Next Generation Sequencing, Tiling Arrays and Predictive Sequence Analysis for Transcriptome Analysis

Gunnar Rätsch

Friedrich Miescher Laboratory
Max Planck Society, Tübingen, Germany

9th Course in Bioinformatics and Systems Biology for Molecular Biologists (March 24, 2009)
Discovery of the Nuclein
(Friedrich Miescher, 1869)

Tübingen, around 1869

“...wants to assume that a single substance is the specific cause of fertilization, then one should undoubtedly first and foremost consider nuclein” (Miescher, 1874)
Introduction

Discovery of the Nuclein
(Friedrich Miescher, 1869)

From lymphocyte & salmon
“multi-basic acid” (≥ 4)

“...wants to assume that a single substance ... is the specific cause of fertilization, then one should undoubtedly first and foremost consider nuclein” (Miescher, 1874)
Transcriptome Analysis

What is encoded on the genome and how is it processed?

Then we can (try to) understand:

- Differences of active components between conditions/organisms?
- What changes when perturbing the biological system?

How to get the transcriptome?

- Infer transcriptome from genomic DNA
- Measure properties of transcriptome
- Combine predictions with measurements
Transcriptome Analysis

What is encoded on the genome and how is it processed?

Then we can (try to) understand:

- Differences of active components between conditions/organisms?
- What changes when perturbing the biological system?

How to get the transcriptome?

- Infer transcriptome from genomic DNA
- Measure properties of transcriptome
- Combine predictions with measurements
Transcriptome Analysis

What is encoded on the genome and how is it processed?

Then we can (try to) understand:

- Differences of active components between conditions/organisms?
- What changes when perturbing the biological system?

How to get the transcriptome?

1. Infer transcriptome from genomic DNA
2. Measure properties of transcriptome
3. Combine predictions with measurements
Transcriptome Analysis

What is encoded on the genome and how is it processed?

Then we can (try to) understand:

- Differences of active components between conditions/organisms?
- What changes when perturbing the biological system?

How to get the transcriptome?

1. Infer transcriptome from genomic DNA
2. Measure properties of transcriptome
3. Combine predictions with measurements
Transcriptome Analysis

What is encoded on the genome and how is it processed?

Then we can (try to) understand:

- Differences of active components between conditions/organisms?
- What changes when perturbing the biological system?

How to get the transcriptome?

1. Infer transcriptome from genomic DNA
2. Measure properties of transcriptome
3. Combine predictions with measurements
Transcriptome Analysis

What is encoded on the genome and how is it processed?

Then we can (try to) understand:

- Differences of active components between conditions/organisms?
- What changes when perturbing the biological system?

How to get the transcriptome?

1. Infer transcriptome from genomic DNA
2. Measure properties of transcriptome
3. Combine predictions with measurements
Newly synthesized pre-mRNA is capped.

[CBP20 & CBP80: cap-binding proteins]

Introns are spliced from pre-mRNA.

[U1, U2, U4-6: spliceosome SF1, U2AF, SR proteins: splicing factors]

A polyA-tail is added to the 3’ terminus of pre-mRNA.

[Bergkessel et al., 2009]
Introduction

Transcription & RNA Processing

- Newly synthesized pre-mRNA is capped.
  [CBP20 & CBP80: cap-binding proteins]

- Introns are spliced from pre-mRNA.
  [U1, U2, U4-6: spliceosome
   SF1, U2AF, SR proteins: splicing factors]

- A polyA-tail is added to the 3’ terminus of pre-mRNA.
  [Bergkessel et al., 2009]
Newly synthesized pre-mRNA is capped.

[CBP20 & CBP80: cap-binding proteins]

Introns are spliced from pre-mRNA.

[U1, U2, U4-6: spliceosome
SF1, U2AF, SR proteins: splicing factors]

A polyA-tail is added to the 3’ terminus of pre-mRNA.

[Bergkessel et al., 2009]
RNA Transcripts

- Protein-coding mRNAs
- Noncoding RNAs
  - Structural RNAs (e.g. rRNAs, tRNAs, ...)
  - Small RNAs (e.g. miRNAs, endogenous siRNAs, ...)
  - Antisense / promoter-associated transcripts
  - ...

© Gunnar Rätsch (FML, Tübingen)
Given a piece of DNA sequence

- Predict protein-coding mRNAs

- Less well developed for non-coding RNAs
Given a piece of DNA sequence
Predict protein-coding mRNAs
Less well developed for non-coding RNAs
DNA Microarrays

Oligonucleotide probes immobilized on a glass slide hybridize to complementary labeled target RNA.
DNA Microarrays

Oligonucleotide probes immobilized on a glass slide hybridize to complementary labeled target RNA.
Experimental Characterization of the Transcriptome

DNA Microarrays

Oligonucleotide probes immobilized on a glass slide hybridize to complementary labeled target RNA.

© Gunnar Rätsch (FML, Tübingen)  Methods for Transcriptome Analysis  Bertinoro Systems Biology 7 / 89
Key Research Questions

- Characterize an organism’s full complement of genes
  - Find new (possibly noncoding) genes
  - Compare genes among organisms

- Characterize transcript isoforms
  - Find new alternative splice forms / transcript ends

- Monitor transcriptome changes between tissues or in response to environmental changes (e.g. stress)
  - Identify significant expression changes

- Understand transcriptome regulation
  - Knock-out / knock-down analysis of regulators
    ~ Identify regulated targets with significant expression changes
  - Identify binding sites used in regulation (e.g. ChIP-on-chip)
Key Research Questions

- Characterize an organism’s full complement of genes
  - Find new (possibly noncoding) genes
  - Compare genes among organisms

- Characterize transcript isoforms
  - Find new alternative splice forms / transcript ends

- Monitor transcriptome changes between tissues or in response to environmental changes (e.g. stress)
  - Identify significant expression changes

- Understand transcriptome regulation
  - Knock-out / knock-down analysis of regulators
  - Identify regulated targets with significant expression changes
  - Identify binding sites used in regulation (e.g. ChIP-on-chip)
Key Research Questions

- Characterize an organism’s full complement of genes
  - Find new (possibly noncoding) genes
  - Compare genes among organisms

- Characterize transcript isoforms
  - Find new alternative splice forms / transcript ends

- Monitor transcriptome changes between tissues
  or in response to environmental changes (e.g. stress)
  - Identify significant expression changes

- Understand transcriptome regulation
  - Knock-out / knock-down analysis of regulators
    - Identify regulated targets with significant expression changes
  - Identify binding sites used in regulation (e.g. ChIP-on-chip)
Key Research Questions

- Characterize an organism’s full complement of genes
  ⇒ Find new (possibly noncoding) genes
  ⇒ Compare genes among organisms

- Characterize transcript isoforms
  ⇒ Find new alternative splice forms / transcript ends

- Monitor transcriptome changes between tissues
  or in response to environmental changes (e.g. stress)
  ⇒ Identify significant expression changes

- Understand transcriptome regulation
  ⇒ Knock-out / knock-down analysis of regulators
    ⇝ Identify regulated targets with significant expression changes
  ⇒ Identify binding sites used in regulation (e.g. ChIP-on-chip)
Roadmap

1. Computational Gene Finding
   - Identification of Genomic Signals
   - Learning to Predict mRNA Transcripts

2. Whole-genome Tiling Arrays
   - Technology and Limitations
   - Identification of Expression Differences
   - De Novo Transcript Discovery

3. Next-generation Sequencing
   - Technology & Limitations
   - Assembly & Read Mapping

4. Extensions
   - Quantification of Transcripts
   - ChIP-on-Chip Studies
Given a piece of DNA sequence
- Predict proteins (or non-coding RNAs)
Given a piece of DNA sequence

Predict the correct corresponding label sequence with labels “intergenic”, “exon”, “intron”, “5’ UTR”, etc.
Hidden Markov Models

Model sequence content:
- One state per segment type
- Allow only plausible transitions
- Content statistics at each state
- Derived from known genes

Prediction:
- Given DNA, find most likely state sequences

Focuses on “content”
- Weak models for “signals”
Hidden Markov Models

Model sequence content:
- One state per segment type
- Allow only plausible transitions
- Content statistics at each state
  - Derived from known genes

Prediction:
- Given DNA, find most likely state sequences
  - Focuses on “content”
  - Weak models for “signals”
Hidden Markov Models

Model sequence content:
- One state per segment type
- Allow only plausible transitions
- Content statistics at each state
  - Derived from known genes

Prediction:
- Given DNA, find most likely state sequences
- Focuses on “content”
- Weak models for “signals”
Hidden Markov Models

Model sequence content:
- One state per segment type
- Allow only plausible transitions
- Content statistics at each state
  - Derived from known genes

Prediction:
- Given DNA, find most likely state sequences

Focuses on “content”
- Weak models for “signals”

\[
p(x, y) = \prod_{i=1}^{L-1} p(x_i | y_i) p(y_{i+1} | y_i)
\]
Hidden Markov Models

Model sequence content:
- One state per segment type
- Allow only plausible transitions
- Content statistics at each state
  - Derived from known genes

Prediction:
- Given DNA, find most likely state sequences
  - Focuses on “content”
  - Weak models for “signals”

\[
p(x, y) = \prod_{i=1}^{L-1} p(x_i | y_i) p(y_{i+1} | y_i)
\]
Computational Gene Finding

Labeling the Genome

DNA

TSS

polyA/cleavage

pre-mRNA

Splice Donor

Splice Acceptor

Splice Donor

Splice Acceptor

mRNA

TIS

Stop

cap

polyA

Protein

© Gunnar Rätsch (FML, Tübingen)
Computational Gene Finding

Labeling the Genome

- DNA
  - TSS
  - TIS
  - Donor
  - Acceptor
  - Donor
  - Acceptor
  - polyA/cleavage

- pre-mRNA

- mRNA
  - cap
  - polyA

- Protein
Computational Gene Finding

Labeling the Genome

DNA

- TSS
- Donor
- Acceptor
- Donor
- Acceptor
- polyA/cleavage

pre-mRNA

- TIS
- Stop

mRNA

- cap
- polyA

Protein

TSS → TIS → Don → Acc → Stop → cleave
Example: Splice Site Recognition

- **True Splice Sites**: fixed window around a true splice site
- **Decoy sites**: all other consensus sites

⇒ Millions of labeled instances from EST databases
Example: Splice Site Recognition

True Splice Sites

CT...GTCGTA...GAAGCTAGGAGCGC...ACGCGT...GA

≈ 150 nucleotides window around dimer

- True sites: fixed window around a true splice site
- Decoy sites: all other consensus sites

Millions of labeled instances from EST databases

© Gunnar Rätsh (FML, Tübingen)
Example: Splice Site Recognition

Potential Splice Sites

CT...GTCGTA...GAAGCTAGGAGCGC...ACGCGT...GA

≈ 150 nucleotides window around dimer

True sites: fixed window around a true splice site
Decoy sites: all other consensus sites
Example: Splice Site Recognition

Potential Splice Sites

CT...GTCGTA...GAAGCTAGGAGCGC...ACGCGT...GA

≈ 150 nucleotides window around dimer

True sites: fixed window around a true splice site
Decoy sites: all other consensus sites

⇒ Millions of labeled instances from EST databases
Example: Splice Site Recognition

Potential Splice Sites

CT...GTCGTA...GAAGCTAGGAGCGC...ACGCGT...GA

≈ 150 nucleotides window around dimer

True sites: fixed window around a true splice site
Decoy sites: all other consensus sites

⇒ Millions of labeled instances from EST databases
Example: Splice Site Recognition

Basic idea:

For instance, exploit that exons have higher GC content or that specific motifs appear near splice sites.

[Sonnenburg et al., 2007]
Example: Splice Site Recognition

Potential Splice Sites

CT...GTCGTA...GAAGCTAGGAGCGC...ACGCCGT...GA

≈ 150 nucleotides window around dimer

Basic idea:

For instance, exploit that exons have higher GC content

or

that specific motifs appear near splice sites.

[Sonnenburg et al., 2007]
Example: Splice Site Recognition

CT...GTCGTA...GAAGCTAGGAGCGC...ACGCGT...GA

≈ 150 nucleotides window around dimer

Basic idea:

In practice: Use one feature per possible substring (e.g. ≤20) at all positions

150 \cdot (4^1 + \ldots + 4^{20}) \approx 2 \cdot 10^{14} \text{ features}

[Sonnenburg et al., 2007]
Results on Splice Site Recognition

<table>
<thead>
<tr>
<th></th>
<th>Worm</th>
<th>Fly</th>
<th>Cress</th>
<th>Fish</th>
<th>Human</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Acc</td>
<td>Don</td>
<td>Acc</td>
<td>Don</td>
<td>Acc</td>
</tr>
<tr>
<td>Markov Chain</td>
<td>92.1</td>
<td>90.0</td>
<td>80.3</td>
<td>78.5</td>
<td>87.4</td>
</tr>
<tr>
<td>auPRC(%)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SVM</td>
<td>95.9</td>
<td>95.3</td>
<td>86.7</td>
<td>87.5</td>
<td>92.2</td>
</tr>
<tr>
<td>auPRC(%)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

[Sonnenburg, Schweikert, Philips, Behr, Rätsch, 2007]

© Gunnar Rätsch (FML, Tübingen)
Example: Predictions in UCSC Browser

Schweikert et al., 2009
Example: Predictions in UCSC Browser

[Schweikert et al., 2009]
Computational Gene Finding
Identification of Genomic Signals

Example: Predictions in UCSC Browser

Based on known genes, learn how to combine predictions for accurate gene structure prediction

[Schweikert et al., 2009]
Discriminative Gene Prediction (simplified)

Simplified Model: Score for splice form $y = \{(p_j, q_j)\}_{j=1}^{J}$:

$$F(y) := \sum_{j=1}^{J-1} S_{GT}(f_{j}^{GT}) + \sum_{j=2}^{J} S_{AG}(f_{j}^{AG}) + \sum_{j=1}^{J-1} S_{L}\left(p_{j+1} - q_{j}\right) + \sum_{j=1}^{J} S_{LE}(q_{j} - p_{j})$$

<table>
<thead>
<tr>
<th>Splice signals</th>
<th>Segment lengths</th>
</tr>
</thead>
</table>

Tune free parameters (in functions $S_{GT}, S_{AG}, S_{LE}, S_{L_i}$) by solving linear program using training set with known splice forms.

[Rätsch, Sonnenburg, Srinivasan, Witte, Müller, Sommer, Schölkopf, 2007]
Simplified Model: Score for splice form $y = \{(p_j, q_j)\}_{j=1}^J$:

$$F(y) := \sum_{j=1}^{J-1} S_{GT}(f_j^{GT}) + \sum_{j=2}^J S_{AG}(f_j^{AG}) + \sum_{j=1}^{J-1} S_{Li}(p_{j+1} - q_j) + \sum_{j=1}^J S_{LE}(q_j - p_j)$$

- **Splice signals**
- **Segment lengths**

Tune free parameters (in functions $S_{GT}, S_{AG}, S_{LE}, S_{Li}$) by solving **linear program** using training set with known splice forms
Results using mGene

- Most accurate \textit{ab initio} method in the nGASP genome annotation challenge (\textit{C. elegans}) [Coghlan et al., 2008]

- Validation of gene predictions for \textit{C. elegans}: [Schweikert et al., 2009]

<table>
<thead>
<tr>
<th></th>
<th>No. of genes</th>
<th>No. of genes analyzed</th>
<th>Frac. of genes w/ expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>New genes</td>
<td>2,197</td>
<td>57</td>
<td>≈ 42%</td>
</tr>
<tr>
<td>Missing unconf. genes</td>
<td>205</td>
<td>24</td>
<td>≈ 8%</td>
</tr>
</tbody>
</table>

- Annotation of other nematode genomes: [Schweikert et al., 2009]

<table>
<thead>
<tr>
<th>Genome</th>
<th>Genome size [Mbp]</th>
<th>No. of genes</th>
<th>No. exons/gene (mean)</th>
<th>mGene accuracy</th>
<th>best other accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>\textit{C. remanei}</td>
<td>235.94</td>
<td>31503</td>
<td>5.7</td>
<td>96.6%</td>
<td>93.8%</td>
</tr>
<tr>
<td>\textit{C. japonica}</td>
<td>266.90</td>
<td>20121</td>
<td>5.3</td>
<td>93.3%</td>
<td>88.7%</td>
</tr>
<tr>
<td>\textit{C. brenneri}</td>
<td>453.09</td>
<td>41129</td>
<td>5.4</td>
<td>93.1%</td>
<td>87.8%</td>
</tr>
<tr>
<td>\textit{C. briggsae}</td>
<td>108.48</td>
<td>22542</td>
<td>6.0</td>
<td>87.0%</td>
<td>82.0%</td>
</tr>
</tbody>
</table>
Results using mGene

- Most accurate *ab initio* method in the nGASP genome annotation challenge (*C. elegans*)  
  
[Coghlan et al., 2008]

- Validation of gene predictions for *C. elegans*:  
  
[Schweikert et al., 2009]

<table>
<thead>
<tr>
<th></th>
<th>No. of genes</th>
<th>No. of genes analyzed</th>
<th>Frac. of genes w/ expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>New genes</td>
<td>2,197</td>
<td>57</td>
<td>∼ 42%</td>
</tr>
<tr>
<td>Missing unconf. genes</td>
<td>205</td>
<td>24</td>
<td>∼ 8%</td>
</tr>
</tbody>
</table>

- Annotation of other nematode genomes:  
  
[Schweikert et al., 2009]

<table>
<thead>
<tr>
<th>Genome</th>
<th>Genome size [Mbp]</th>
<th>No. of genes</th>
<th>No. exons/gene (mean)</th>
<th>mGene accuracy</th>
<th>best other accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>C. remanei</em></td>
<td>235.94</td>
<td>31503</td>
<td>5.7</td>
<td>96.6%</td>
<td>93.8%</td>
</tr>
<tr>
<td><em>C. japonica</em></td>
<td>266.90</td>
<td>20121</td>
<td>5.3</td>
<td>93.3%</td>
<td>88.7%</td>
</tr>
<tr>
<td><em>C. brenneri</em></td>
<td>453.09</td>
<td>41129</td>
<td>5.4</td>
<td>93.1%</td>
<td>87.8%</td>
</tr>
<tr>
<td><em>C. briggsae</em></td>
<td>108.48</td>
<td>22542</td>
<td>6.0</td>
<td>87.0%</td>
<td>82.0%</td>
</tr>
</tbody>
</table>
Results using mGene

- Most accurate *ab initio* method in the nGASP genome annotation challenge (*C. elegans*) [Coghlan et al., 2008]

- Validation of gene predictions for *C. elegans*: [Schweikert et al., 2009]

<table>
<thead>
<tr>
<th></th>
<th>No. of genes</th>
<th>No. of genes analyzed</th>
<th>Frac. of genes w/ expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>New genes</td>
<td>2,197</td>
<td>57</td>
<td>≈ 42%</td>
</tr>
<tr>
<td>Missing unconf. genes</td>
<td>205</td>
<td>24</td>
<td>≈ 8%</td>
</tr>
</tbody>
</table>

- Annotation of other nematode genomes: [Schweikert et al., 2009]

<table>
<thead>
<tr>
<th>Genome</th>
<th>Genome size [Mbp]</th>
<th>No. of genes</th>
<th>No. exons/gene (mean)</th>
<th>mGene accuracy</th>
<th>best other accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>C. remanei</em></td>
<td>235.94</td>
<td>31503</td>
<td>5.7</td>
<td>96.6%</td>
<td>93.8%</td>
</tr>
<tr>
<td><em>C. japonica</em></td>
<td>266.90</td>
<td>20121</td>
<td>5.3</td>
<td>93.3%</td>
<td>88.7%</td>
</tr>
<tr>
<td><em>C. brenneri</em></td>
<td>453.09</td>
<td>41129</td>
<td>5.4</td>
<td>93.1%</td>
<td>87.8%</td>
</tr>
<tr>
<td><em>C. briggsae</em></td>
<td>108.48</td>
<td>22542</td>
<td>6.0</td>
<td>87.0%</td>
<td>82.0%</td>
</tr>
</tbody>
</table>
mGene.web: Gene Finding for Everybody ;-)  
(Schweikert et al., 2009)

http://mgene.org/webservice
Gene finding accuracy still far from perfect

- Misses genes, predicts incorrect gene models
- Does not (yet) predict alternative transcripts
- Cannot predict when transcripts are expressed/modified/degraded...

Need experimental data for condition specific transcriptomes.

Then we can learn to predict (hopefully).
Limitations/Extensions

Gene finding accuracy still far from perfect
- Misses genes, predicts incorrect gene models
- Does not (yet) predict alternative transcripts
- Cannot predict when transcripts are expressed/modified/degraded . . .

Need experimental data for condition specific transcriptomes.

Then we can learn to predict (hopefully).
Limitations/Extensions

Gene finding accuracy still far from perfect

- Misses genes, predicts incorrect gene models
- Does not (yet) predict alternative transcripts
- Cannot predict when transcripts are expressed/modified/degraded...

Need experimental data for condition specific transcriptomes.

Then we can learn to predict (hopefully).
1. Computational Gene Finding
   - Identification of Genomic Signals
   - Learning to Predict mRNA Transcripts

2. Whole-genome Tiling Arrays
   - Technology and Limitations
   - Identification of Expression Differences
   - De Novo Transcript Discovery

3. Next-generation Sequencing
   - Technology & Limitations
   - Assembly & Read Mapping

4. Extensions
   - Quantification of Transcripts
   - ChIP-on-Chip Studies
Directly measure the transcriptome

- Whole genome tiling arrays
- Transcriptome sequencing (Sanger or Next Generation Sequencing)
Directly measure the transcriptome

- Whole genome tiling arrays
- Transcriptome sequencing (Sanger or Next Generation Sequencing)
Whole-genome Tiling Arrays

- Whole-genome, quantitative measurements of expression
- Allows to cost-effectively analyze many conditions (replicates)
- Hybridization data is noisy, analysis challenging
- Variants: exon arrays

see Mockler et al. [2005], Yazaki et al. [2007] for comprehensive reviews
Whole-genome Tiling Arrays

- Whole-genome, quantitative measurements of expression
- Allows to cost-effectively analyze many conditions (replicates)
- Hybridization data is noisy, analysis challenging
- Variants: exon arrays,

see Mockler et al. [2005], Yazaki et al. [2007] for comprehensive reviews
Whole-genome, quantitative measurements of expression
Allows to cost-effectively analyze many conditions (replicates)
Hybridization data is noisy, analysis challenging
Variants: exon arrays, exon junction arrays

see Mockler et al. [2005], Yazaki et al. [2007] for comprehensive reviews
Whole-genome Tiling Arrays

- Whole-genome, quantitative measurements of expression
- Allows to cost-effectively analyze many conditions (replicates)
- Hybridization data is noisy, analysis challenging
- Variants: exon arrays, exon junction arrays

see Mockler et al. [2005], Yazaki et al. [2007] for comprehensive reviews
Whole-genome, quantitative measurements of expression

Allows to cost-effectively analyze many conditions (replicates)

Hybridization data is noisy, analysis challenging

Variants: exon arrays, exon junction arrays

see Mockler et al. [2005], Yazaki et al. [2007] for comprehensive reviews
Repeats cause **cross-hybridization**

\[ \text{Hybridizing mRNA transcript} \]

\[ \text{Hybridization intensity} \]

\[ \Rightarrow \text{Discard tiling probes with high sequence similarity to } >1 \text{ location in the genome} \]
Repeats cause **cross-hybridization**

⇒ Discard tiling probes with high sequence similarity to $>1$ location in the genome
Repeats cause **cross-hybridization**

⇒ Discard tiling probes with high sequence similarity to >1 location in the genome
Repeats cause **cross-hybridization**

⇒ Discard tiling probes with high sequence similarity to >1 location in the genome
Hybridization intensity exhibits a **probe sequence bias**

Sequence-normalization approaches:
- Rescaling by probe GC content [Samanta et al., 2006]
- Rescaling using genomic DNA hybridization [David et al., 2006].

\[
\displaystyle n_{ij} = \frac{x_{ij} - b_j(y_i)}{y_i}
\]

for probe \(i\) on replicate array \(j\) with RNA hybridization signal \(x_{ij}\) to obtain normalized signal \(n_{ij}\); DNA hybridization signal \(y_i\) is transformed into RNA background signal by \(b_j\) estimated from intergenic probes.

- Regression techniques [Royce et al., 2007b, Zeller et al., 2008c]
Hybridization intensity exhibits a \textbf{probe sequence bias}

\begin{itemize}
  \item Sequence-normalization approaches:
    \begin{itemize}
      \item Rescaling by probe GC content \cite{Samanta et al., 2006}
      \item Rescaling using genomic DNA hybridization \cite{David et al., 2006}.
    \end{itemize}
  \item Regression techniques \cite{Royce et al., 2007b, Zeller et al., 2008c}
\end{itemize}

\[ n_{ij} = \frac{x_{ij} - b_j(y_i)}{y_i} \]

for probe $i$ on replicate array $j$ with RNA hybridization signal $x_{ij}$ to obtain normalized signal $n_{ij}$; DNA hybridization signal $y_i$ is transformed into RNA background signal by $b_j$ estimated from intergenic probes.
Hybridization intensity exhibits a probe sequence bias

Sequence-normalization approaches:

- Rescaling by probe GC content [Samanta et al., 2006]
- Rescaling using genomic DNA hybridization [David et al., 2006].

\[ n_{ij} = \frac{x_{ij} - b_j(y_i)}{y_i} \]

for probe \( i \) on replicate array \( j \) with RNA hybridization signal \( x_{ij} \) to obtain normalized signal \( n_{ij} \); DNA hybridization signal \( y_i \) is transformed into RNA background signal by \( b_j \) estimated from intergenic probes.

- Regression techniques [Royce et al., 2007b, Zeller et al., 2008c]

© Gunnar Rätsch (FML, Tübingen)
Transcript normalization assumes constant transcript intensities $\bar{y}_i$ (median estimate) \cite{Zeller2008c}.

- Learns intensity deviation from transcript intensity $\delta_i := y_i - \bar{y}_i$.
- Takes probe sequence $x_i$ (positional information on mono-, di- and tri-mer occurrence) as input for regression. Models probe sequence effect depending on $y_i$: $f(x_i, y_i) \approx \delta_i$.
**Tiling Array Analysis Challenges (III)**

- **Transcript normalization** assumes constant transcript intensities $\bar{y}_i$ (median estimate) [Zeller et al., 2008c]

- Learns intensity deviation from transcript intensity $\delta_i := y_i - \bar{y}_i$

- Takes probe sequence $x_i$ (positional information on mono-, di- and tri-mer occurrence) as input for regression.

Mathematically, models probe sequence effect depending on $y_i$: $f(x_i, y_i) \approx \delta_i$

![Graph showing observed intensity, transcript intensity, and fold difference δ between observed and transcript intensity.](image)
**Tiling Array Analysis Challenges (III)**

- **Transcript normalization** assumes constant transcript intensities $\bar{y}_i$ (median estimate) [Zeller et al., 2008c]
- Learns intensity deviation from transcript intensity $\delta_i := y_i - \bar{y}_i$
- Takes probe sequence $x_i$ (positional information on mono-, di- and tri-mer occurrence) as input for regression.

Models probe sequence effect depending on $y_i$: $f(x_i, y_i) \approx \delta_i$

---

![Graph showing log-intensity vs. observed intensity, annotated exonic, annotated intronic, and transcript intensity with fold difference $\delta$ between observed and transcript intensity.](image)
Transcript normalization assumes constant transcript intensities $\bar{y}_i$ (median estimate) \cite{Zeller2008}

- Learns intensity deviation from transcript intensity $\delta_i := y_i - \bar{y}_i$
- Takes probe sequence $\mathbf{x}_i$ (positional information on mono-, di- and tri-mer occurrence) as input for regression.

Models probe sequence effect depending on $y_i$: $f(\mathbf{x}_i, y_i) \approx \delta_i$

Discretize $y$ into $Q = 20$ quantiles and estimate $Q$ independent functions $f_1(\mathbf{x}), \ldots, f_Q(\mathbf{x})$.

Linear regression

$$f_q(\mathbf{x}) = \mathbf{w}_q^T \mathbf{x}$$
Identification of Expression Changes

1. Map tiling probes to annotated transcripts (define probe sets)
2. Use standard microarray tools to analyze gene expression

Gene expression values are typically computed using robust “summarization methods” that account for probe noise [e.g. Irizarry et al., 2003]

Significant expression changes are typically identified with a statistical test. Results have to be corrected for multiple testing [e.g. Storey and Tibshirani, 2003]

Advantages of tiling arrays:
- Annotations change, only remapping is needed to obtain expression measurements for the latest annotation.
- Expression can be measured per exon, not only per gene.
- Expression can be measured for introns (detect retention).
Identification of Expression Changes

1. Map tiling probes to annotated transcripts (define probe sets)
2. Use standard microarray tools to analyze gene expression

Gene expression values are typically computed using robust “summarization methods” that account for probe noise [e.g. Irizarry et al., 2003]

Significant expression changes are typically identified with a statistical test. Results have to be corrected for multiple testing [e.g. Storey and Tibshirani, 2003]

Advantages of tiling arrays:

- Annotations change, only remapping is needed to obtain expression measurements for the latest annotation.
- Expression can be measured per exon, not only per gene.
- Expression can be measured for introns (⇒ detect retention).
**Goal:** Identify exon/intron segments that are *differentially* spliced in the analyzed samples.
**Detection of Alternative Transcripts (II)**

**Goal:** Identify exon/intron segments that show different intensities than *other exons/introns* in at least one analyzed sample.
Detecting Alternative Exons

Fit a gene expression model to exon array data [Irizarry et al., 2003]:

\[ x_{ik} = g_k + p_i + \epsilon_{ik} \]

- RNA hybridization signal \( x_{ij} \)
- \( g_k \) gene-wide expression effect in sample \( k \)
- \( p_i \) effect of probe \( i \), error terms \( \epsilon_{ik} \).

Detect alternatively spliced exons as outliers [Purdom et al., 2008] from large residuals \( \epsilon_{i'k'} \) for alternative exon probes \( i' \) in sample \( k' \).

Test exon junction probes for different transcript isoforms for differential expression using e.g. the Kruskal Wallis test [Sugnet et al., 2006].

More sophisticated methods use classification techniques.

[Eichner, 2008, Eichner et al., 2009]
Detecting Alternative Exons

Fit a gene expression model to **exon array data** [Irizarry et al., 2003]:

\[ x_{ik} = g_k + p_i + \epsilon_{ik} \]

- RNA hybridization signal \( x_{ij} \)
- \( g_k \) gene-wide expression effect in sample \( k \)
- \( p_i \) effect of probe \( i \)
- error terms \( \epsilon_{ik} \).

Detect alternatively spliced exons as outliers [Purdom et al., 2008] from large residuals \( \epsilon_{i'k'} \) for alternative exon probes \( i' \) in sample \( k' \).

Test **exon junction probes** for different transcript isoforms for differential expression using e.g. the Kruskal Wallis test [Sugnet et al., 2006].

More sophisticated methods use classification techniques.

[Eichner, 2008, Eichner et al., 2009]
Detecting Alternative Exons

Fit a gene expression model to exon array data [Irizarry et al., 2003]:

\[ x_{ik} = g_k + p_i + \epsilon_{ik} \]

- RNA hybridization signal \( x_{ij} \),
- \( g_k \) gene-wide expression effect in sample \( k \),
- \( p_i \) effect of probe \( i \), error terms \( \epsilon_{ik} \).

Detect alternatively spliced exons as outliers [Purdom et al., 2008] from large residuals \( \epsilon_{i'k'} \) for alternative exon probes \( i' \) in sample \( k' \).

Test exon junction probes for different transcript isoforms for differential expression using e.g. the Kruskal Wallis test [Sugnet et al., 2006].

More sophisticated methods use classification techniques.

[Eichner, 2008, Eichner et al., 2009]
De novo transcript identification is needed to re-annotate expressed genes.
De novo segmentation is needed to re-annotate expressed genes.
De novo segmentation is needed to re-annotate expressed genes.
Identify “positive probes” in local neighborhood. Smooth data locally, across replicates (Pseudomedian\(^1\) [Royce et al., 2007a])

Two approaches:

- define an *ad hoc* threshold on smoothed signal intensity (e.g. 90th signal percentile) [Kampa et al., 2004]
- estimate a threshold from negative bacterial control probes to adjust an empirical false discovery rate [He et al., 2007]

Combine positive probes into “transfrags” in case of a run of consecutive positive probes (minRun) interrupted by a limited number of negative probes (maxGap) [Bertone et al., 2004, Kampa et al., 2004]

Problem: Manual parameter “tuning”

\(^1\)median of all pairwise averages of probe signals within a sliding window
1. Identify “positive probes” in local neighborhood. Smooth data locally, across replicates (Pseudomedian\(^1\) [Royce et al., 2007a]).

Two approaches:

- Define an *ad hoc* threshold on smoothed signal intensity (e.g., 90th signal percentile) [Kampa et al., 2004].
- Estimate a threshold from negative bacterial control probes to adjust an empirical false discovery rate [He et al., 2007].

2. Combine positive probes into “transfrags” in case of a run of consecutive positive probes (minRun) interrupted by a limited number of negative probes (maxGap) [Bertone et al., 2004, Kampa et al., 2004].

**Problem:** Manual parameter “tuning”

\(^1\)Median of all pairwise averages of probe signals within a sliding window
Identify “positive probes” in local neighborhood. Smooth data locally, across replicates (Pseudomedian\(^1\) [Royce et al., 2007a])

Two approaches:

- define an *ad hoc* threshold on smoothed signal intensity (e.g. 90th signal percentile) [Kampa et al., 2004]
- estimate a threshold from negative bacterial control probes to adjust an empirical false discovery rate [He et al., 2007]

Combine positive probes into “transfrags” in case of a run of consecutive positive probes (minRun) interrupted by a limited number of negative probes (maxGap) [Bertone et al., 2004, Kampa et al., 2004]

**Problem:** Manual parameter “tuning”

\(^1\)median of all pairwise averages of probe signals within a sliding window
Dynamic Programming Segmentation

Model intensities as piecewise constant function [Huber et al., 2006]:

\[ x_{ij} = \mu_s + \epsilon_{ij} \]

for probe \( i \) on replicate array \( j \) with RNA hybridization signal \( x_{ij} \); \( t_s \) and \( t_{s+1} \), \( \mu_s \) is the mean signal of the \( s \)th segment, \( \epsilon_{ij} \) error terms.

Minimize the sum of squared residuals:

\[ G(t_1, \ldots, t_S) = \sum_{s=1}^S \sum_{j=1}^R \sum_{i=t_s}^{t_{s+1}-1} (x_{ij} - \mu_s)^2 \]

where \( S \) is the number of segments and \( R \) the number of replicates.

The optimal segmentation can be computed in \( O(n^2) \) time using dynamic programming [Huber et al., 2006].

Problem: S is to be user-specified
Dynamic Programming Segmentation

Model intensities as piecewise constant function [Huber et al., 2006]:

\[ x_{ij} = \mu_s + \epsilon_{ij} \]

for probe \( i \) on replicate array \( j \) with RNA hybridization signal \( x_{ij} \); segment boundaries \( t_s \) and \( t_{s+1} \). \( \mu_s \) is the mean signal of the \( s \)th segment, \( \epsilon_{ij} \) error terms.

Minimize the sum of squared residuals:

\[
G(t_1, \ldots, t_S) = \sum_{s=1}^{S} \sum_{j=1}^{R} \sum_{i=t_s}^{t_{s+1}-1} (x_{ij} - \mu_s)^2
\]

where \( S \) is the number of segments and \( R \) the number of replicates.

The optimal segmentation can be computed in \( O(n^2) \) time using dynamic programming [Huber et al., 2006].

Problem: \( S \) is to be user-specified.
Dynamic Programming Segmentation

Model intensities as piecewise constant function [Huber et al., 2006]:

\[ x_{ij} = \mu_s + \epsilon_{ij} \]

for probe \( i \) on replicate array \( j \) with RNA hybridization signal \( x_{ij} \); segment boundaries \( t_s \) and \( t_{s+1} \). \( \mu_s \) is the mean signal of the \( s \)th segment, \( \epsilon_{ij} \) error terms.

Minimize the sum of squared residuals:

\[ G(t_1, \ldots, t_S) = \sum_{s=1}^{S} \sum_{j=1}^{R} \sum_{i=t_s}^{t_{s+1}-1} (x_{ij} - \mu_s)^2 \]

where \( S \) is the number of segments and \( R \) the number of replicates.

The optimal segmentation can be computed in \( O(n^2) \) time using dynamic programming [Huber et al., 2006].

Problem: \( S \) is to be user-specified.
Dynamic Programming Segmentation

Model intensities as piecewise constant function [Huber et al., 2006]:

\[ x_{ij} = \mu_s + \epsilon_{ij} \]

for probe \( i \) on replicate array \( j \) with RNA hybridization signal \( x_{ij} \); segment boundaries \( t_s \) and \( t_{s+1} \). \( \mu_s \) is the mean signal of the \( s \)th segment, \( \epsilon_{ij} \) error terms.

Minimize the sum of squared residuals:

\[ G(t_1, \ldots, t_S) = \sum_{s=1}^{S} \sum_{j=1}^{R} \sum_{i=t_s}^{t_{s+1}-1} (x_{ij} - \mu_s)^2 \]

where \( S \) is the number of segments and \( R \) the number of replicates.

The optimal segmentation can be computed in \( O(n^2) \) time using dynamic programming [Huber et al., 2006].

Problem: \( S \) is to be user-specified
Hidden Markov Models

- Learn to label each probe given its hybridization signal and local context
  
  [Ji and Wong, 2005a, Du et al., 2006]

- Train transition and emission probabilities on annotated genes

- Explicit intron model [Zeller et al., 2008c]

- Q discrete expression levels [Zeller et al., 2008c]
Hidden Markov Models

- Learn to label each probe given its hybridization signal and local context
  [Ji and Wong, 2005a, Du et al., 2006]

- Train transition and emission probabilities on annotated genes

- Explicit intron model [Zeller et al., 2008c]

- Q discrete expression levels
  [Zeller et al., 2008c]
Hidden Markov Models

- Learn to label each probe given its hybridization signal and local context
  [Ji and Wong, 2005a, Du et al., 2006]
- Train transition and emission probabilities on annotated genes
- Explicit intron model [Zeller et al., 2008c]
- $Q$ discrete expression levels
  [Zeller et al., 2008c]
Hidden Markov Models

- Learn to label each probe given its hybridization signal and local context
  [Ji and Wong, 2005a, Du et al., 2006]
- Train transition and emission probabilities on annotated genes
- Explicit intron model [Zeller et al., 2008c]
- $Q$ discrete expression levels
  [Zeller et al., 2008c]
Parametrization and Decoding

Log transition probabilities $\phi(k, l)$

between states $k$ and $l$

Log emission probabilities $g_k(x)$ in state $k$

for (discretized) hybridization signal $x$

Scoring a sequence of hybridization signals $x$

with a given labeling $\pi$ and parametrization $\theta$:

$$F_\theta(x, \pi) = \sum_{p=1}^{\mid \pi \mid} g_{\pi_p}(x_p) + \phi(\pi_{p-1}, \pi_p)$$

$S$ denotes set of states, $\mid \pi \mid$ the length of $\pi$

Decoding to obtain the best-scoring labeling for $x$:

$$\arg\max_{\pi} F_\theta(x, \pi)$$

(Viterbi decoding [Durbin et al., 1998])
Parametrization and Decoding

Log transition probabilities $\phi(k, l)$ between states $k$ and $l$

Log emission probabilities $g_k(x)$ in state $k$ for (discretized) hybridization signal $x$

**Scoring a sequence** of hybridization signals $x$ with a given labeling $\pi$ and parametrization $\theta$:

$$ F_\theta(x, \pi) = \sum_{p=1}^{\left|\pi\right|} g_{\pi_p}(x_p) + \phi(\pi_{p-1}, \pi_p) $$

$S$ denotes set of states, $|\pi|$ the length of $\pi$

**Decoding** to obtain the best-scoring labeling for $x$:

$$ \arg\max_{\pi} F_\theta(x, \pi) $$ (Viterbi decoding [Durbin et al., 1998])
Parametrization and Decoding

Log transition probabilities $\phi(k, l)$ between states $k$ and $l$

Log emission probabilities $g_k(x)$ in state $k$ for (discretized) hybridization signal $x$

Scoring a sequence of hybridization signals $x$ with a given labeling $\pi$ and parametrization $\theta$:

$$F_\theta(x, \pi) = \sum_{p=1}^{\vert \pi \vert} g_{\pi_p}(x_p) + \phi(\pi_{p-1}, \pi_p)$$

$S$ denotes set of states, $\vert \pi \vert$ the length of $\pi$

Decoding to obtain the best-scoring labeling for $x$:

$$\arg\max_{\pi} F_\theta(x, \pi) \quad \text{(Viterbi decoding [Durbin et al., 1998])}$$
Training an HMM

Training sequences: signals $x^i$ and labels $\pi^i$ for $i = 1, \ldots, n$.

Log transition probabilities [Durbin et al., 1998]:
\[
\phi(k, l) = \log\left(\frac{A_{k,l}}{\sum_{l'} A_{k,l'}}\right)
\]
for all state pairs $(k, l) \in S^2$

counting observed transitions:
\[
A_{k,l} = \sum_{i=1}^n \sum_{p=1}^{\left|\pi^i\right|} \left[\pi^i_p = k \land \pi^i_{p+1} = l\right]
\]

Log emission probabilities [Durbin et al., 1998]

for piece-wise constant $g_k$ with $L$ levels (ranging from $t_l$ to $t_{l+1}$):
\[
g_{k,l} = \log\left(\frac{E_l}{\sum_{l'} E_{l'}}\right)
\]

counting discrete signal values:
\[
E_l = \sum_{i=1}^n \sum_{p=1}^{\left|\pi^i\right|} \left[\pi^i_p = k \land t_l < x^i_p \leq t_{l+1}\right]
\]

HMMs can also be (re-)trained in an unsupervised fashion
[e.g. Munch et al., 2006]
Hidden Markov SVMs

Enforce a **large margin** (cf. gene finding) between the correct one $\pi^{(i)}$ and any other labeling $\overline{\pi} \neq \pi^{(i)}$:

$$F_{\theta}(x^{(i)}, \pi^{(i)}) - F_{\theta}(x^{(i)}, \overline{\pi}) \gg 0 \quad \forall \overline{\pi} \neq \pi^{(i)} \quad \forall i = 1, \ldots, n$$
Method Comparison

Recall: Proportion of annotated exons/introns covered by predictions.
Precision: Proportion of predictions covered by annotated exons/introns.
Apply statistical test for significant expression change to signal from transcriptionally active regions (TARs) defined by previous segmentation [Zeller et al., 2009].
Roadmap

1. Computational Gene Finding
   - Identification of Genomic Signals
   - Learning to Predict mRNA Transcripts

2. Whole-genome Tiling Arrays
   - Technology and Limitations
   - Identification of Expression Differences
   - De Novo Transcript Discovery

3. Next-generation Sequencing
   - Technology & Limitations
   - Assembly & Read Mapping

4. Extensions
   - Quantification of Transcripts
   - ChIP-on-Chip Studies
Applications of DNA/RNA sequencing:

- *De novo* genome sequencing
- Genome resequencing
- Transcriptome sequencing
- Methylation analysis

Sequencing Technology

- Capillary/Sanger sequencing
- Pyrosequencing (Roche/454)
- SOLiD sequencing (ABI)
- Flow cell sequencing (Illumina)
- Single molecule sequencing (Nano pores, etc.)
Sequencing Techniques & Applications

Applications of DNA/RNA sequencing:
- De novo genome sequencing
- Genome resequencing
- Transcriptome sequencing
- Methylation analysis

Sequencing Technology
- Capillary/Sanger sequencing
  - Pyrosequencing (Roche/454)
  - SOLiD sequencing (ABI)
  - Flow cell sequencing (Illumina)
  - Single molecule sequencing (Nano pores, etc.)
Sequencing Techniques & Applications

Applications of DNA/RNA sequencing:
- *De novo* genome sequencing
- Genome resequencing
- Transcriptome sequencing
- Methylation analysis

Sequencing Technology
- **Capillary/Sanger sequencing**
- Pyrosequencing (Roche/454)
- SOLiD sequencing (ABI)
- Flow cell sequencing (Illumina)
- Single molecule sequencing (Nano pores, etc.)
Applications of DNA/RNA sequencing:

- De novo genome sequencing
- Genome resequencing
- Transcriptome sequencing
- Methylation analysis

Sequencing Technology

- Capillary/Sanger sequencing
- Pyrosequencing (Roche/454)
- SOLiD sequencing (ABI)
- Flow cell sequencing (Illumina)
- Single molecule sequencing (Nano pores, etc.)
Illumina Sequencing

- Solexa released a sequencing machine in 2006
- Fragment sizes from 28 – 75
- Probes are fixed to a glass plate “flow cell”
- Reagents are directed through flow cell

(see Movie)
Illumina Sequencing

- Flow cell preparation
- Bridge amplification
- Synthesize second strand
- Denaturate to single-stranded samples
- After several cycles clusters are ready for sequencing
- Sequence the fragments

(see Movie)

[Ossowski, 2007]
Illumina Sequencing

- Flow cell preparation
- Bridge amplification
- Synthesize second strand
- Denaturate to single-stranded samples
- After several cycles clusters are ready for sequencing
- Sequence the fragments

(see Movie)

[Ossowski, 2007]
Illumina Sequencing

- Flow cell preparation
- Bridge amplification
- Synthesize second strand
- Denaturate to single-stranded samples
- After several cycles clusters are ready for sequencing
- Sequence the fragments

(see Movie)

[Ossowski, 2007]
Illumina Sequencing

- Flow cell preparation
- Bridge amplification
- Synthesize second strand
- Denature to single-stranded samples
- After several cycles clusters are ready for sequencing
- Sequence the fragments

(see Movie)

[Ossowski, 2007]
Illumina Sequencing

- Flow cell preparation
- Bridge amplification
- Synthesize second strand
- Denaturate to single-stranded samples
- After several cycles clusters are ready for sequencing
- Sequence the fragments

(see Movie)

[Ossowski, 2007]
Illumina Sequencing

- Flow cell preparation
- Bridge amplification
- Synthesize second strand
- Denaturate to single-stranded samples
- After several cycles, clusters are ready for sequencing
- Sequence the fragments

(see Movie)

[Ossowski, 2007]
SOLiD Sequencing

- Sequencing by ligation: Fragments ligated to “beads”
- PCR, beads enriched with fragments, ends of the templates modified to allow for an attachment to the slide
- Beads are deposited onto a glass slide
- Di-base probes compete for ligation to the sequencing primer

![Diagram of SOLiD Sequencing process]

© Gunnar Rätzsch (FML, Tübingen)
**SOLiD Sequencing**

- Sequencing by ligation: Fragments ligated to "beads"
- PCR, beads enriched with fragments, ends of the templates modified to allow for an attachment to the slide
- Beads are deposited onto a glass slide
- Di-base probes compete for ligation to the sequencing primer
**SOLiD Sequencing**

- Sequencing by ligation: Fragments ligated to “beads”
- PCR, beads enriched with fragments, ends of the templates modified to allow for an attachment to the slide
- Beads are deposited onto a glass slide
  - Di-base probes compete for ligation to the sequencing primer
**SOLiD Sequencing**

- Sequencing by ligation: Fragments ligated to “beads”
- PCR, beads enriched with fragments, ends of the templates modified to allow for an attachment to the slide
- Beads are deposited onto a glass slide
- Di-base probes compete for ligation to the sequencing primer
SOLiD Sequencing - Color Space

- 4 fluorescent dyes for 16 possible 2-mers
- Reverse, complement and reverse complement are always of same color

Possible Dinucleotides Encoded By Each Color

Double Interrogation

With 2 base encoding each base is defined twice
SOLiD Sequencing - Color Space

- 4 fluorescent dyes for 16 possible 2-mers
- Reverse, complement and reverse complement are always of same color

### SNP site indicated by 2 adjacent color changes

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A</td>
<td>C</td>
</tr>
<tr>
<td>A</td>
<td>C</td>
<td>T</td>
</tr>
<tr>
<td>A</td>
<td>C</td>
<td>T</td>
</tr>
<tr>
<td>A</td>
<td>C</td>
<td>T</td>
</tr>
</tbody>
</table>

SNP

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>G</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Reference in base space
Reference in color space
Read in color space
Read in base space

### Single color change is typically a measurement error

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>A</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>C</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>C</td>
<td>T</td>
<td></td>
</tr>
</tbody>
</table>

Error

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>C</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>C</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>C</td>
<td>C</td>
<td></td>
</tr>
</tbody>
</table>

Reference in base space
Reference in color space
Read in color space
Read in base space

### 1 Base Deletion

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A</td>
<td>C</td>
</tr>
<tr>
<td>A</td>
<td>A</td>
<td>C</td>
</tr>
<tr>
<td>A</td>
<td>C</td>
<td>T</td>
</tr>
<tr>
<td>A</td>
<td>C</td>
<td>T</td>
</tr>
</tbody>
</table>

Deletion

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A</td>
<td>C</td>
</tr>
<tr>
<td>A</td>
<td>A</td>
<td>C</td>
</tr>
<tr>
<td>A</td>
<td>C</td>
<td>T</td>
</tr>
<tr>
<td>A</td>
<td>C</td>
<td>T</td>
</tr>
</tbody>
</table>

Reference in base space
Reference in color space
Read in color space
Read in base space

© Gunnar Rätsch (FML, Tübingen)
## Overview / Extensions

<table>
<thead>
<tr>
<th>Technology</th>
<th>Read length</th>
<th>Output/run</th>
<th>Run time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Illumina GA II</td>
<td>40 – 75 bp</td>
<td>≈6-20 Gbp</td>
<td>5 – 8d</td>
</tr>
<tr>
<td>ABI SOLiD</td>
<td>35 – 50 bp</td>
<td>≈6-15 Gbp</td>
<td>6 – 7d</td>
</tr>
<tr>
<td>Roche/454</td>
<td>300 – 500 bp</td>
<td>≈100 Mbp</td>
<td>7h</td>
</tr>
<tr>
<td>Sanger</td>
<td>1000 bp</td>
<td>≈67 kbp</td>
<td>1h</td>
</tr>
</tbody>
</table>

There are several extensions available:
- mate-pair / paired-end
- bisulfite treatment
- multiplexing $8 \times 12$
  $\Rightarrow 96$ samples per flow cell

[Sutskever, 2008]
## Next-generation Sequencing

### Technology & Limitations

**Overview / Extensions**

<table>
<thead>
<tr>
<th>Technology</th>
<th>Read length</th>
<th>Output/run</th>
<th>Run time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Illumina GA II</td>
<td>40 – 75 bp</td>
<td>≈6-20 Gbp</td>
<td>5 – 8d</td>
</tr>
<tr>
<td>ABI SOLiD</td>
<td>35 – 50 bp</td>
<td>≈6-15 Gbp</td>
<td>6 – 7d</td>
</tr>
<tr>
<td>Roche/454</td>
<td>300 – 500 bp</td>
<td>≈100 Mbp</td>
<td>7h</td>
</tr>
<tr>
<td>Sanger</td>
<td>1000 bp</td>
<td>≈67 kbp</td>
<td>1h</td>
</tr>
</tbody>
</table>

There are several extensions available:

- mate-pair / paired-end
- bisulfite treatment
- multiplexing 8 × 12  
  \[75 \rightarrow 96 \text{ samples per flow cell}\]

[Sutskever, 2008]
Overview / Extensions

<table>
<thead>
<tr>
<th>Technology</th>
<th>Read length</th>
<th>Output/run</th>
<th>Run time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Illumina GA II</td>
<td>40 − 75 bp</td>
<td>≈6-20 Gbp</td>
<td>5 − 8d</td>
</tr>
<tr>
<td>ABI SOLiD</td>
<td>35 − 50 bp</td>
<td>≈6-15 Gbp</td>
<td>6 − 7d</td>
</tr>
<tr>
<td>Roche/454</td>
<td>300 − 500 bp</td>
<td>≈100 Mbp</td>
<td>7h</td>
</tr>
<tr>
<td>Sanger</td>
<td>1000 bp</td>
<td>≈67 kbp</td>
<td>1h</td>
</tr>
</tbody>
</table>

There are several extensions available:

- mate-pair / paired-end
- bisulfite treatment
- multiplexing 8 × 12
  \(\sim 96\) samples per flow cell

[Sutskever, 2008]
Overview / Extensions

<table>
<thead>
<tr>
<th>Technology</th>
<th>Read length</th>
<th>Output/run</th>
<th>Run time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Illumina GA II</td>
<td>40 – 75 bp</td>
<td>≈6-20 Gbp</td>
<td>5 – 8d</td>
</tr>
<tr>
<td>ABI SOLiD</td>
<td>35 – 50 bp</td>
<td>≈6-15 Gbp</td>
<td>6 – 7d</td>
</tr>
<tr>
<td>Roche/454</td>
<td>300 – 500 bp</td>
<td>≈100 Mbp</td>
<td>7h</td>
</tr>
<tr>
<td>Sanger</td>
<td>1000 bp</td>
<td>≈67 kbp</td>
<td>1h</td>
</tr>
</tbody>
</table>

There are several extensions available:

- mate-pair / paired-end
- bisulfite treatment
- multiplexing $8 \times 12$
  $\sim 96$ samples per flow cell

[Sutskever, 2008]
Given read data the following analysis steps are possible:
- Assembly
- Mapping/Alignments

Assembled genome

ACGTACCGTTTGTACCTAGATATTTTCTTCTTCTAGTAGATATTTTTTTTTTTTTTAGATAAAA

[Sutskever, 2008]
Short Reads Analysis - Methods

Given read data the following analysis steps are possible:

- Assembly
- Mapping/Alignments

...GCAAACCAGTGACCTGACTACTACGTCGTAACGTAACGTAACCGGTAAGCT...
GCAAACCAGTGACCTGACTACTACGTCGTAACGTAACGTAACCGGTAAGCT...
CAAACCAGTGACCTGACTACTACGTCGTAACGTAACGTAACCGGTAAGCT...
AAACCAGTGACCTGACTACTACGTCGTAACGTAACGTAACCGGTAAGCT...
AACCAGTGACCTGACTACTACGTCGTAACGTAACGTAACCGGTAAGCT...
ACCAGTGACCTGACTACTACGTCGTAACGTAACGTAACCGGTAAGCT...
Short Reads Analysis - Methods

Given read data the following analysis steps are possible:
- Assembly
- Mapping/Alignments

...GCAAACCGACCTGACTAATGTCGTAACGTACACGGTAGCT...
GCAAACCGACCTGACTGCTACGTCGTAACGTAC
CAAACCGACCTGACTGCTACGTCGTCAACGTACA
AAACCATGGACCTGACTGCTACGTCGTAACGTACAC
AAACCATGGACCTGACTGCTACGTCGTAACGTACAC
AAACCATGGACCTGACTGCTACGTCGTAACGTACAC
AAACCATGGACCTGACTGCTACGTCGTAACGTACAC
AAACCATGGACCTGACTGCTACGTCGTAACGTACAC
AAACCATGGACCTGACTGCTACGTCGTAACGTACAC
Given read data the following analysis steps are possible:

- Assembly
- Mapping/Alignments

Problem: 100 million reads of short length

⇒ Big computational challenge
Short Reads Analysis - Problems

- Experiment leads to millions of reads 36 – 75nt
- Reads may have a position-wise varying quality

Quality corresponds to error probability:

\[ q = -10 \cdot \log_{10}(\frac{p}{1-p}) \]

Example: If we have an error probability of \(10^{-3}\) per base then the quality is 30
Read assembly problem

For a set of reads stemming from a reference genome find maximally overlapping parts in order to reconstruct the genomic sequence

Classical assembly: \[\Rightarrow\text{Too inefficient for short reads}\]

1. **Overlap phase**: Every read is compared with every other read and the overlap graph is computed.
2. **Layout phase**: Pairs are determined that position every read in the assembly.
3. **Consensus phase**: Multi-alignment of all the placed reads is produced to obtain the final sequence.

New techniques: Plethora of tools available (EULER, VELVET, SHARCGS, SSAKE/VCAKE, ...)

**Idea**: de Bruijn Graphs
Short Reads Assembly

Read assembly problem

For a set of reads stemming from a reference genome find maximally overlapping parts in order to reconstruct the genomic sequence

Classical assembly: ⇒ Too inefficient for short reads

1. Overlap phase: Every read is compared with every other read and the overlap graph is computed
2. Layout phase: Pairs are determined that position every read in the assembly
3. Consensus phase: Multi-alignment of all the placed reads is produced to obtain the final sequence

New techniques: Plethora of tools available (EULER, VELVET, SHARCGS, SSAKE/VCAKE, ...)

Idea: de Bruijn Graphs
Short Reads Assembly

Read assembly problem

For a set of reads stemming from a reference genome find maximally overlapping parts in order to reconstruct the genomic sequence.

Classical assembly: ⇒ **Too inefficient for short reads**

1. **Overlap phase**: Every read is compared with every other read and the overlap graph is computed.
2. **Layout phase**: Pairs are determined that position every read in the assembly.
3. **Consensus phase**: Multi-alignment of all the placed reads is produced to obtain the final sequence.

New techniques: Plethora of tools available (EULER, VELVET, SHARCGS, SSAKE/VCAKE, ...)

**Idea**: de Bruijn Graphs
Read assembly problem

For a set of reads stemming from a reference genome find maximally overlapping parts in order to reconstruct the genomic sequence.

Classical assembly: \( \Rightarrow \) **Too inefficient for short reads**

1. **Overlap phase**: Every read is compared with every other read and the overlap graph is computed.
2. **Layout phase**: Pairs are determined that position every read in the assembly.
3. **Consensus phase**: Multi-alignment of all the placed reads is produced to obtain the final sequence.

**New techniques**: Plethora of tools available (EULER, VELVET, SHARCGS, SSAKE/VCAKE, . . . )

**Idea**: de Bruijn Graphs
Reads are mapped as a path in the graph

Number of reads does not influence number of nodes

⇒ Use de-Bruijn graphs to solve the problem
**de Bruijn Graphs**

**Example:**

```
TAGAC
AGACT
AGACT
ACTGA
CTGAT
TGATT
GATTG
TTGAC
TGACC
GACCA
ATTGC
TTGCC
```

**Nodes represent** $k$-mers **smaller than read length**
- A $k$-mer can refer to thousands of reads containing it
- Read errors or ambiguities lead to branching of paths
- Each node also stores the reverse complement

[Zerbino and Birney, 2008]
de Bruijn Graphs

Example:

Nodes represent $k$-mers smaller than read length
- A $k$-mer can refer to thousands of reads containing it
- Read errors or ambiguities lead to branching of paths
- Each node also stores the reverse complement

[Zerbino and Birney, 2008]
de Bruijn Graphs

Example:

- Nodes represent $k$-mers smaller than read length
- A $k$-mer can refer to thousands of reads containing it
- Read errors or ambiguities lead to branching of paths
- Each node also stores the reverse complement

[Zerbino and Birney, 2008]
de Bruijn Graphs

Example:

- Nodes represent $k$-mers smaller than read length
- A $k$-mer can refer to thousands of reads containing it
- Read errors or ambiguities lead to branching of paths
- Each node also stores the reverse complement

[Zerbino and Birney, 2008]
Consider a constructed de Bruijn graph

- Unconnected nodes
- Ambiguous paths
- Erroneous edges

Read off genome sequence (if everything goes well ;-) [Zerbino and Birney, 2008]
Consider a constructed de Bruijn graph

- Unconnected nodes
- Ambiguous paths
- Erroneous edges

Read off genome sequence (if everything goes well ;-) )

[Zerbino and Birney, 2008]
Consider a constructed de Bruijn graph

- Unconnected nodes
- Ambiguous paths
- Erroneous edges

Read off genome sequence (if everything goes well ;-) )

[Zerbino and Birney, 2008]
Consider a constructed de Bruijn graph

- Unconnected nodes
- Ambiguous paths
- Erroneous edges

Read off genome sequence (if everything goes well ;-)  

[Zerbino and Birney, 2008]
Consider a constructed de Bruijn graph

- Unconnected nodes
- Ambiguous paths
- Erroneous edges

Read off genome sequence (if everything goes well ;-) )

[Zerbino and Birney, 2008]
Results for Velvet

Table 3. Comparison of short read assemblers on experimental *Streptococcus suis* Solexa reads

<table>
<thead>
<tr>
<th>Assembler</th>
<th>No. of contigs</th>
<th>N50</th>
<th>Average error rate</th>
<th>Memory</th>
<th>Time</th>
<th>Seq. Cov.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Velvet 0.3</td>
<td>470</td>
<td>8661 bp</td>
<td>0.02%</td>
<td>2.0G</td>
<td>2 min 57 sec</td>
<td>97%</td>
</tr>
<tr>
<td>SSAKE 2.0</td>
<td>265</td>
<td>1727 bp</td>
<td>0.20%</td>
<td>1.7G</td>
<td>1 h 47 min</td>
<td>16%</td>
</tr>
<tr>
<td>VCAKE 1.0</td>
<td>7675</td>
<td>1137 bp</td>
<td>0.64%</td>
<td>1.8G</td>
<td>4 h 25 min</td>
<td>134%</td>
</tr>
</tbody>
</table>

Considerably shorter fragments for larger genomes

[Zerbino and Birney, 2008]
Results for Velvet

Table 3. Comparison of short read assemblers on experimental *Streptococcus suis* Solexa reads

<table>
<thead>
<tr>
<th>Assembler</th>
<th>No. of contigs</th>
<th>N50</th>
<th>Average error rate</th>
<th>Memory</th>
<th>Time</th>
<th>Seq. Cov.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Velvet 0.3</td>
<td>470</td>
<td>8661 bp</td>
<td>0.02%</td>
<td>2.0G</td>
<td>2 min 57 sec</td>
<td>97%</td>
</tr>
<tr>
<td>SSAKE 2.0</td>
<td>265</td>
<td>1727 bp</td>
<td>0.20%</td>
<td>1.7G</td>
<td>1 h 47 min</td>
<td>16%</td>
</tr>
<tr>
<td>VCAKE 1.0</td>
<td>7675</td>
<td>1137 bp</td>
<td>0.64%</td>
<td>1.8G</td>
<td>4 h 25 min</td>
<td>134%</td>
</tr>
</tbody>
</table>

Considerably shorter fragments for larger genomes

[Zerbino and Birney, 2008]
Short Reads Analysis - Mapping

Reads mapping problem
For each read find its target regions on the reference genome such that are at most $k$ mismatches between read and target.

- Global/local alignment of all reads prohibitive
- A read stems from a certain small region
- Find this region and then do an alignment
  - spaced seeds
  - suffix trees/arrays
- Common tools: GenomeMapper, Shrimp, SOAP, VMATCH, …
Reads mapping problem

For each read find its target regions on the reference genome such that are at most $k$ mismatches between read and target.

- Global/local alignment of all reads prohibitive
- A read stems from a certain small region
- Find this region and then do an alignment
  - spaced seeds
  - suffix trees/arrays
- Common tools: GenomeMapper, Shrimp, SOAP, VMATCH, ...
Given a long fixed string of length $n$ and smaller patterns of lengths $m$ to be searched for.

Construction in $O(n)$, Patterns can be detected in $O(m)$.
Spliced vs. Unspliced Alignments

pre-mRNA

Exon

Intron

mRNA

Short reads

Short read is split by intron when aligning to reference Genome

[Wikipedia]
Spliced vs. Unspliced Alignments

- Find matching region on genome with a few mismatches
- Efficient data structures for mapping many reads
- Most current mapping techniques are limited to unspliced reads
Find matching region on genome with a few mismatches

Efficient data structures for mapping many reads

Most current mapping techniques are limited to unspliced reads
Spliced vs. Unspliced Alignments

- Find matching region on genome with a few mismatches
- Efficient data structures for mapping many reads
- Most current mapping techniques are limited to unspliced reads
Extended Smith-Waterman Algorithm

Classical scoring $f : \Sigma \times \Sigma \rightarrow \mathbb{R}$

Source of Information
- Sequence matches
- Computational splice site predictions
- Intron length model
- Read quality information

<table>
<thead>
<tr>
<th></th>
<th>gap</th>
<th>A</th>
<th>C</th>
<th>G</th>
<th>T</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>gap</td>
<td>0.33</td>
<td>0.3</td>
<td>0.12</td>
<td>0.3</td>
<td>0.3</td>
<td>0.55</td>
</tr>
<tr>
<td>A</td>
<td>0.31</td>
<td>0.12</td>
<td>0.12</td>
<td>0.3</td>
<td>0.55</td>
<td>0.33</td>
</tr>
<tr>
<td>C</td>
<td>0.44</td>
<td>0.12</td>
<td>0.44</td>
<td>0.3</td>
<td>0.59</td>
<td>0.12</td>
</tr>
<tr>
<td>G</td>
<td>0.13</td>
<td>0.85</td>
<td>0.31</td>
<td>0.33</td>
<td>0.51</td>
<td>0.3</td>
</tr>
<tr>
<td>T</td>
<td>0.55</td>
<td>0.12</td>
<td>0.13</td>
<td>0.12</td>
<td>0.11</td>
<td>0.1</td>
</tr>
<tr>
<td>N</td>
<td>0.12</td>
<td>0.01</td>
<td>0.3</td>
<td>0.12</td>
<td>0.3</td>
<td>0.01</td>
</tr>
</tbody>
</table>
Extended Smith-Waterman Algorithm

Classical scoring $f: \Sigma \times \Sigma \rightarrow \mathbb{R}$

<table>
<thead>
<tr>
<th></th>
<th>gap</th>
<th>A</th>
<th>C</th>
<th>G</th>
<th>T</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>gap</td>
<td>0.33</td>
<td>0.3</td>
<td>0.12</td>
<td>0.3</td>
<td>0.3</td>
<td>0.55</td>
</tr>
<tr>
<td>A</td>
<td>0.31</td>
<td>0.12</td>
<td>0.12</td>
<td>0.3</td>
<td>0.55</td>
<td>0.33</td>
</tr>
<tr>
<td>C</td>
<td>0.44</td>
<td>0.12</td>
<td>0.44</td>
<td>0.3</td>
<td>0.59</td>
<td>0.12</td>
</tr>
<tr>
<td>G</td>
<td>0.13</td>
<td>0.85</td>
<td>0.31</td>
<td>0.33</td>
<td>0.51</td>
<td>0.3</td>
</tr>
<tr>
<td>T</td>
<td>0.55</td>
<td>0.12</td>
<td>0.13</td>
<td>0.12</td>
<td>0.11</td>
<td>0.1</td>
</tr>
<tr>
<td>N</td>
<td>0.12</td>
<td>0.01</td>
<td>0.3</td>
<td>0.12</td>
<td>0.3</td>
<td>0.01</td>
</tr>
</tbody>
</table>

Source of Information
- Sequence matches
- Computational splice site predictions
- Intron length model
- Read quality information
Extended Smith-Waterman Algorithm

Classical scoring $f : \Sigma \times \Sigma \rightarrow \mathbb{R}$

Source of Information
- Sequence matches
- Computational splice site predictions
- Intron length model
- Read quality information

<table>
<thead>
<tr>
<th></th>
<th>gap</th>
<th>A</th>
<th>C</th>
<th>G</th>
<th>T</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>gap</td>
<td>0.33</td>
<td>0.3</td>
<td>0.12</td>
<td>0.3</td>
<td>0.3</td>
<td>0.55</td>
</tr>
<tr>
<td>A</td>
<td>0.31</td>
<td>0.12</td>
<td>0.12</td>
<td>0.3</td>
<td>0.55</td>
<td>0.33</td>
</tr>
<tr>
<td>C</td>
<td>0.44</td>
<td>0.12</td>
<td>0.44</td>
<td>0.3</td>
<td>0.59</td>
<td>0.12</td>
</tr>
<tr>
<td>G</td>
<td>0.13</td>
<td>0.85</td>
<td>0.31</td>
<td>0.33</td>
<td>0.51</td>
<td>0.3</td>
</tr>
<tr>
<td>T</td>
<td>0.55</td>
<td>0.12</td>
<td>0.13</td>
<td>0.12</td>
<td>0.11</td>
<td>0.1</td>
</tr>
<tr>
<td>N</td>
<td>0.12</td>
<td>0.01</td>
<td>0.3</td>
<td>0.12</td>
<td>0.3</td>
<td>0.01</td>
</tr>
</tbody>
</table>
Extended Smith-Waterman Algorithm

Quality scoring $f : (\Sigma \times \mathbb{R}) \times \Sigma \rightarrow \mathbb{R}$

Source of Information
- Sequence matches
- Computational splice site predictions
- Intron length model
- Read quality information

De Bona et al., 2008
QPalma’s Accurate Alignments

Generate set of artificially spliced reads
- Genomic reads with quality information
- Genome annotation for artificially splicing the reads
- Use 10,000 reads for training and 30,000 for testing

Alignment Error Rate
- SmithW: 14.19%
- Intron: 9.96%
- Intron+Splice: 1.94%
- Intron+Splice +Quality: 1.78%

[De Bona et al., 2008]
An Alignment Pipeline

Sequence input

Short transcript reads
(with quality information)

Genomic sequence

Scores for potential splice sites

Optimized parameters for spliced alignments

Alignments using *vmatch* & *QPalma* filter

*QPALMA*

Final unspliced reads
(full length match)

Final spliced alignment
(with one intron)

Realign reads against splice junctions using *vmatch* for quality control

[De Bona et al., 2008]
ARTICLES

Alternative isoform regulation in human tissue transcriptomes

Eric T. Wang\textsuperscript{1,2*}, Rickard Sandberg\textsuperscript{1,3*}, Shujun Luo\textsuperscript{4}, Irina Khrebtukova\textsuperscript{4}, Lu Zhang\textsuperscript{4}, Christine Mayr\textsuperscript{5}, Stephen F. Kingsmore\textsuperscript{6}, Gary P. Schroth\textsuperscript{4} & Christopher B. Burge\textsuperscript{1}

A Global View of Gene Activity and Alternative Splicing by Deep Sequencing of the Human Transcriptome

Marc Sultan,\textsuperscript{1*} Marcel H. Schulz,\textsuperscript{2,3*} Hugues Richard,\textsuperscript{2*} Alon Magen,\textsuperscript{1} Andreas Klingenhoff,\textsuperscript{4} Matthias Scherf,\textsuperscript{4} Martin Seifert,\textsuperscript{4} Tatjana Borodina,\textsuperscript{1} Aleksey Soldatov,\textsuperscript{1} Dmitri Parkhomchuk,\textsuperscript{1} Dominic Schmidt,\textsuperscript{1} Sean O’Keeffe,\textsuperscript{2} Stefan Haas,\textsuperscript{2} Martin Vingron,\textsuperscript{2} Hans Lehrach,\textsuperscript{1} Marie-Laure Yaspo\textsuperscript{1†}
### Transcriptome Studies in Human

<table>
<thead>
<tr>
<th>Alternative transcript events</th>
<th>Total events ((x10^5))</th>
<th>Number detected ((x10^5))</th>
<th>Both isoforms detected</th>
<th>Number tissue-regulated</th>
<th>% Tissue-regulated (\text{observed})</th>
<th>% Tissue-regulated (\text{estimated})</th>
</tr>
</thead>
<tbody>
<tr>
<td>Skipped exon</td>
<td>37</td>
<td>35</td>
<td>10,436</td>
<td>6,822</td>
<td>65</td>
<td>72</td>
</tr>
<tr>
<td>Retained intron</td>
<td>1</td>
<td>1</td>
<td>167</td>
<td>96</td>
<td>57</td>
<td>71</td>
</tr>
<tr>
<td>Alternative 5’ splice site (A5SS)</td>
<td>15</td>
<td>15</td>
<td>2,168</td>
<td>1,386</td>
<td>64</td>
<td>72</td>
</tr>
<tr>
<td>Alternative 3’ splice site (A3SS)</td>
<td>17</td>
<td>16</td>
<td>4,181</td>
<td>2,655</td>
<td>64</td>
<td>74</td>
</tr>
<tr>
<td>Mutually exclusive exon (MXE)</td>
<td>4</td>
<td>4</td>
<td>167</td>
<td>95</td>
<td>57</td>
<td>66</td>
</tr>
<tr>
<td>Alternative first exon (AFE)</td>
<td>14</td>
<td>13</td>
<td>10,281</td>
<td>5,311</td>
<td>52</td>
<td>63</td>
</tr>
<tr>
<td>Alternative last exon (ALE)</td>
<td>9</td>
<td>8</td>
<td>5,246</td>
<td>2,491</td>
<td>47</td>
<td>52</td>
</tr>
<tr>
<td>Tandem 3’ UTRs</td>
<td>7</td>
<td>7</td>
<td>5,136</td>
<td>3,801</td>
<td>74</td>
<td>80</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>105</strong></td>
<td><strong>100</strong></td>
<td><strong>37,782</strong></td>
<td><strong>22,657</strong></td>
<td><strong>60</strong></td>
<td><strong>68</strong></td>
</tr>
</tbody>
</table>

- **Constitutive exon or region**
- **Body read**
- **Junction read**
- **pA Polyadenylation site**
- **Alternative exon or extension**
- **Inclusive/extended isoform**
- **Exclusive isoform**
- **Both isoforms**

[Wang et al., 2008]
Roadmap

1. Computational Gene Finding
   - Identification of Genomic Signals
   - Learning to Predict mRNA Transcripts

2. Whole-genome Tiling Arrays
   - Technology and Limitations
   - Identification of Expression Differences
   - De Novo Transcript Discovery

3. Next-generation Sequencing
   - Technology & Limitations
   - Assembly & Read Mapping

4. Extensions
   - Quantification of Transcripts
   - ChIP-on-Chip Studies
Combine *ab initio* predictions with transcriptome measurements

- Higher accuracy
- Condition/tissue specific predictions
RNA-seq Data

Read coverage

RNA-Seq data

Gene structure

ACGGTG GTCAATGTACCTAAATGG GT
GTCAATGTACCTAAATGG GTTAATT TG
ATGG GTAAATTTGACCACACGTGAAGA
...ACGGTG GTCAATGTACCTAAATGG GTTAATT TGACCACACGTGAAGAGAGAGGCCCTCC...

Extensions
Quantification of Transcripts
RNA-seq reads for *A. thaliana* (provided by Weigel lab, MPI Devel. Biology)

- 4 lanes from Illumina Genome Analyzer
- 38nt reads, polyA enriched
- Strand unspecific, young leaves
- Read mapping using **ShoRe** [Ossowski et al., 2008]
- Spliced alignments using **QPalma** [De Bona et al., 2008]
- \(\approx 30\) million unspliced and \(\approx 1\) million spliced reads (\(\approx 50\times\) coverage)
Tiling Array Data

Gene structure

35 bp
25-mer probes

A — T
C — G
G — C
A — T
G — C
A — T
T — T
G — C
G — C
T — A
T — A

Tiling array data

cDNA fragments with fluorescence markers

© Gunnar Rätsch (FML, Tübingen)
Tiling arrays for *A. thaliana* (provided by Weigel lab, MPI Devel. Biology)

- 25nt probes, 35nt spacing, 3 replicates
- Strand unspecific, polyA enriched
- 12 different tissues (young leaves, root, etc.)
- 12 conditions/mutants (e.g. abiotic stresses)
- Quantile and sequence dependent normalization [Zeller et al., 2008b]
Learning to Integrate Data Sources

[Behr et al., 2008]

STEP 1: SVM Signal Predictions

tss
tis
acc
don
stop

True gene model

---

Methods for Transcriptome Analysis

© Gunnar Rätsch (FML, Tübingen)
Learning to Integrate Data Sources

[Behr et al., 2008]

STEP 1: SVM Signal Predictions

STEP 2: Integration

transform features

F(x,y)
Learning to Integrate Data Sources

[Behr et al., 2008]

STEP 1: SVM Signal Predictions

STEP 2: Integration

F(x,y)

transform features

large margin

genomic position

True gene model

Wrong gene model

© Gunnar Rätsch (FML, Tübingen)
Learning to Integrate Data Sources

[Behr et al., 2008]

STEP 1: SVM Signal Predictions
- tss
- tis
- acc
- don
- stop

STEP 2: Integration

Transform features

Large margin

F(x,y)

Tiling Array Data

Genomic position

True gene model

Wrong gene model
Results: How much does the data help?
[Behr et al., 2008]

Experimental setup (*Arabidopsis thaliana*):
- 60% of known genes for training signals in step 1
- 400 genes for training of combination of data
- 300 regions around cDNA confirmed genes for evaluation
Results: How much does the data help?  
[Behr et al., 2008]

Experimental setup (*Arabidopsis thaliana*):
- 60% of known genes for training signals in step 1
- 400 genes for training of combination of data
- 300 regions around cDNA confirmed genes for evaluation

<table>
<thead>
<tr>
<th>Method</th>
<th>transcript level [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SN</strong></td>
<td><strong>SP</strong></td>
</tr>
<tr>
<td><em>Ab initio</em></td>
<td>73.7</td>
</tr>
<tr>
<td>Tiling arrays (young leaves)</td>
<td>78.0</td>
</tr>
<tr>
<td>Tiling arrays (inflorescence)</td>
<td>77.1</td>
</tr>
<tr>
<td>Tiling arrays (root)</td>
<td>76.2</td>
</tr>
<tr>
<td>Tiling arrays (combined)</td>
<td>77.4</td>
</tr>
<tr>
<td>RNA-seq (w/o spliced reads)</td>
<td>76.8</td>
</tr>
<tr>
<td>RNA-seq (with spliced reads)</td>
<td>79.6</td>
</tr>
</tbody>
</table>
Results: How much does the data help? [Behr et al., 2008]

Experimental setup (*Arabidopsis thaliana*):
- 60% of known genes for training signals in step 1
- 400 genes for training of combination of data
- 300 regions around cDNA confirmed genes for evaluation

<table>
<thead>
<tr>
<th></th>
<th>transcript level [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SN</td>
</tr>
<tr>
<td><strong>Ab initio</strong></td>
<td>73.7</td>
</tr>
<tr>
<td><strong>Tiling arrays</strong></td>
<td>(young leaves)</td>
</tr>
<tr>
<td></td>
<td>(inflorescence)</td>
</tr>
<tr>
<td></td>
<td>(root)</td>
</tr>
<tr>
<td></td>
<td>(combined)</td>
</tr>
<tr>
<td>RNA-seq (w/o spliced reads)</td>
<td>76.8</td>
</tr>
<tr>
<td>RNA-seq (with spliced reads)</td>
<td>79.6</td>
</tr>
</tbody>
</table>
Results: How much does the data help?

[Behr et al., 2008]

Experimental setup (Arabidopsis thaliana):
- 60% of known genes for training signals in step 1
- 400 genes for training of combination of data
- 300 regions around cDNA confirmed genes for evaluation

<table>
<thead>
<tr>
<th></th>
<th>transcript level [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SN</td>
</tr>
<tr>
<td>Ab initio</td>
<td>73.7</td>
</tr>
<tr>
<td>Tiling arrays (young leaves)</td>
<td>78.0</td>
</tr>
<tr>
<td>Tiling arrays (inflorescence)</td>
<td>77.1</td>
</tr>
<tr>
<td>Tiling arrays (root)</td>
<td>76.2</td>
</tr>
<tr>
<td>Tiling arrays (combined)</td>
<td>77.4</td>
</tr>
<tr>
<td>RNA-seq (w/o spliced reads)</td>
<td>76.8</td>
</tr>
<tr>
<td>RNA-seq (with spliced reads)</td>
<td>79.6</td>
</tr>
</tbody>
</table>
Results: How much data is needed?  
[Behr et al., 2008]

Experimental setup (*Arabidopsis thaliana)*:
- 60% of known genes for training signals in step 1
- 400 genes for training of combination of data
- 300 regions around cDNA confirmed genes for evaluation

<table>
<thead>
<tr>
<th>Method</th>
<th>Transcript level [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SN</td>
</tr>
<tr>
<td>Ab initio</td>
<td></td>
</tr>
<tr>
<td>RNA-seq 1/128</td>
<td>75.5</td>
</tr>
<tr>
<td>RNA-seq 1/64</td>
<td>76.8</td>
</tr>
<tr>
<td>RNA-seq 1/32</td>
<td>76.2</td>
</tr>
<tr>
<td>RNA-seq 1/16</td>
<td>77.1</td>
</tr>
<tr>
<td>RNA-seq 1/8</td>
<td>78.6</td>
</tr>
<tr>
<td>RNA-seq 1/4</td>
<td>77.4</td>
</tr>
<tr>
<td>RNA-seq 1/2</td>
<td>79.3</td>
</tr>
<tr>
<td>RNA-seq</td>
<td>79.6</td>
</tr>
</tbody>
</table>
Results: How much data is needed?

[Behr et al., 2008]

Experimental setup (*Arabidopsis thaliana*):
- 60% of known genes for training signals in step 1
- 400 genes for training of combination of data
- 300 regions around cDNA confirmed genes for evaluation

<table>
<thead>
<tr>
<th>Methods</th>
<th>transcript level [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SN</td>
</tr>
<tr>
<td><strong>Ab initio</strong></td>
<td>73.7</td>
</tr>
<tr>
<td>RNA-seq 1/128</td>
<td>75.5</td>
</tr>
<tr>
<td>RNA-seq 1/64</td>
<td>76.8</td>
</tr>
<tr>
<td>RNA-seq 1/32</td>
<td>76.2</td>
</tr>
<tr>
<td>RNA-seq 1/16</td>
<td>77.1</td>
</tr>
<tr>
<td>RNA-seq 1/8</td>
<td>78.6</td>
</tr>
<tr>
<td>RNA-seq 1/4</td>
<td>77.4</td>
</tr>
<tr>
<td>RNA-seq 1/2</td>
<td>79.3</td>
</tr>
<tr>
<td>RNA-seq</td>
<td>79.6</td>
</tr>
</tbody>
</table>
Results: Combining helps!

[Behr et al., 2008]

Methods for Transcriptome Analysis

Bertinoro Systems Biology

© Gunnar Rätsch (FML, Tübingen)
Results: Combining helps!

[Behr et al., 2008]
Results: Combining helps!

[Behr et al., 2008]
Results: Combining helps!

[Behr et al., 2008]
Quantification of Transcripts

**Given:** Accurate short reads alignments
We can use exon/intron read coverage to:

1. Improve gene finder predictions?
2. Predict transcript abundances?

**First Step:** Given a set of known transcripts, we predict transcript abundances by solving a *linear programming* problem:

- Optimizes the weights for each transcript
- Exploits additive nature of the read coverage
- Minimizing the residual error
Quantification of Transcripts

**Given:** Accurate short reads alignments
We can use exon/intron read coverage to:

1. Improve gene finder predictions?
2. Predict transcript abundances?

**First Step:** Given a set of known transcripts, we predict transcript abundances by solving a **linear programming** problem:

- **Optimizes the weights** for each transcript
- Exploits additive nature of the read coverage
- Minimizing the residual error
Quantification of Transcripts

(Preliminary)

AT1G01630
Chromosome CHR1 +

228971 229372 229773 230174 230575

Read count

10^1

10^0

2.29 2.292 2.294 2.296 2.298 2.3 2.302 2.304 2.306 2.308 x 10^5
Quantification of Transcripts

(Preliminary)
Combination of gene finding and transcript quantification:
Detect alternative transcripts including their abundance without
relying on a genome annotation.

Example for A. thaliana.

The two isoforms were correctly determined (upper panel) and the transcript abundances are estimated well (lower panel).
Roadmap

1. Computational Gene Finding
   - Identification of Genomic Signals
   - Learning to Predict mRNA Transcripts

2. Whole-genome Tiling Arrays
   - Technology and Limitations
   - Identification of Expression Differences
   - De Novo Transcript Discovery

3. Next-generation Sequencing
   - Technology & Limitations
   - Assembly & Read Mapping

4. Extensions
   - Quantification of Transcripts
   - ChIP-on-Chip Studies
ChIP-on-chip and ChIP-seq

- ChIP = Chromatin Immunoprecipitation
- Established technique, now used for genome-wide screens on a chip (ChIP-on-chip) or through Next-Generation Sequencing (ChIP-seq)
- Analyze binding of a single transcription factor (TF)
- **Goal:** Identify parts of the chromatin that this TF binds to

RNA Immunoprecipitation to understand RNA processing.
ChIP-on-chip and ChIP-seq

- ChIP = Chromatin Immunoprecipitation
- Established technique, now used for genome-wide screens on a chip (ChIP-on-chip) or through Next-Generation Sequencing (ChIP-seq)
- Analyze binding of a single transcription factor (TF)
- **Goal:** Identify parts of the chromatin that this TF binds to

RNA Immunoprecipitation to understand RNA processing.
ChIP Protocol: Preparations

ChIP: Chromatin Immunoprecipitation

Create antibody against a certain TF:

- Identify the TF-coding gene
- Transfer gene sequence to a cloning vector
- Get cells (*E. coli*, yeast, ...) to express the protein
- Extract (correctly folded) protein from cells, purify, then purify again
  - Inject in animal, extract, purify, ...
  \[ \Rightarrow \text{Obtain (poly-clonal) antibody} \]
ChIP: Chromatin Immunoprecipitation

Create antibody against a certain TF:

- Identify the TF-coding gene
- Transfer gene sequence to a cloning vector
- Get cells (*E. coli*, yeast, . . . ) to express the protein
- Extract (correctly folded) protein from cells, purify, then purify again
- Inject in animal, extract, purify, . . .  
  ⇒ Obtain (poly-clonal) antibody
ChIP Protocol: Overview

ChIP-on-chip wet-lab portion of the workflow

DNA microarray
**ChIP-on-chip: Detection**

- Tiling arrays or Sequencing analysis pipelines can be used
  \( \Rightarrow \) *Similar problems as in transcriptome analysis*
- Compare with control experiment, e.g. LOF of the TF, calculate p-values of binding probability using a smoothing window
- ChIP arrays often analyzed with *specialized methods*:
  Model-based Analysis of Tiling-array (MAT), TileMap; TiMAT
- ML approaches: Learn expected distribution from regulatory region, where binding peak is to be expected (often difficult as labeled data is scarce)

[Provided by Sebastian Schultheiss]
ChIP-on-chip: Detection

- Tiling arrays or Sequencing analysis pipelines can be used ⇒ *Similar problems as in transcriptome analysis*
- Compare with control experiment, e.g. LOF of the TF, calculate p-values of binding probability using a smoothing window
- ChIP arrays often analyzed with *specialized methods*:
  Model-based Analysis of Tiling-array (MAT), TileMap; TiMAT
- ML approaches: Learn expected distribution from regulatory region, where binding peak is to be expected (often difficult as labeled data is scarce)

[Provided by Sebastian Schultheiss]
ChIP-on-chip: Detection

- Tiling arrays or Sequencing analysis pipelines can be used ⇒ *Similar problems as in transcriptome analysis*
- Compare with control experiment, e.g. LOF of the TF, calculate p-values of binding probability using a smoothing window
- ChIP arrays often analyzed with *specialized methods*: Model-based Analysis of Tiling-array (MAT), TileMap; TiMAT
- ML approaches: Learn expected distribution from regulatory region, where binding peak is to be expected (often difficult as labeled data is scarce)
**MAT**: Model-based analysis of tiling arrays for ChIP-chip

**Idea**: Majority of signal is due to non-specific binding, there are strong probe sequence effects

⇒ Formulate an array probe affinity model

[Johnson et al., 2006]
ChIP-on-chip: MAT

Algorithm:

- Divide probes into **affinity bins**
- Sample probe signal **variance** for every probe per bin (more numbers makes this more stable than just comparing replicates)
- Calculate a **t-value** $t(k)$ per probe $k$
- Compute **trimmed mean** $TM$ in sliding window
- The **central probe** of the window from probes $i$ to $j$ will be assigned a MATscore $i,j = \sqrt{j-i} \cdot TM_{k=i}^j t(k)$
- Define a MATscore **threshold** above which a probe is classified as enriched, subtract control experiments if available
- Threshold can be found by MAT using a **p-value** from a non-enriched null sample or a user-supplied **FDR**
- MAT **merges** all enriched regions within 300 bp and assigns them the highest MATscore of the region
ChIP-on-chip: TileMap HMM

\begin{align*}
\pi_0 & \quad 1 \\
1 - a_0 & \quad f_0(t) \\
1 - a_1 & \quad f_1(t)
\end{align*}

\begin{align*}
\text{if } d_{i,i+1} \leq d_0 \\
\text{if } d_{i,i+1} > d_0
\end{align*}

[Ji and Wong, 2005b]
Algorithm:

- Compute **test-statistic** for each normalized, log-transformed probe $X_{ijk}$, which is the hybridization intensity for probe $i$ under condition $j$ in replicate $k$
- $X_{ijk} | \mu_i^2 \sim N(\mu_{ij}, \sigma_i^2)$ estimate every $\sigma_i^2$ to approximate posterior distribution of $\mu_{ij}$
- Use a formula akin to a t-statistic, which uses not only information for probe $i$ to estimate standard deviation but pools information from all probes for higher sensitivity
- Combine information from neighboring probes (moving average/sliding window or an HMM)

[Ji and Wong, 2005b]
ChIP-on-chip experiment on known TF (plant stem cell regulator)

WUSCHEL

Key: Transcription Factor

Models should be based on transcripts not genes!
ChIP-on-chip: What do we learn?

ChIP-on-chip experiment on known TF (plant stem cell regulator)

Motif finding
TF targets

Expression levels of bound genes
regulative direction

in silico  Wetlab

WUSCHEL

Key:  Transcription Factor

Models should be based on transcripts not genes!
ChIP-on-chip: What do we learn?

ChIP-on-chip experiment on known TF (plant stem cell regulator)

Motif finding

Expression levels of bound genes

Regulative direction

in silico

Wetlab

Key:

Transcription Factor
Response Regulator
Peptide, Ligand

Models should be based on transcripts not genes!
ChIP-on-chip: What do we learn?

ChIP-on-chip experiment on known TF (plant stem cell regulator)

Motif finding

TF targets

Expression levels of bound genes

Regulative direction

in silico

Wetlab

Key:

Transcription Factor

Response Regulator

activates

Peptide, Ligand

represses

Hormone

Models should be based on transcripts not genes!
ChIP-on-chip: What do we learn?

ChIP-on-chip experiment on known TF (plant stem cell regulator)

Motif finding 
TF targets

Expression levels of bound genes 
regulative direction

Annotation: Are targets TFs? Use existing biological knowledge

Databases in silico Wetlab

Key: Transcription Factor
Response Regulator
activates Peptide, Ligand
represses Hormone

Models should be based on transcripts not genes!
ChIP-on-chip: What do we learn?

- ChIP-on-chip experiment on known TF (plant stem cell regulator)
  - Motif finding
    - TF targets
  - Expression levels of bound genes
    - Regulative direction
  - Annotation: Are targets TFs? Use existing biological knowledge
  - Infer regulatory network
    - Identify putative targets
    - Expand biological knowledge

Key:
- Transcription Factor
- Response Regulator
- Peptide, Ligand
- Hormone

Models should be based on transcripts not genes!
ChIP-on-chip: What do we learn?

- ChIP-on-chip experiment on known TF (plant stem cell regulator)
- Motif finding: TF targets
- Expression levels of bound genes: regulative direction
- Annotation: Are targets TFs? Use existing biological knowledge
- Infer regulatory network: identify putative targets, expand biological knowledge

Key:
- CLAVATA3
- WUSCHEL
- BHLH88
- HAPs
- AGAMOUS
- PERIANTHIA
- Cytokinin
- A-type ARR

Models should be based on transcripts not genes!
ChIP-on-chip: What do we learn?

ChIP-on-chip experiment on known TF (plant stem cell regulator)

Motif finding
TF targets

Expression levels of bound genes
regulative direction

Annotation: Are targets TFs? Use existing biological knowledge

Infer regulatory network
Identify putative targets
Expand biol. knowledge

Key:
Transcription Factor
Response Regulator
activates
Peptide, Ligand
represses
Hormone

Databases  in silico  Wetlab

Models should be based on transcripts not genes!
ChIP-on-chip: What do we learn?

ChIP-on-chip experiment on known TF (plant stem cell regulator)

Motif finding & TF targets
Expression levels of bound genes & regulative direction

Annotation: Are targets TFs? Use existing biological knowledge

Infer regulatory network
Identify putative targets
Expand biological knowledge

Databases in silico Wetlab

Key:
- Transcription Factor
- Response Regulator
- Peptide, Ligand
- Hormone

Models should be based on transcripts not genes!
Summary & Conclusions

- Methods for characterizing transcriptomes
  - in different organisms
  - under different conditions (development/environment)

- Gene finding methods
  - Improved accuracy due to novel inference methods
  - Limitations: no alternative transcripts or expression information

- Analysis of tiling array data & short reads
  - Identification of alternative and differential splicing
  - Segmentation of tiling array data to identify transcribed regions
  - Read alignments difficult, but very promising data

- Combination of predictions and transcriptome measurements
  - Lead to improved gene finding
  - Condition specific transcriptome predictions
  - Help to uncover the full complexity of transcriptomes
Methods for characterizing transcriptomes
- in different organisms
- under different conditions (development/environment)

Gene finding methods
- Improved accuracy due to novel inference methods
- Limitations: no alternative transcripts or expression information

Analysis of tiling array data & short reads
- Identification of alternative and differential splicing
- Segmentation of tiling array data to identify transcribed regions
- Read alignments difficult, but very promising data

Combination of predictions and transcriptome measurements
- Lead to improved gene finding
- Condition specific transcriptome predictions
- Help to uncover the full complexity of transcriptomes
Summary & Conclusions

- Methods for characterizing transcriptomes
  - in different organisms
  - under different conditions (development/environment)

- Gene finding methods
  - Improved accuracy due to novel inference methods
  - Limitations: no alternative transcripts or expression information

- Analysis of tiling array data & short reads
  - Identification of alternative and differential splicing
  - Segmentation of tiling array data to identify transcribed regions
  - Read alignments difficult, but very promising data

- Combination of predictions and transcriptome measurements
  - Lead to improved gene finding
  - Condition specific transcriptome predictions
  - Help to uncover the full complexity of transcriptomes
Methods for characterizing transcriptomes

- in different organisms
- under different conditions (development/environment)

Gene finding methods

- Improved accuracy due to novel inference methods
- Limitations: no alternative transcripts or expression information

Analysis of tiling array data & short reads

- Identification of alternative and differential splicing
- Segmentation of tiling array data to identify transcribed regions
- Read alignments difficult, but very promising data

Combination of predictions and transcriptome measurements

- Lead to improved gene finding
- Condition specific transcriptome predictions
- Help to uncover the full complexity of transcriptomes
# Acknowledgments

## Gene Finding
- **Gabi Schweikert** (FML)
- **Jonas Behr** (FML)
- **Alex Zien** (FML & FIRST)
- **Georg Zeller** (FML & MPI)

## Tiling Arrays
- **Georg Zeller** (FML & MPI)
- **Johannes Eichner** (FML)
- **Sascha Laubinger** (MPI)
- **Detlef Weigel** (MPI)

## Short Read Analysis
- **Fabio De Bona** (FML)
- **Stephan Ossowski** (MPI)
- **Korbinian Schneeberger** (MPI)

## CHiP-on-Chip
- **Sebastian Schultheiss** (FML & Uni HD)
- **Jan Lohmann** (Uni HD)

---

**More Information**
- [http://www.fml.mpg.de/raetsch](http://www.fml.mpg.de/raetsch)
- Slides with references are available online
Acknowledgments

Gene Finding
- Gabi Schweikert (FML)
- Jonas Behr (FML)
- Alex Zien (FML & FIRST)
- Georg Zeller (FML & MPI)

Tiling Arrays
- Georg Zeller (FML & MPI)
- Johannes Eichner (FML)
- Sascha Laubinger (MPI)
- Detlef Weigel (MPI)

Short Read Analysis
- Fabio De Bona (FML)
- Stephan Ossowski (MPI)
- Korbinian Schneeberger (MPI)

CHiP-on-Chip
- Sebastian Schultheiss (FML & Uni HD)
- Jan Lohmann (Uni HD)

More Information
- http://www.fml.mpg.de/raetsch
- Slides with references are available online
Acknowledgments

**Gene Finding**
- Gabi Schweikert (FML)
- Jonas Behr (FML)
- Alex Zien (FML & FIRST)
- Georg Zeller (FML & MPI)

**Tiling Arrays**
- Georg Zeller (FML & MPI)
- Johannes Eichner (FML)
- Sascha Laubinger (MPI)
- Detlef Weigel (MPI)

**Short Read Analysis**
- Fabio De Bona (FML)
- Stephan Ossowski (MPI)
- Korbinian Schneeberger (MPI)

**CHiP-on-Chip**
- Sebastian Schultheiss (FML & Uni HD)
- Jan Lohmann (Uni HD)

**More Information**
- [http://www.fml.mpg.de/raetsch](http://www.fml.mpg.de/raetsch)
- Slides with references are available online
References I


