Towards an Optimized Illumina Microarray Data Analysis Pipeline

Pan Du, Simon Lin

Robert H. Lurie Comprehensive Cancer Center,
Northwestern University

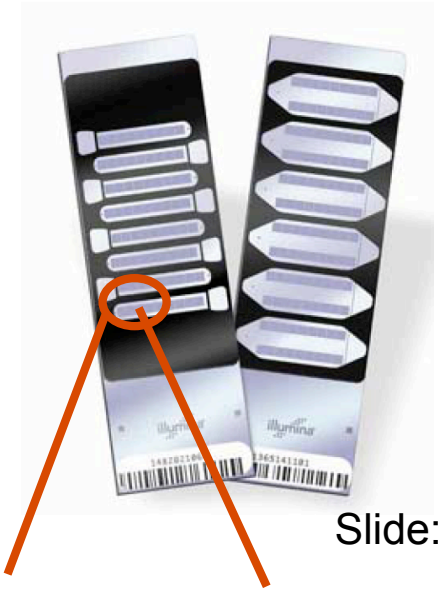
Oct 01, 2007

Outline

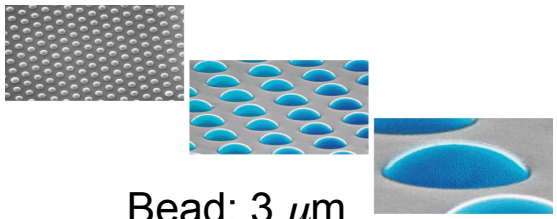
- Introduction of Illumina Beadarray technology
- Lumi package overview
- nuID and related annotation packages
- VST (variance stabilizing transform)
- RSN (robust spline normalization)

Illumina BeadArray Technology

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



Slide: 2 x 7cm



Bead: 3 μm

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 on average

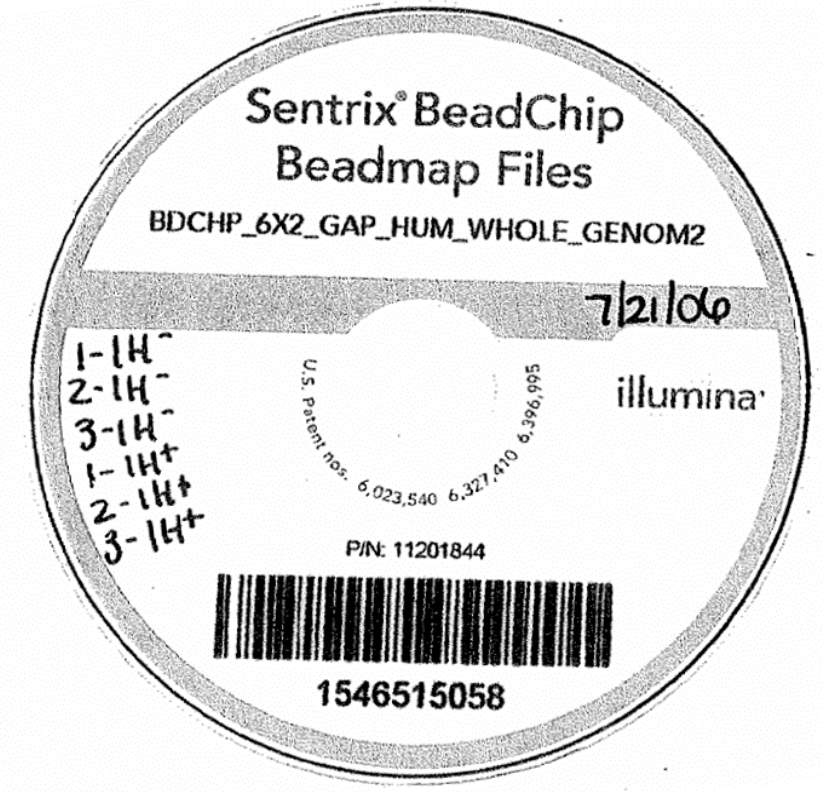
Beads are randomly assembled and held in these microwells

Multiple arrays on the same slide

Cost: < $200

Each array is different

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



(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 ?
Probe Specificity	50-mer design
Normalization issues	6 to 12 arrays on the same slide

Affymetrix vs. Illumina

	Affymetrix	Illumina
Redundancy	Low (usually one)	High (tens of replicates)
Probe location	Fixed	Random
Probe length	25 mer	50 mer
Probe vs. gene	probe → probe-set → gene	probe → gene
Array layout	One array per chip	Multiple arrays per chip

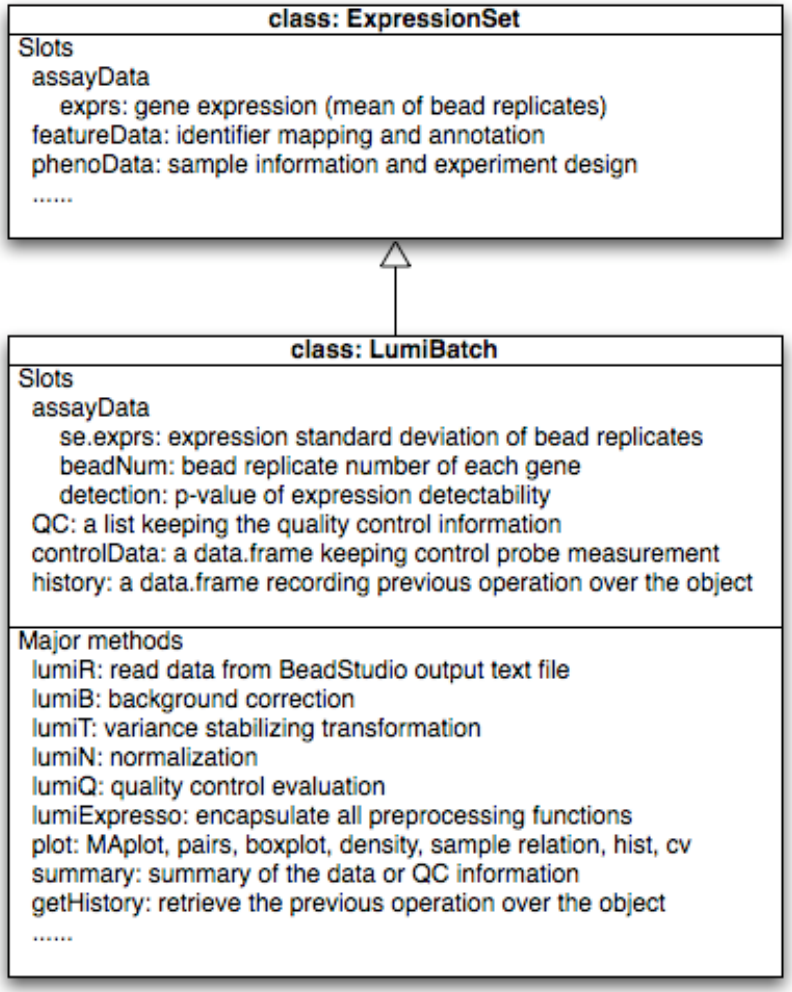
Overview of *lumi* package

Design Objectives of *lumi* Package

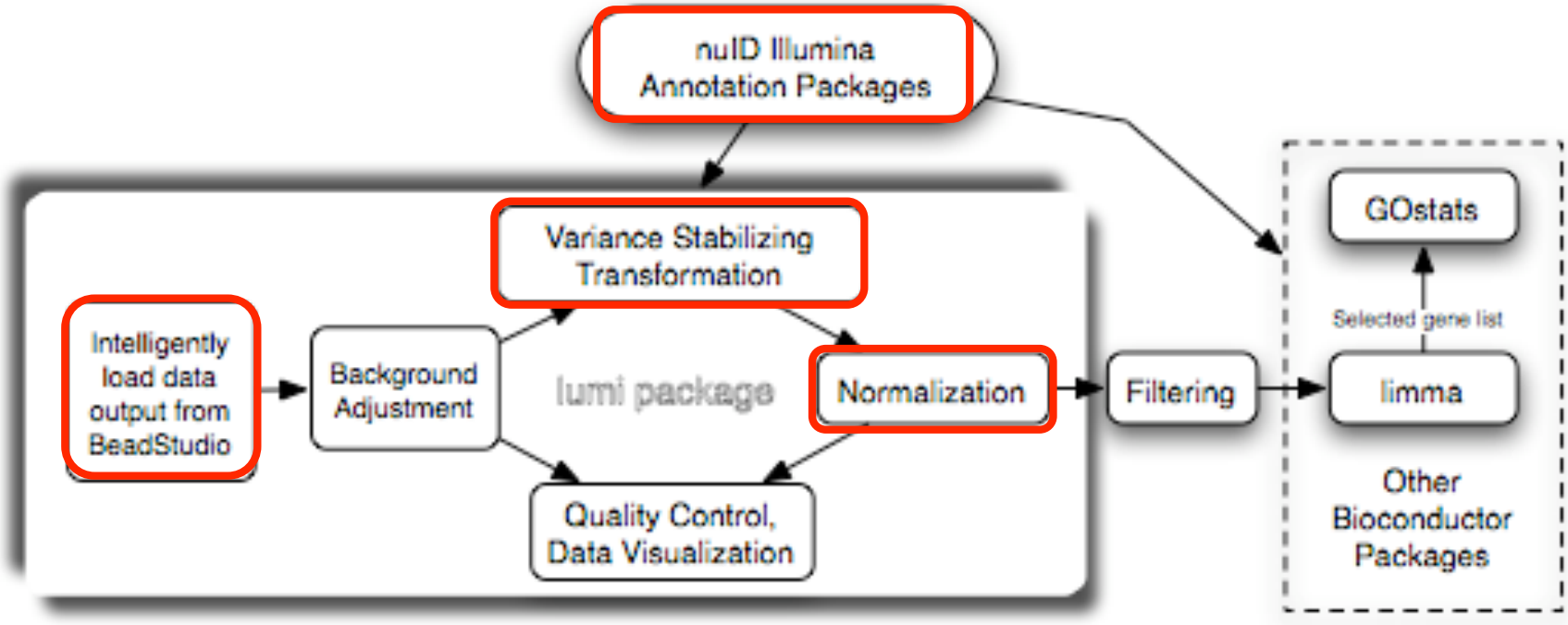
- To provide algorithms uniquely designed for Illumina
- To best utilize the existing functionalities by following the class infrastructure and identifier management framework in Bioconductor

Object Models

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

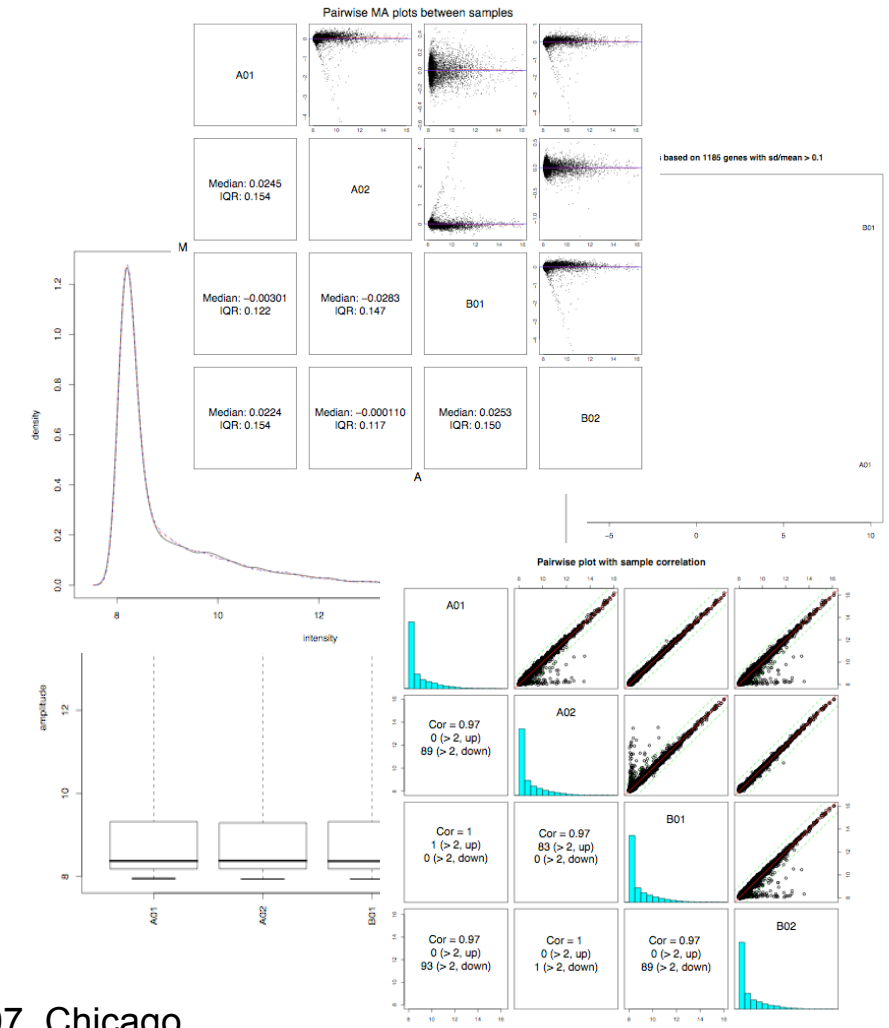


Analysis Pipeline



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)
```

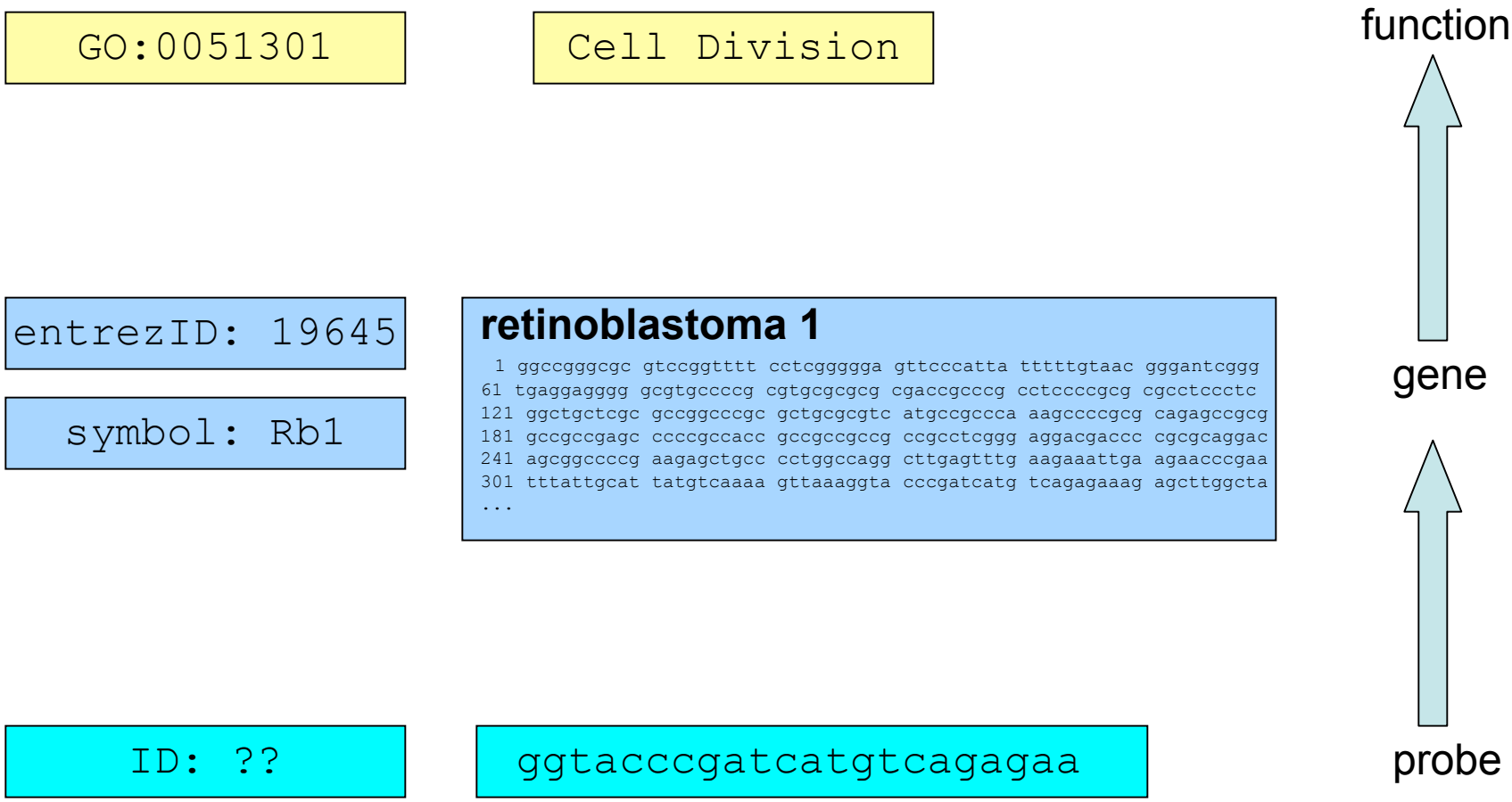


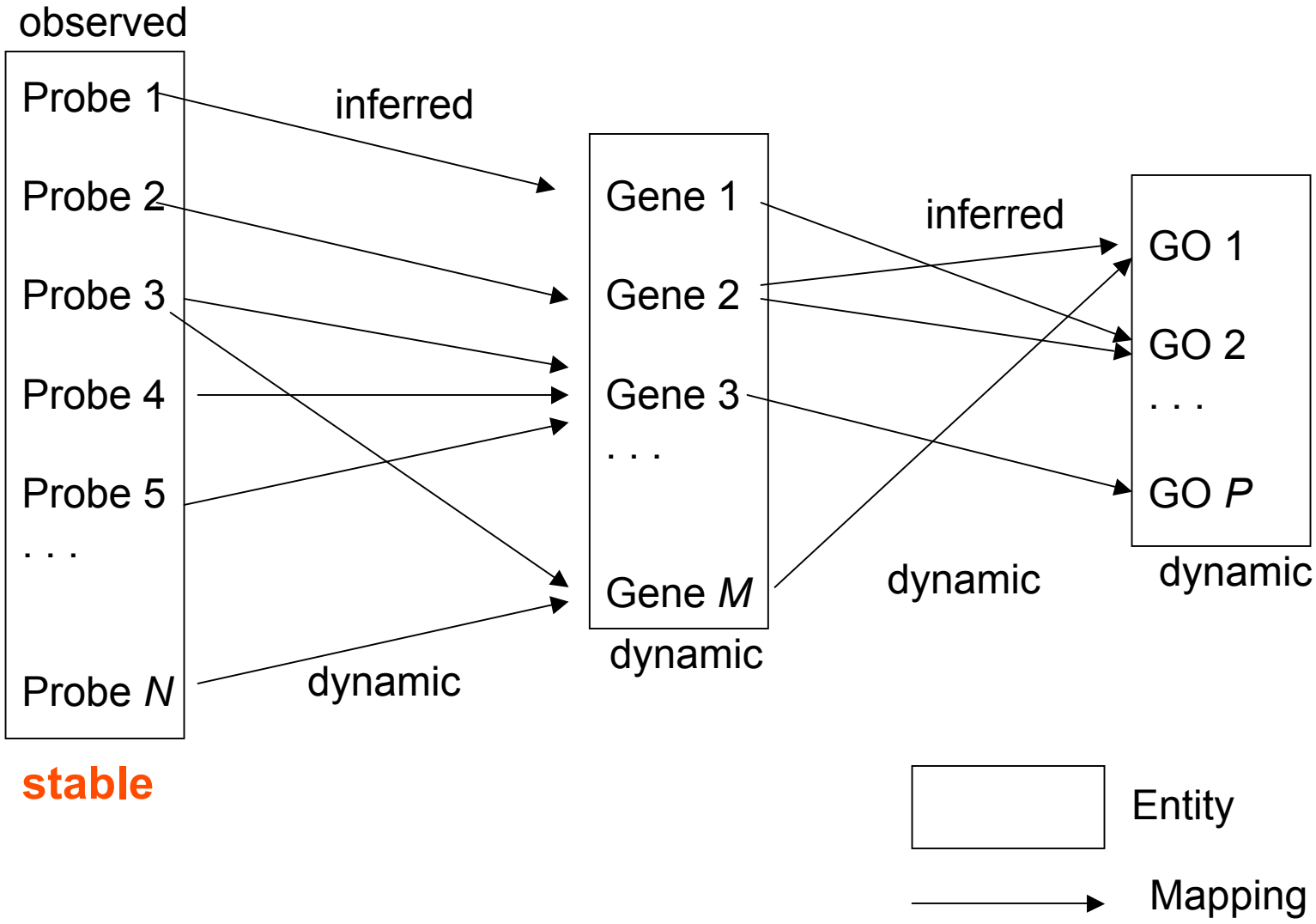
nuID and Illumina Annotation Packages

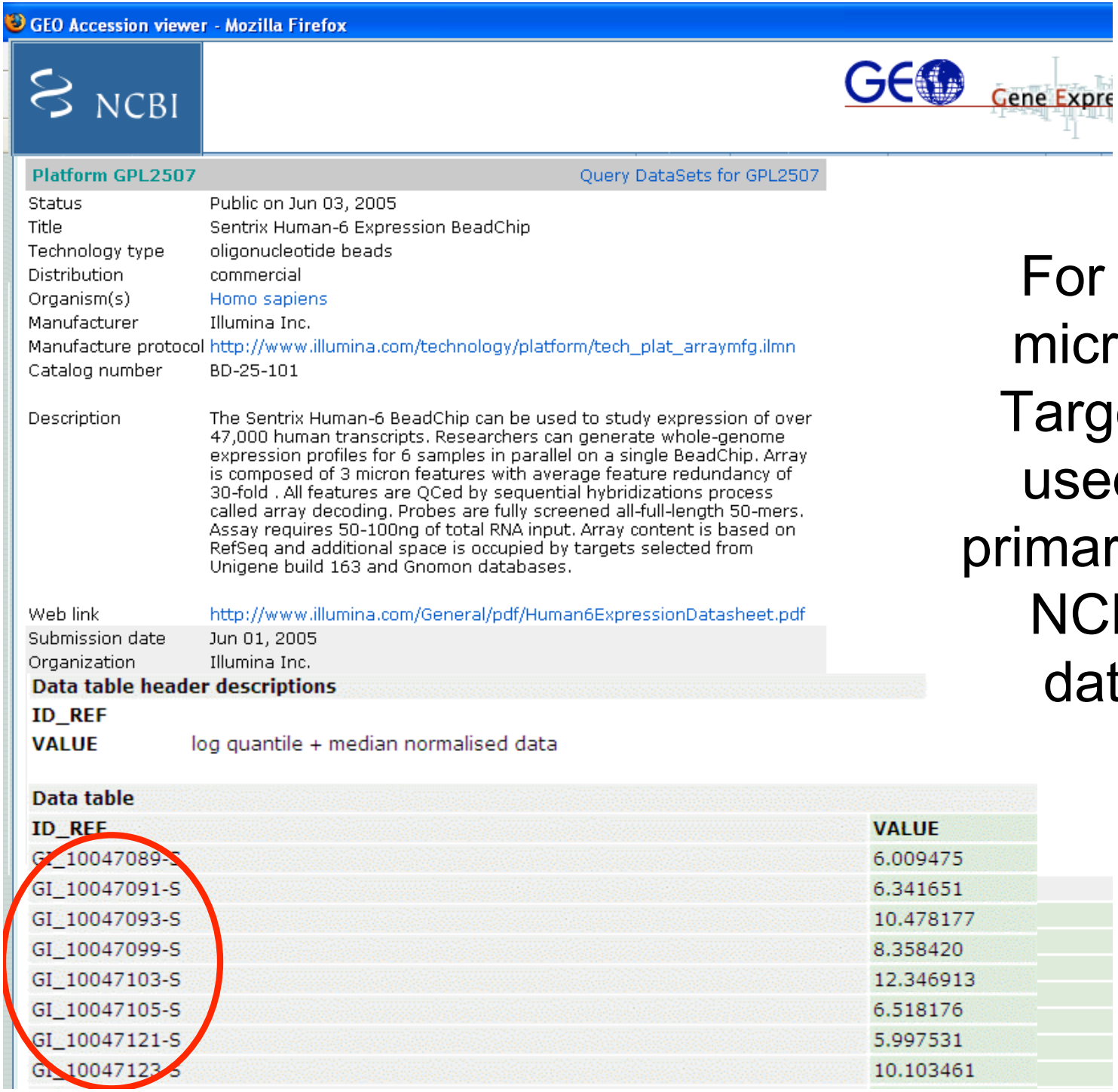
What is nuID

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

Microarray Information Flow




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



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

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?

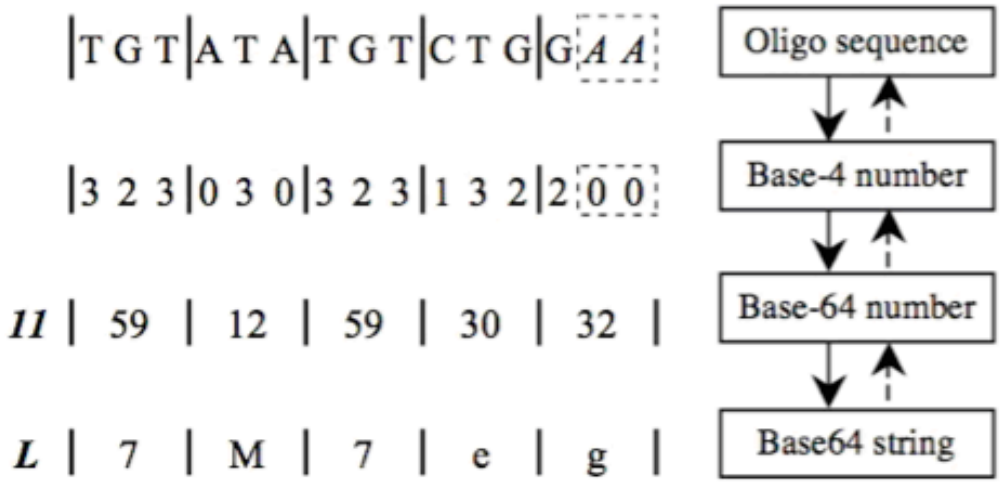


Figure 2
The encoding and decoding process of nuID. The solid arrows represent the encoding process, and the dashed arrows represent the decoding process. The bold-italic number 11 is the numeric value of the checking code "L". The "AA" at the end of sequence is the padded nucleotides.

Example of nuID

Array Type	Manufacturer's Proprietary Identifier	Nucleotide Sequence	nuID
Affymetrix Human	206064_s_at_probe1	TGTATATGTCTGGTTTTCTTACCCC	a7M7ev98VQ
Illumina Human	GI_23097300-A	GCTTCACTCGCTTCCCAGGGGCTCCGTTCACCAACTACATGAGCTACACG	cn0dn1Sqdb0UHE4nEY
Illumina Mouse	TRBV23_AE000664_T_cell_receptor_beta_variable_23_106-S	GACCCTTCGAAGTGAAAGAACACAGTCATGTTATATGGTATAGTCATGGT	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.

Illumina Annotation Packages

- Produced nuID indexed annotation packages for all Illumina expression chips. (named as lumiHumanV1, …)
- In the future, the packages will be based on the most updated RefSeq matches with nuID and their annotations.

Summary

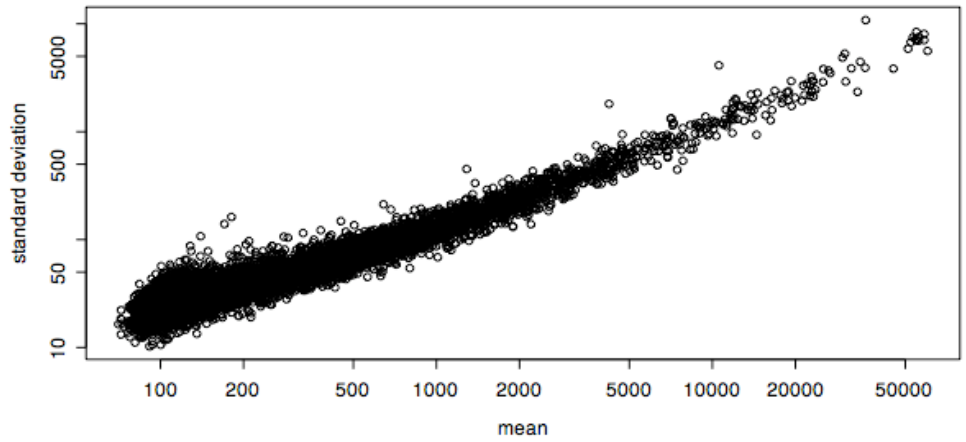
- For microarray reporting: Probe-level data is preferred over gene-level data.
- nuID is universal, globally unique, and permanent.
- Do not need a central authority to issue nuID.

Variance Stabilization Transformation

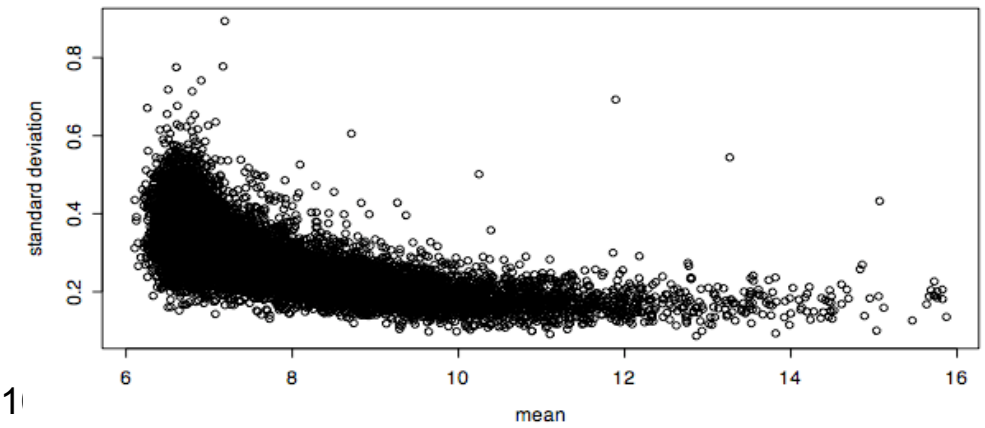
Variance Stabilization

- 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
- Variance stabilization through a generalized log transformation

Example of Mean and Variance Relation



(A) Raw



(B) Log2 transformed

Variance Stabilization

- Mathematical model

$$Y = \alpha + \mu e^{\eta} + \varepsilon \qquad (1)$$

- Asymptotic variance-stabilizing transformation

$$h(y) = \int^{y} 1/\sqrt{v(u)}\,du$$

- Mean and variance relation

$$h(y) = \begin{cases} 1/c_1 \ \operatorname{arcsin} h(c_2/\sqrt{c_3} + c_1/\sqrt{c_3}\ y), & \text{when } c_3 > 0 \\ 1/c_1 \ln(c_2 + c_1 y), & \text{when } c_3 = 0 \end{cases}$$

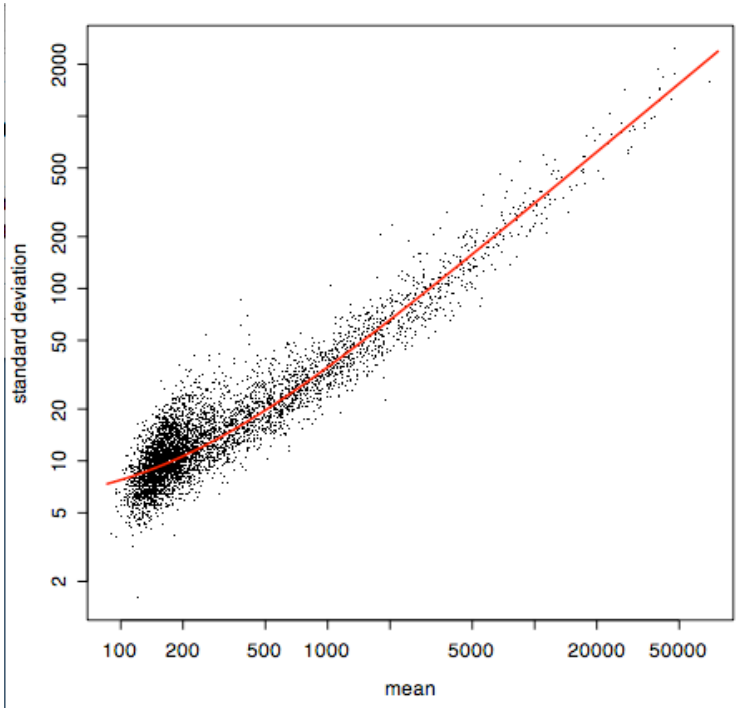
VSN (Variance Stabilizing Normalization)

- Estimation the mean and variance relation based on limited technique replicates
- Combines variance stabilizing and normalization based on the limited replicates across chips
- Assumption: most genes are not differentially expressed
- Sometimes unstable due to the above reasons.

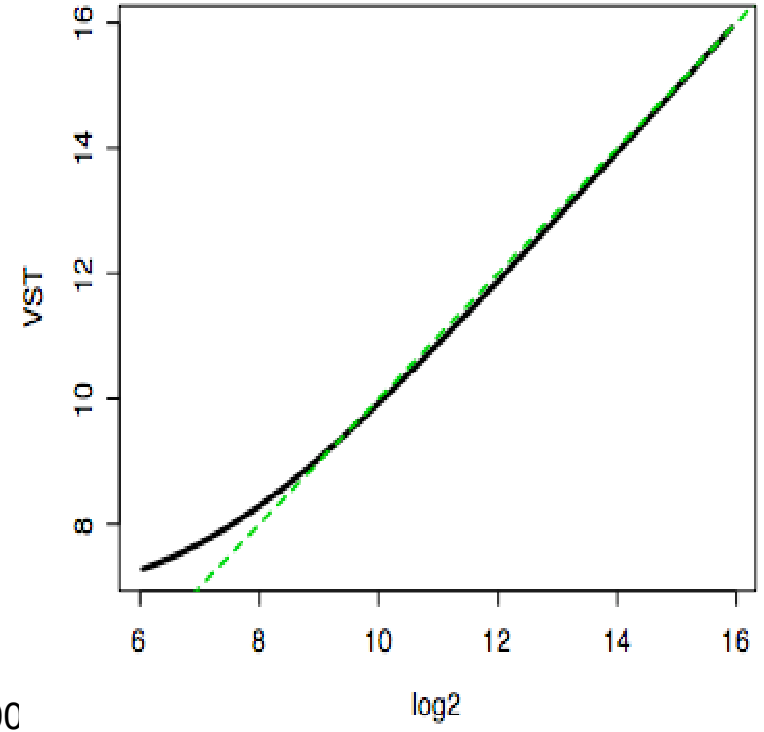
Variance Stabilizing Transformation (VST)

Illumina BeadArray technology enables better variance stabilizing

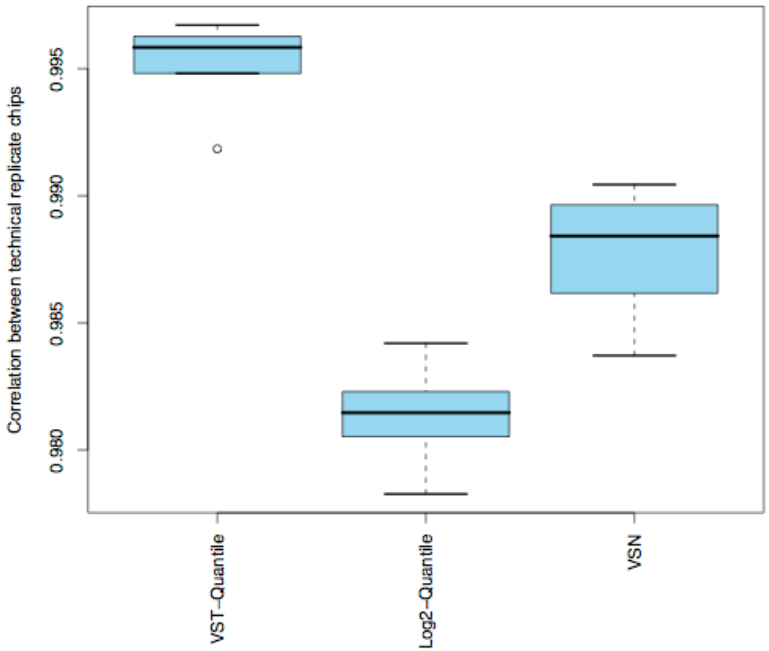
Better fit the relations between mean and standard deviation

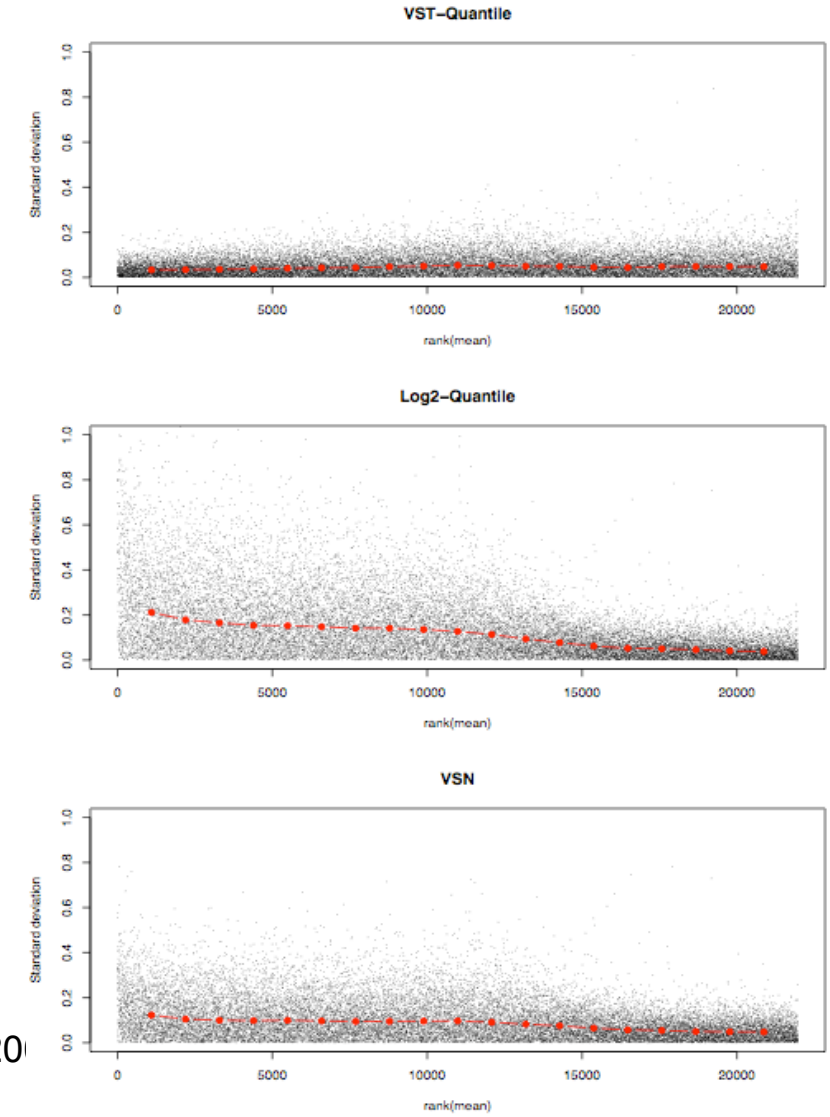


Relations between log2 and VST (arcsinh)



Variance Stabilization of the Technical Replicates





Comparison of Log2, VSN and VST

Table 1. Comparisons of log2, VSN and VST

	log2	**VSN**	**VST**
Error model for each individual array	None	Equation (1)	Equation (1)
Estimated from	None	Between-array replicates	Within-array replicates
Requires built-in normalization	No	Yes	No
Negative value	No	Yes	Yes
Parameter estimation method	Fixed mathematical transformation	Maximum likelihood integrated with normalization	Linear fitting
Assumptions of the replicates	None	Most of the genes are not differentially expressed; thus, they can be treated as replicates.	No such assumption required because the probes are in the same array.
Observed or assumed replicates	Not used	Usually less than a dozen	Usually over 30

Robust Spline Normalization

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)

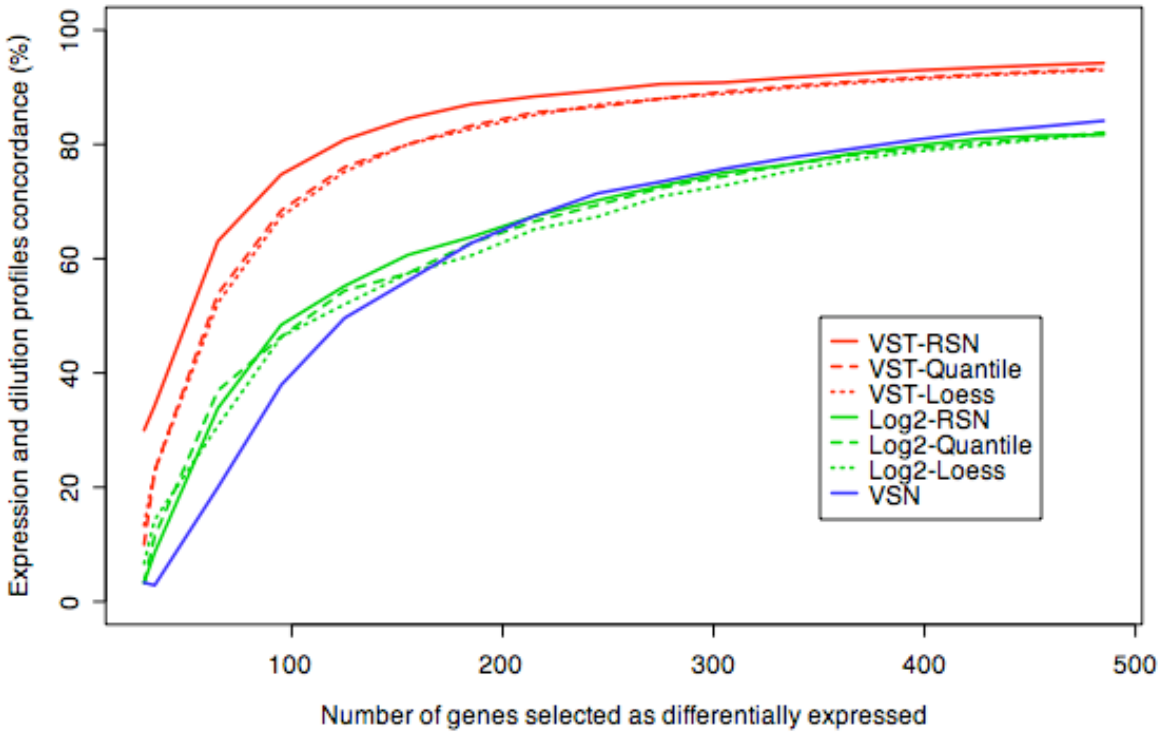
- Combining the strength of curve fitting and quantile normalization
  - Continuous mapping
  - Rank invariant
  - Insensitive to differentially expressed genes.
- Basic Ideas of RSN
  - Perform a quantile normalization of the entire microarray dataset for the purpose of estimating the fold-changes between 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

Algorithms Evaluation

Evaluation Data Sets

- Barnes data: (Barnes, M., et al., 2005)
  - measured a dilution series (two replicates and six dilution ratios: 100:0, 95:5, 75:25, 50:50, 25:75 and 0:100) of two human tissues: blood and placenta.

Performance Evaluation



Based on Barnes titration data

VST improves the concordance between the expression profiles and the real dilution ratio profiles

Conclusions and Future Plan

- Lumi package provides a pipeline of Illumina microarray preprocessing and annotation
- Provide algorithms uniquely designed for Illumina
- Options to use other traditional algorithms and compatible with other Bioconductor packages
- In the future,
  - enhance the quality control part
  - extend the lumi package to other Illumina data:
    - DNA copy number analysis
    - Methylation profiling
    - SNP and genotyping

Acknowledgements

- Robert H. Lurie Comprehensive Cancer Center, Northwestern University
  - Warren A. Kibbe and other members in the Bioinformatics group
  - Nadereh Jafari, microarray core
- European Bioinformatics Institute, UK
  - Wolfgang Huber
- The Walter and Eliza Hall Institute of Medical Research, Australia
  - Gordon Smyth