Parallelized preprocessing algorithms for high-density oligonucleotide array data

Markus Schmidberger
Ulrich Mansmann

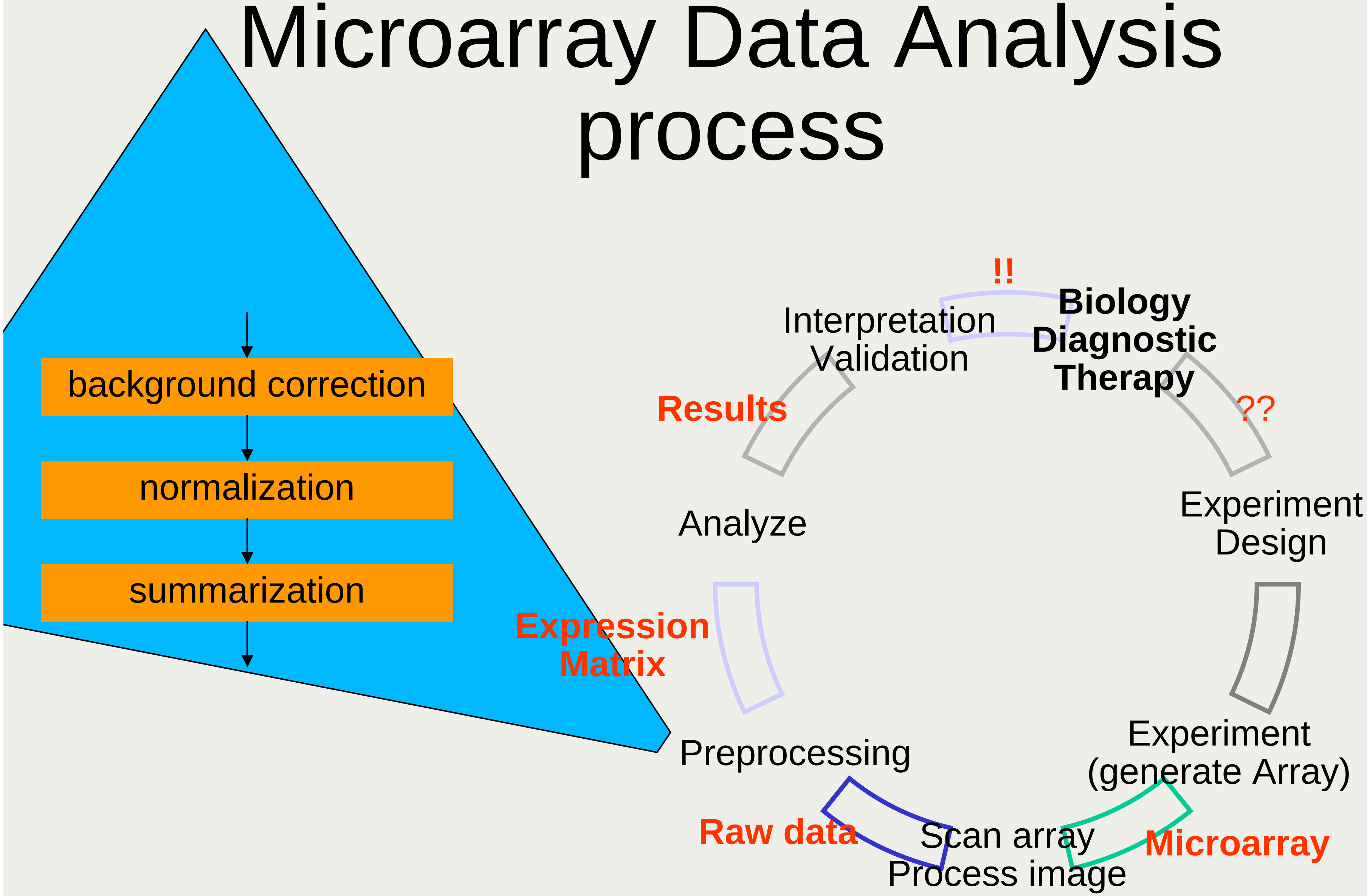
IBE
http://ibe.web.med.uni-muenchen.de

HiCOMB 2008
Seventh IEEE International Workshop on High Performance Computational Biology
April 14, 2008; Miami, Florida, USA

Overview

1. Introduction to Microarrays
2. Problems & Challenges & Solutions
3. Parallelization and R
4. Parallelization for preprocessing
5. Results

Microarray Data Analysis process



Sources of errors

amount of RNA in the biopsy
efficiencies of

- RNA extraction
- reverse transcription
- Labeling
- fluorescent detection

- probe purity and length
- distribution
- spotting efficiency, spot size
- cross-/unspecific hybridization
- stray signal

Systematic

- similar effect on many measurements
- corrections can be estimated from data

↓

Calibration

Stochastic

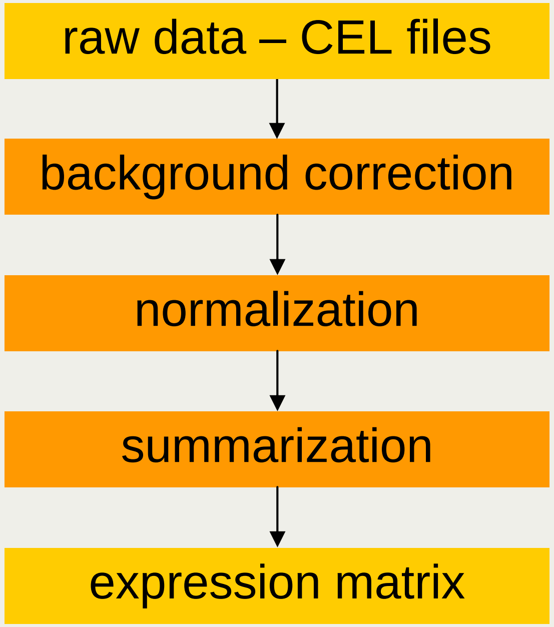
- too random to be explicitely accounted for
- remain as “noise”

↓

Error model

Preprocessing

- Background correction
  - remove noise of optical detection system
- Normalization
  - make measurements comparable from different array hybridizations
- Summarization
  - transcripts are represented in multiple probes



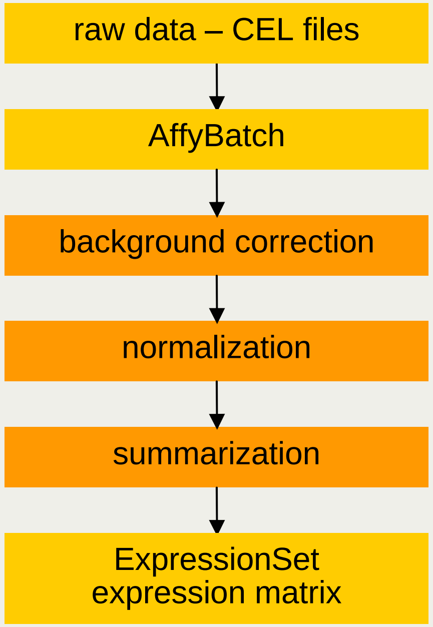
Existing software

- R and BioConductor mostly used in research (Open Source)

- Various different algorithms
  - With different advantages and disadvantages
  - No optimal solution (quality <-> effectiveness)

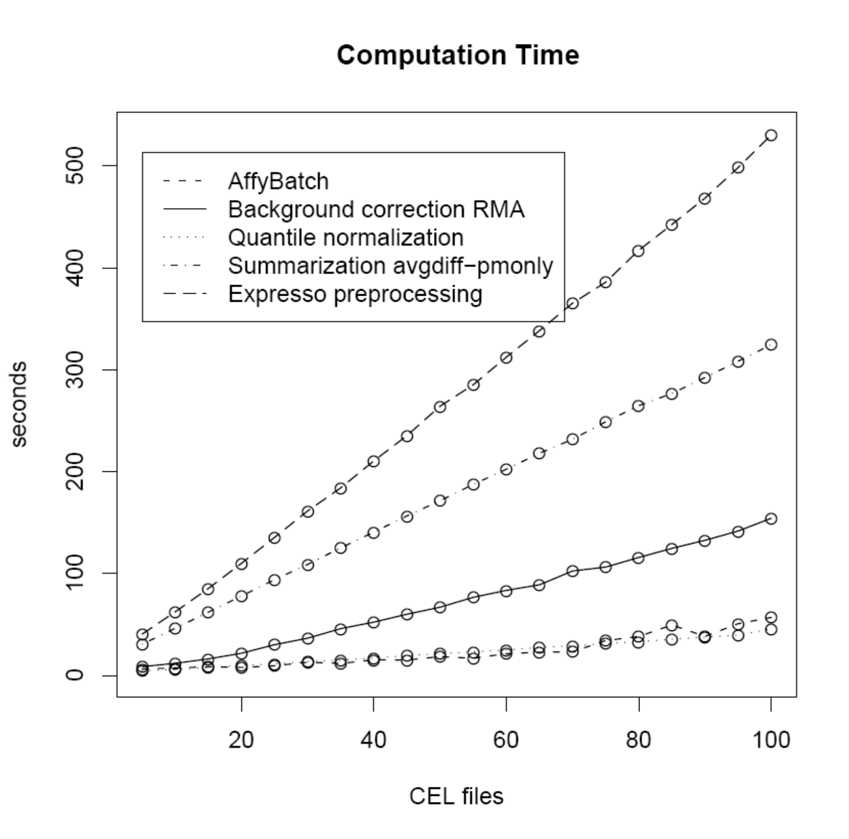
- Some approved and often used methods RMA, MAS 5.0, Quantile normalization, Cyclic loess, VSN, expresso, GCRMA

Problems



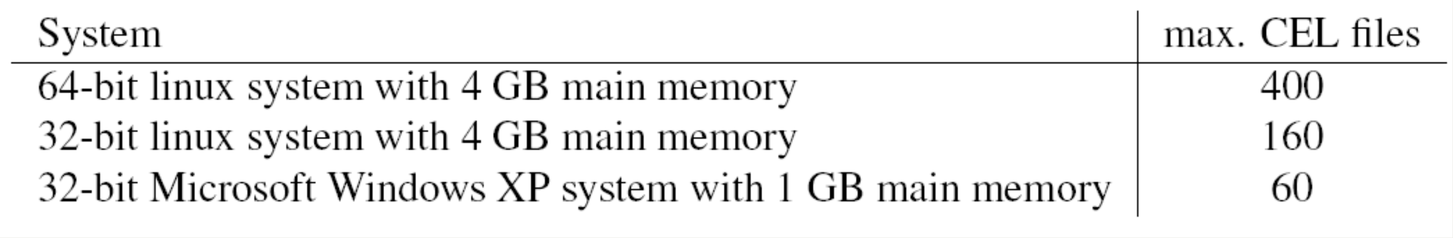
- Data-structure of R
  - data are stored in class ‘AffyBatch’
  - complex class with a lot of different slots
  - AffyBatch is memory intensiv
- Performance of algorithms
  - Inefficient program structure
  - Long computation time

Problems



How many arrays can I RMA process?
(Ben Bolstad)

http://bmbolstad.com/misc/ComputeRMAFAQ/size.html



System	max. CEL files
64-bit linux system with 4 GB main memory	400
32-bit linux system with 4 GB main memory	160
32-bit Microsoft Windows XP system with 1 GB main memory	60

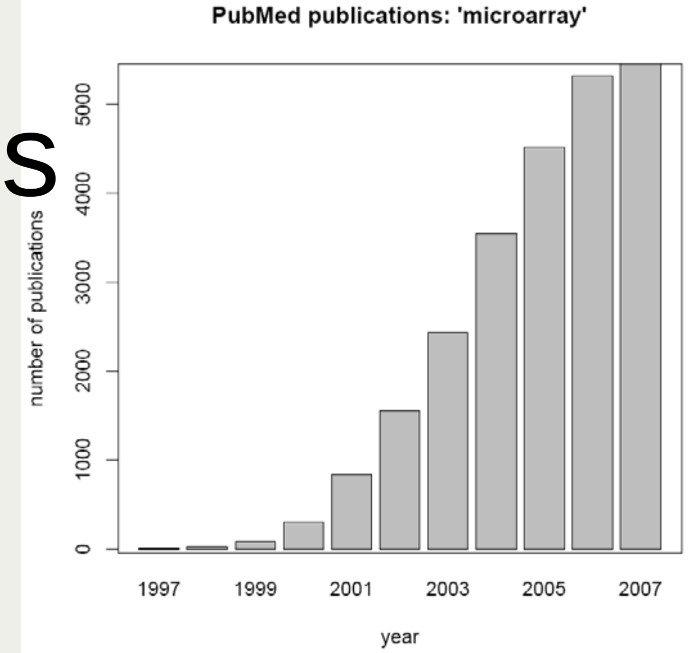
Chip: hgu133

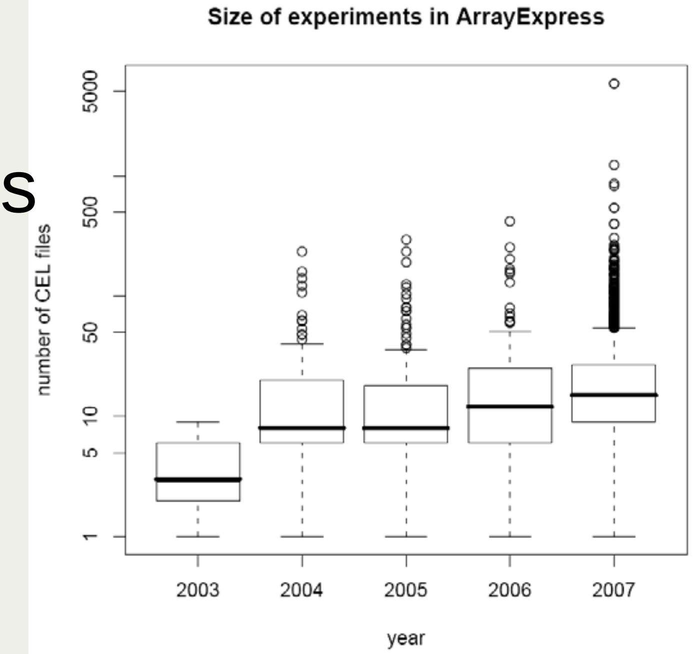
Challenges

- Microarray experiments more and more popular

- Microarray chips become cheaper
  - Experiments grow in size
  - EBI experiment: 6000 Arrays

- Microarray chips grow in size
  - More genes per chip





Possible Solutions

- Business applications
  - Expensive, not adaptable
- Faster and bigger computers
  - Expensive, limited
  - Main memory 256 GB: 60t €
- Better coding
  - C, DB
- Distribution to several computers / processors
  - Concurrent calculation of parts at different processors
  - Main memory 8 GB: 2000 € -> 60t € = 30 computers

Parallelization

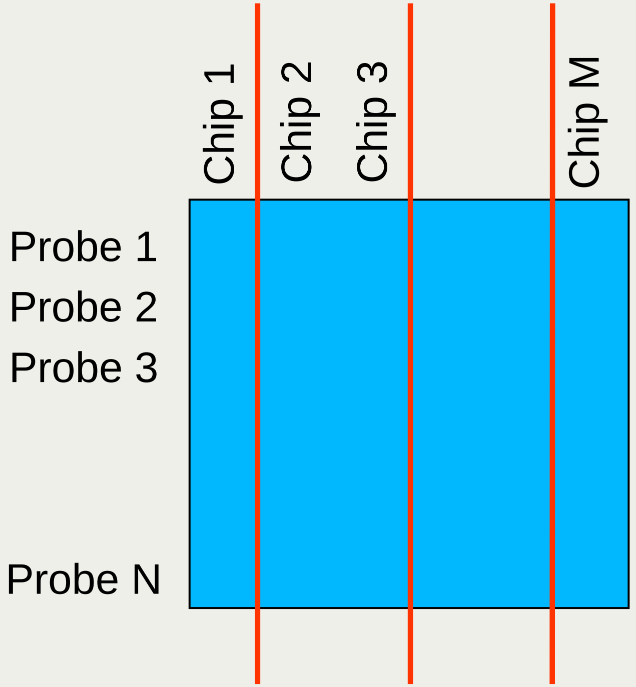
- Multiprocessors
  - the use of two or more central processing units (CPUs) within a single computer system
  - Today: Two-processors get a standard for workstations
- Multicomputers = Cluster
  - different parts of a program run simultaneously on two or more computers that are communicating with each other over a network
  - Computer, network, software

Software: MPI

- **M**essage **P**assing **I**nterface
- MPI is an API for parallel programming based on the message passing model
- MPI processes execute in parallel
- MPI is a standard for libraries
- Libraries exists for
  - FORTRAN, C, C++
  - R: Rmpi, Snow, papply
- IBE Cluster: LAM/MPI 7.1.3

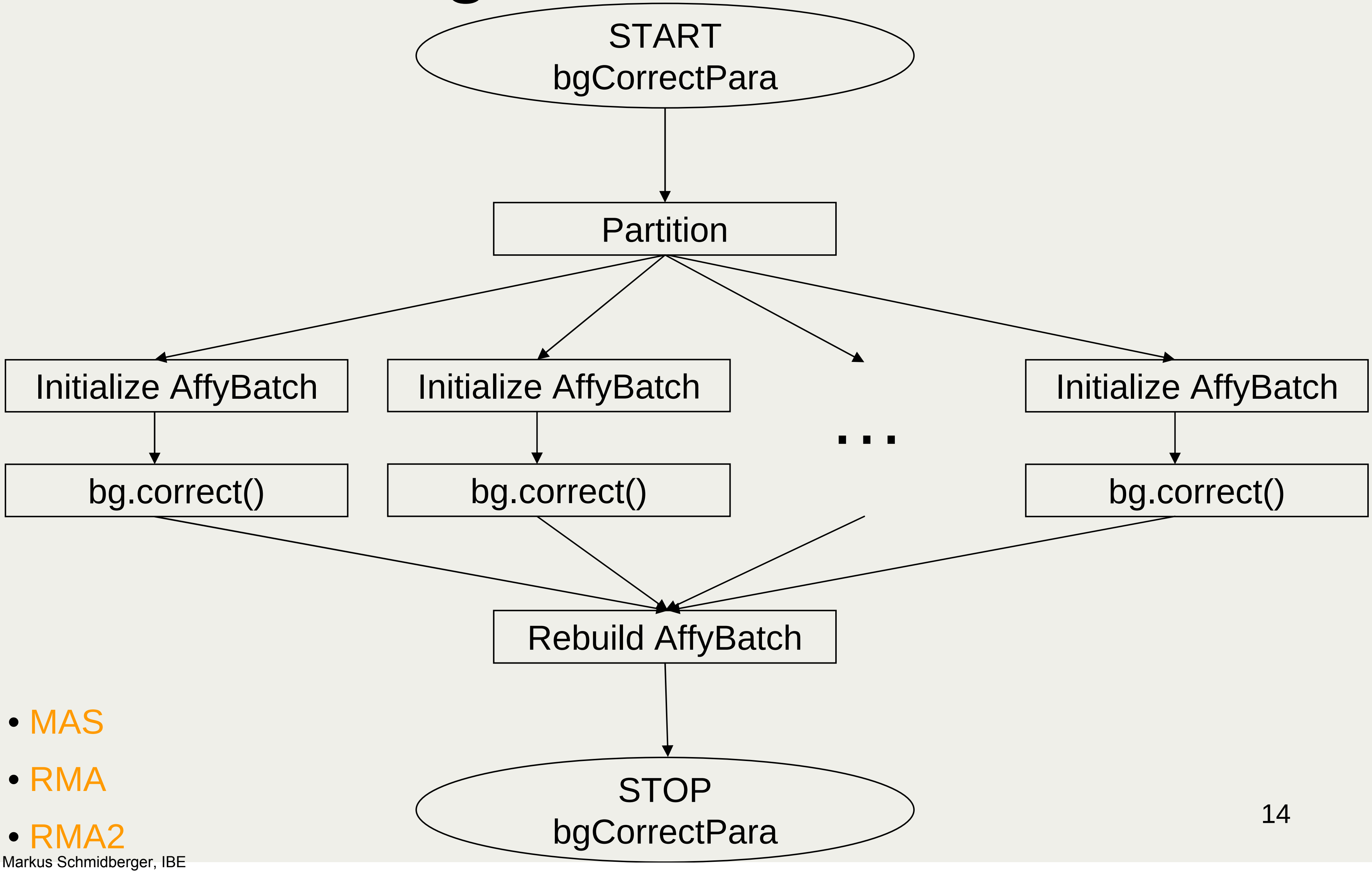
Decomposition of AffyBatch

- AffyBatch = intensities from multiple arrays



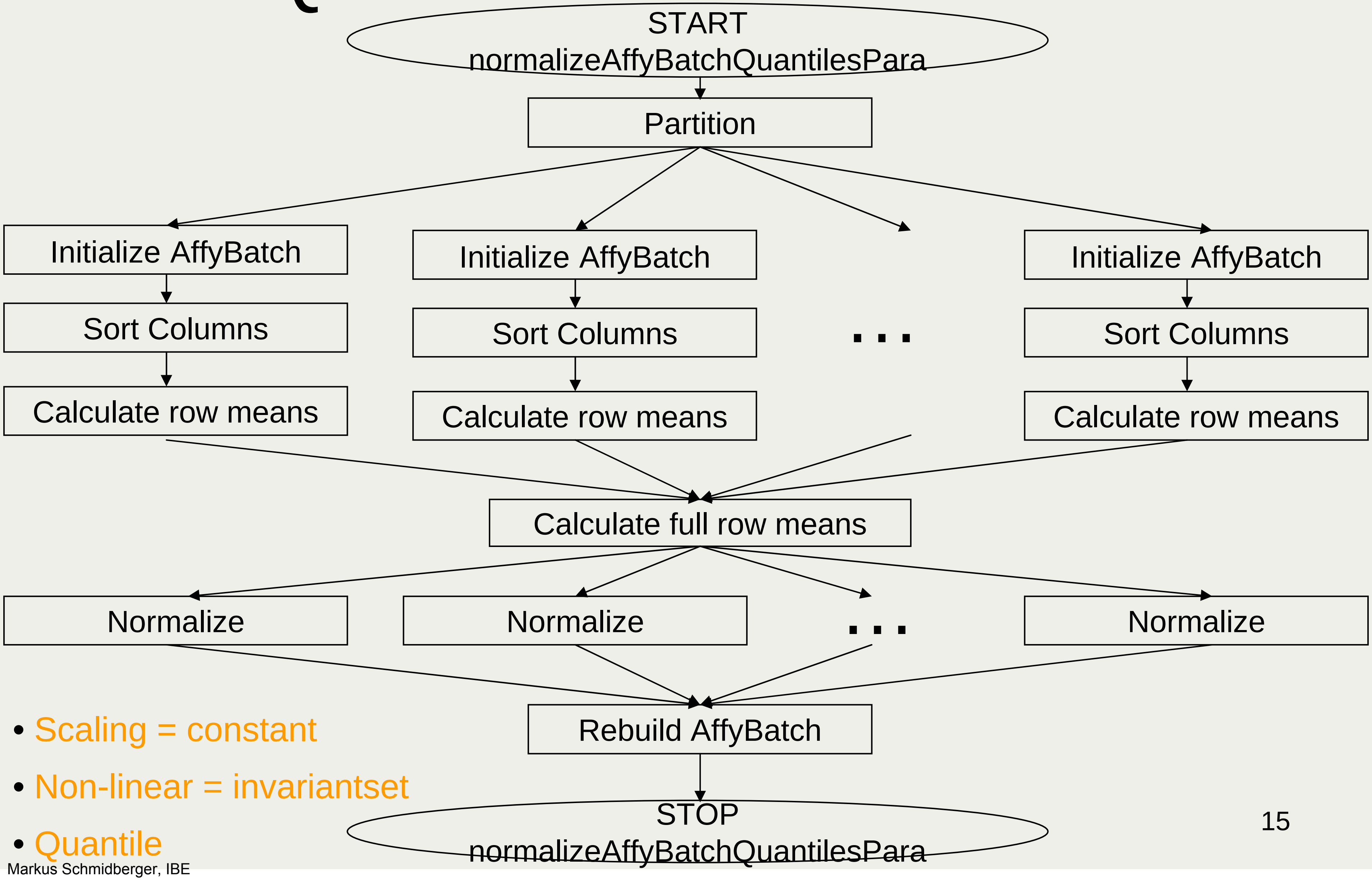
- Which decomposition is the best ?
  - Partition by chips
  - Partition by probes
  - Partition of CEL file name list

Background Correction



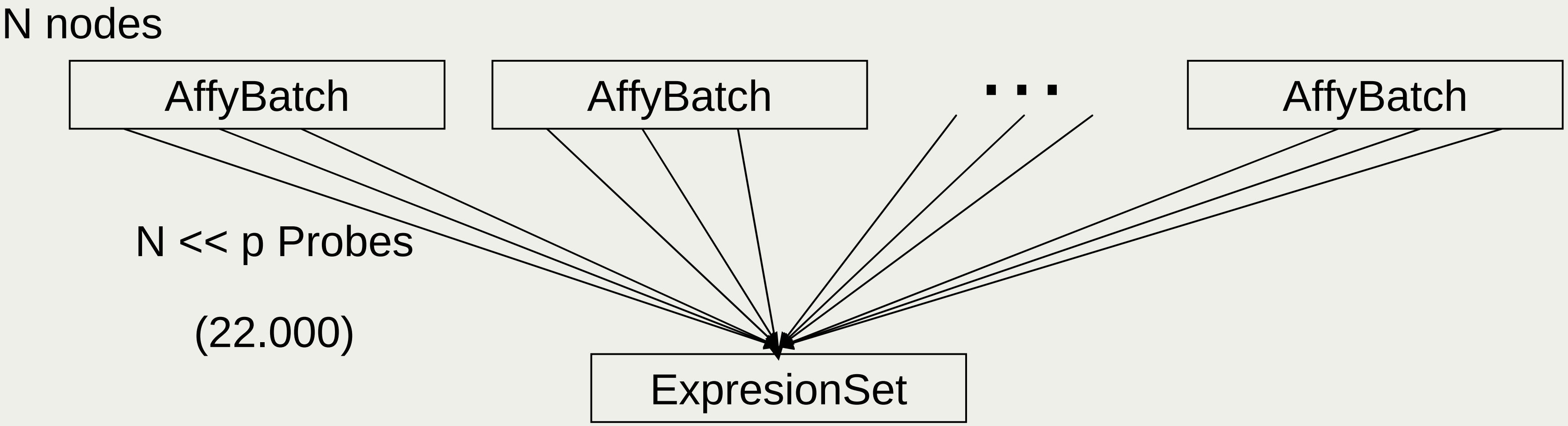
- MAS
- RMA
- RMA2

Quantile Normalization



- Scaling = constant
- Non-linear = invariantset
- Quantile

Summarization



- avgdiff
- liwong
- mas
- medianpolish

Communication Overhead

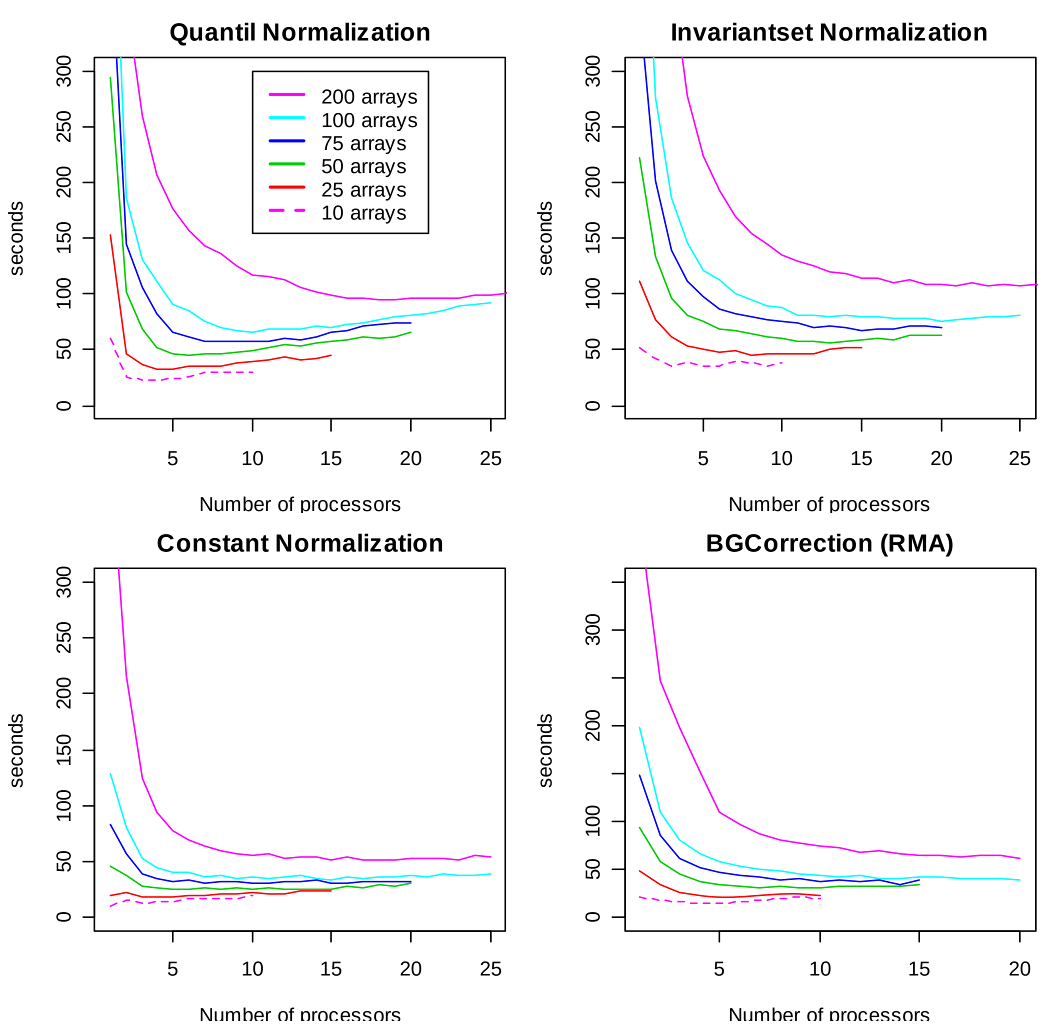
- AffyBatch memory intensive
  - A lot of data to transfer
  - Create AffyBatches at nodes
    - CEL Files available over samba device
  - Complete preprocessing method: preproPara()
    - Reducing the exchange of data
    - At no point a complete AffyBatch required
- R functions and environments
  - Define global functions

Results

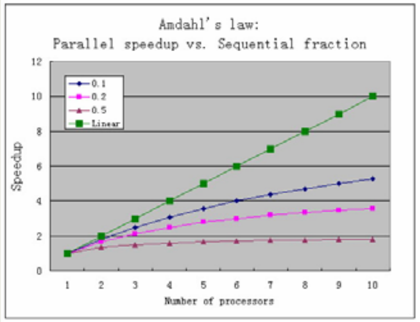
- Package **affyPara** with parallelized affy-functions
  - More CEL Files preprocessable
  - Speedup

- Parallelization methods produce in view of machine accuracy the same results as serialized methods.
  - All.equal(), machine's precision.

Results - Computation Time



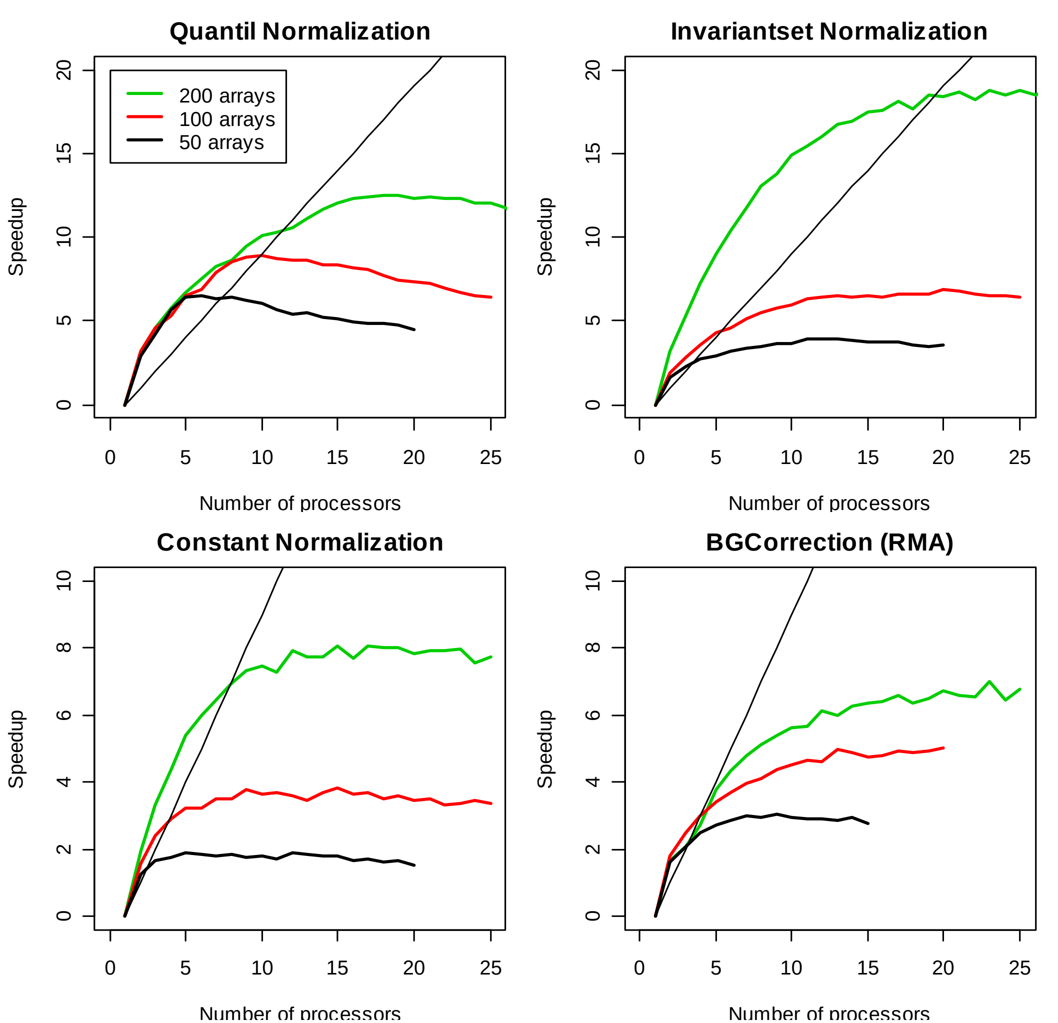
Results – Speedup



Sp = T_1 / T_p

Sp ~ 1 / [s +p/N]

- Speedup of the methods up to factor 15



Results - Conclusion

- Partition of data and distribution to several nodes solves the main memory problems
  - IBE Cluster: ~ 16.000 microarrays

    (32 workstations, 8 GB main memory, 2 dual core, Intel Xeon DP 5150, 1 Gbit Network, Linux 2.6.18 openSuse, LAM/MPI 7.1.3)
  - Expansion of the cluster
    -> more data processible

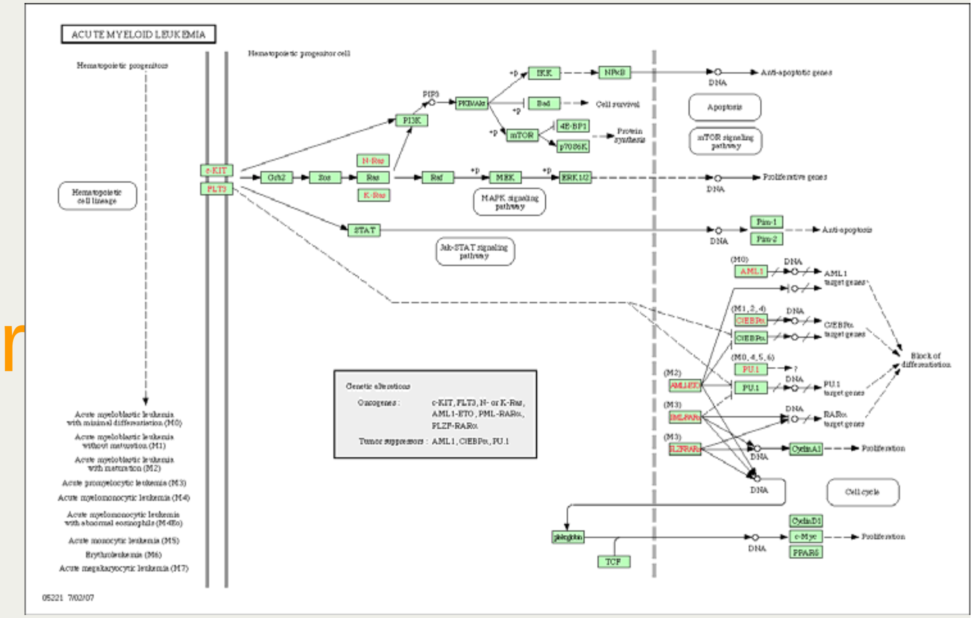
- affyPara package available in the next BioConductor release in April 2008

Large project in applied bioinformatics in preparation

- Collecting cancer data from public libraries
  – ArrayExpress, GEO, …
  -> more than 5000 microarrays

- Preprocessing all together

- Analyzes all together
  – pathways ?


ACUTE MYELOID LEUKEMIA
Hematopoietic progenitors
Hematopoietic progenitor cell
Hematopoietic cell lineage
c-KIT
FLT3
Grb2
Sos
N-Ras
Ras
K-Ras
PI3K
PIP3
PKB/Akt
IKK
NFκB
Bad
Cell survival
mTOR
4E-BP1
p70S6K
Protein synthesis
Raf
MEK
ERK1/2
+p
MAPK signaling pathway
STAT
Jak-STAT signaling pathway
DNA
Anti-apoptotic genes
Apoptosis
mTOR signaling pathway
Proliferative genes
Pim-1
Pim-2
Anti-apoptosis
(M0)
AML1
AML1 target genes
(M1,2,4)
C/EBPα
C/EBPα target genes
(M0,4,5,6)
PU.1
?
PU.1 target genes
Block of differentiation
(M2)
AML1-ETO
(M3)
PML-RARα
RARα target genes
PLZF-RARα
CyclinA1
Proliferation
Cell cycle
plakoglobin
TCF
CyclinD1
c-Myc
PPARδ
Genetic alterations
Oncogenes : c-KIT, FLT3, N- or K-Ras, AML1-ETO, PML-RARα, PLZF-RARα
Tumor suppressors : AML1, C/EBPα, PU.1
Acute myeloblastic leukemia with minimal differentiation (M0)
Acute myeloblastic leukemia without maturation (M1)
Acute myeloblastic leukemia with maturation (M2)
Acute promyelocytic leukemia (M3)
Acute myelomonocytic leukemia (M4)
Acute myelomonocytic leukemia with abnormal eosinophils (M4Eo)
Acute monocytic leukemia (M5)
Erythroleukemia (M6)
Acute megakaryocytic leukemia (M7)
05221 7/02/07

Parallelized preprocessing algorithms for high-density oligonucleotide array data

Thanks for your attention

Dipl.-Tech. Math. Markus Schmidberger
schmidb@ibe.med.uni-muenchen.de

HiCOMB 2008
Seventh IEEE International Workshop on High Performance Computational Biology
April 14, 2008; Miami, Florida, USA

IBE
http://ibe.web.med.uni-muenchen.de

[redacted]

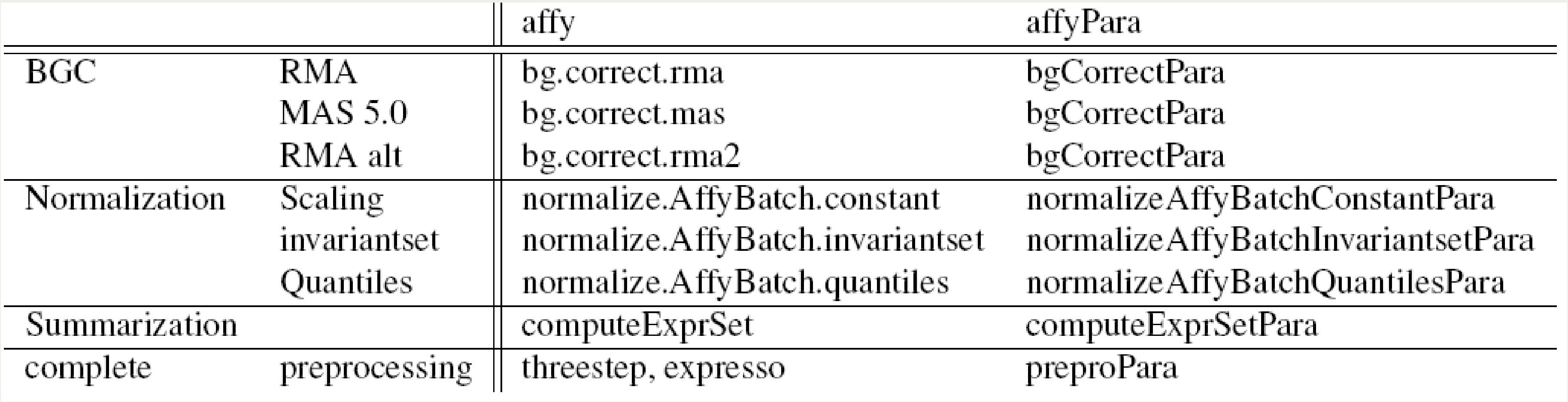
Literature R and Preprocessing

- Gentleman et. All; Bioinformatics and Computational Biology Solutions, *Using R and Bioconductor;* Springer, 2005 (Statistics for Biology and Health)
- Berrar et. All.; *A Practical Approach to Microarray Data Analysis;* Kluwer Academic Publishers, 2004
- Bolstad, Irizarry, Astrand, Speed; *A comparison of normalization methods for high density oligonucleotide array data based on variance and bias; Bioinformatics,* 2003*, 19(2)*, 185-193
- Irizarry, Wu, Jaffee; *Comparison of Affymetrix GeneChip expression measures;* Bioinformatics*,* 2006*, 22 no. 7*, 789-794

Literature Parallel Computing

- Sloan, J. D.; *High Performance Linux Clusters with OSCAR, Rocks, OpenMosix, and MPI O'Reilly*, 2004
- Tierney, Luke: *Code Analysis and Parallelizing Vector Operations in R*. In: DSC 2007. Auckland, New Zealand, February 2007
- Rossini, Anthony: *Simple Parallel Statistical Computing in R*. In: UW Biostatistics Working Paper Series 193 (2003)
- Sevcikova, Hana: *Statistical Simulations on Parallel Computers*. In: Journal of Computational and Graphical Statistics 13 (2003), Nr. 4, 886-906.

Package affyPara



		affy	affyPara
BGC	RMA	bg.correct.rma	bgCorrectPara
	MAS 5.0	bg.correct.mas	bgCorrectPara
	RMA alt	bg.correct.rma2	bgCorrectPara
Normalization	Scaling	normalize.AffyBatch.constant	normalizeAffyBatchConstantPara
	invariantset	normalize.AffyBatch.invariantset	normalizeAffyBatchInvariantsetPara
	Quantiles	normalize.AffyBatch.quantiles	normalizeAffyBatchQuantilesPara
Summarization		computeExprSet	computeExprSetPara
complete	preprocessing	threestep, expresso	preproPara