- S4 class based and compatible with other Bioconductor packages
- Support the general microarray preprocessing algorithms

Unique features of *lumi* package

- Intelligent read data from BeadStudio output file
- A variance-stabilizing transformation (VST) algorithm
- A robust spline normalization (RSN) method
- The nucleotide universal identifier (nuID) identified annotation packages

Object Models

- Design based on the S4 Classes.
- One major class: lumiBatch
- Compatible with other Bioconductor packages;



Intelligently Load the Data

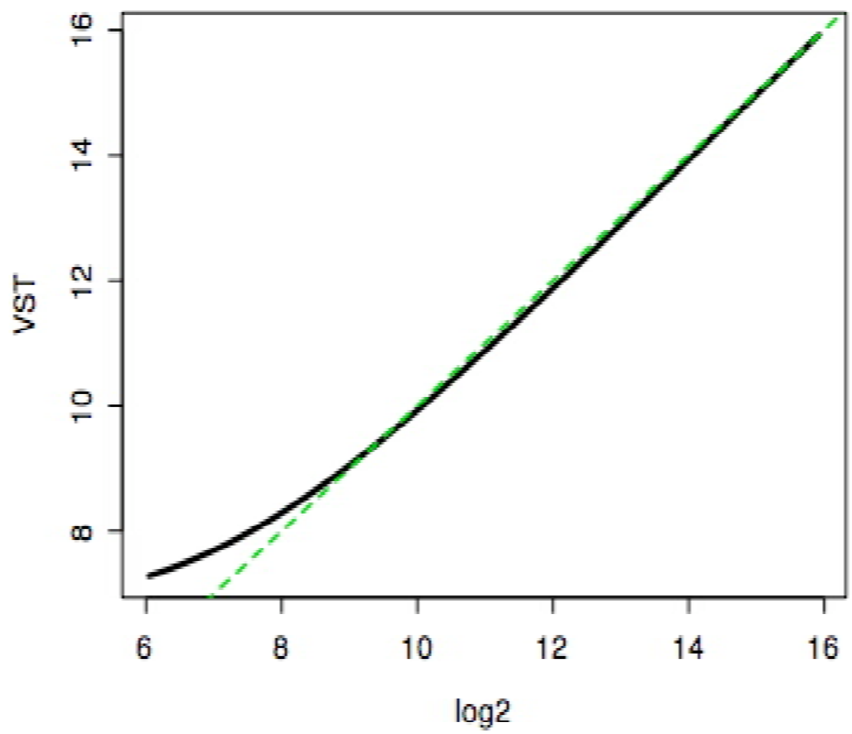
- Automatically recognize the format and different versions of the BeadStudio output
- Automatically check the identifiers (TargetID or ProbeID) used, and replaced them as nuID if the annotation library is provided.
- Extract the sample information from the sample IDs.
- lumi.x <- lumiR(fileName)

Variance Stabilization

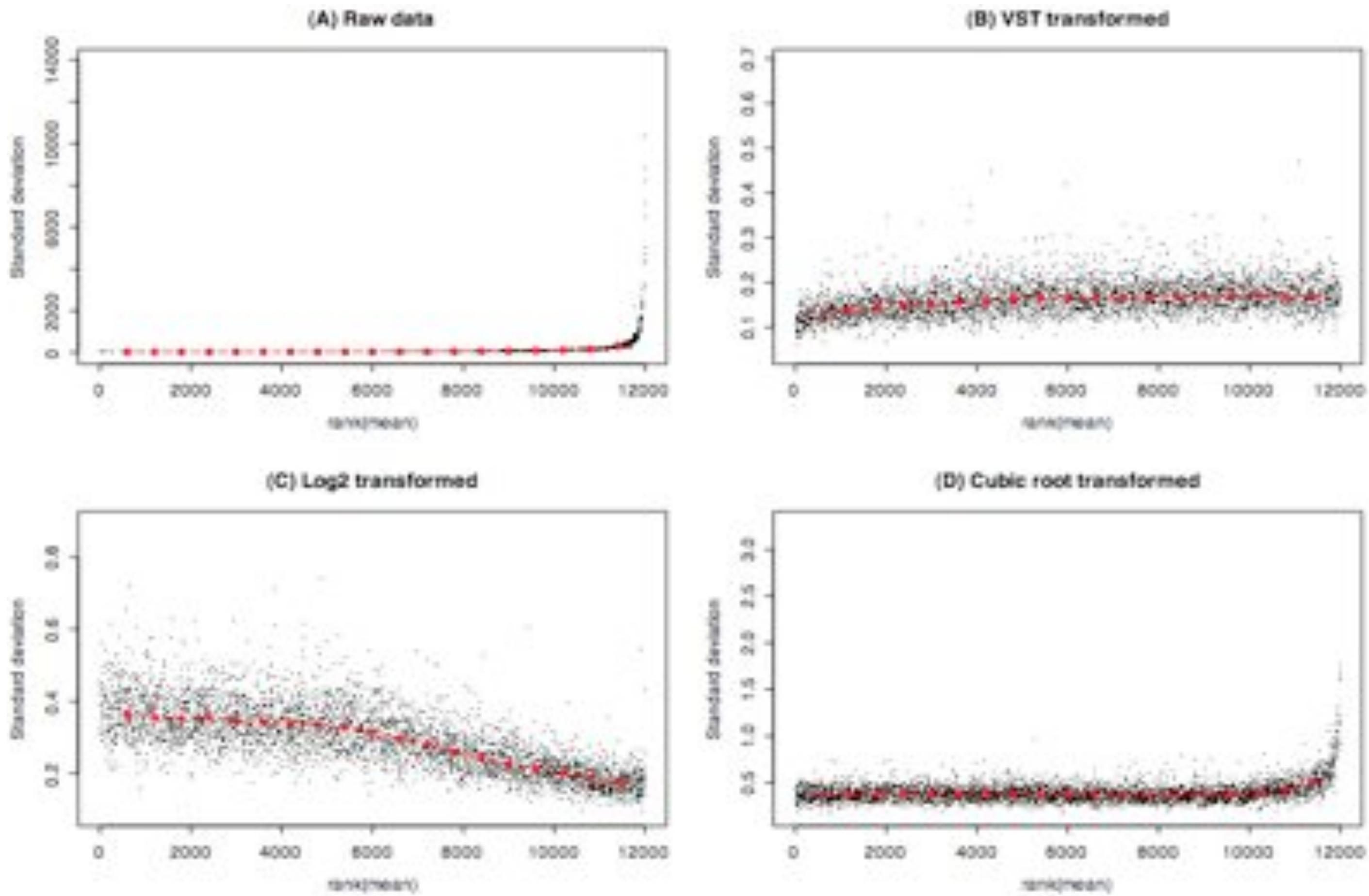
- General assumption of statistical tests to microarray data: variance is independent of intensity
- In reality, larger intensities tend to have larger variations
- Current implementation:
  - Log2 transform is widely used
  - VSN (Variance Stabilizing Normalization) combines variance stabilizing and normalization based on the limited replicates across chips

Variance Stabilizing Transformation (VST)

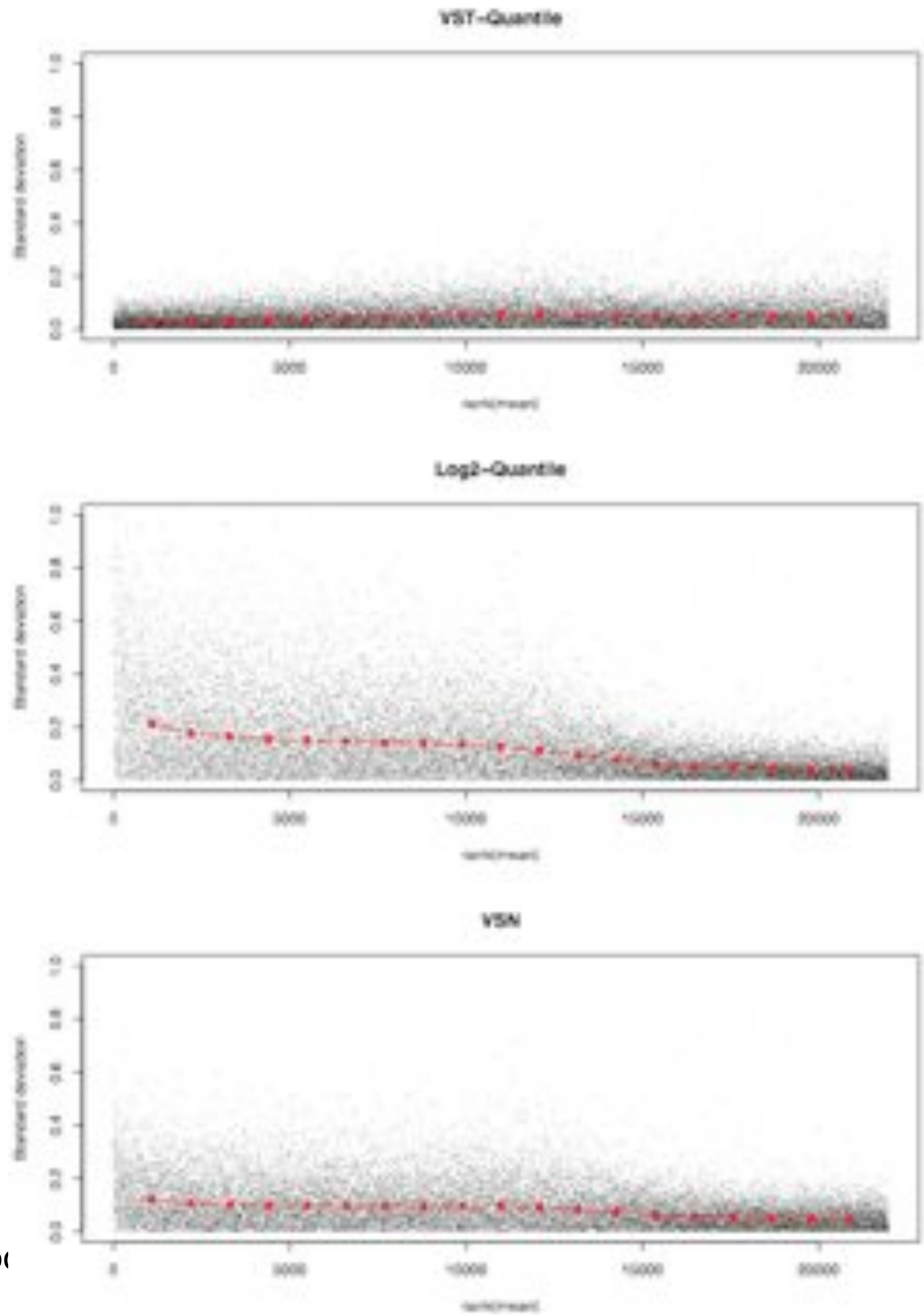
- Taking the advantage of larger number of technical replicates available on the Illumina microarray
- A generalized log transformation (arcsinh function)



Variance Stabilization of the Bead Level Data



Variance Stabilization of the Technical Replicates



Robust Spline Normalization (RSN)

- Quantile normalization:
  - Pros: computational efficiency, preserves the rank order
  - Cons: The intensity transformation is discontinuous
- Loess and other curve-fitting based normalization:
  - Pros: continuous
  - Cons: cannot guarantee the rank order. Strong assumption (majority genes unexpressed and symmetric distributed)
- RSN combines the good features of the quanitle and loess normalization

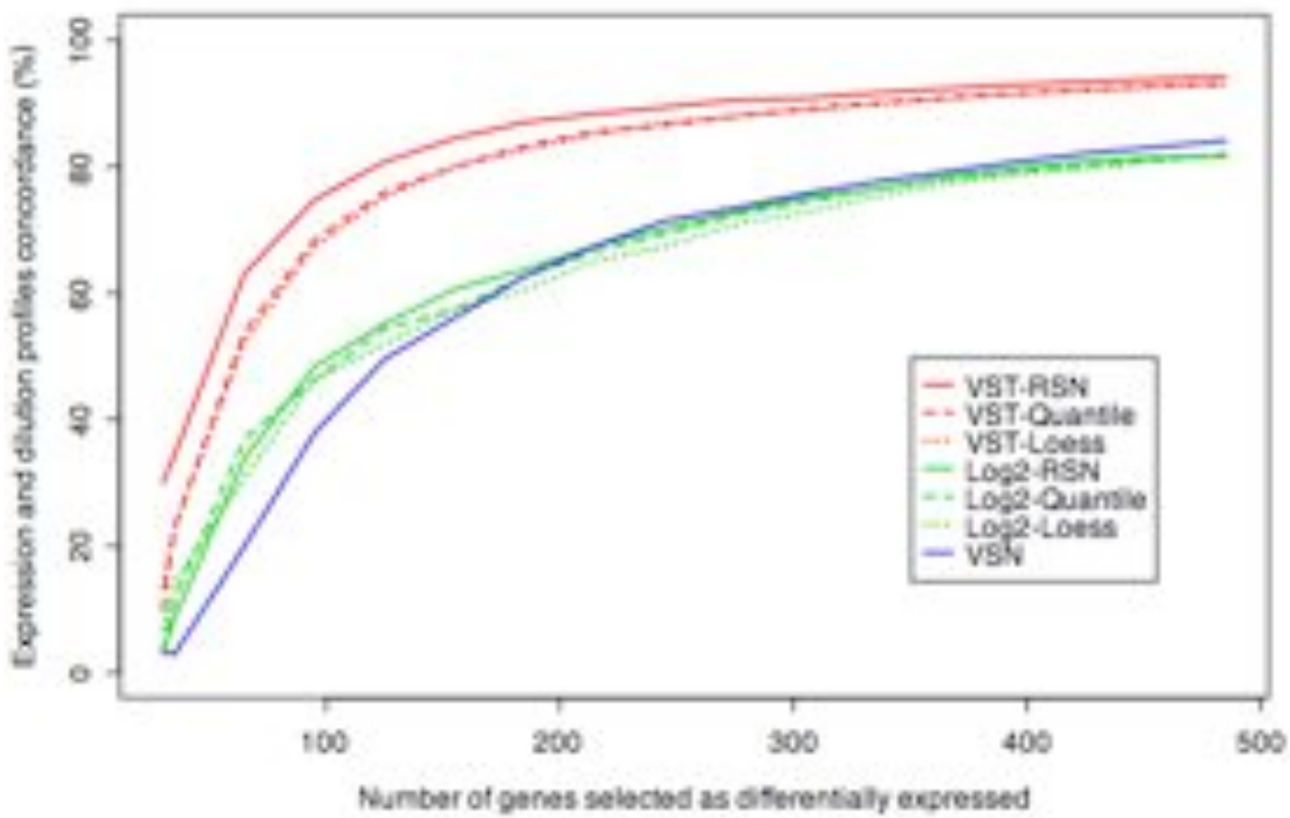
Comparison of curve fitting and quantile normalization

	Curve fitting based normalization	**Quantile normalization**
Assumption	Most genes are not differentially expressed.	All samples have the same distribution.
Approximation	Based on curve fitting	Replaced by the average of the probes with the same rank
Problems	Does not work well when lots of genes are differentially expressed.	Will lose small difference between samples, and the change is un-recoverable. Normalize across all samples, memory intensive.
Strengths	The value mapping is continuous. Normalize in pairwise, memory save.	Rank invariant Computationally efficient

Robust Spline Normalization (RSN)

- Combining the strength of curve fitting and quantile normalization
  - Continuous mapping
  - Rank invariant
  - Insensitive to differentially expressed genes.
- Basic Ideas of RSN
  - Perform a quantile normalization of the entire microarray dataset for the purpose of estimating the fold-changes between samples
  - Fit a weighted monotonic-constraint spline by Gaussian window to down-weight the probes with high fold-changes
  - Normalize each microarray against a reference microarray

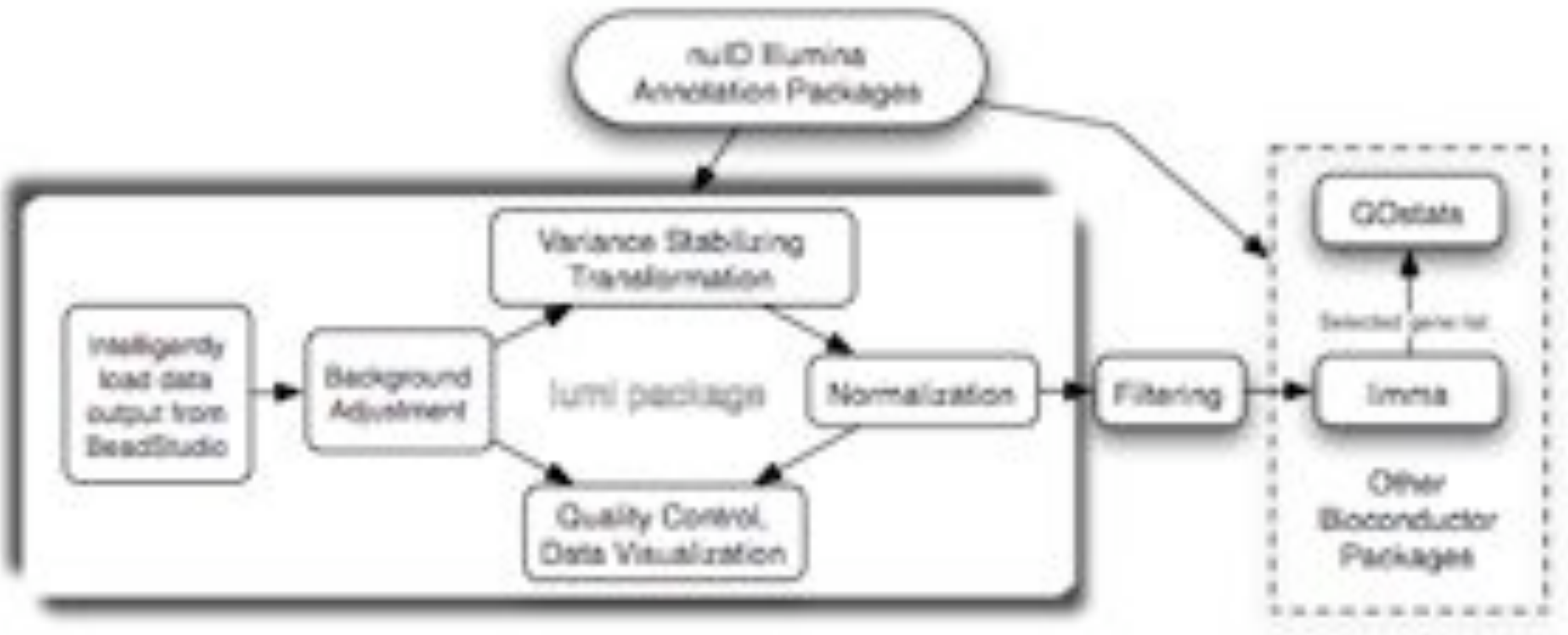
Performance Evaluation



nuID Illumina annotation packages

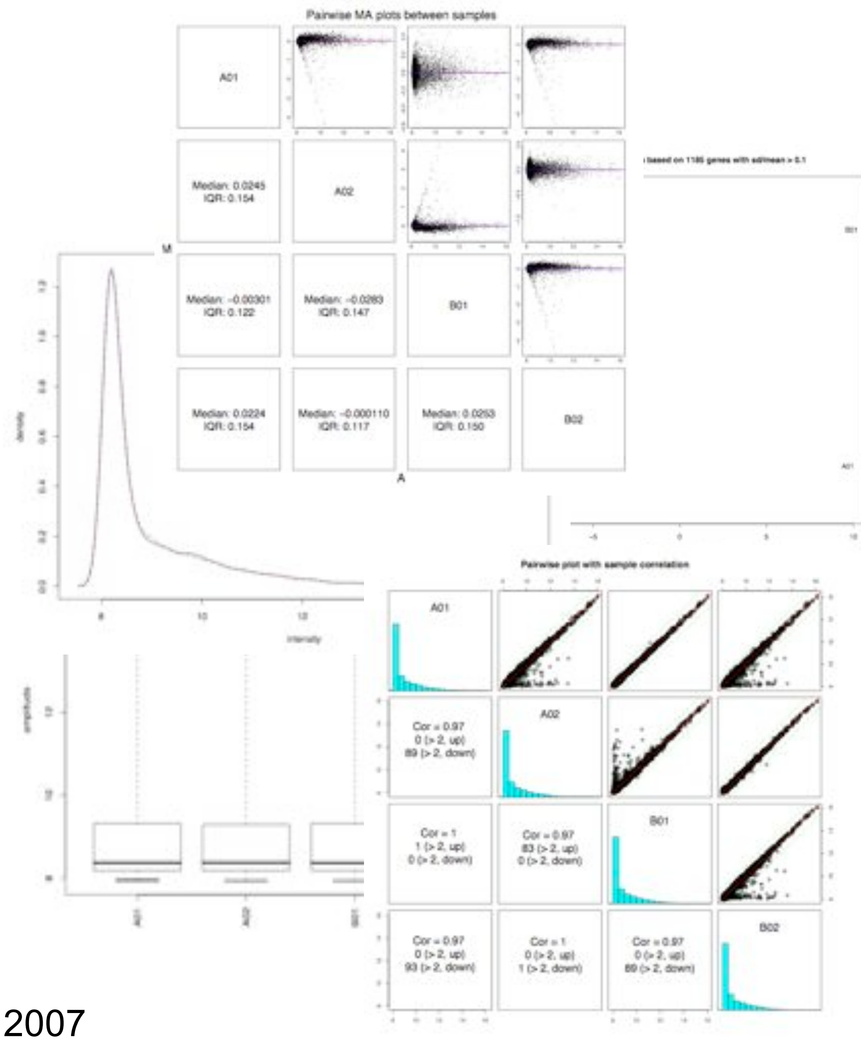
- Avoiding the inconsistence of the identification system currently used by Illumina microarray
- nuID is one-to-one correspondence to the probe sequence
- nuID is self-identifiable and including error checking
- Get the most updated probe annotation by blasting the RefSeq database

Analysis Pipeline



Example Code

```
> # load the library
> library(lumi)

> # specify the file name output from Bead Studio
>  fileName <- 'Barnes_gene_profile.txt'
> # Read the data and create a LumiBatch object
>  example.lumi <- lumiR(fileName, lib='lumiHumanV1')

> ## summary of data
> example.lumi
> ## summary of quality control information
> summary(example.lumi, QC )

> ## preprocessing and quality control after normalization
> lumi.N.Q <- lumiExpresso(example.lumi)
> ## summary of quality control information after preprocessing
> summary(lumi.N.Q, QC )

> ## plot different plots
> pairs(lumi.N.Q)
> plot(lumi.N.Q, what='sampleRelation')
> boxplot(lumi.N.Q)

> # Extract expression data for further processing
> dataMatrix <- exprs(lumi.N)
```



Benchmark Data Sets to compare Methods

	library (affy)	library (lumi)
Data sets	affycomp	lumiBarnes
Latin-square spike-in?	yes	no
Titration?	yes	yes

Part II: nuID

A Novel Identifier for Oligos, Ideal for Oligonucleotide-based Microarrays

Outline

- What is nuID?
- Why nuID?
- How does nuID work?

What is nuID

- nuID is the abbreviation of Nucleotide Universal Identifier
- nuID is a novel identifier for oligos, ideal for oligonucleotide-based microarrays

Why nuID?

Microarray Information Flow

GO:0051301

Cell Division

function

entrezID: 19645

symbol: Rb1

retinoblastoma 1

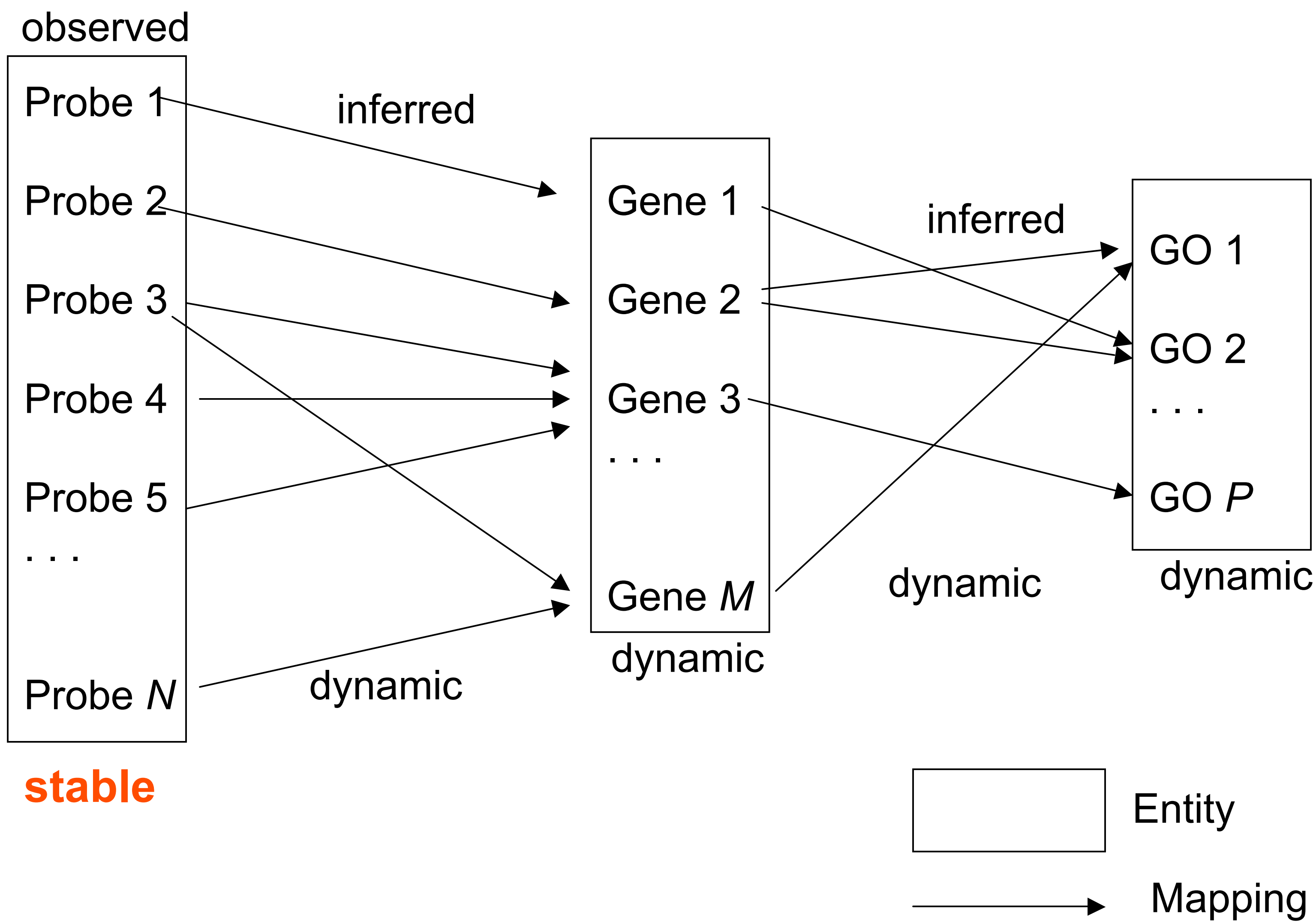
```
 1 ggccgggcgc gtccggtttt cctcgggga gttcccatta tttttgtaac gggantcggg
61 tgaggagggg gcgtgccccg cgtgcgcgcg cgaccgcccg cctccccgcg cgcctccctc
121 ggctgctcgc gccggcccgc gctgcgcgtc atgccgccca aagccccgcg cagagccgcg
181 gccgccgagc ccccgccacc gccgccgccg ccgcctcggg aggacgaccc cgcgcaggac
241 agcggccccg aagagctgcc cctggccagg cttgagtttg aagaaattga agaacccgaa
301 tttattgcat tatgtcaaaa gttaaaggta cccgatcatg tcagagaaag agcttggcta
...
```

gene

ID: ??

ggtacccgatcatgtcagagaa

probe


observed
Probe 1
Probe 2
Probe 3
Probe 4
Probe 5
. . .
Probe N
stable
inferred
dynamic
Gene 1
Gene 2
Gene 3
. . .
Gene M
dynamic
inferred
dynamic
GO 1
GO 2
. . .
GO P
dynamic
Entity
Mapping

GEO Accession viewer - Mozilla Firefox

NCBI

GEO Gene Expre

Platform GPL2507 | Query DataSets for GPL2507

Status	Public on Jun 03, 2005
Title	Sentrix Human-6 Expression BeadChip
Technology type	oligonucleotide beads
Distribution	commercial
Organism(s)	Homo sapiens
Manufacturer	Illumina Inc.
Manufacture protocol	http://www.illumina.com/technology/platform/tech_plat_arraymfg.ilmn
Catalog number	BD-25-101
Description	The Sentrix Human-6 BeadChip can be used to study expression of over 47,000 human transcripts. Researchers can generate whole-genome expression profiles for 6 samples in parallel on a single BeadChip. Array is composed of 3 micron features with average feature redundancy of 30-fold . All features are QCed by sequential hybridizations process called array decoding. Probes are fully screened all-full-length 50-mers. Assay requires 50-100ng of total RNA input. Array content is based on RefSeq and additional space is occupied by targets selected from Unigene build 163 and Gnomon databases.
Web link	http://www.illumina.com/General/pdf/Human6ExpressionDatasheet.pdf
Submission date	Jun 01, 2005
Organization	Illumina Inc.

Data table header descriptions

ID_REF	
VALUE	log quantile + median normalised data

Data table

ID_REF	VALUE
GI_10047089-S	6.009475
GI_10047091-S	6.341651
GI_10047093-S	10.478177
GI_10047099-S	8.358420
GI_10047103-S	12.346913
GI_10047105-S	6.518176
GI_10047121-S	5.997531
GI_10047123-S	10.103461

For Illumina microarrays, TargetID was used as the primary ID in the NCBI GEO database.

Challenges of Target IDs

- Not unique: “GI_28476905” and “scl0076846.1_142” are the same gene on Mouse_Ref-8_V1 chip.

  -- Synonyms.

- Not stable over time: “GI_21070949-S” in the Mouse_Ref-8_V1 chip but as “scl022190.1_154-S” in the later Mouse-6_V1 chip.

  -- IDs can be recycled or retired.

- Not universal across manufacturers

  -- Homonyms.

- Not interpretable without metadata: However, metadata (lookup table) is not always available in reality.

How to ensure one ID per item?

- How to enforce 1:1 mapping?

- How can it be globally unique?

- How can it be permanent?

Solution I: Central Authority

- GenBank/ EMBL / DDBJ
- May help enforcing 1:1 mapping of an ID and an entity
  - HUGO Nomenclauter Committee
  - "Givining unique and meaningful names to every human gene"
- May be infeasible either technically or socially

Solution II: nuID

- Unique, guaranteed
  - Each name identifies only one entity
  - Inherently enforces 1:1 mapping
  - Uniquely resolvable
- Globally unique, guaranteed
  - Decentralized
  - No ID registry necessary
- Permanent, guaranteed
- Carries information about the entity
  - White box
  - no need for a lookup table

How does nuID work?

nuID: the idea

- Sequence itself as the ID
- Combined with the following four features
  - Compression: make it shorter
  - Checksum
    - Prevent transmission error
    - Provide self-identification
  - Encryption: in cases where the sequence identity is proprietary
  - Digital watermark: identify issuer

How does nuID work?

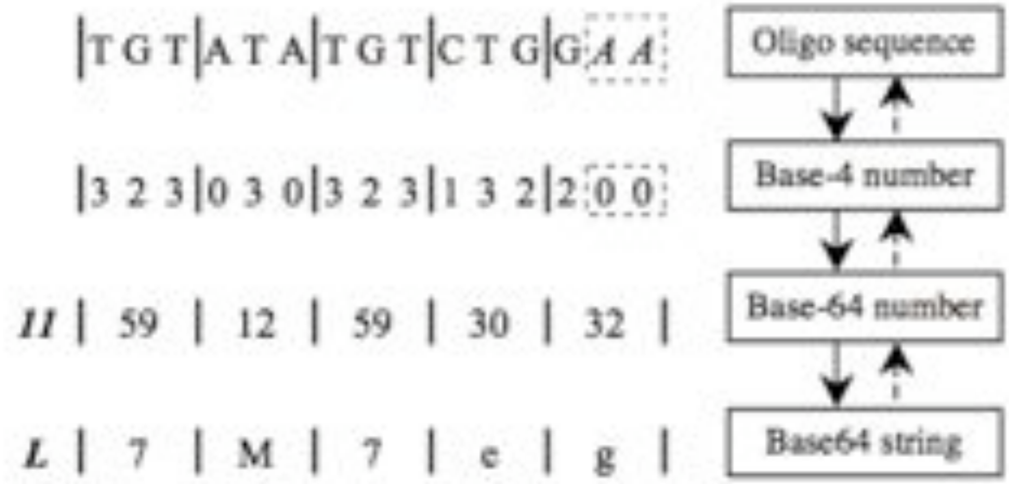


Figure 2
The encoding and decoding process of nuID. The solid arrows represent the encoding process, and the dashed arrows represent the decoding process. The bold-italic number 11 is the numeric value of the checking code "L". The "AA" at the end of sequence is the padded nucleotides.

Example of nuID

Array Type	Manufacturer's Proprietary Identifier	Nucleotide Sequence	nuID
Affymetrix Human	206064_s_at_probe1	TGTATATGTCTGGTTTTCTT ACCCC	a7M7ev98VQ
Illumina Human	GI_23097300-A	GCTTCACTCGCTTCCCAGG GGCTCCGTTCACCAACTAC ATGAGCTACACG	cn0dn1Sqdb0UHE4nEY
Illumina Mouse	TRBV23_AE000664_T _cell_receptor_beta_vari able_23_106-S	GACCCTTCGAAGTGAAAGA ACACAGTCATGTTATATGG TATAGTCATGGT	9hX2C4CBEtO8zrMtOs

Performance of checksum

Table 2: The error detection power of the nuID checksum algorithm (N = 21)

L	1-character	2-character	3-character	Random
25mer	0.97780	0.97918	0.98689	0.99924
50mer	0.97724	0.97838	0.98607	0.99997
100mer	0.97894	0.97825	0.98617	1*

L and N are defined in Equation (3) and (4) in Methods. The column "1-character" is the error detection rate of an nuID with only one character mutated. Similar definition for column "2-character" and "3-character". "Random" column is error detection rate of a random ASCII string. The optimum detection power is 1.0.

* We realize the detection of nuIDs for 100mers is not guaranteed, but in none of our simulations did we ever encounter a randomly assembled string that was a valid nuID.

Implementation of nuID

- We have build nuID based annotation packages for all Illumina expression chips.
- We have set up a website for nuID conversion and check latest annotation for the probe.
- The implementation is also included in the lumi package.

Conclusions

- For microarray reporting: Probe-level data is preferred over gene-level data.
- nuID is universal, globally unique, and permanent.
- Do not need a central authority to issue nuID.