Cluster Analysis of Heterogeneous Gene Expression Datasets

Pratyaksha Wirapati, Hamid Sadouki, Mauro Delorenzi

Bioinformatics Core Facility
http://bcf.isb-sib.ch

Swiss Institute of Bioinformatics
http://www.isb-sib.ch

Statistical Advances in Genome-Scale Data Analysis
3-8 May 2009, Monte Verità, Ascona, Switzerland

The problem

Agglomerative hierachical clustering is a versatile workhorse method for exploratory analysis of multivariate data.

Although clustering is routinely used for single-study analysis of gene expression microarray data, the extension to co-analysis of multiple datasets with different platforms and study designs is not yet clear.

Statistical challenges: How to summarize co-expression across studies (non i.i.d) with possible study-specific biases? How to quantify consistency and systematic heterogeneity? How to deal with genes missing from some platforms?

Computational challenges: How to compute the clustering tree efficiently (time- and space-wise)? How to design and implement the appropriate data structure and algorithm?

Statistical Methods

- Consider multiple studies to be hierarchically stratified

  e.g., diseases (cancer types) $\longrightarrow$ studies $\longrightarrow$ arrays

- For each strata (disease i, dataset j), calculate all pairwise Pearson's correlations

- For each pair of genes (a, b), use Fisher's inverse hyperbolic tangent transform to make the Pearson's correlation normal with constant variance:

$$z_{ij,ab} = \tanh^{-1} r_{ij,ab} \qquad \text{Var}(z_{ij,ab}) = 1/(n_{ij} - 3)$$

- Combine the z-transformed correlations into a single matrix,

- Use multi-stage random-effects meta-analysis to combine z's

- Construct one hierarchical clustering tree, using top-level summaries of correlations as the similarity measures

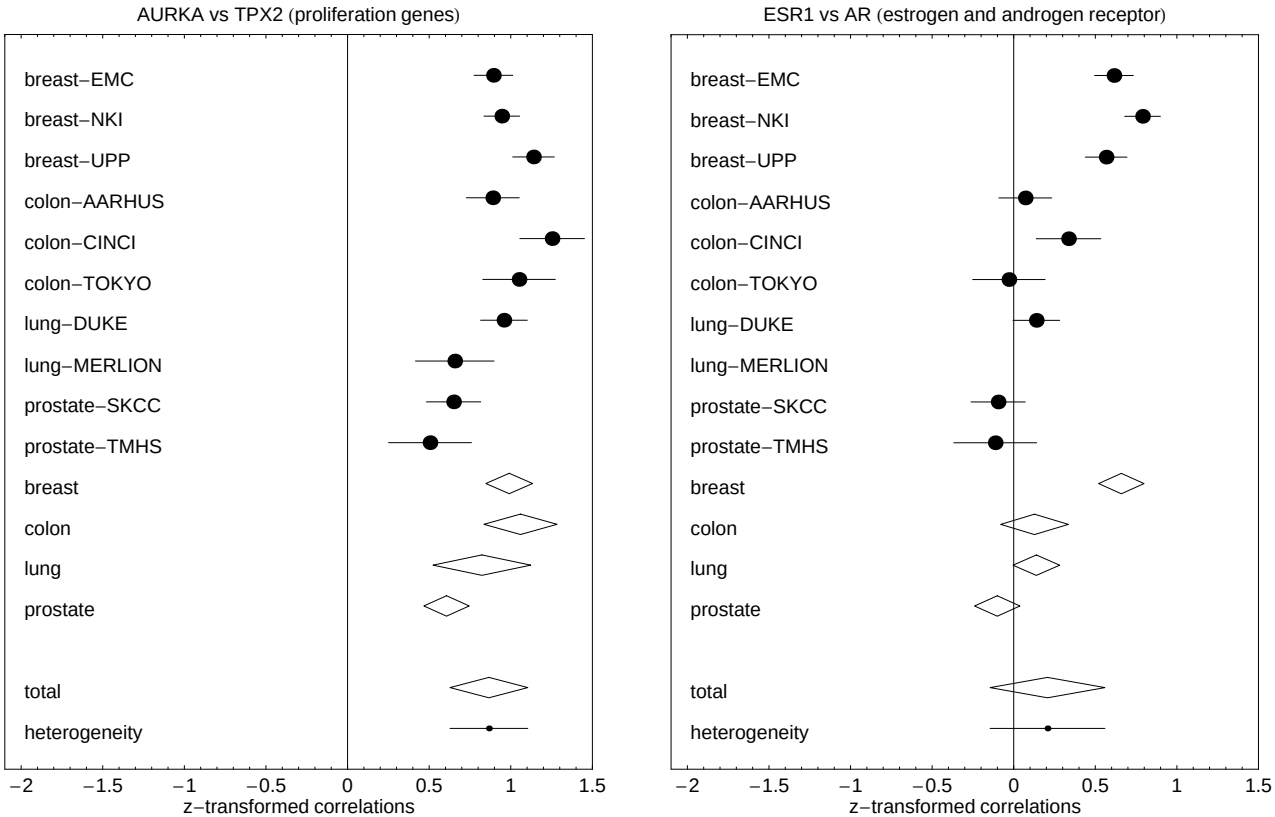
Bioinformatics and Computational Methods

- Preprocessing: not critical (can be any method)
- Gene mapping across platforms: align probe sequences to the same version of RefSeq (high quality NM series), choose a unique probe per gene by maximum variance
- Use *the union* of all genes instead of the intersection. If a gene is missing from a platform, correlations involving this gene is considered a missing data at the disease-level analysis
- Hierarchical clustering of tens of thousands of variables: reciprocal nearest-neighbor algorithm, using average linkage.
- Tree branch reordering: "nearest-nephew" rule, applied recursively (computation is $O(n)$ time complexity).

Example (breast, prostate, colon and lung cancer)

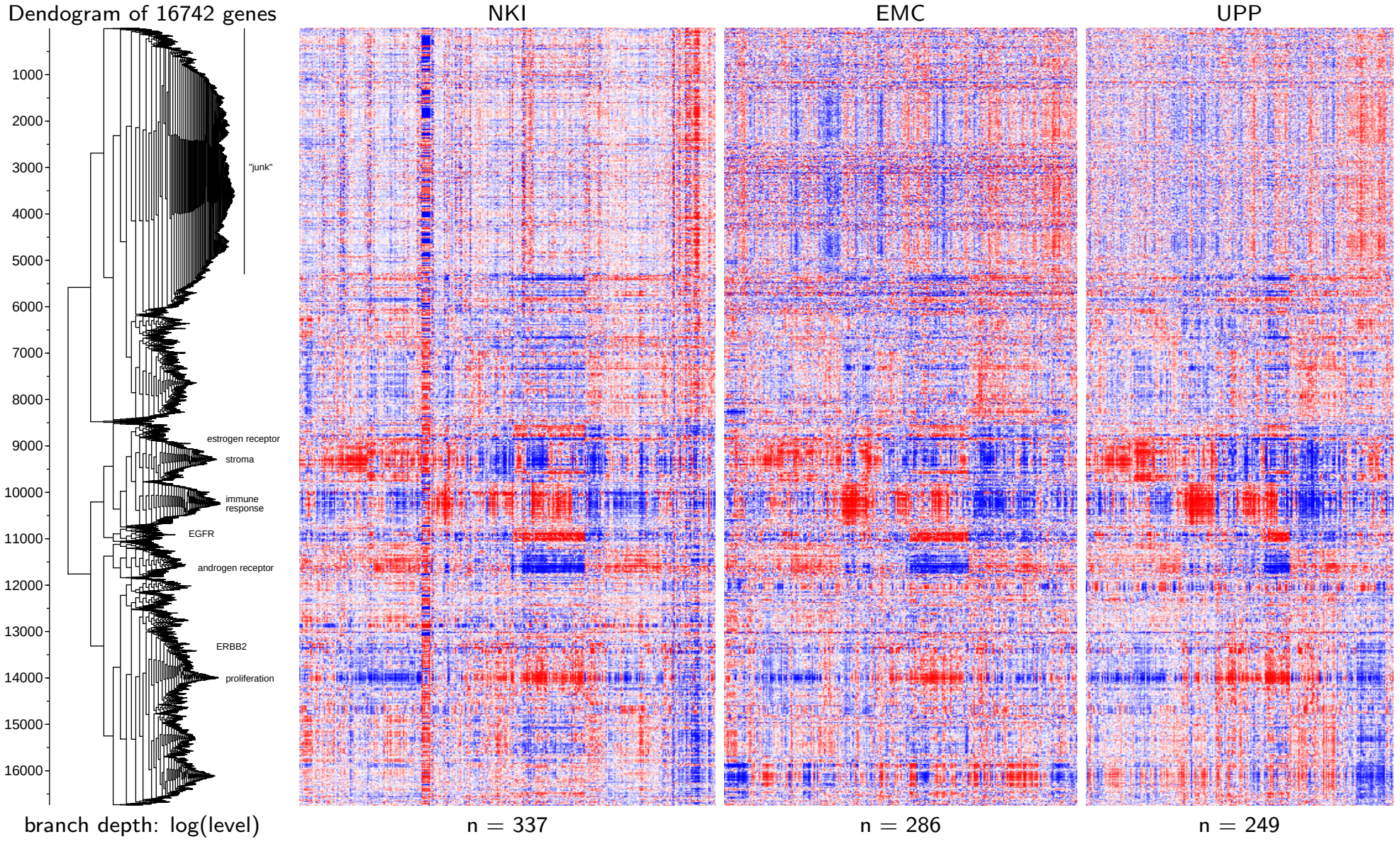
Multi-stage random-effects meta-analysis can be use to both combine the correlations (by "automatic weighting" using the within- and between-strata variances) and assess *differential co-expression* using the between-strate variance.

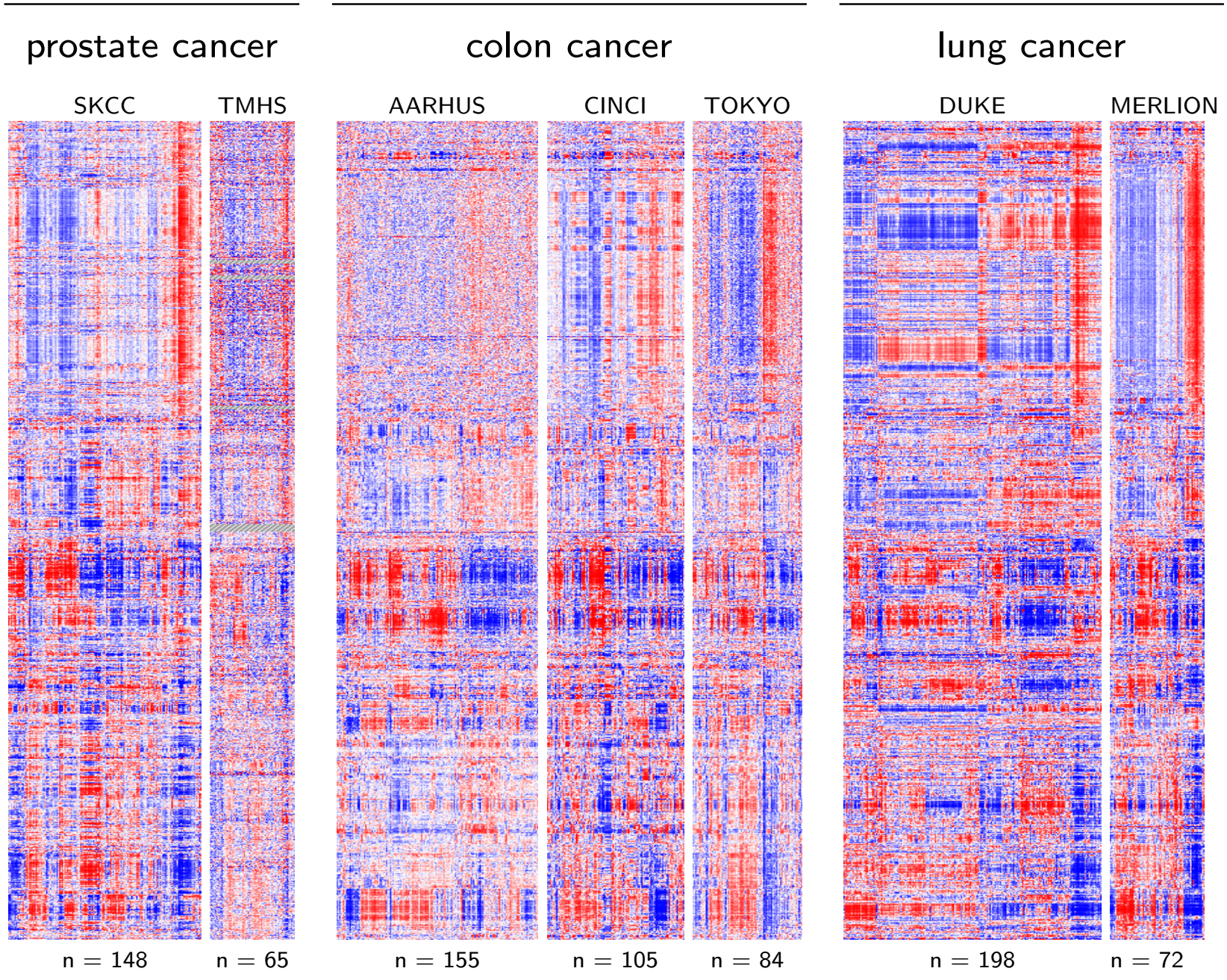
Examples of consistently correlated pairs (left) and breast-cancer-only pairs (right)



*between-strata variance estimator: DerSimonian-Laird moment estimator (the choice doesn't seem to be critial; similar results to REML and empirical Bayes, which require iterative calculation)

breast cancer




prostate cancer
colon cancer
lung cancer
SKCC
TMHS
AARHUS
CINCI
TOKYO
DUKE
MERLION
n = 148
n = 65
n = 155
n = 105
n = 84
n = 198
n = 72

Some observations

- In whole-genome clustering of expression datasets, typically a third of genes are "junk" (not variably expressed). See the top one third of the heatmaps. Patterns in this area can be used to identify measurement artefacts.

  Note: The DUKE dataset contains substantial batch effects (not yet corrected, stratification may be needed in re-analysis)

- Outlier arrays can be easily seen

- Some modules are highly conserved (e.g. proliferation, stroma, immune response). Some are disease-specific (e.g., the one containing androgen receptor).

- Large estrogen receptor cluster in breast cancer can be split into submodules when co-analyzed with other diseases

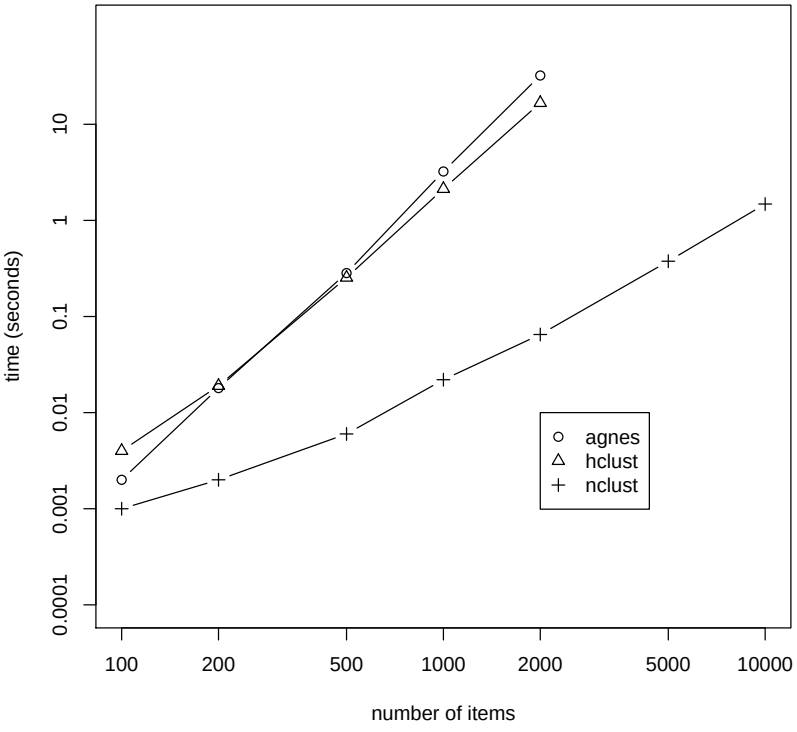
Potential Applications

- **Quality control**: identify bad arrays, bad preprocessing, batch effects, etc. Advantage: we're looking at artefacts affecting the joint distribution (not just the marginal, per-array distribution)
- **Basic research**: refinement of co-expression module inference

  A large module of highly correlated genes based on single disease analysis can be split into multiple modules according to multi-disease analysis.
- **Biomarker development**: reusable "cassettes" of signatures (based on correlated expression) that can be mixed and matched according to their relevance in a particular disease.

Computational issues

The reciprocal neareast-neighbor algorithm produces exactly the same tree (if the answer is unique) as the classical brute force algorithm, but has $O(n^2)$ time complexity (instead of $O(n^3)$), and does not require storing all pairs of similarities/distances (they can be computed on the fly).
Comparison of `nclust` (this algorithm) with R's `hclust` and `agnes` is shown on the right (on small simulated data), with expected results



On the example analysis: 1699 arrays $\times$ 16742 genes computed on a laptop with 2.4GHz Intel CoreDuo CPU (using single core):

Clustering of genes: $\sim$10.5 minutes (289Mb peak memory usage)

Clustering of arrays: $\sim$2.5 minutes (147Mb peak memory usage)

Ongoing works

- Packaging for R (currently it's based on Unix-command line)
- Multicore implementation (parallelization is straightforward)
- Automatic selection of differential coexpression (or the lack of it) by propagating the heterogeneity measures up the clustering tree
- Extending the dataset collection (broadly and deeply), including correction of artefacts (outliers, batch effects) found in the first round of analysis