Computational Systems Biology
Deep Learning in the Life Sciences

6.802 6.874 20.390 20.490 HST.506

David Gifford
Lecture 8

March 5, 2018

Single-cell RNA seqencing

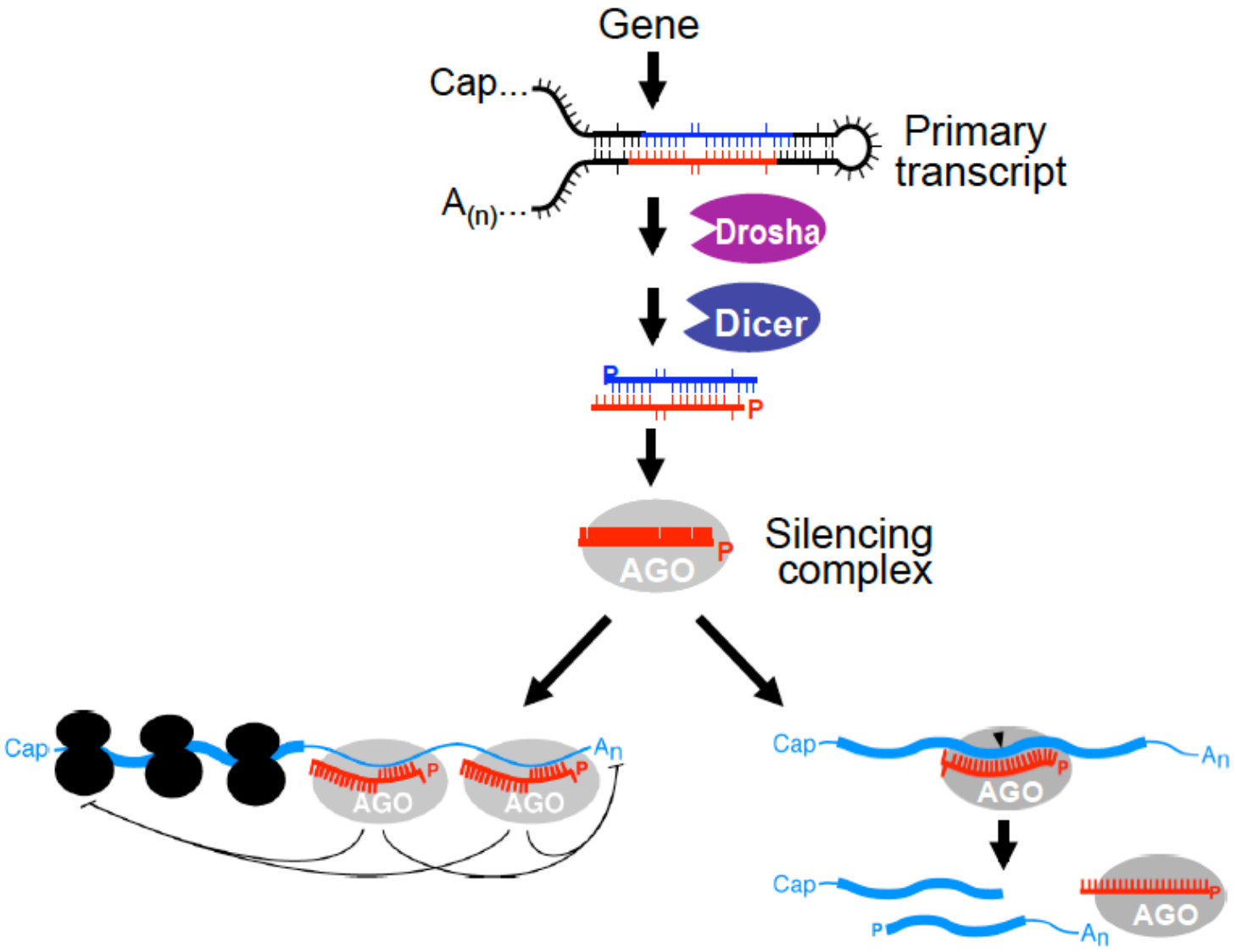
http://mit6874.github.io

What's on tap today!

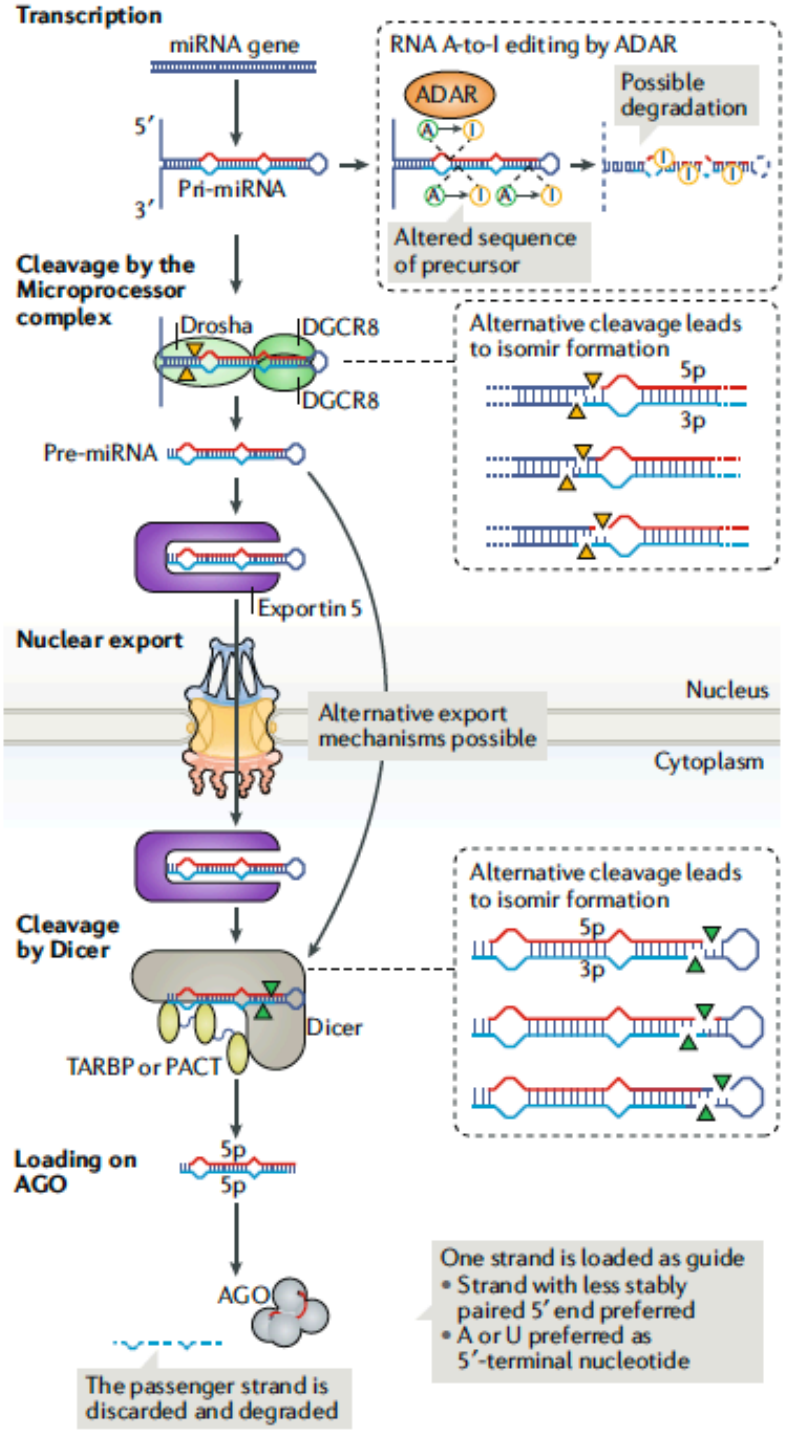
- miRNAs regulate translation
- Counting molecules instead of reads (UMIs)
- Single-cell protocol details
  - 3' based methods
  - Whole transcript method
- Pseudo time presentation of cells in expression state
- Modeling RNA expression with regression trees

miRNAs regulate mRNA translation

miRNAs regulate gene translation

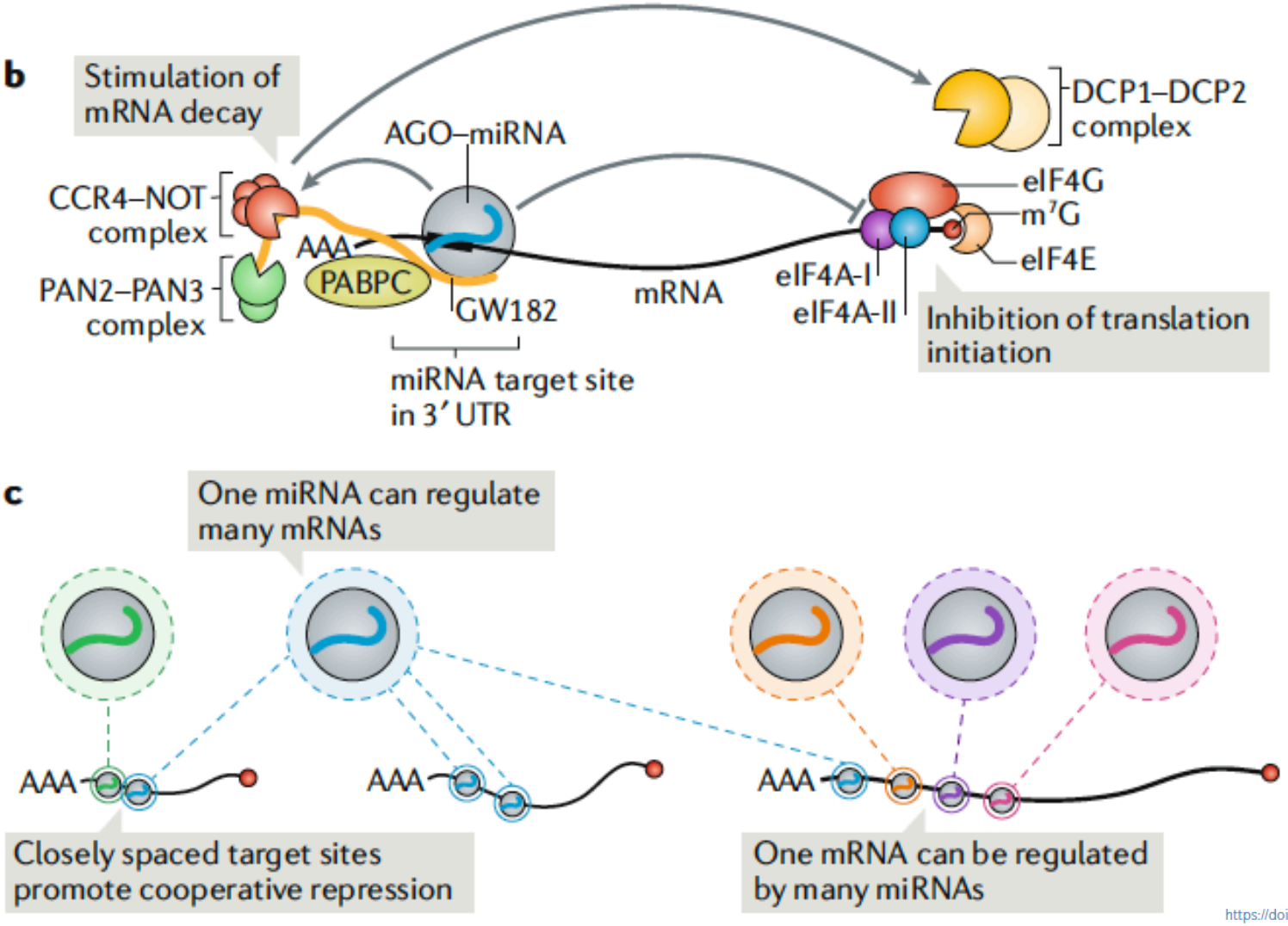


miRNAs are expressed and processed



https://doi.org/10.1038/s41580-018-0045-7

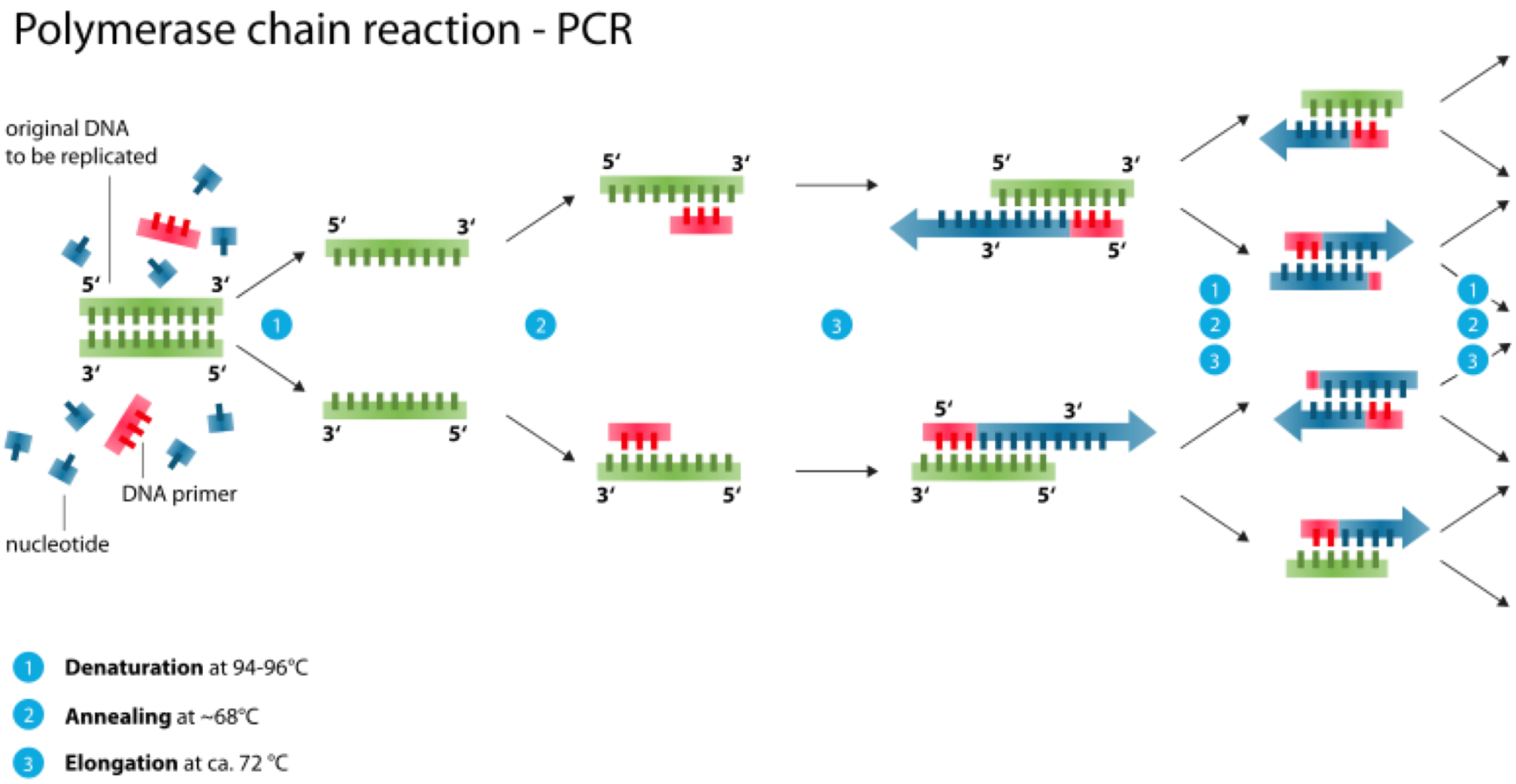
A miRNA can regulate one or more genes



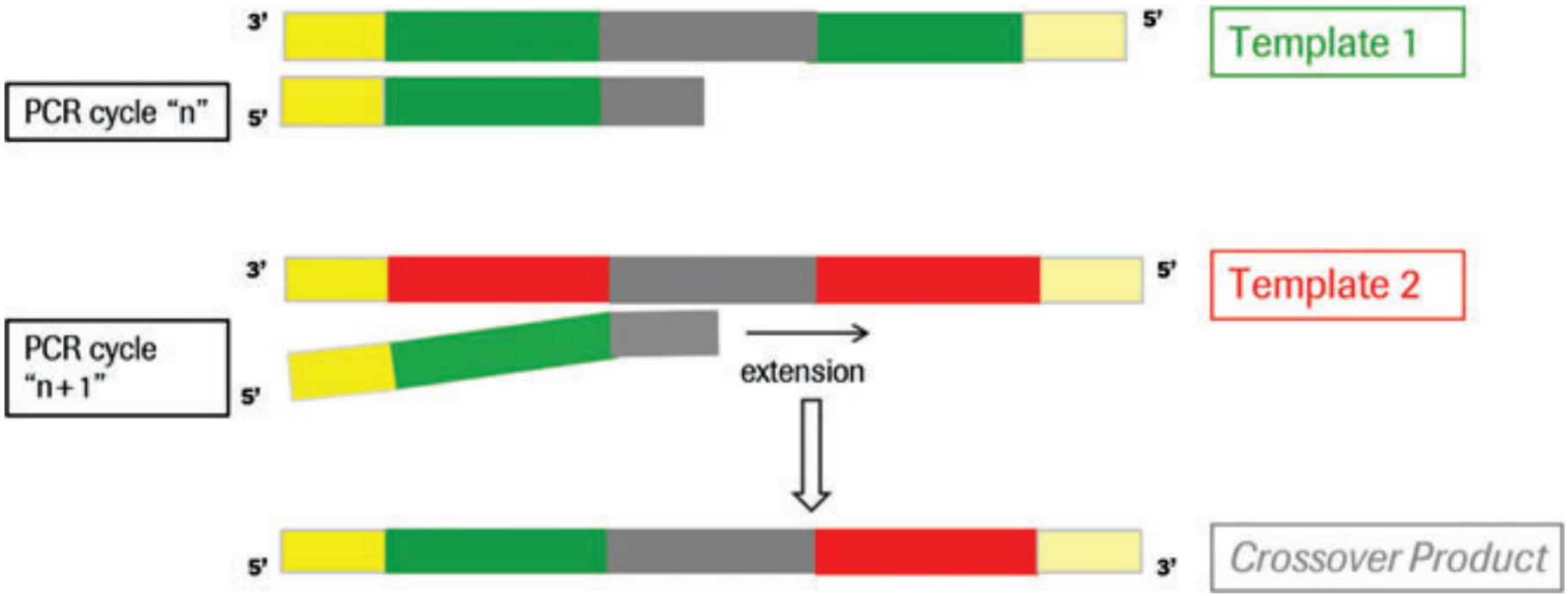
https://doi.org/10.1038/s41580-018-0045-7

How to count molecules and not reads

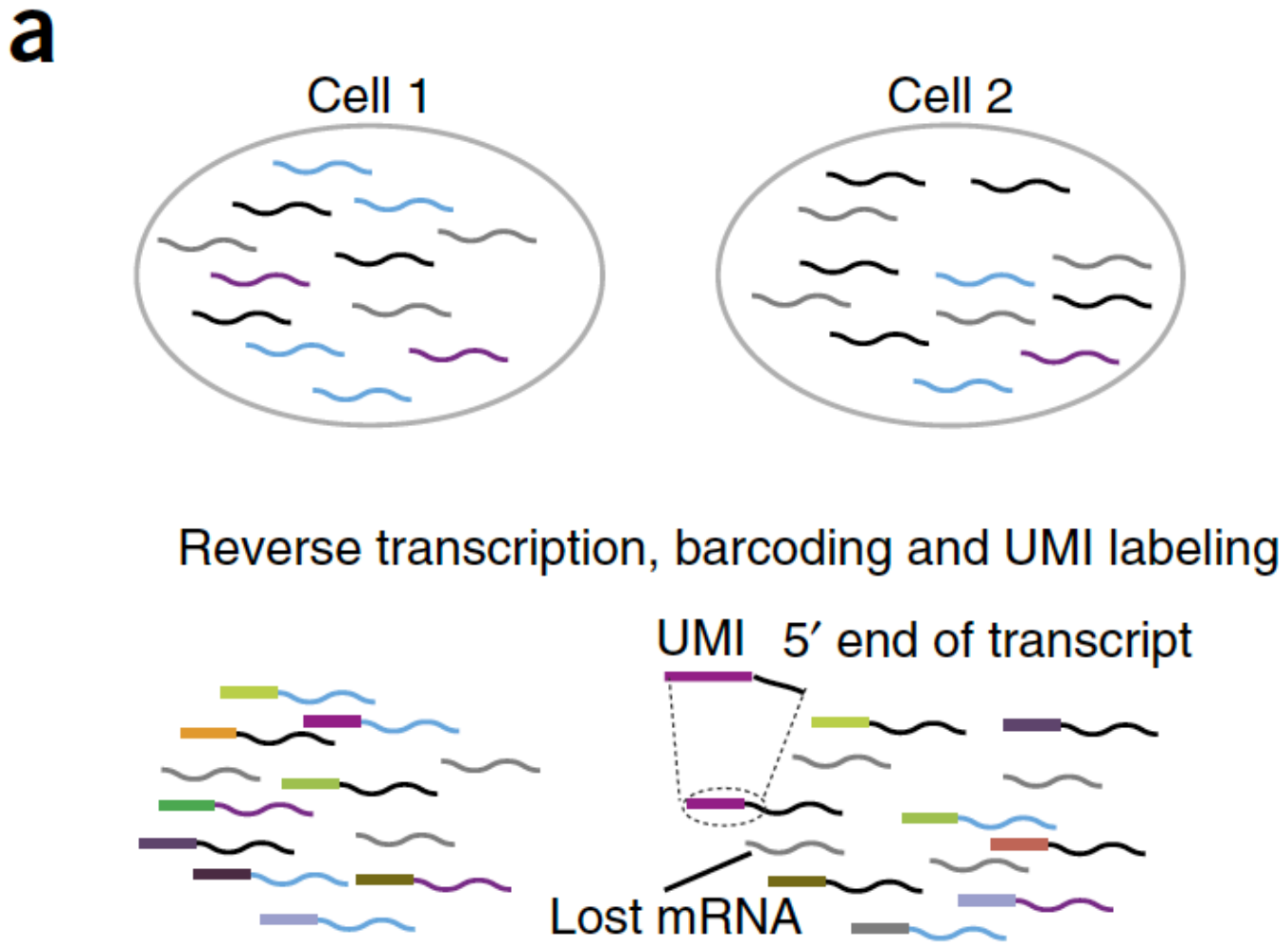
PCR does not uniformly amplify all molecules because of hybridization differences



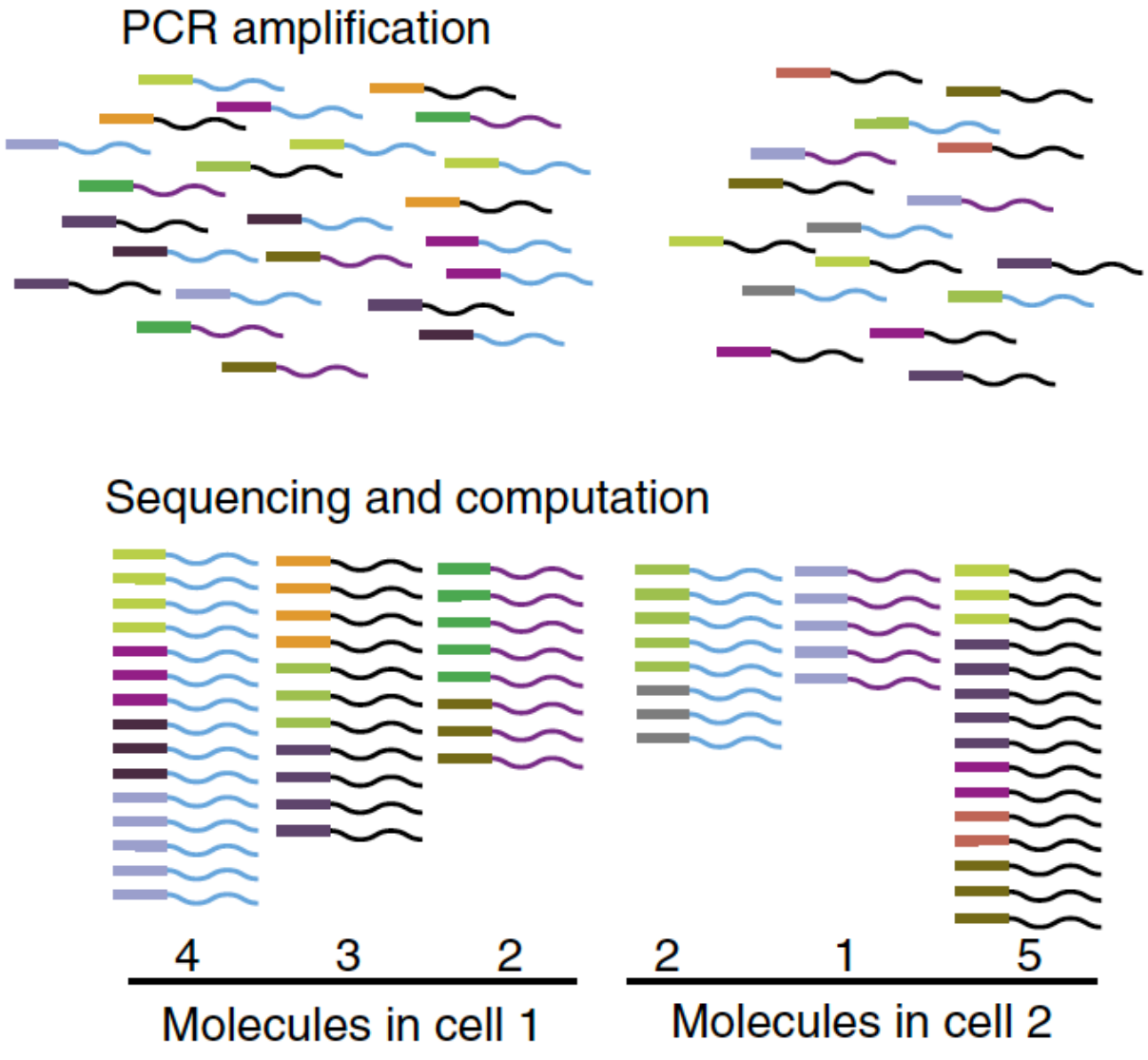
PCR chimeras are molecules that are not in the starting population



Unique molecular identifiers (UMIs) label molecules



Reads with the same sequence and UMI are counted once



TPM and FPKM metrics

RPKM or FPKM (Fragments Per Kilobase Million)

$$\mathrm{RPKM}_g = \frac{\frac{r_g 10^3}{\mathrm{fl}_g}}{\frac{R}{10^6}} = \frac{r_g \times 10^9}{\mathrm{fl}_g \times R}$$

TPM (Transcripts Per kilobase Million)

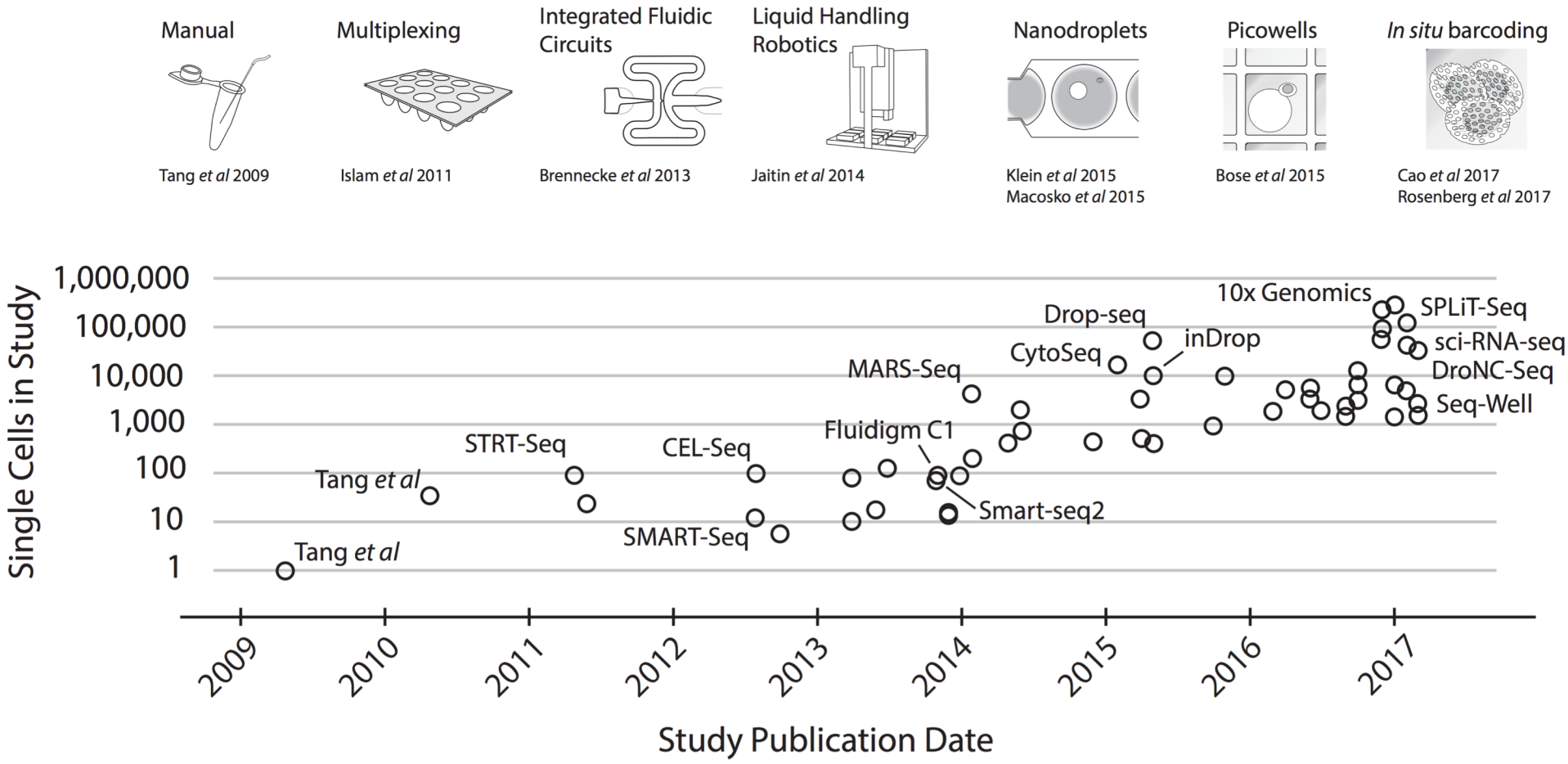
$$\mathrm{TPM} = \frac{r_g \times \mathrm{rl} \times 10^6}{\mathrm{fl}_g \times T}$$

$$T = \sum_{g \in G} \frac{r_g \times \mathrm{rl}}{\mathrm{fl}_g}$$

Measurement of mRNA abundance using RNA-seq data: RPKM measure is inconsistent among samples

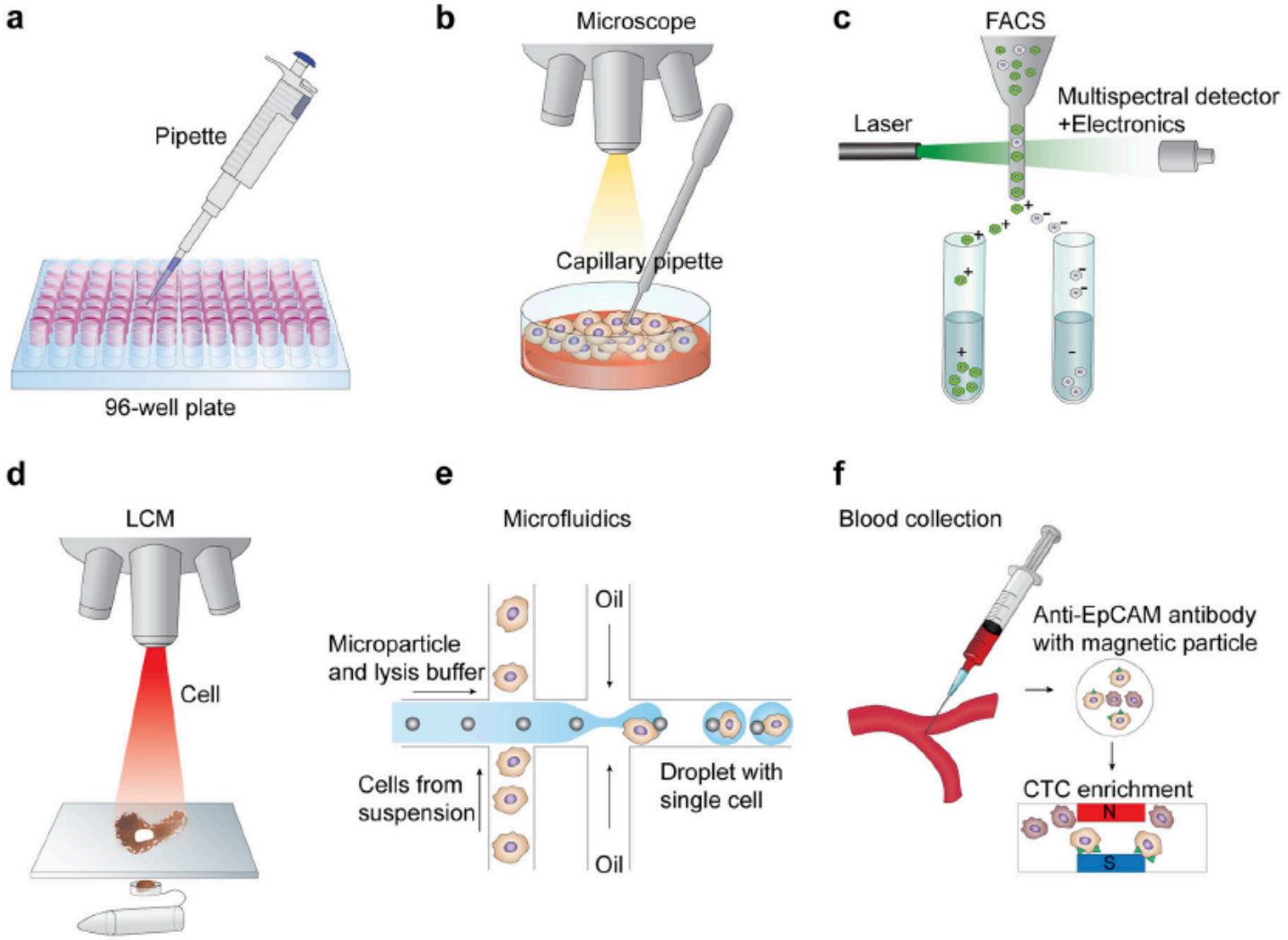
Günter P. Wagner · Koryu Kin · Vincent J. Lynch

Single-cell RNA sequencing

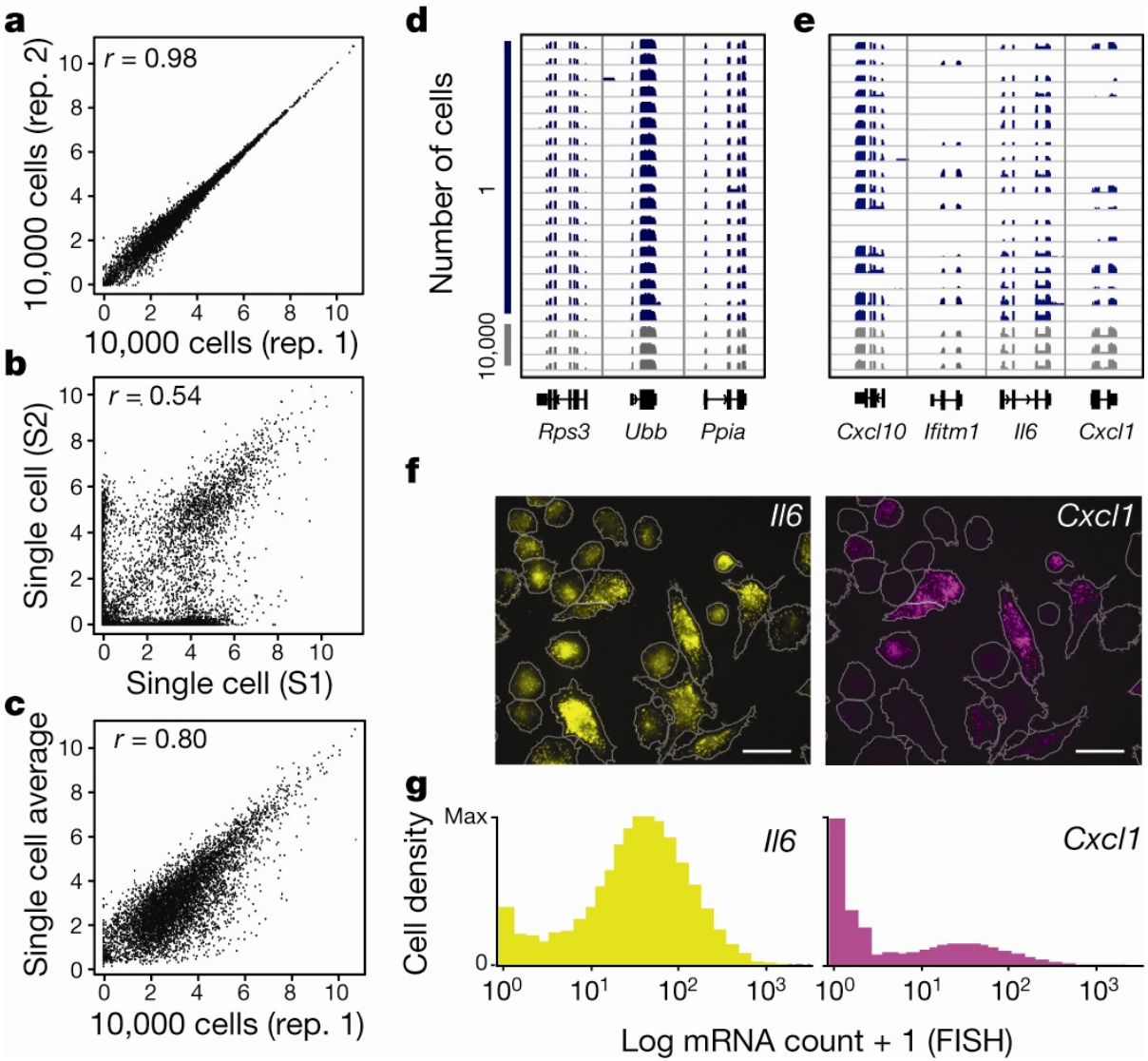


Svensson et al 2017

Different approaches to isolating single-cells for sequencing their RNA

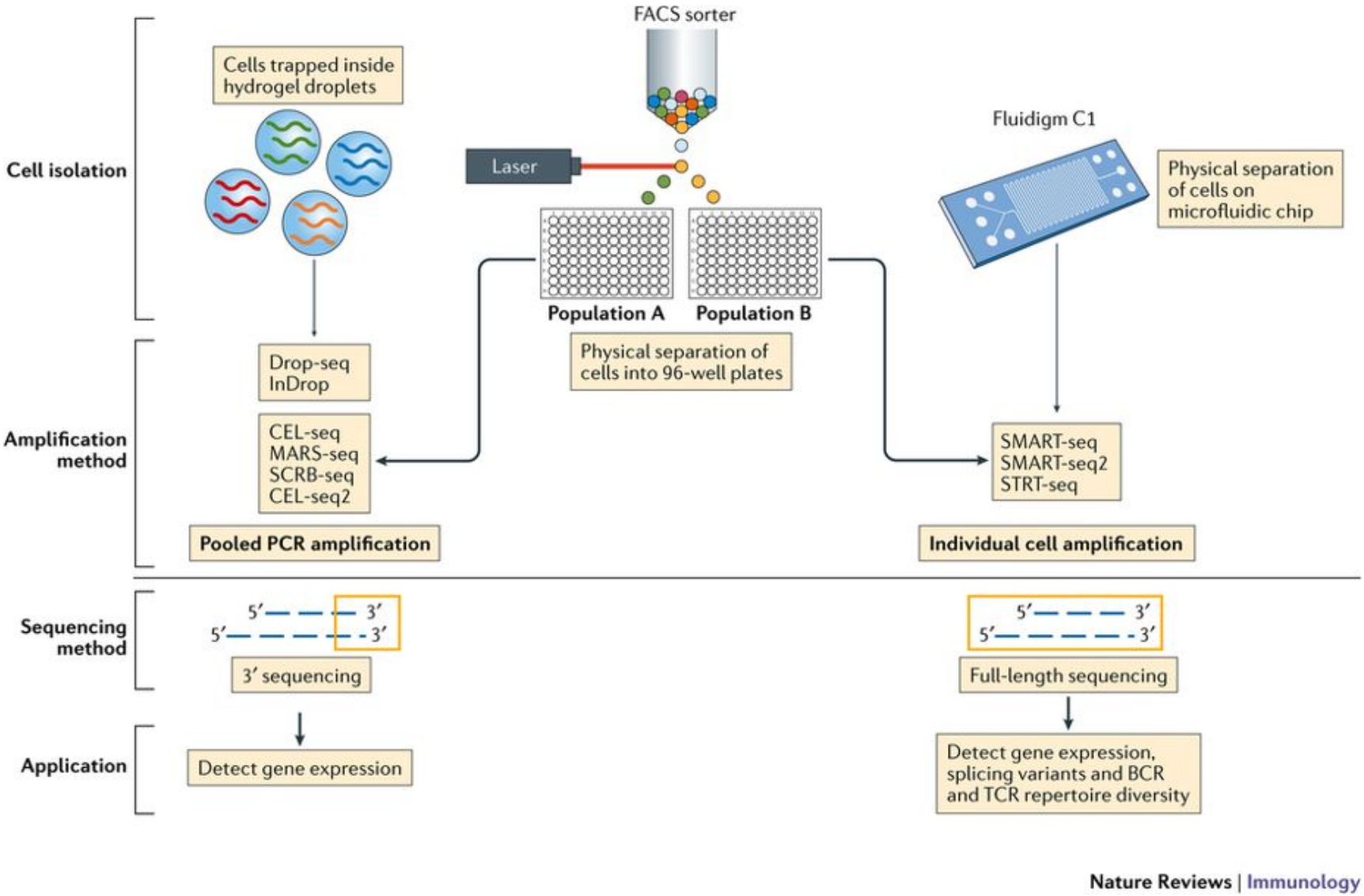


Single-cell RNA-Seq of LPS-stimulated bone-marrow-derived dendritic cells reveals extensive transcriptome heterogeneity.



AK Shalek *et al. Nature* **000**, 1-5 (2012) doi:10.1038/nature12172

Single-cell RNA-sequencing technologies mainly differ in cell isolation and transcript amplification



Paplexi et al. 2017

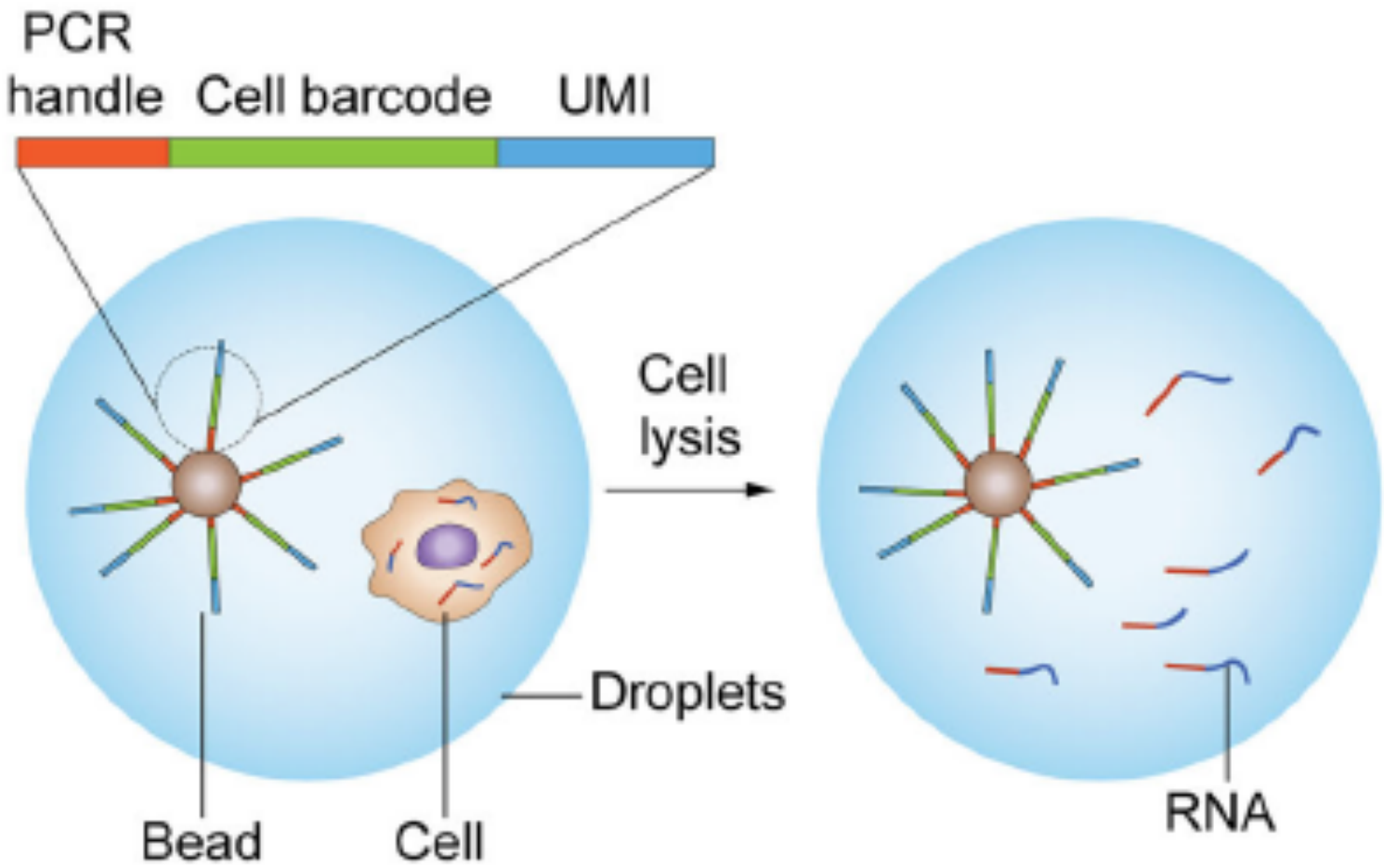
Single-cell RNA-sequencing technologies also vary in cost and sensitivity

	FACS	CyTOF	qPCR	Plate-based protocols (STRT-seq, SMART-seq, SMART-seq2)	Fluidigm C1	Pooled approaches (CEL-seq, MARS-seq, SCRB-seq, CEL-seq2)	Massively parallel approaches (Drop-seq, InDrop)
Cell capture method	Laser	Mass cytometry	Micropipettes	FACS	Microfluidics	FACS	Microdroplets
Number of cells per experiment	Millions	Millions	300–1,000	50–500	48–96	500–2,000	5,000–10,000
Cost	$0.05 per cell	$35 per cell	$1 per cell	$3–6 per well	$35 per cell	$3–6 per well	$0.05 per cell
Sensitivity	Up to 17 markers	Up to 40 markers	10–30 genes per cell	7,000–10,000 genes per cell for cell lines; 2,000–6,000 genes per cell for primary cells	6,000–9,00 genes per cell for cell lines; 1,000–5,000 genes per cell for primary cells	7,000–10,000 genes per cell for cell lines; 2,000–6,000 genes per cell for primary cells	5,000 genes per cell for cell lines; 1,000–3,000 genes per cell for primary cells

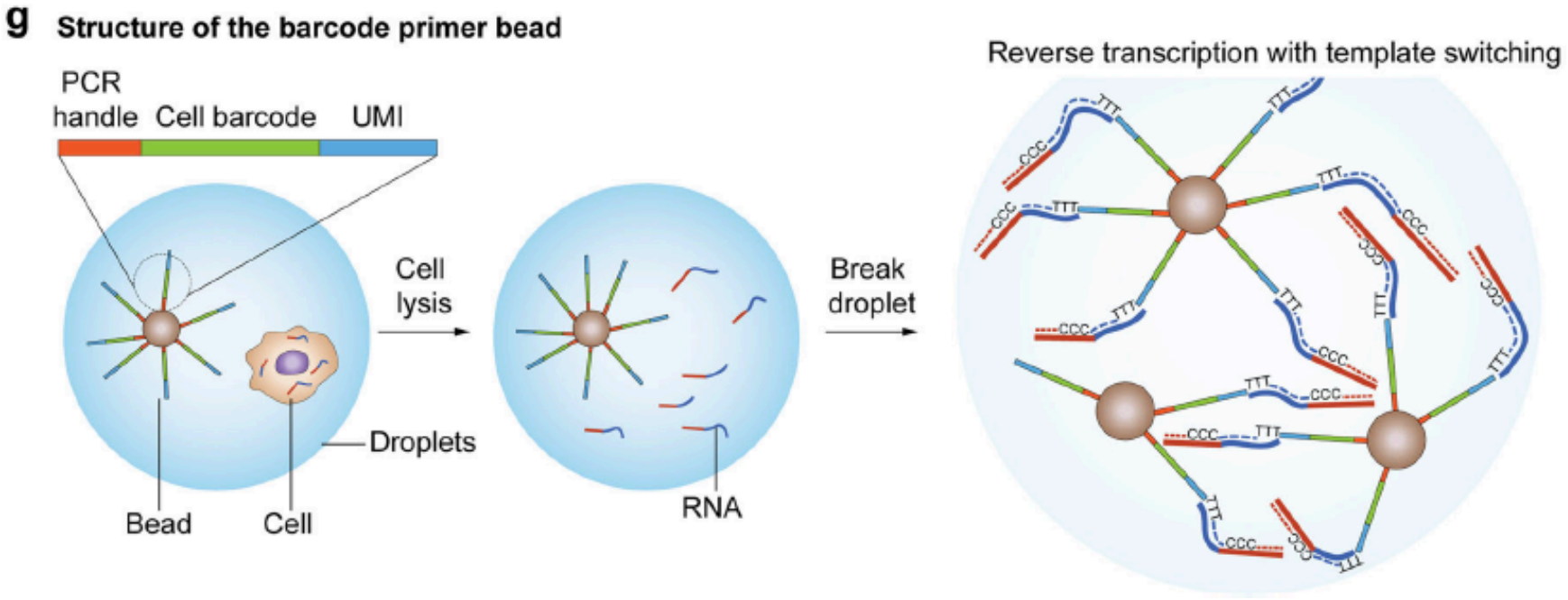
CEL-seq, cell expression by linear amplification and sequencing; CyTOF, cytometry by time of flight (mass cytometry); FACS, fluorescence-activated cell sorting; InDrop, indexing droplets sequencing; MARS-seq, massively parallel single-cell RNA sequencing; qPCR, quantitative PCR; SCRB-seq, single-cell RNA barcoding and sequencing; STRT-seq, single-cell tagged reverse transcription sequencing.

Paplexi et al. 2017

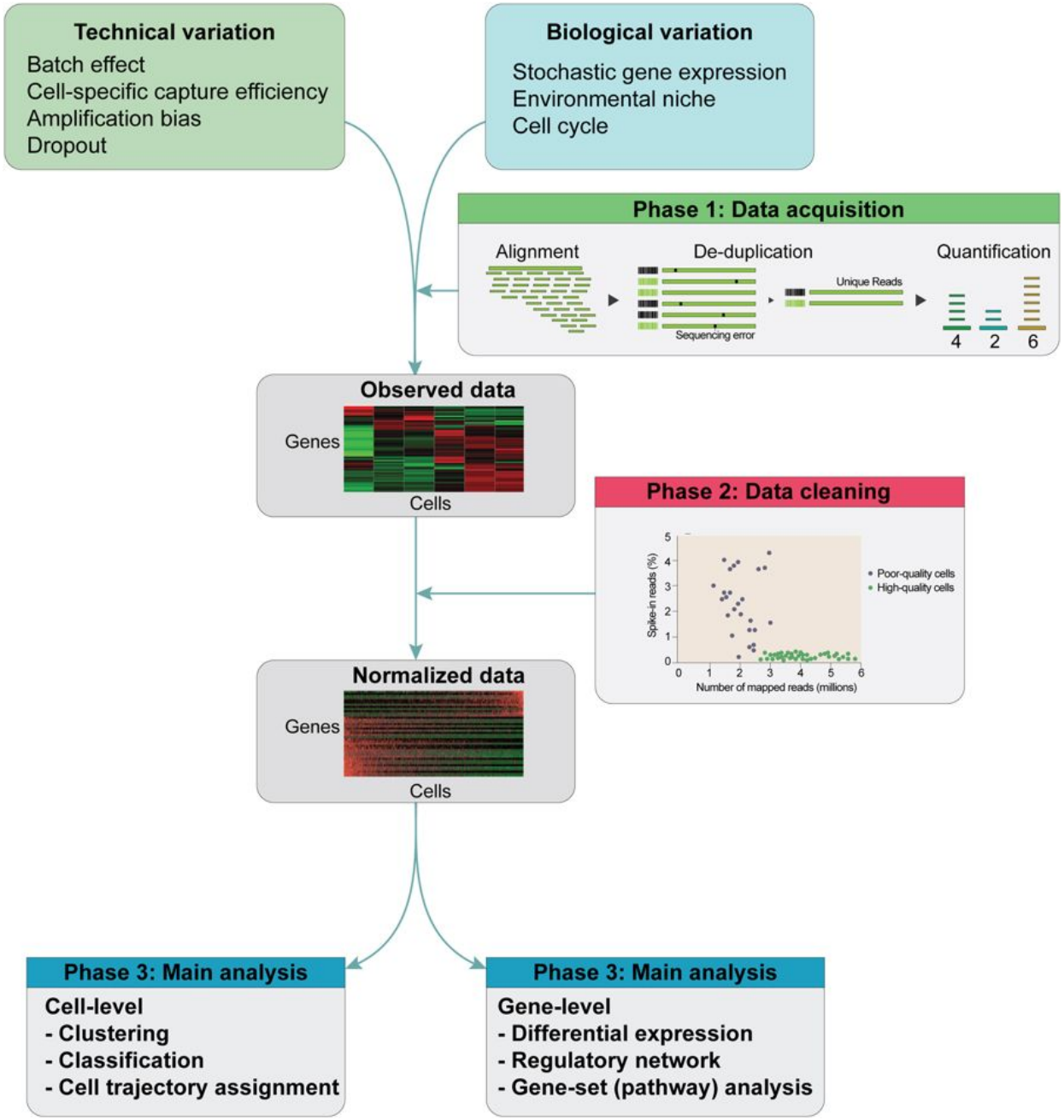
We can use droplets as reaction chambers
10x Genomics Chromium



Each droplet contains a unique cell barcode

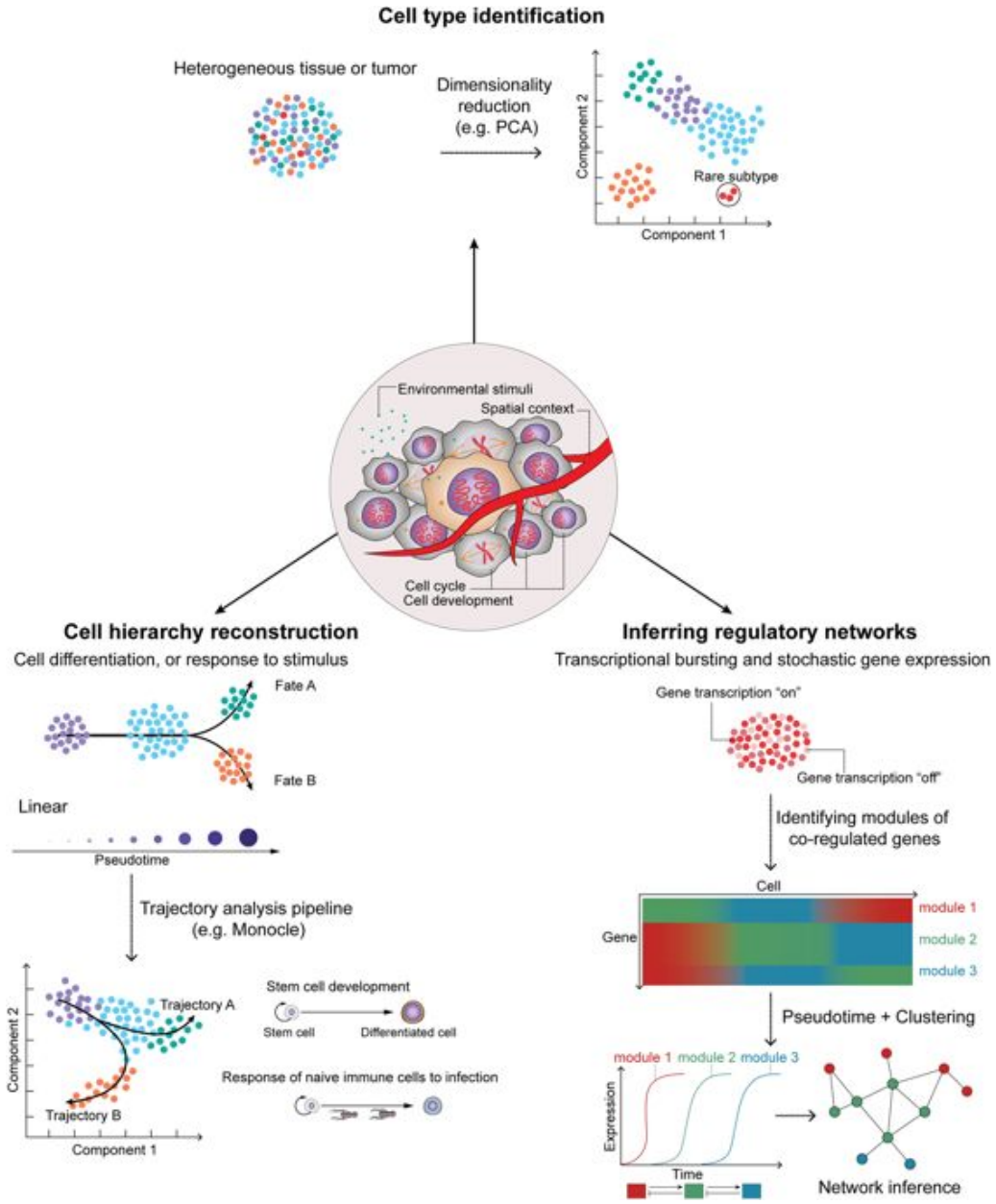


Common steps in scRNA-seq analysis



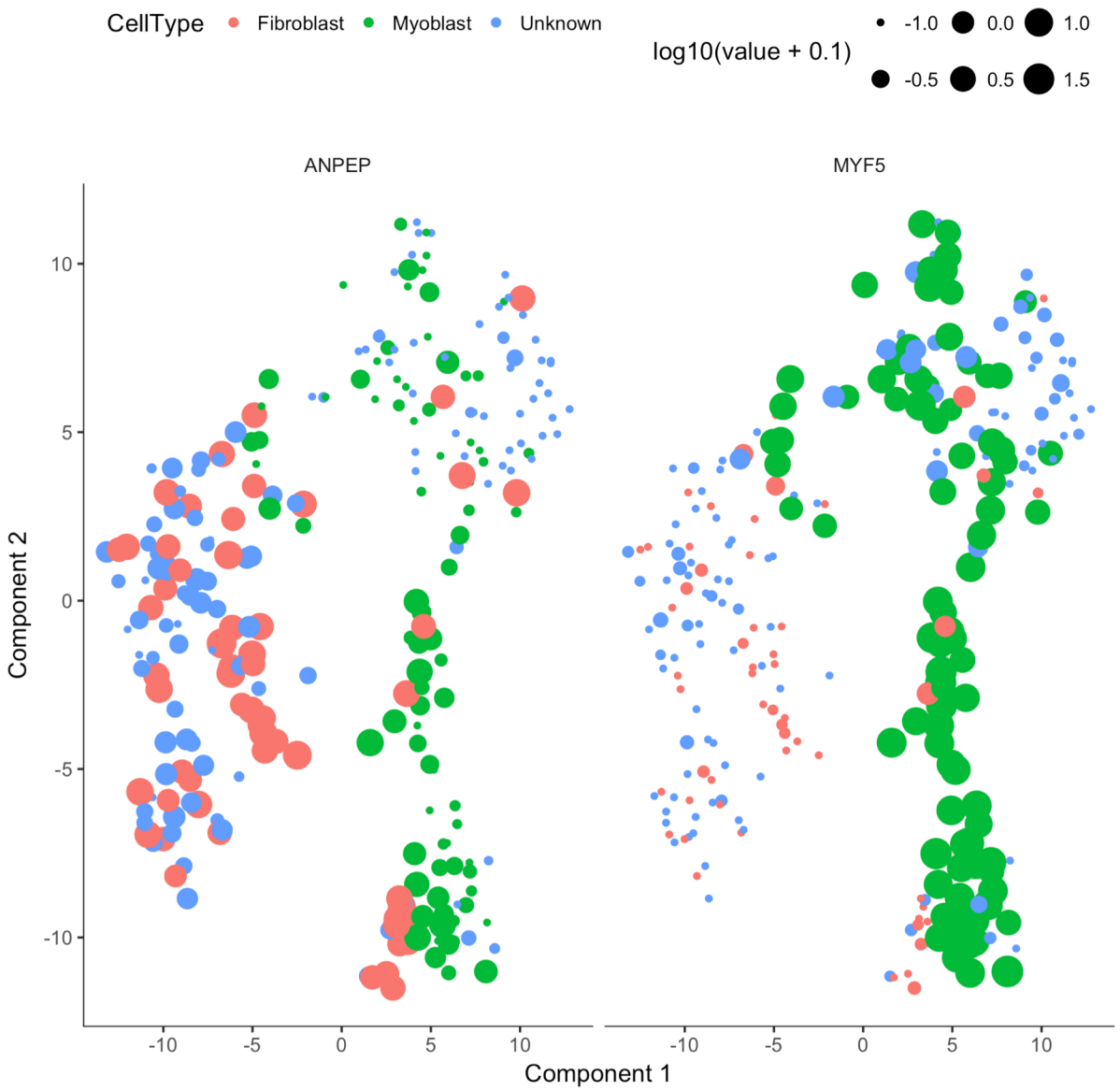
Hwang et al. 2018

Downstream biological analyses of scRNA-seq

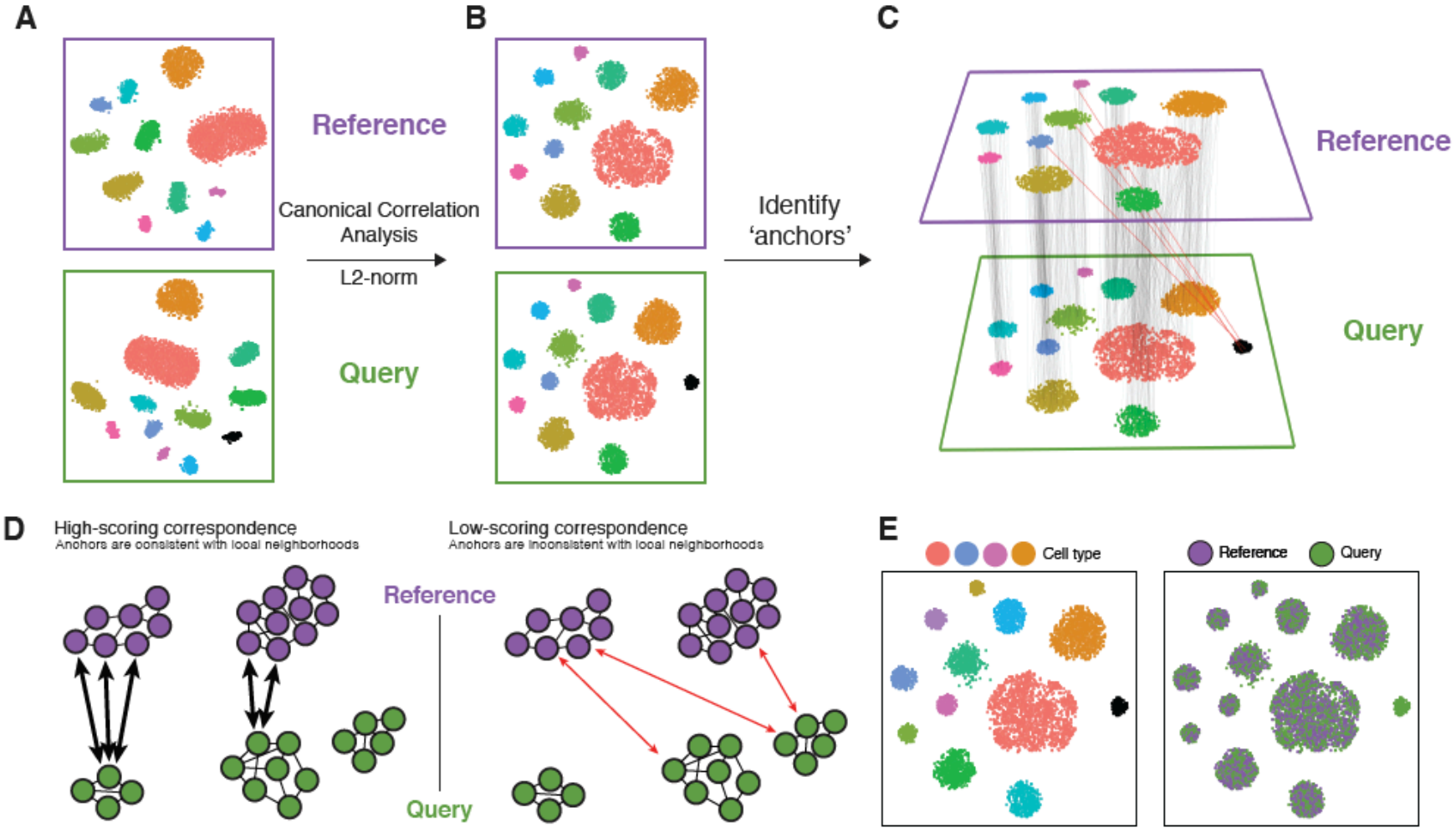


Hwang et al. 2018

Labeling by gene expression

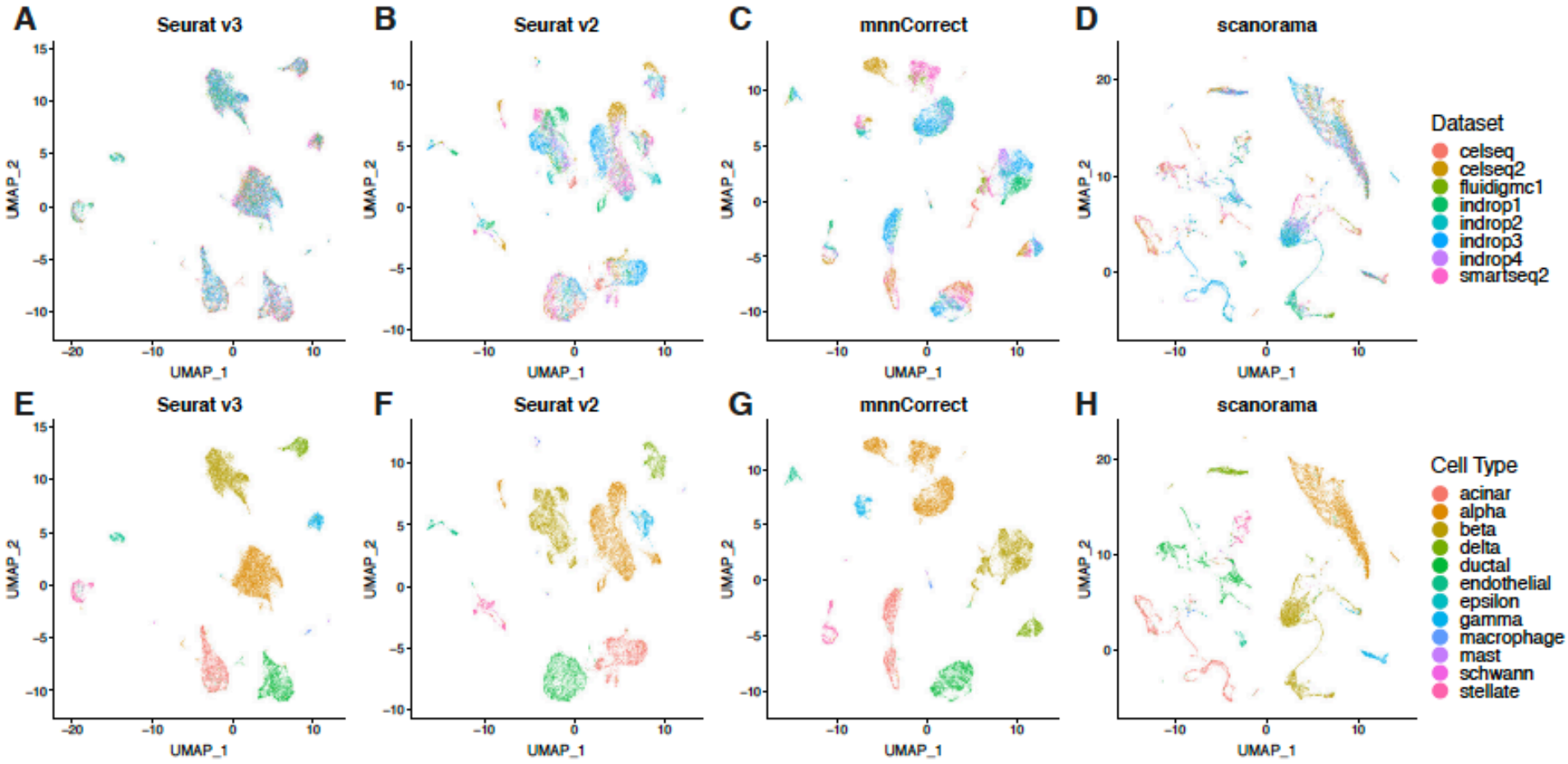


Matching clusters from a reference to a query data set

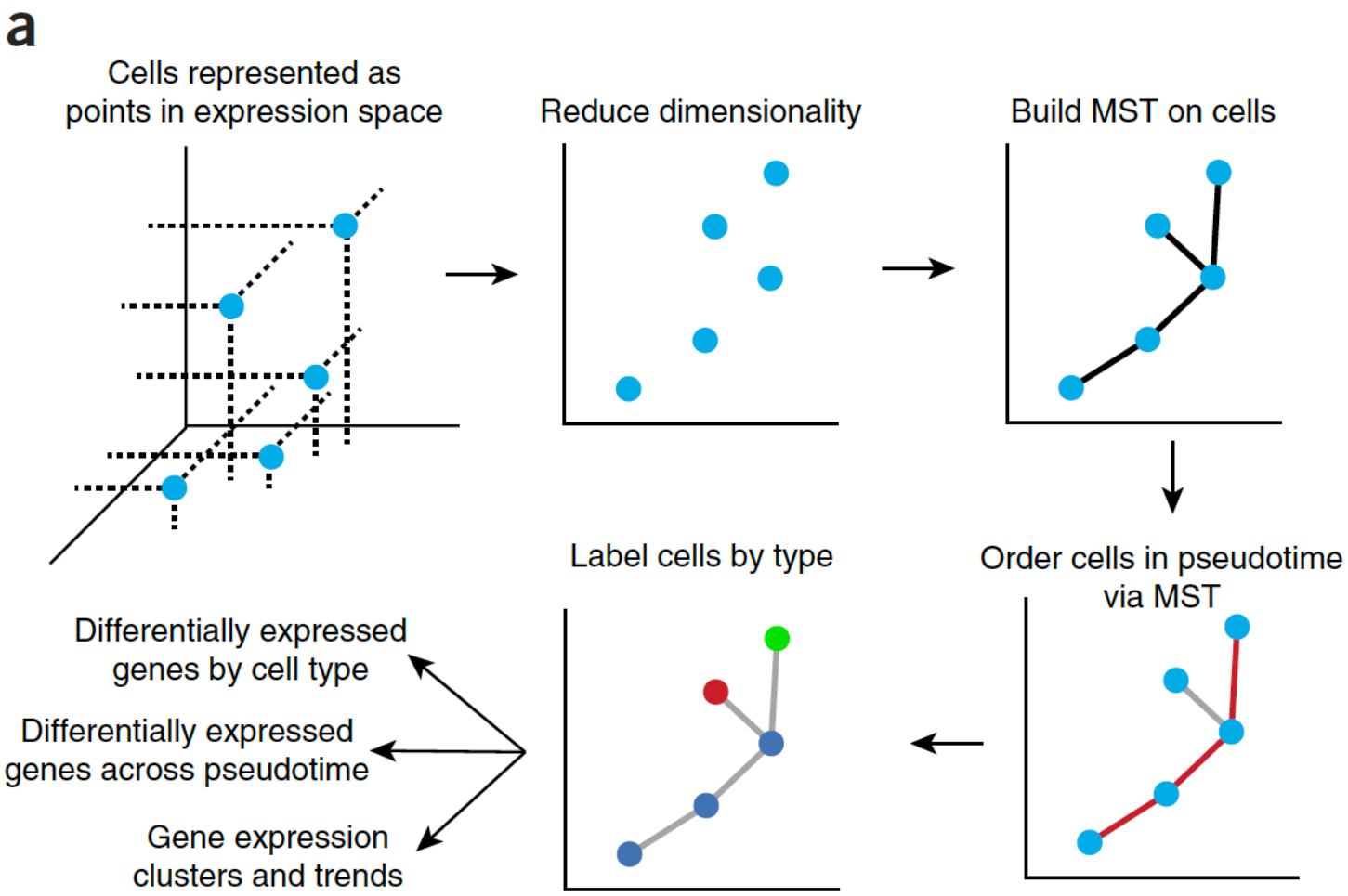


https://www.biorxiv.org/content/biorxiv/early/2018/11/02/460147.full.pdf

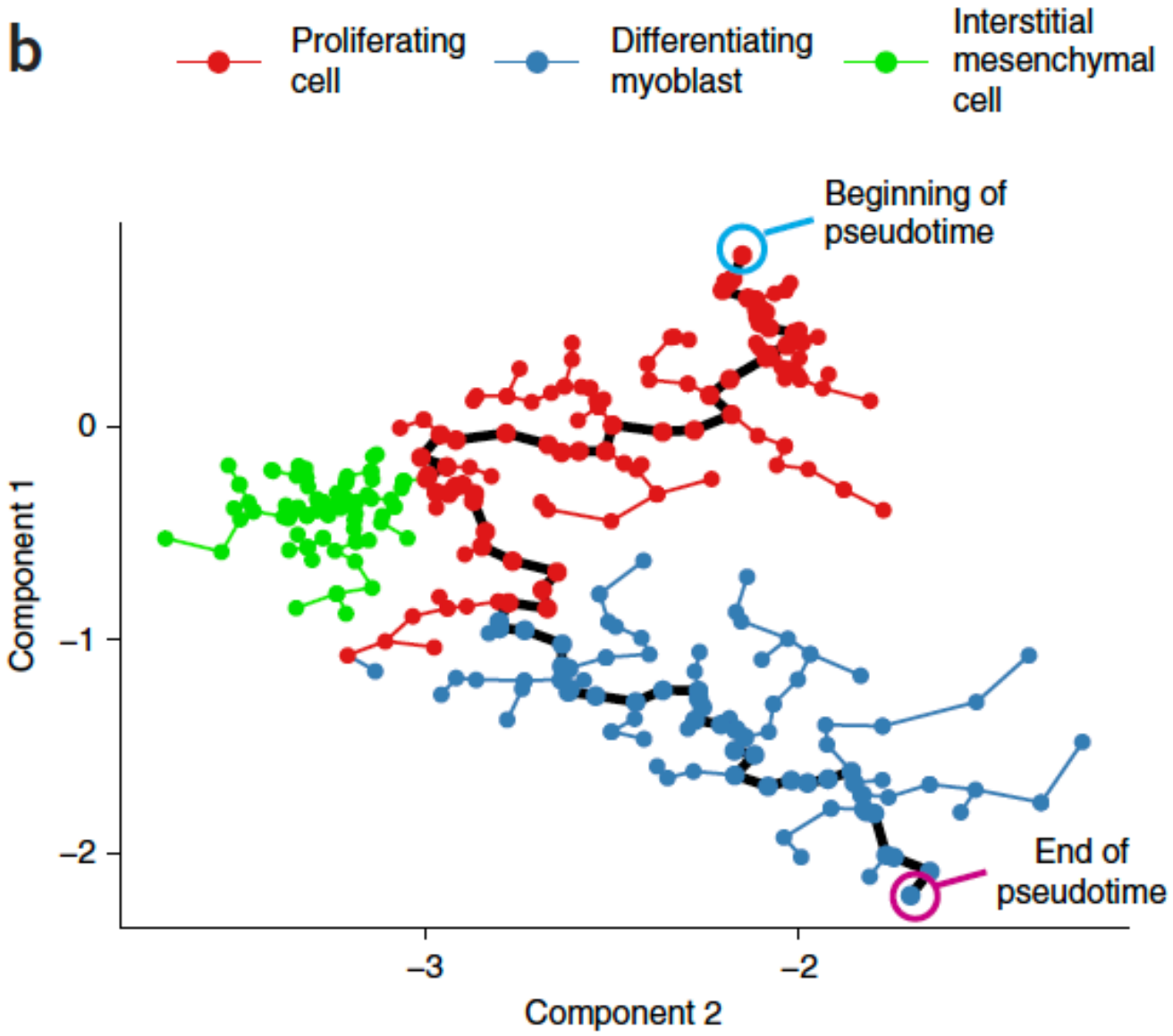
Transferring labels from a reference to a query data set



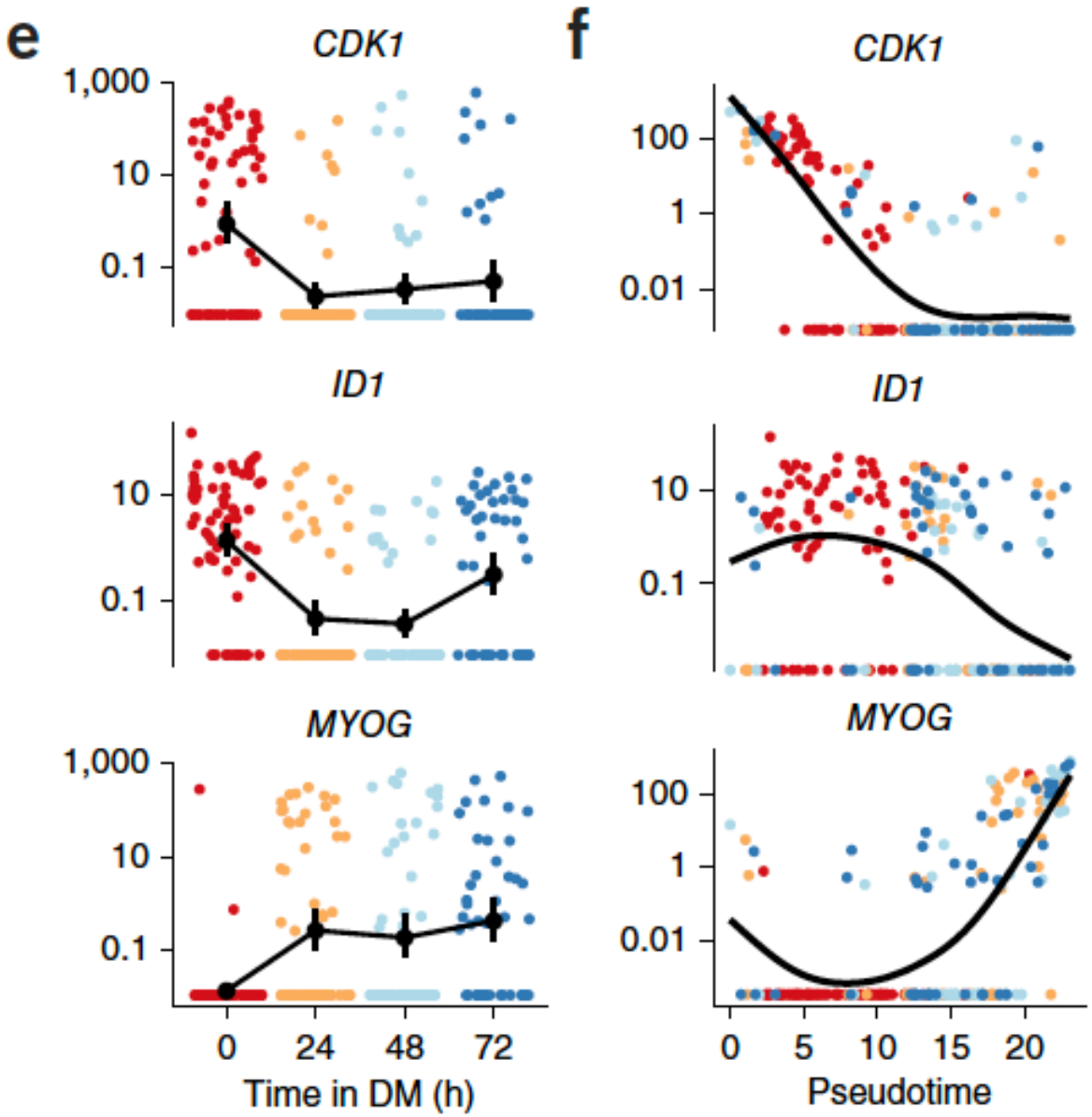
Cells can be ordered by ”pseudotime”



Cells ordered by pseudotime



Real-time and pseudotime ordering of key muscle differentiation genes

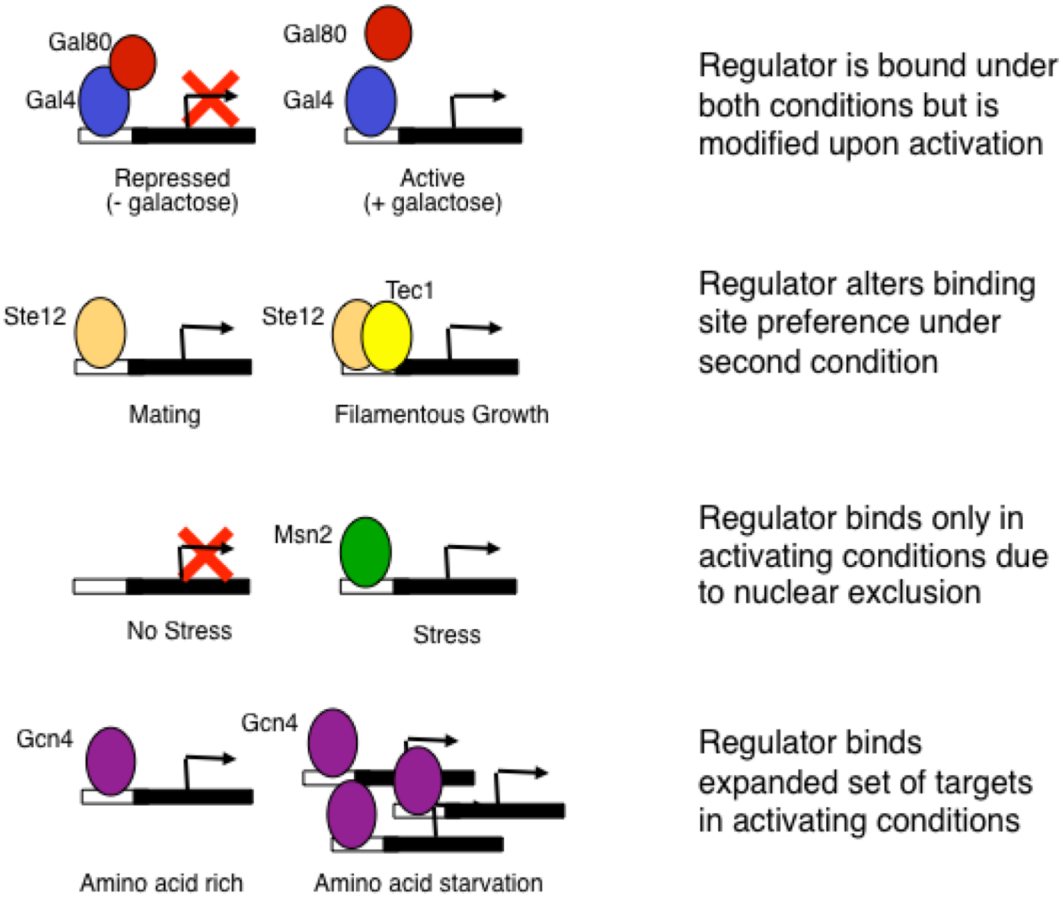


Regression trees are a modular approach for modeling RNA expression

Modules and modularity

- Modularity guides most engineering systems
  - separates implementation from interface/function
  - creates building blocks, easy combination
- Biological systems are more "integrated" and involve greater robustness (compensation)
- Finding (abstracting) units of (partial) autonomy in cells is therefore a more challenging task
  - the focus on larger groups of genes nevertheless gives us some statistical power

TF expression, mechanisms



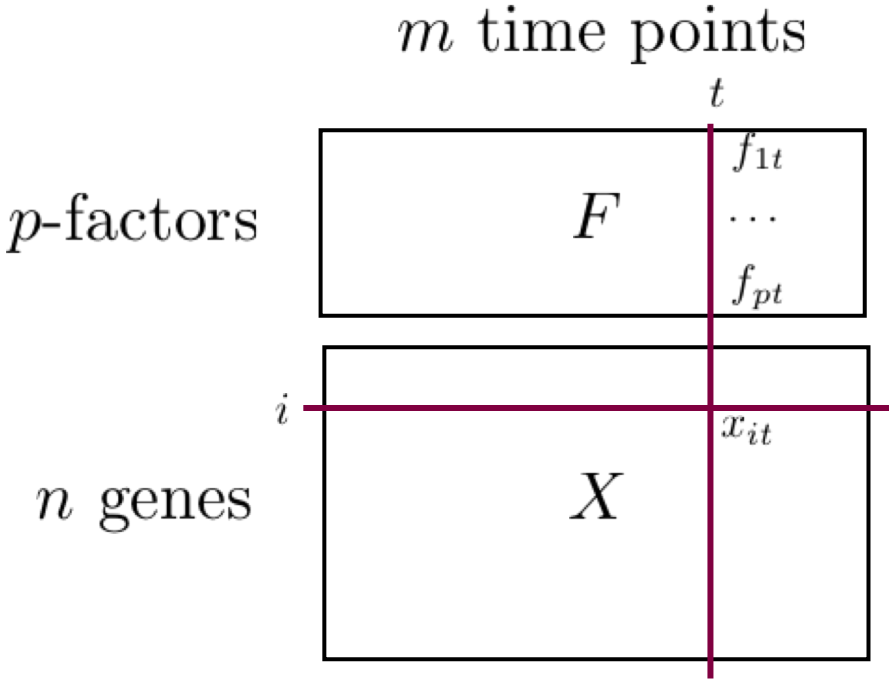
(Young, 2003)

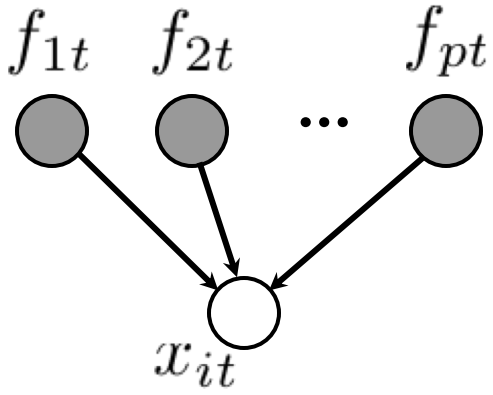
Expression programs

- We can try to relate the expression of transcription factors (TFs) and genes they regulate
- The key assumption we have to make is that TFs are themselves transcriptionally regulated, i.e., their mRNA levels are indicative of their activity
- The problem:

  Find a subset of TFs and the genes they regulate along with the function (program) that relates their expression

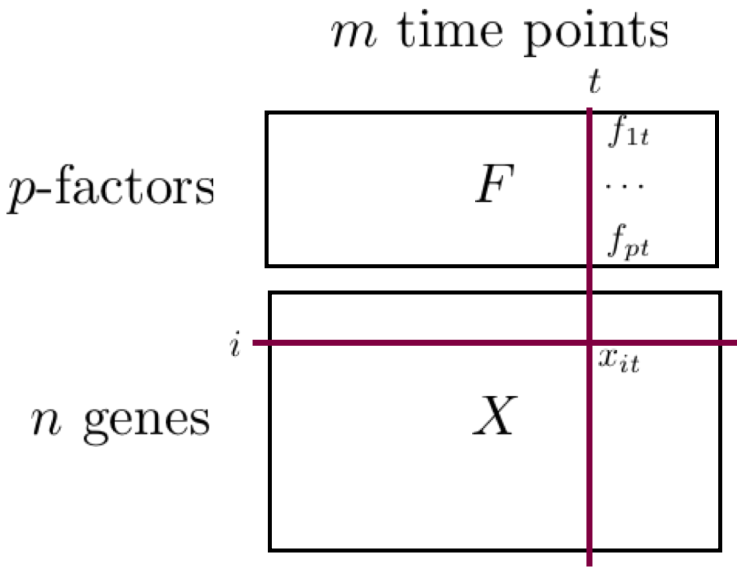
A regression model

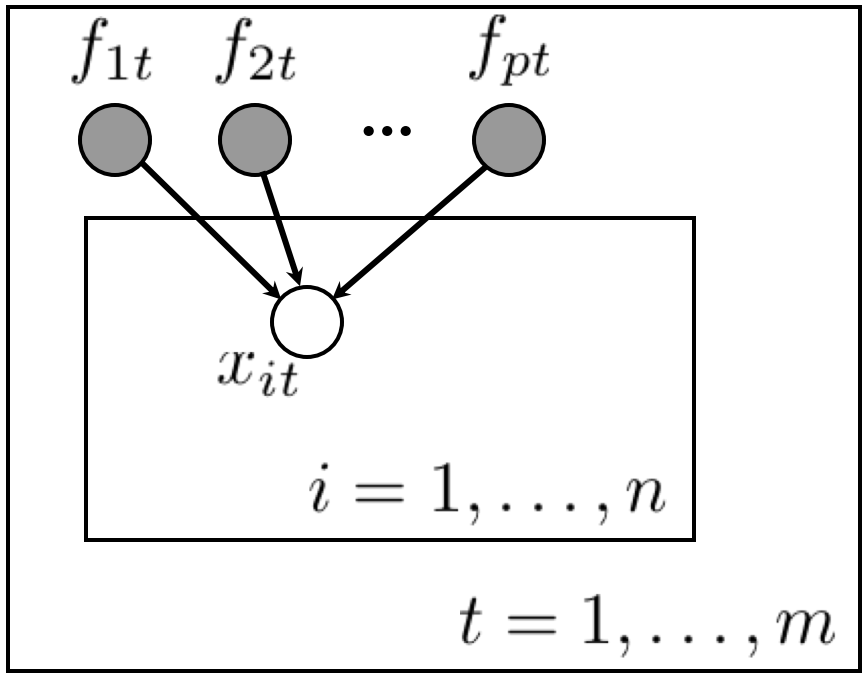




$$P(x_{it}|f_{1t},\dots,f_{pt},\Theta)$$

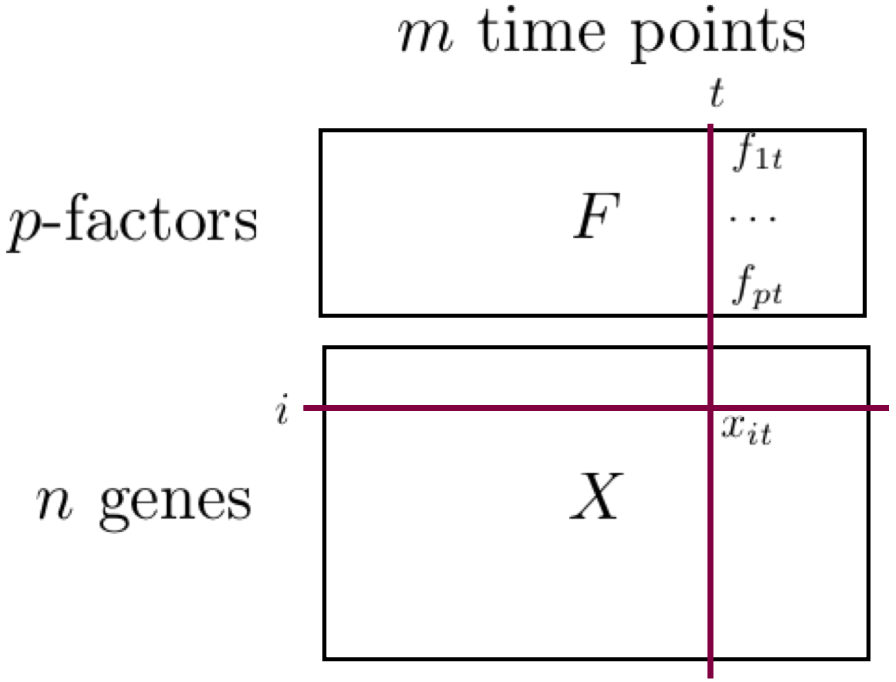
A regression model: sampling

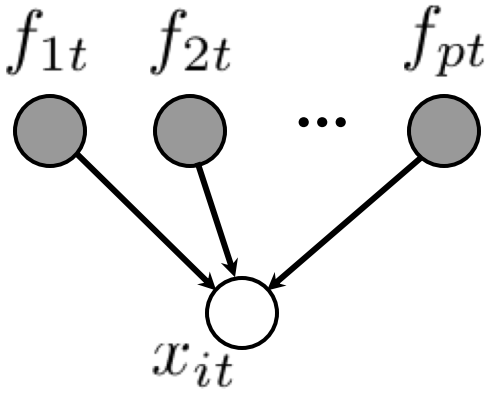




$$P(X|F,\Theta_1) = \prod_{t=1}^{m} \left[\prod_{i=1}^{n} P(x_{it}|f_{1t},\ldots,f_{pt},\Theta_1) \right]$$

Possible regression models



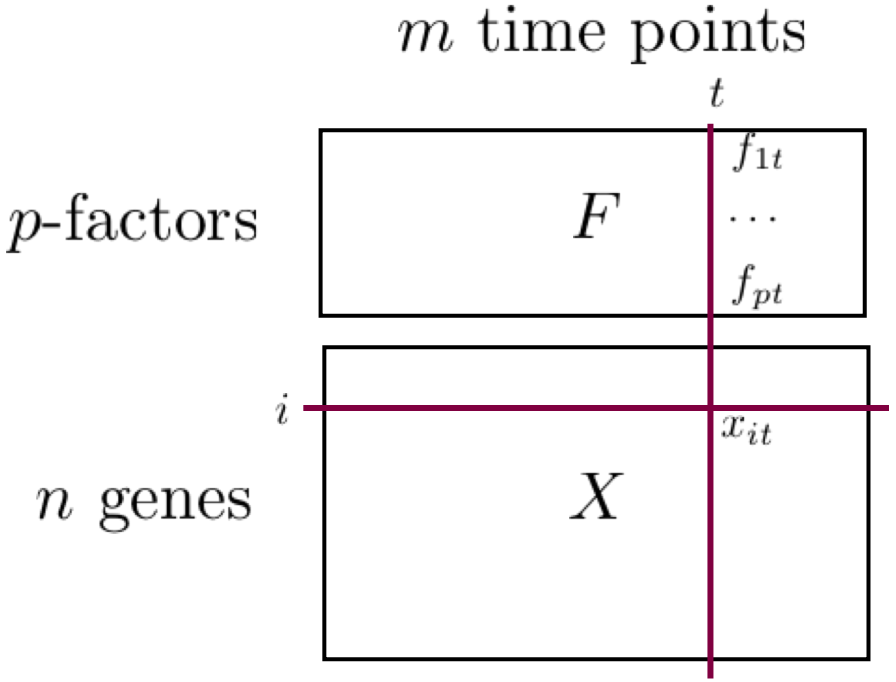


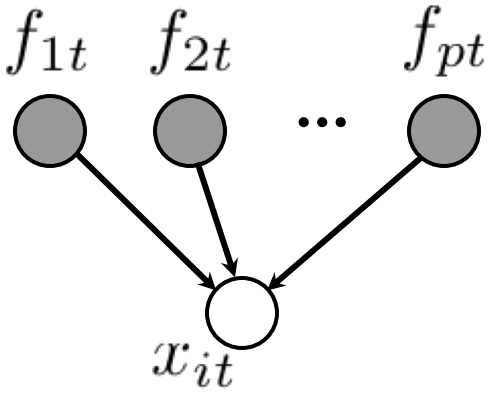
$$P(x_{it}|f_{1t},\dots,f_{pt},\Theta_1) =$$

- Linear regression (with feature selection)
- Regression tree
- Discretization + conditional table

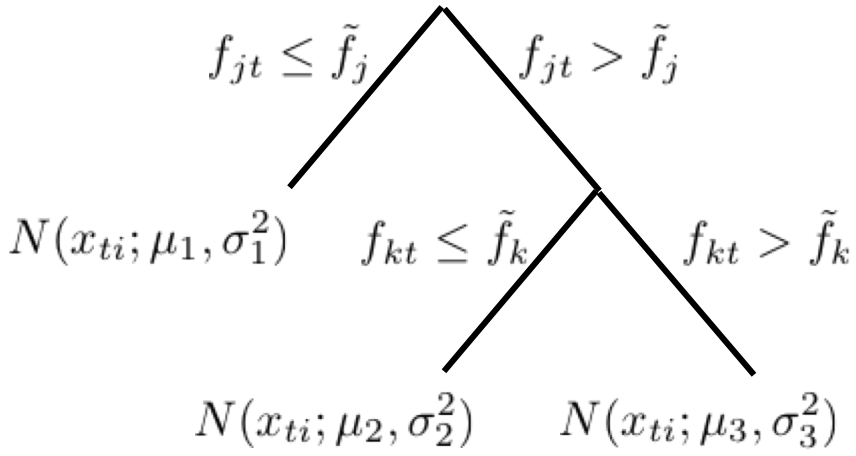
etc.

A regression tree

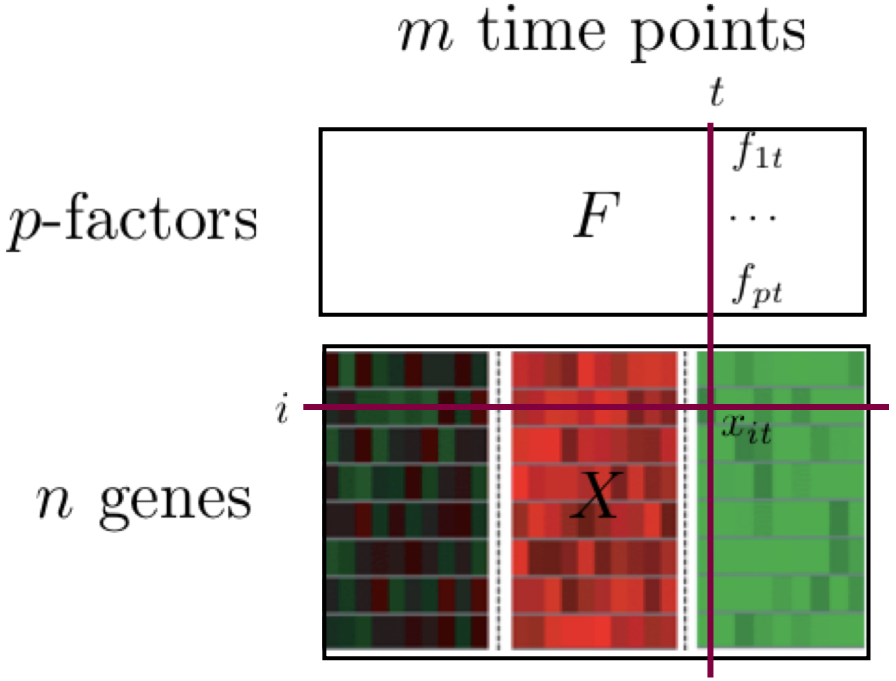


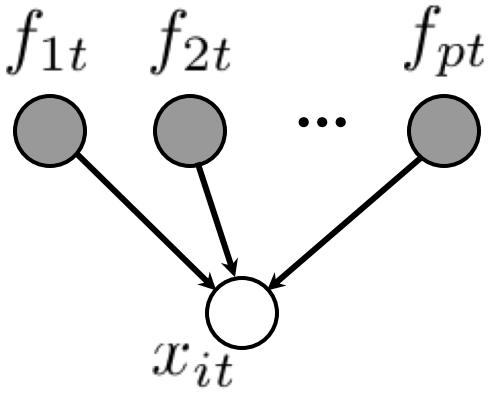


$$P(x_{it}|f_{1t},\dots,f_{pt},\Theta_1) =$$

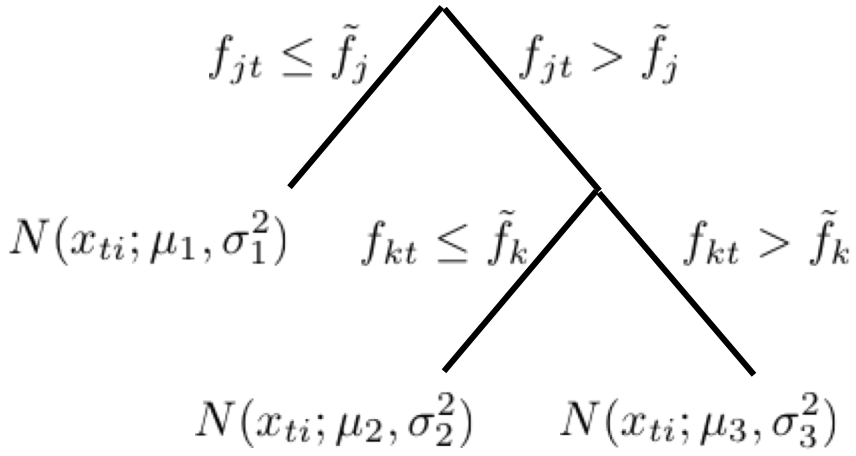


A regression tree

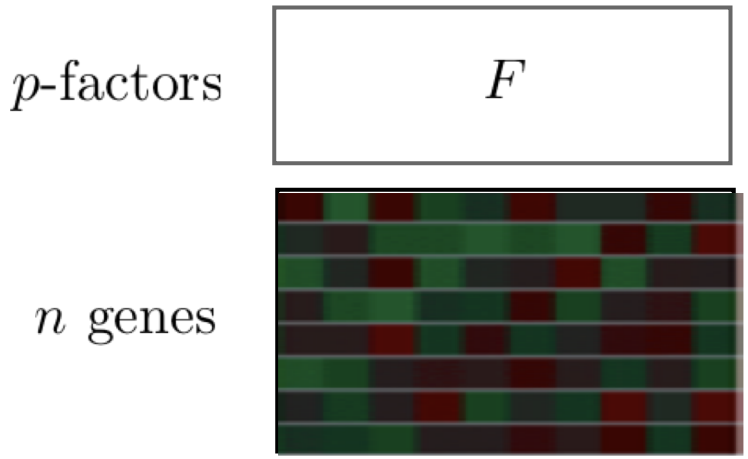




$$P(x_{it}|f_{1t},\dots,f_{pt},\Theta_1) =$$



Learning regression trees: null model

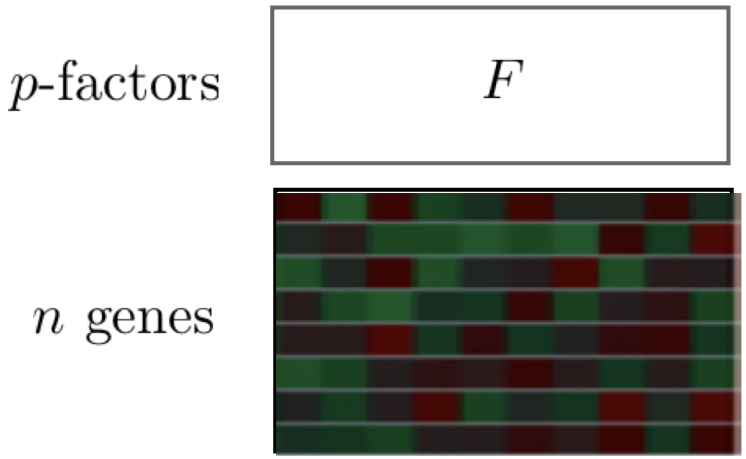


- The simplest case (null model): no dependence on the factors

$$P(X;\theta) = \prod_{i=1}^{n} \prod_{t=1}^{m} N(x_{it};\, \mu, \sigma^2)$$

where $\theta = \{\mu, \sigma^2\}$.

Learning regression trees: null model



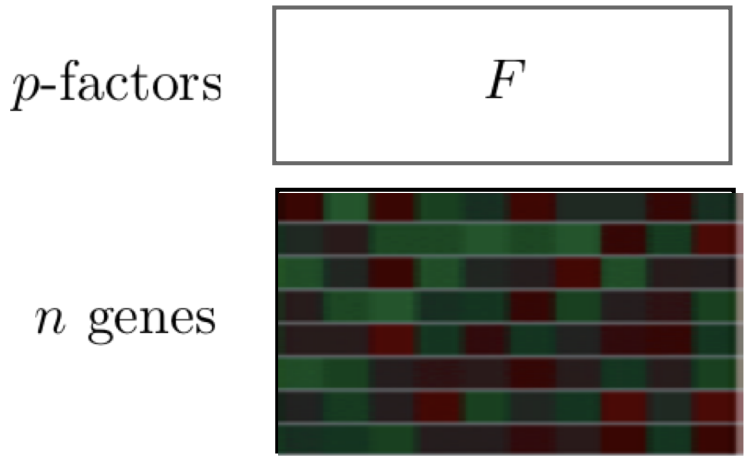
- The simplest case (null model): no dependence on the factors

$$P(X;\theta) = \prod_{i=1}^{n}\prod_{t=1}^{m} N(x_{it};\, \mu, \sigma^2)$$

where $\theta = \{\mu, \sigma^2\}$.

$$f(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Learning regression trees: null model



- The simplest case (null model): no dependence on the factors

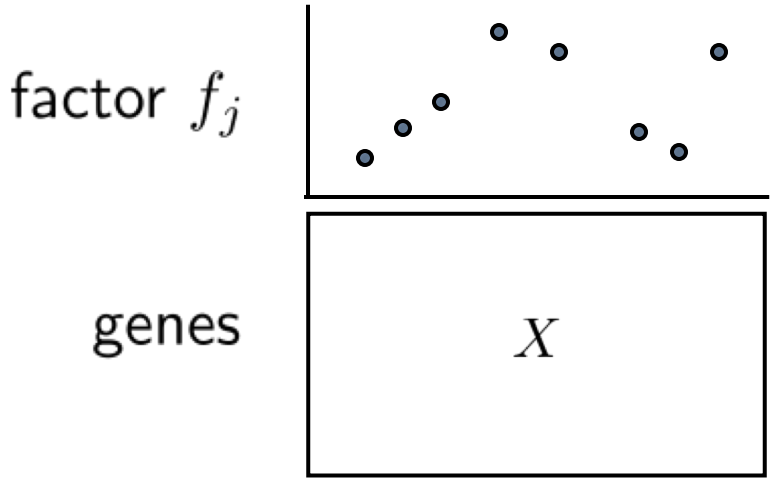
$$P(X;\theta) = \prod_{i=1}^{n}\prod_{t=1}^{m} N(x_{it};\, \mu, \sigma^2)$$

where $\theta = \{\mu, \sigma^2\}$.

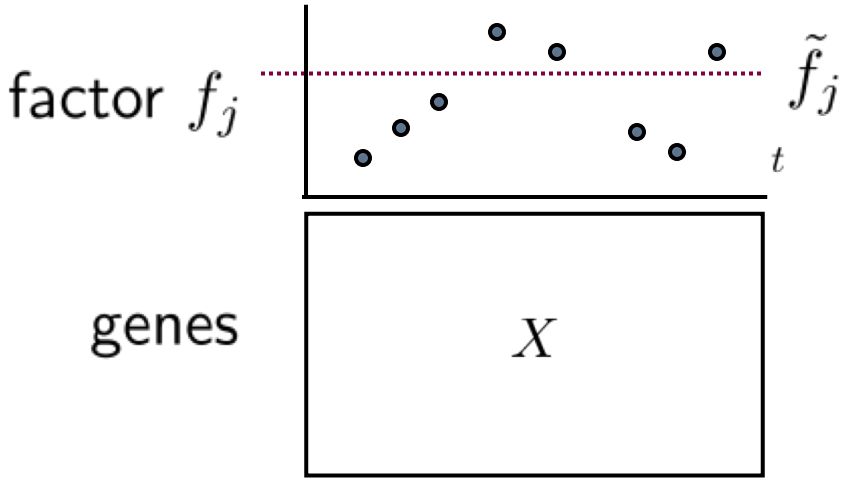
- By maximizing $l(X;\theta) = \log P(X;\theta)$ with respect to the parameters (mean and variance), we get

$$l(X;\hat{\theta}) = nm\left[-\frac{1}{2} - \frac{1}{2}\log(2\pi\hat{\sigma}^2)\right]$$

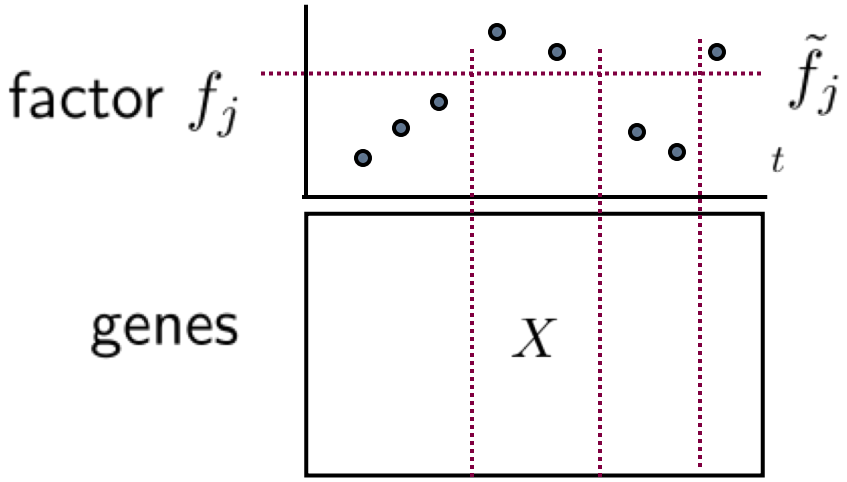
Learning regression trees: one factor



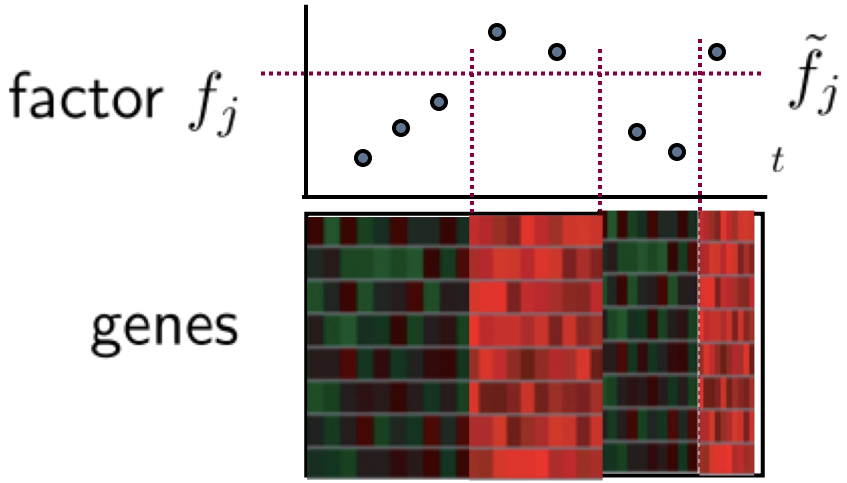
Learning regression trees: one factor



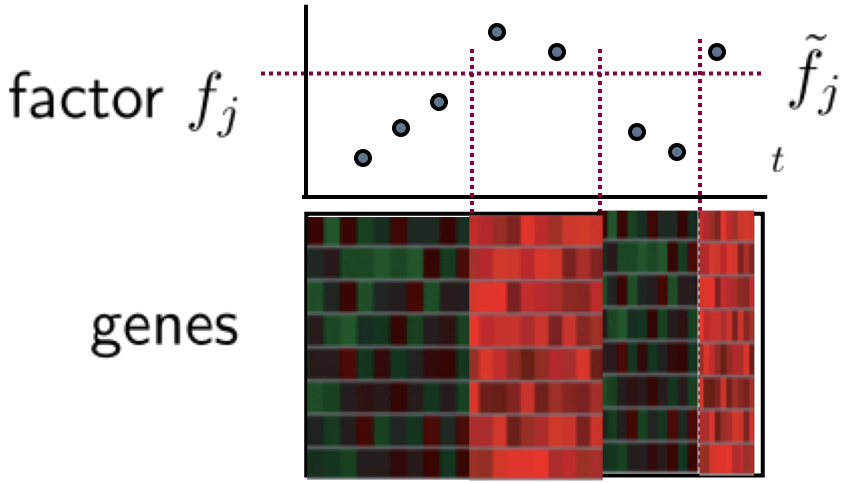
Learning regression trees: one factor



Learning regression trees: one factor



Learning regression trees: one factor



partitions

$I_1 = \{t : f_{jt} \leq \tilde{f}_{jt}\}$,

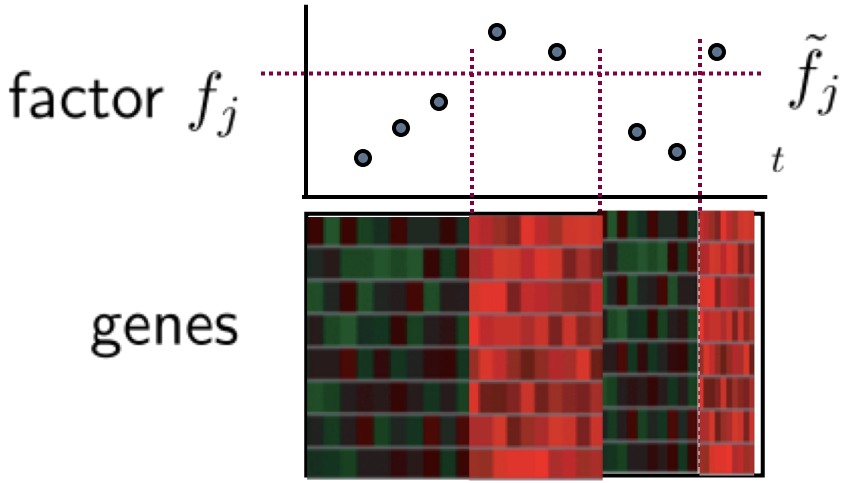
$I_2 = \{t : f_{jt} > \tilde{f}_{jt}\}$

- Conditioning on the factor value partitions the expression matrix into two types of segments

$$P(X|f_j;\Theta) = \left[\prod_{t\in I_1}\prod_{i=1}^{n} N(x_{it};\, \mu_1, \sigma_1^2)\right]\left[\prod_{t\in I_2}\prod_{i=1}^{n} N(x_{it};\, \mu_2, \sigma_2^2)\right]$$

where $\Theta = \{\tilde{f}_j, \mu_1, \mu_2, \sigma_1^2, \sigma_2^2\}$.

Learning regression trees: one factor



partitions

$I_1 = \{t : f_{jt} \leq \tilde{f}_{jt}\}$,

$I_2 = \{t : f_{jt} > \tilde{f}_{jt}\}$

- Conditioning on the factor value partitions the expression matrix into two types of segments

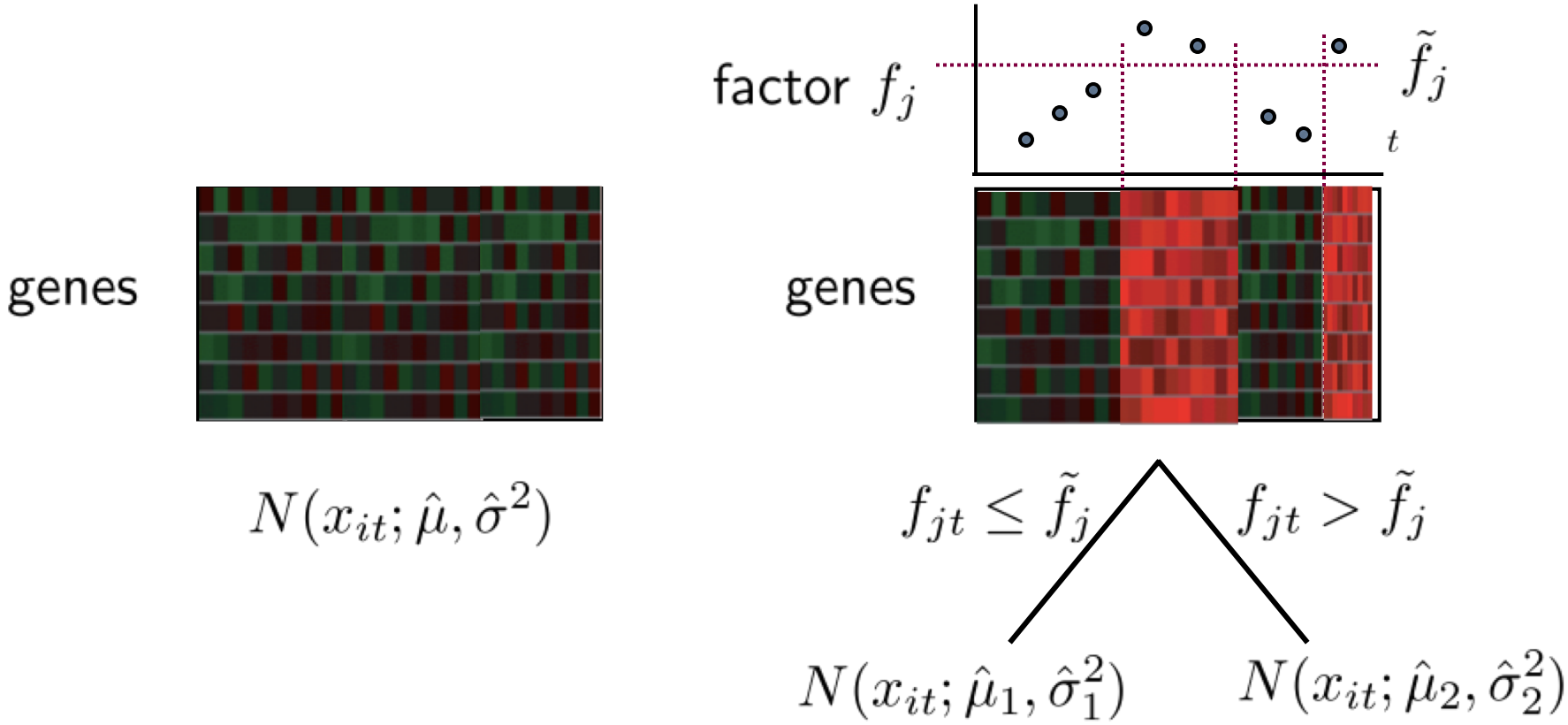
$$P(X|f_j;\Theta) = \left[\prod_{t\in I_1}\prod_{i=1}^{n} N(x_{it};\,\mu_1,\sigma_1^2)\right]\left[\prod_{t\in I_2}\prod_{i=1}^{n} N(x_{it};\,\mu_2,\sigma_2^2)\right]$$

where $\Theta = \{\tilde{f}_j, \mu_1, \mu_2, \sigma_1^2, \sigma_2^2\}$.

- By maximizing $l(X|f_j;\Theta) = \log P(X|f_j;\Theta)$ with respect to the parameters (threshold, means and variances), we get

$$l(X|f_j;\hat{\Theta}) = n|\hat{I}_1|\left[-\frac{1}{2} - \frac{1}{2}\log(2\pi\hat{\sigma}_1^2)\right] + n|\hat{I}_2|\left[-\frac{1}{2} - \frac{1}{2}\log(2\pi\hat{\sigma}_2^2)\right]$$

Competing "programs"



We can penalize a likelihood based upon the complexity of a model

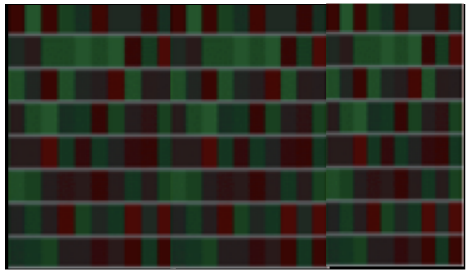
k – Number of parameters in the model

n - Number of observations in X

$$BIC_M = \log \mathcal{L}_M(X|\hat{\Theta}) - \frac{k}{2} \log n$$

Competing "programs"

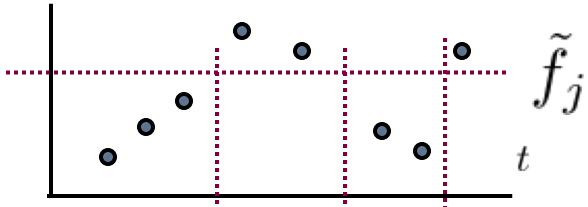
genes



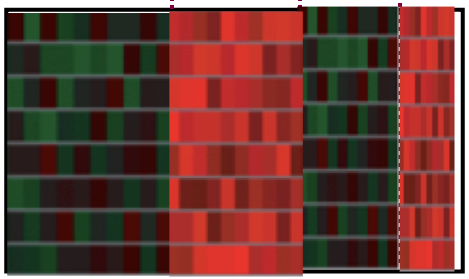
$N(x_{it}; \hat{\mu}, \hat{\sigma}^2)$

BIC score =
$l(X; \hat{\theta}) - \frac{2}{2} \log(nm)$

factor f_j



genes

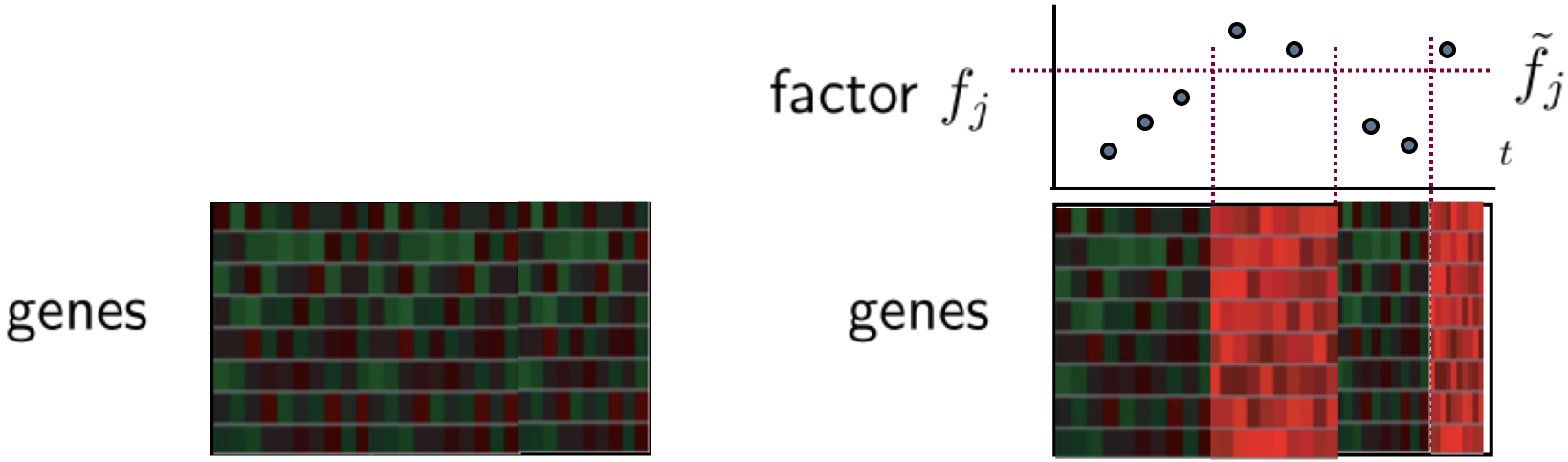


$f_{jt} \leq \tilde{f}_j$ $\quad$ $f_{jt} > \tilde{f}_j$

$N(x_{it}; \hat{\mu}_1, \hat{\sigma}_1^2)$ $\quad$ $N(x_{it}; \hat{\mu}_2, \hat{\sigma}_2^2)$

BIC score =
$l(X|f_j; \hat{\Theta}) - \frac{5}{2} \log(nm)$

The algorithm



$N(x_{it}; \hat{\mu}, \hat{\sigma}^2)$

BIC score $=$
$l(X; \hat{\theta}) - \frac{2}{2} \log(nm)$

$f_{jt} \leq \tilde{f}_j$ $\quad$ $f_{jt} > \tilde{f}_j$

$N(x_{it}; \hat{\mu}_1, \hat{\sigma}_1^2)$ $\quad$ $N(x_{it}; \hat{\mu}_2, \hat{\sigma}_2^2)$

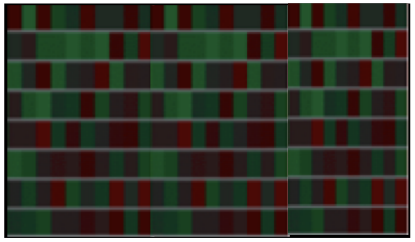
BIC score $=$
$l(X|f_j; \hat{\Theta}) - \frac{5}{2} \log(nm)$

Algorithm:

1) Calculate BIC scores with and without factor
2) If null model wins, stop, otherwise recurse

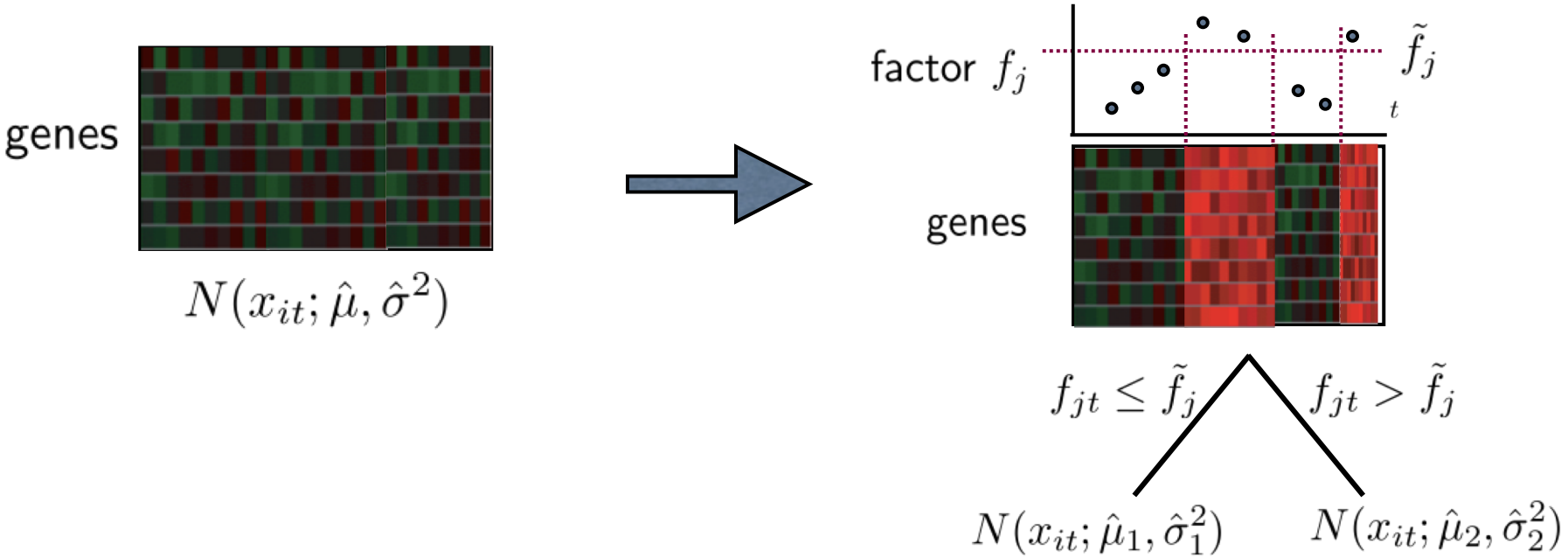
Recursive estimation

genes

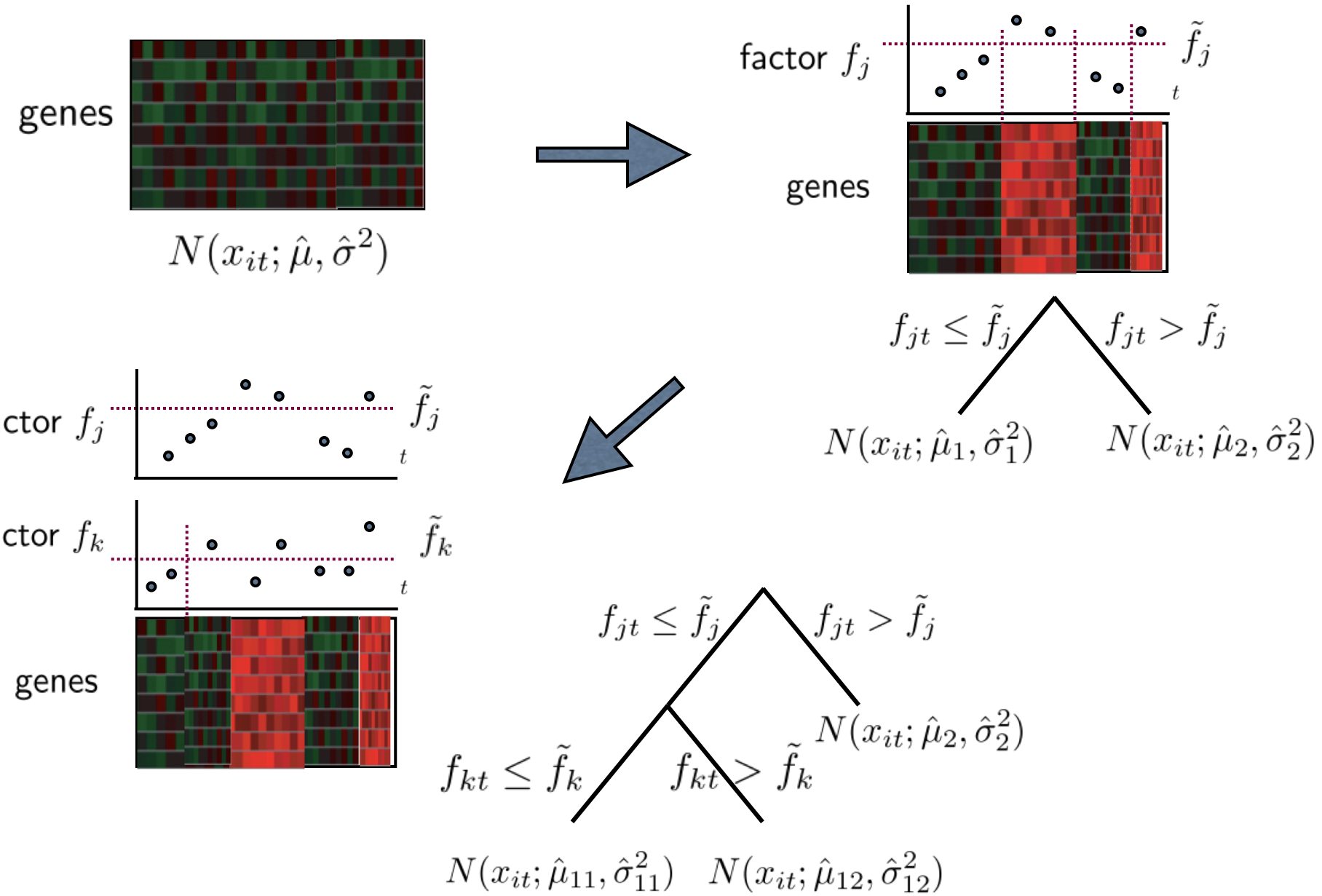


$$N(x_{it}; \hat{\mu}, \hat{\sigma}^2)$$

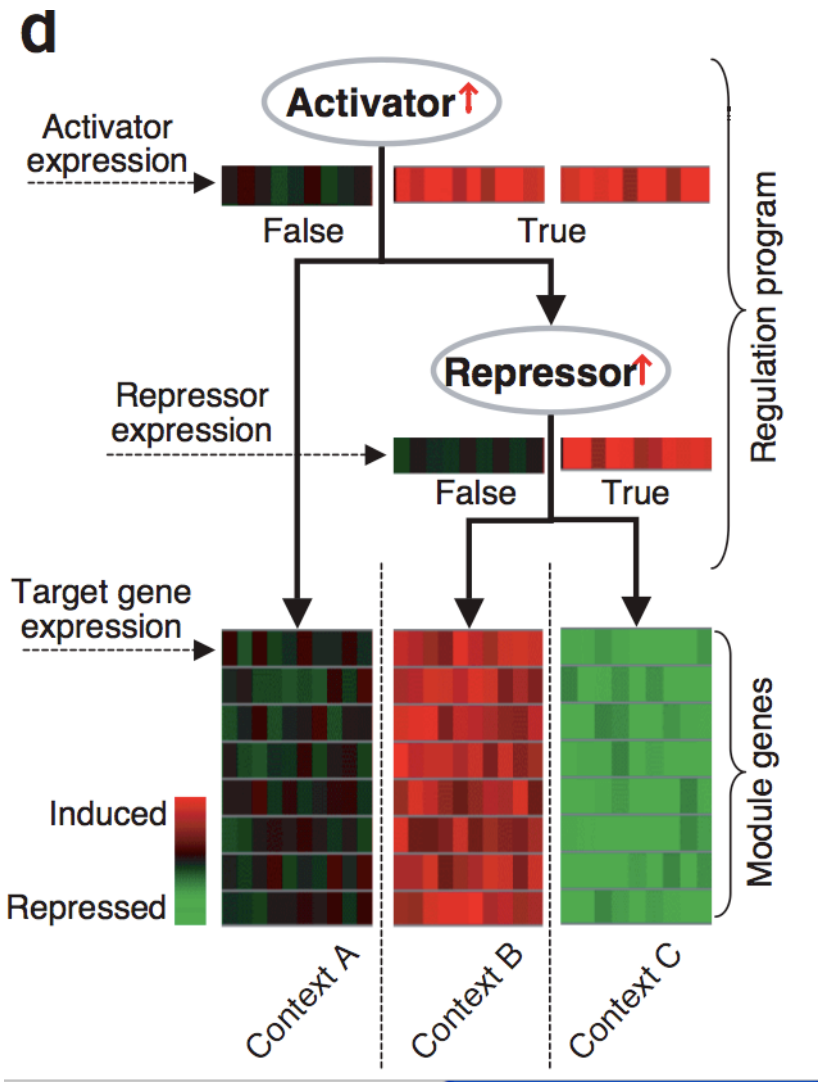
Recursive estimation



Recursive estimation



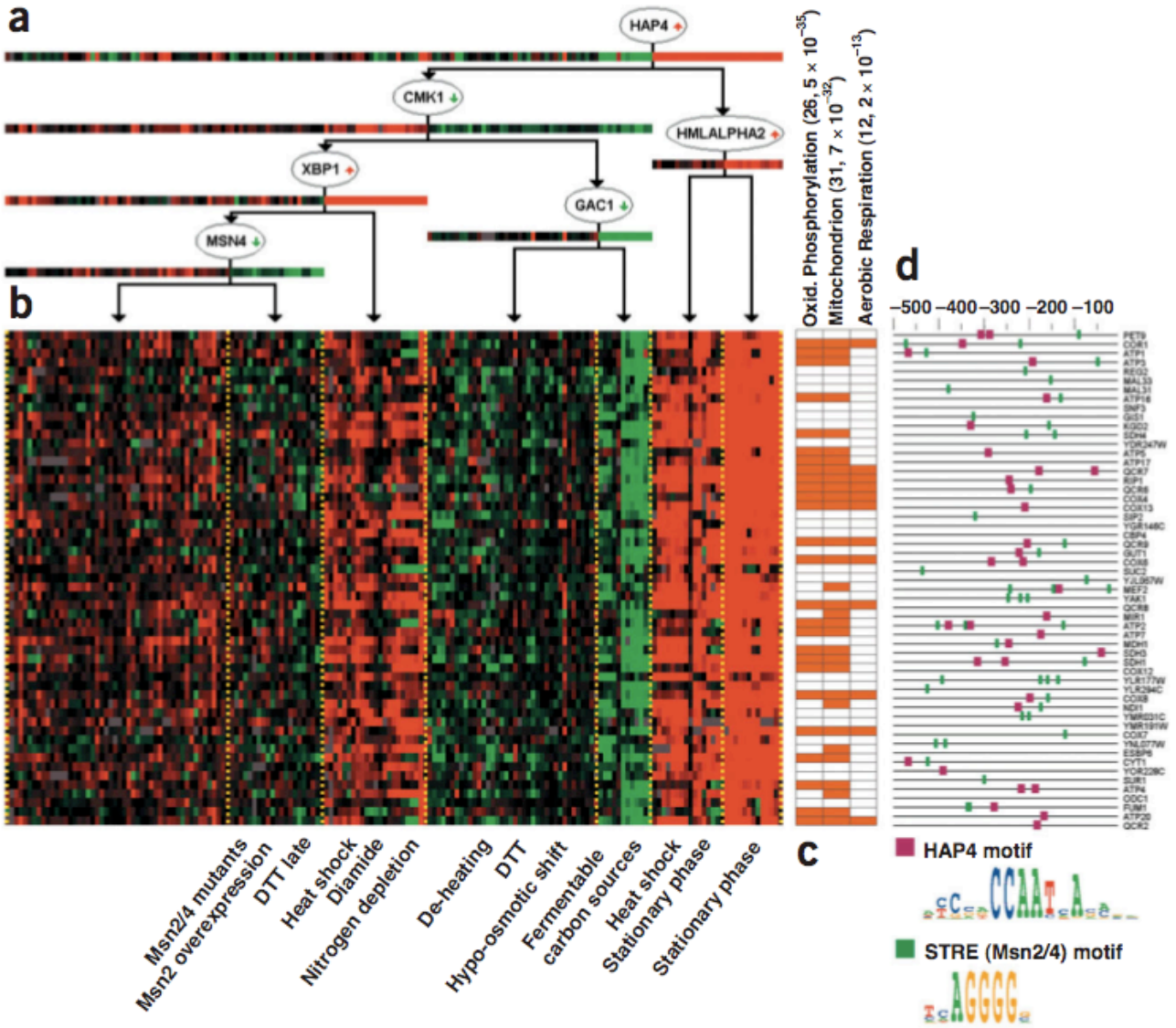
An ideal regression tree



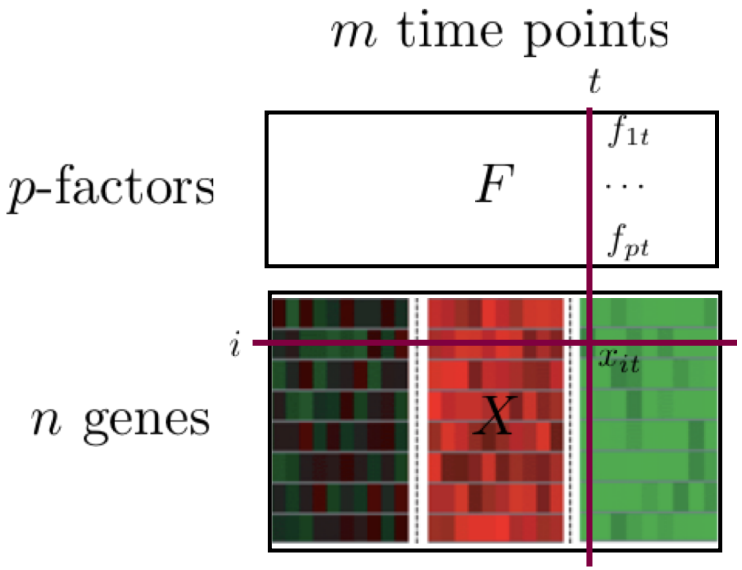
(Segal et al. 2003)

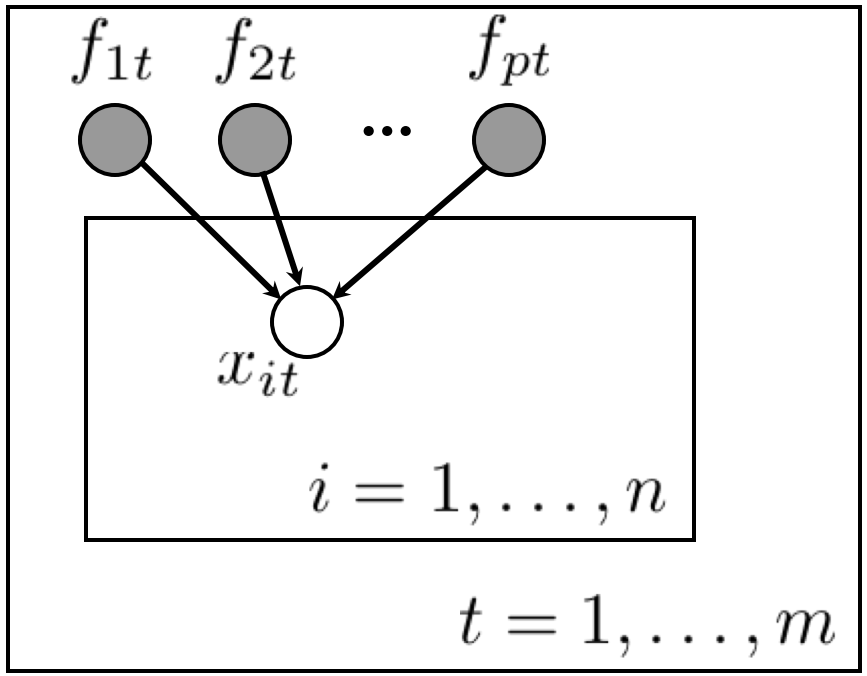
Expression programs

- An example expression program found in yeast



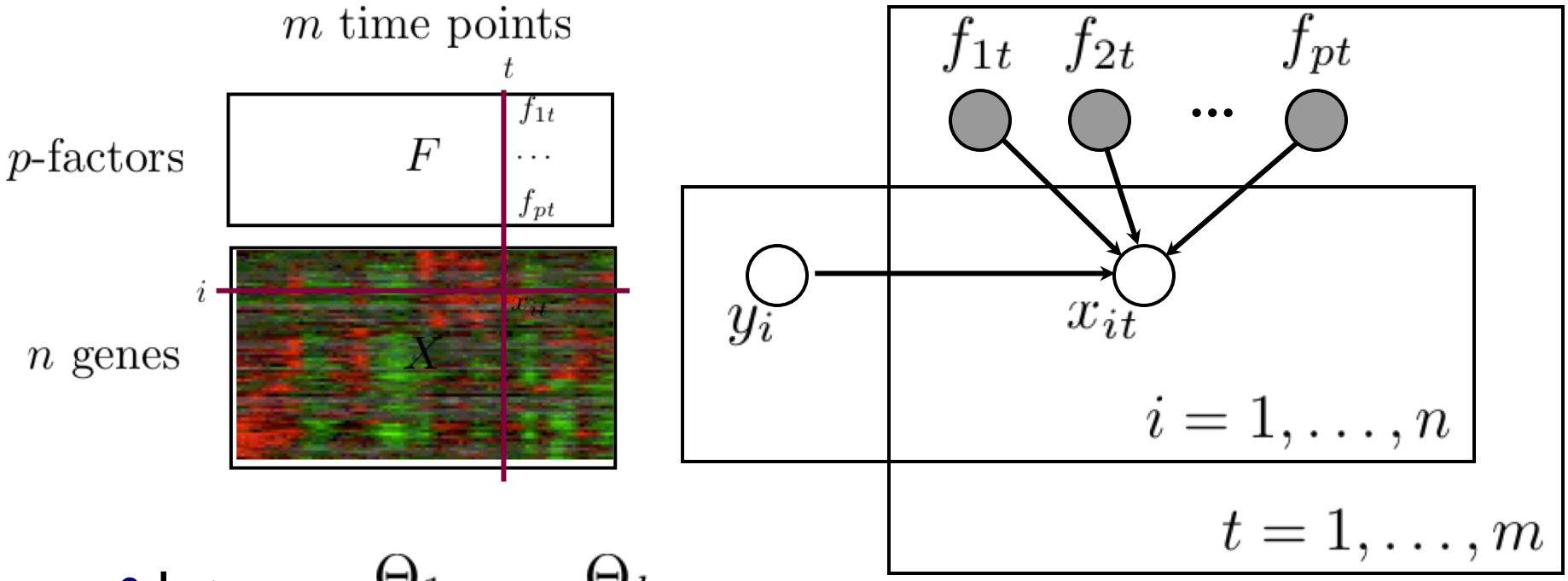
A single regression tree: sampling model





$$P(X|F,\Theta_1) = \prod_{t=1}^{m} \left[\prod_{i=1}^{n} P(x_{it}|f_{1t},\ldots,f_{pt},\Theta_1) \right]$$

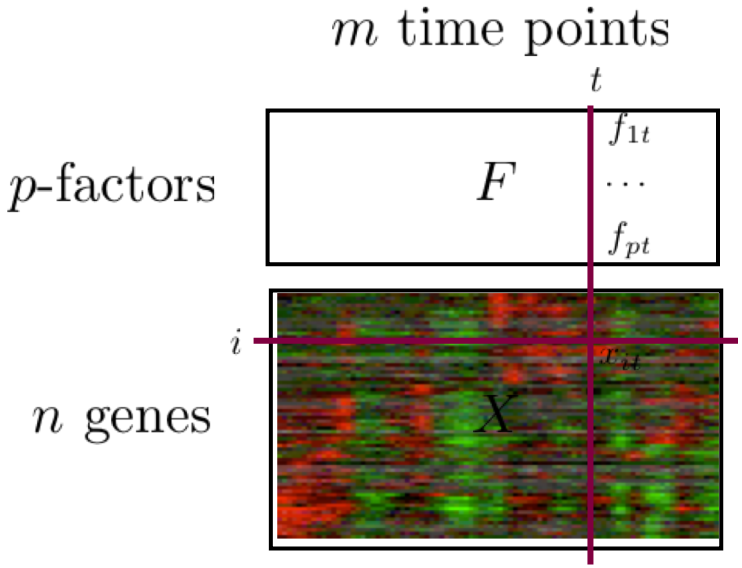
A mixture of trees: sampling model

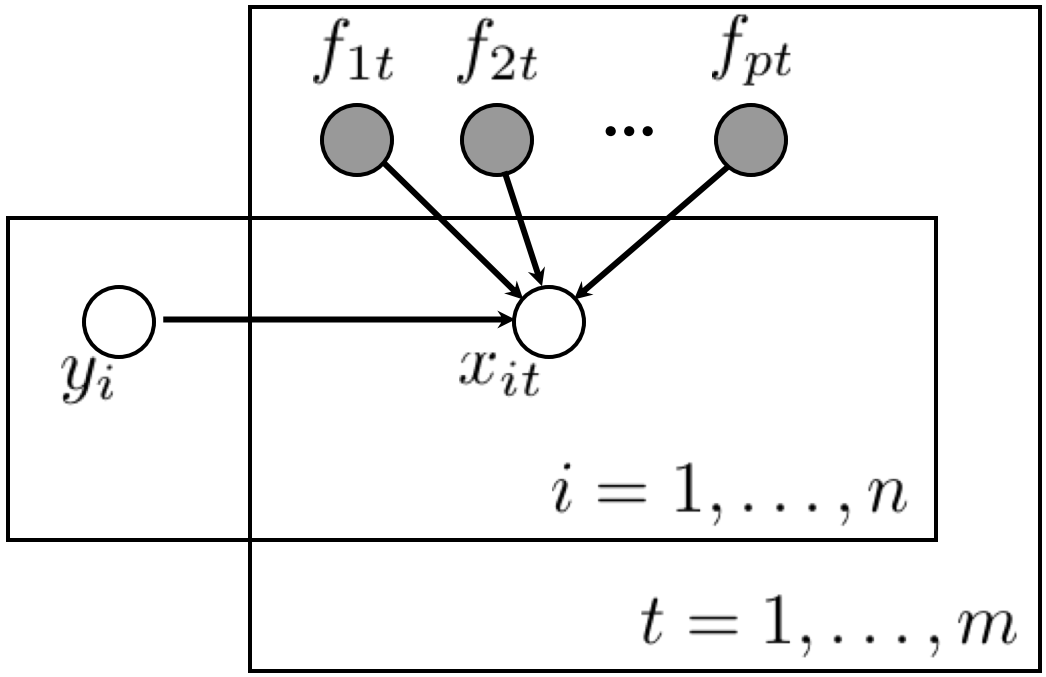


- k trees $\Theta_1, \ldots, \Theta_k$

$$P(x_{it}|f_{1t}, \ldots, f_{pt}, y) = P(x_{it}|f_{1t}, \ldots, f_{pt}, \Theta_y)$$

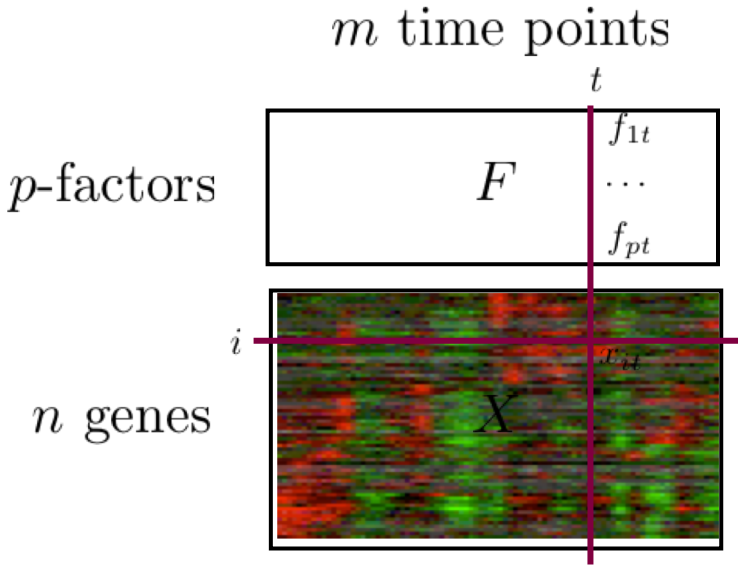
A mixture of trees: sampling model

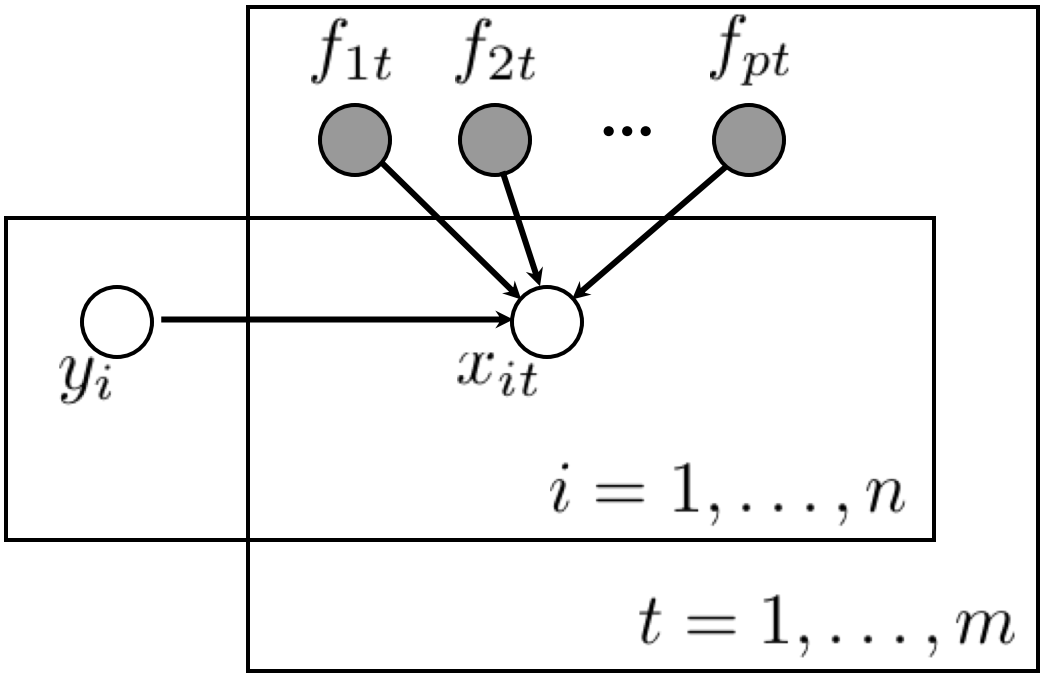




$$P(X|F,\Theta) = \prod_{i=1}^{n} \sum_{y_i=1}^{k} P(y_i) \prod_{t=1}^{m} P(x_{it}|f_{1t},\ldots,f_{pt},\Theta_{y_i})$$

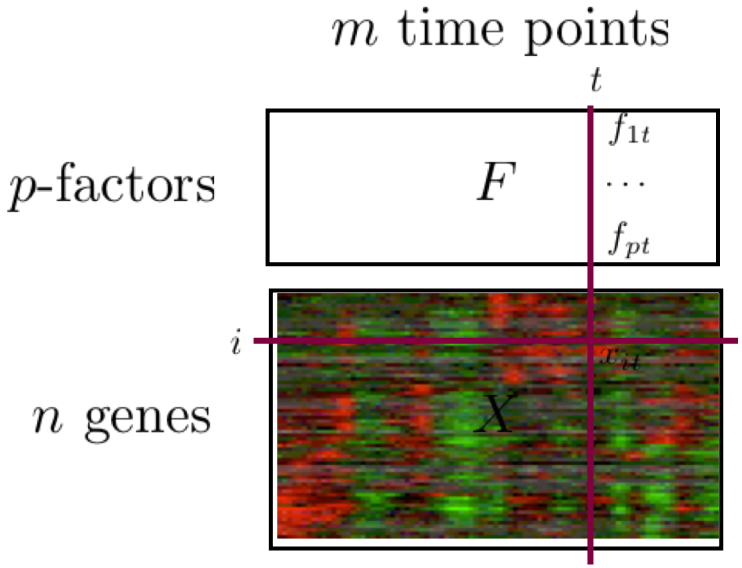
A mixture of trees: sampling model

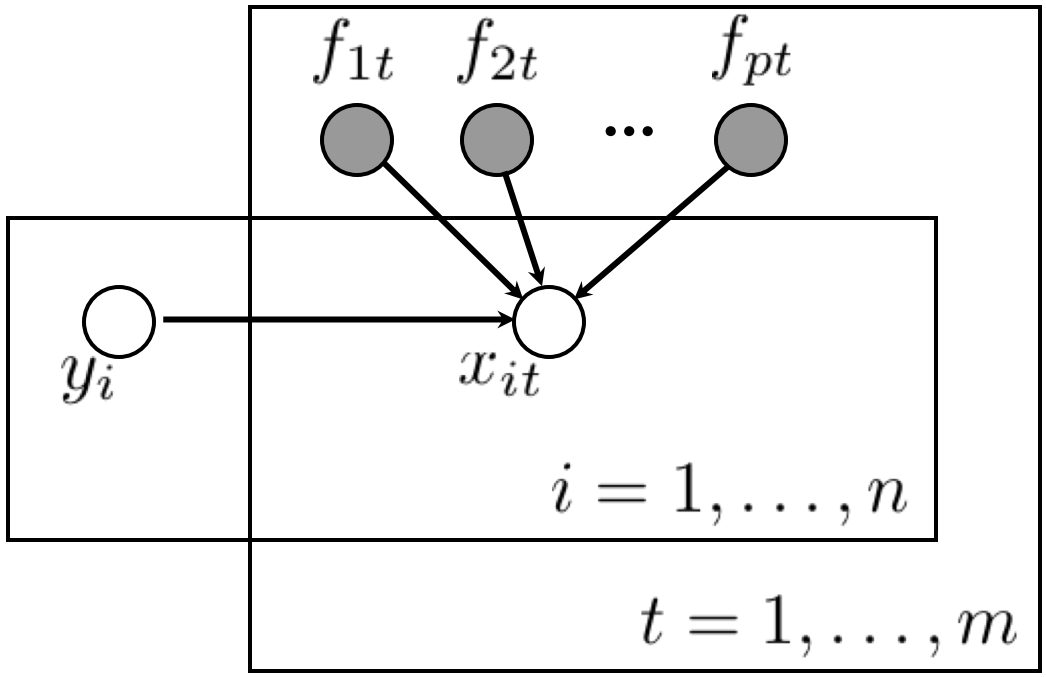




$$P(X|F,\Theta) = \prod_{i=1}^{n} \sum_{y_i=1}^{k} P(y_i) \prod_{t=1}^{m} P(x_{it}|f_{1t},\ldots,f_{pt},\Theta_{y_i})$$

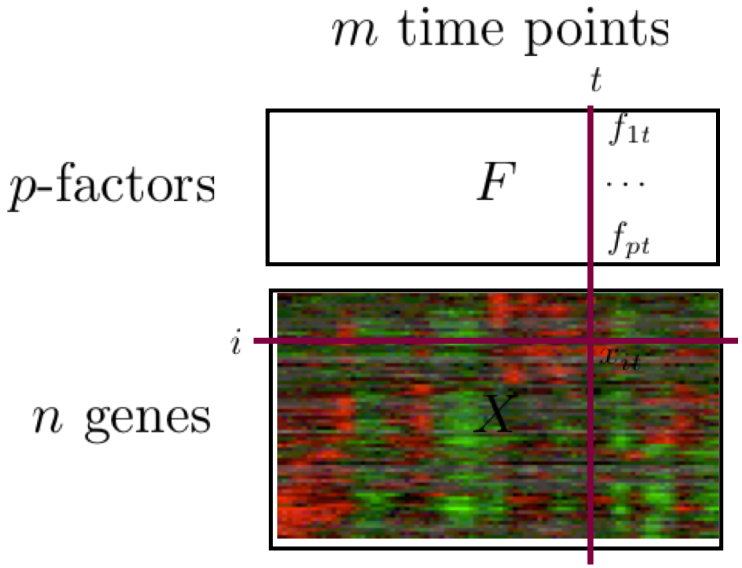
A mixture of trees: sampling model

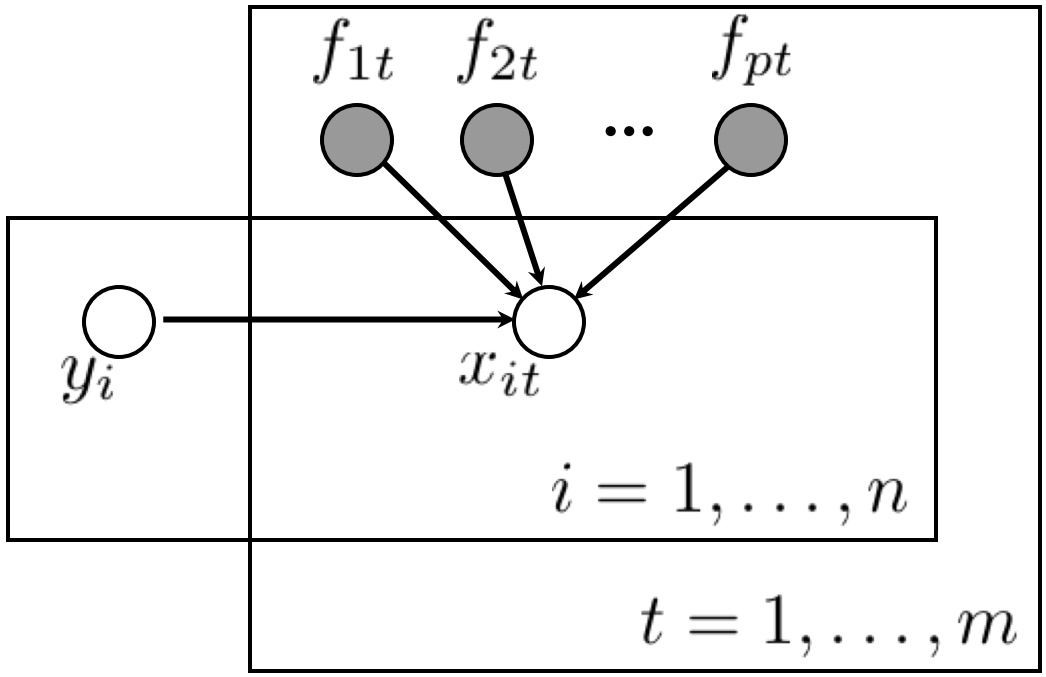




$$P(X|F,\Theta) = \prod_{i=1}^{n} \sum_{y_i=1}^{k} P(y_i) \prod_{t=1}^{m} P(x_{it}|f_{1t},\ldots,f_{pt},\Theta_{y_i})$$

A mixture of trees: sampling model

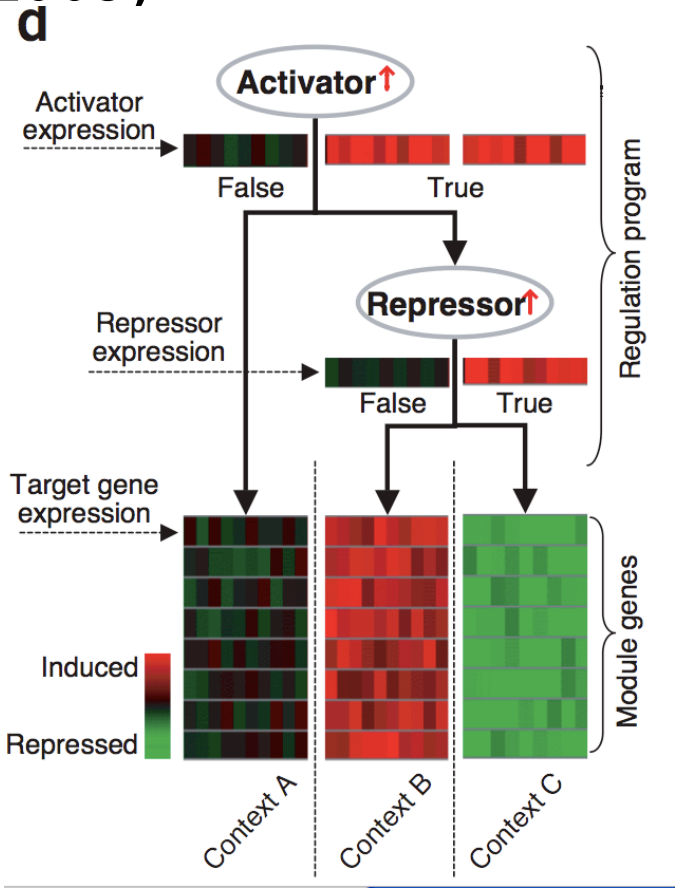




$$P(X|F,\Theta) = \prod_{i=1}^{n} \sum_{y_i=1}^{k} P(y_i) P(\underline{x}_i|F,\Theta_{y_i})$$

Finding expression programs

- We can jointly search for transcriptionally regulated subsets of genes and the corresponding programs (Segal et al., 2003)



EM algorithm

- We model the program assignments and expression profiles as

$$P(y_1, \ldots, y_n, \underline{x}_1, \ldots, \underline{x}_n | F, \Theta) = \prod_{i=1}^{n} P(y_i)\, P(\underline{x}_i | F, \Theta_{y_i})$$

- We can optimize the parameters of this model via the EM algorithm

(0) cluster genes into k clusters
(1) re-estimate regression trees for each cluster (program)
(2) (softly) re-assign each gene to the best program

$$p(y = j|i) \propto P(y) P(\underline{x}_i | F, \Theta_y)$$

Regression trees as programs

- Each regression tree can be represented as an "if-then-else" program

Cond. distribution $P(x|f_1, \ldots, f_p) = P(x|F)$

if $f_j \leq \tilde{f}_j$

$x \sim N(x; \mu_1, \sigma_1^2)$

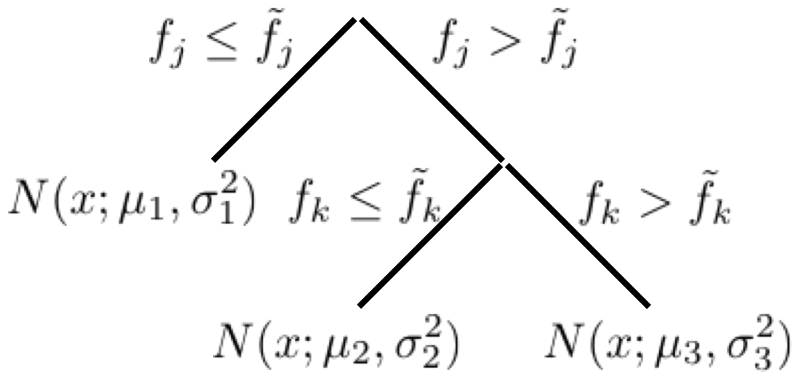
else

if $f_k \leq \tilde{f}_k$

$x \sim N(x; \mu_2, \sigma_2^2)$

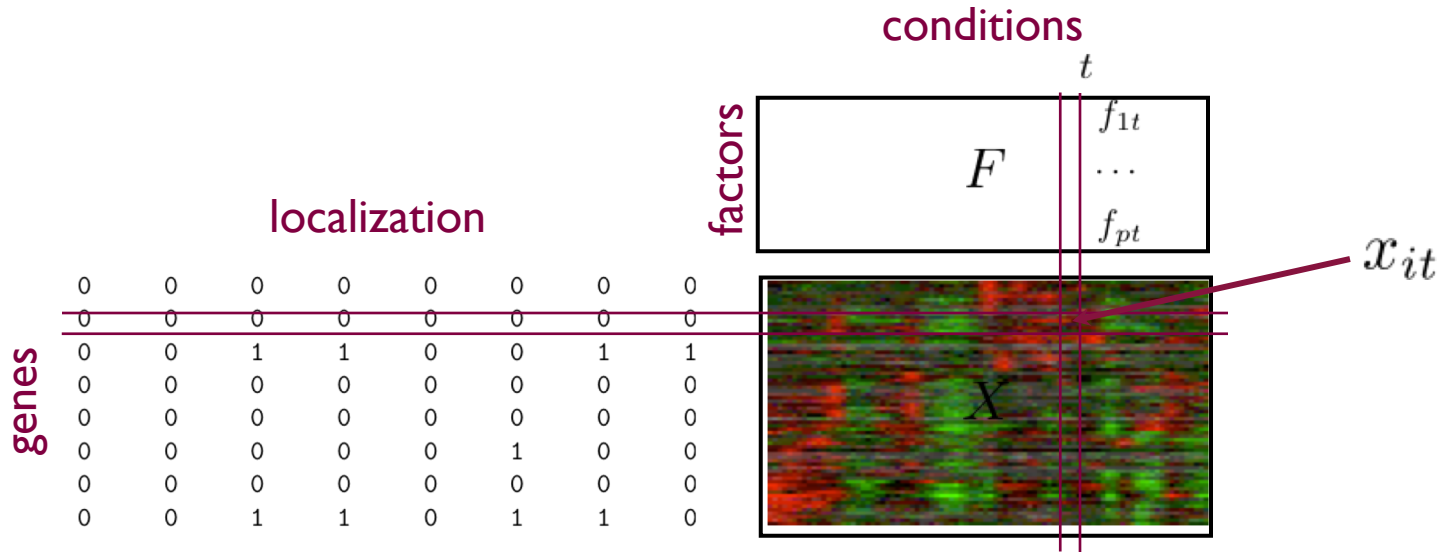
else

$x \sim N(x; \mu_3, \sigma_3^2)$



Extensions...

- The basic regression trees formulation permits us to estimate various types of "programs"
  - e,g, TF expression & localization, etc.



FIN - Thank You