Today's and Tomorrow's Sequencing Technologies and their Bioinformatic Challenges

Sven Rahmann

Genome Informatics, Institute of Human Genetics
Faculty of Medicine, University of Duisburg-Essen

and

Bioinformatics for High-Throughput Technologies
Computer Science XI, TU Dortmund

Essen, 11.+12.10.2011

Overview

Part 1

Sequencing Technologies

"New" Sequencing Technologies...

...produce a flood of short DNA reads from pepared samples.

Technology Selection

- Roche: 454
- Illumina: Illumina
- Applied Biosystems: SOLiD
- Life Technologies: Ion Torrent
- Pacific Biosciences: SMRT
- Helicos HeliScope

Roche 454

GS FLX with Titanium reagents

- 1 M reads
- 400 bp read length
- run needs 10 h
- 16 regions, 16 barcodes: 256 samples

Roche 454

GS FLX with Titanium reagents

- 1 M reads
- 400 bp read length
- run needs 10 h
- 16 regions, 16 barcodes: 256 samples

GS FLX+ with Titanitum XL+ reagents

- 1 M reads
- 700 bp read length, up to 1 000
- run needs 20 h (?)

Roche 454

GS FLX with Titanium reagents

- 1 M reads
- 400 bp read length
- run needs 10 h
- 16 regions, 16 barcodes: 256 samples

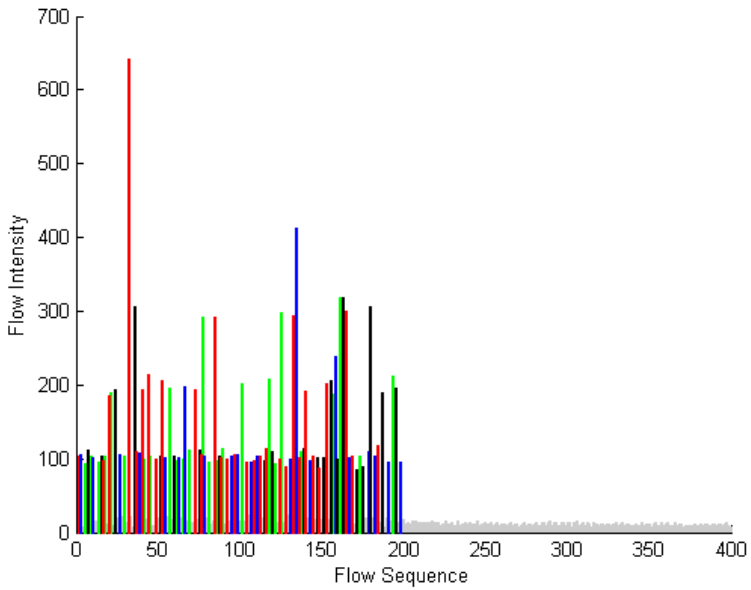
GS FLX+ with Titanitum XL+ reagents

- 1 M reads
- 700 bp read length, up to 1 000
- run needs 20 h (?)

long reads, but few. Throughput $<$ 1 Gbp / day.

Roche 454 Output

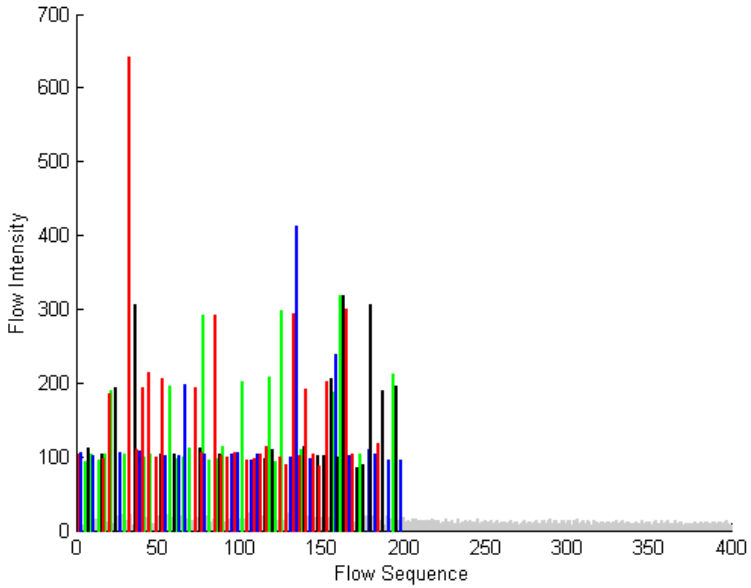
Flowgrams, not sequences!



repeatedly attempt T,A,C,G,. . .
measure light intensity.

Roche 454 Output

Flowgrams, not sequences!

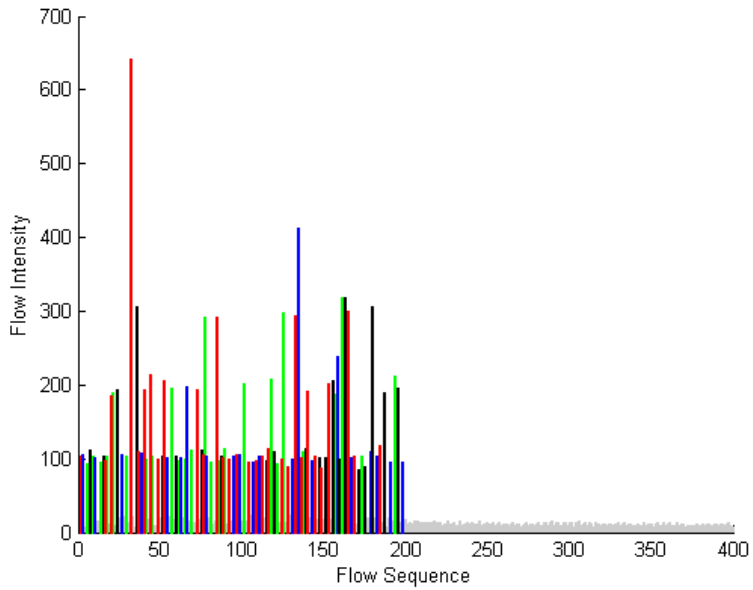


repeatedly attempt T,A,C,G,. . .
measure light intensity.

1 base $\approx$ intensity 100,
proportional for homopolymers.

Roche 454 Output

Flowgrams, not sequences!



repeatedly attempt T,A,C,G,. . .
measure light intensity.

1 base $\approx$ intensity 100,
proportional for homopolymers.

Output for sequence TCAG. . . could be:
(105,3,99,7,0,112,9,103,. . .)

Illumina

Illumina HiSeq 2000

- HiSeq Control Software (HCS) 1.4, Flow cell v3
- up to 3 G reads or read pairs
- up to 2x 100 bp
- run needs up to 11 days
- 8 lanes, 12 barcodes (up to 96 samples)

Illumina

Illumina HiSeq 2000

- HiSeq Control Software (HCS) 1.4, Flow cell v3
- up to 3 G reads or read pairs
- up to 2x 100 bp
- run needs up to 11 days
- 8 lanes, 12 barcodes (up to 96 samples)

short reads, but many. Throughput 50 Gbp / day.

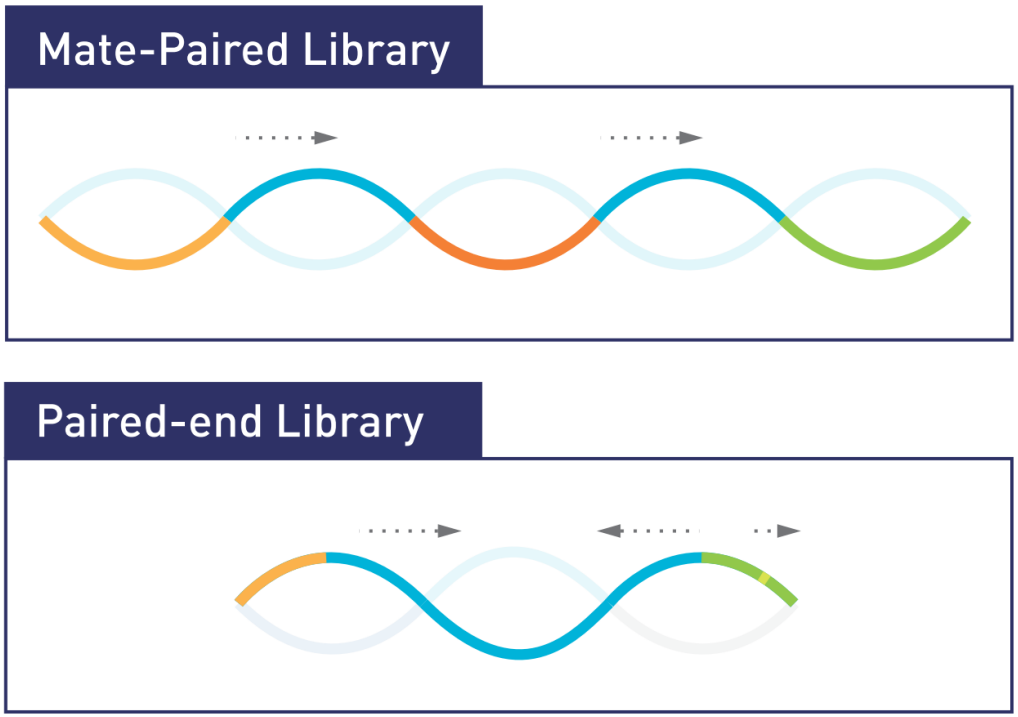
LifeTechnologies (ABI) SOLiD

Current 5500xl System

- number of reads not given
- MP: $60 + 60$ bp; PE: $75 + 35$ bp; F: 75 bp
- run needs up to 3 weeks (12 lanes parallel)
- 2 slides à 6 lanes

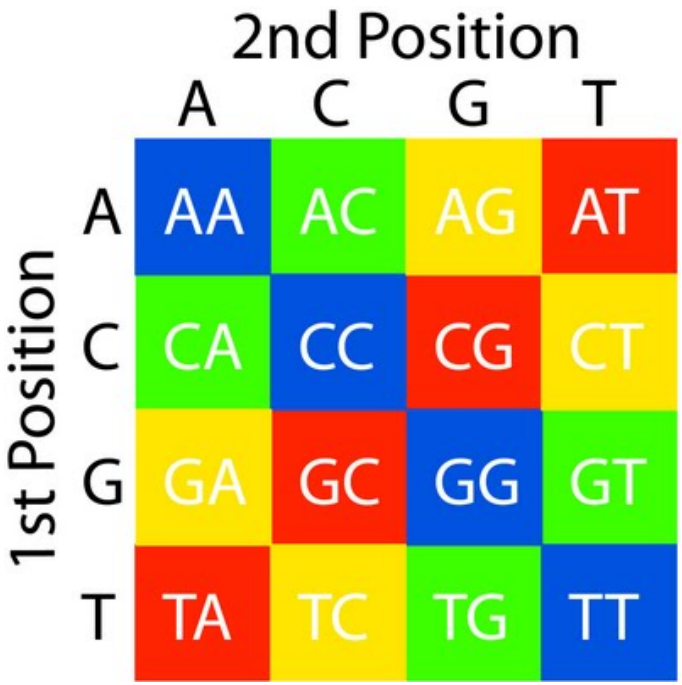
short reads, but many. Throughput up to 20 Gbp / day.

Mate Pairs vs. Paired End Reads



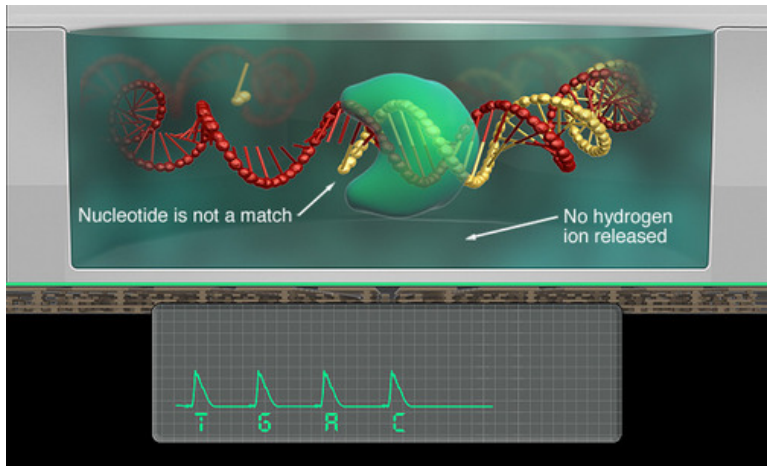
SOLiD Output

No nucleotides, but "colors" encoding di-nucleotides.
Colors represented as numbers 0,1,2,3.
Symmetries: reverse and complement has same color.
Example: T02301 = TTCGGT



LifeTechnologies' IonTorrent PGM

- Personal Genome Machine (PGM) 314, 316, 318
- only a few million reads, but high accuracy
- read length: 250; up to 400 in 2012?
- fastest workflow (prep in 6h, sequence in 2h)
- throughput: $<$ 1 Gbp per day, may scale up
- inexpensive: no fluorescent chemistry;
  detects hydrogen ions on semiconductor chips.



Quality Values

Idea: interpret like phred scores (Ewing & Green, 1998)

quality	error rate	percent ok
Q 10	1 / 10	90 %
Q 20	1 / 100	99 %
Q 30	1 / 1 000	99.9 %
Q 40	1 / 10 000	99.99 %
Q 50	1 / 100 000	99.999 %

Quality Values

Idea: interpret like phred scores (Ewing & Green, 1998)

quality	error rate	percent ok
Q 10	1 / 10	90 %
Q 20	1 / 100	99 %
Q 30	1 / 1 000	99.9 %
Q 40	1 / 10 000	99.99 %
Q 50	1 / 100 000	99.999 %
Q q	$10^{-q/10}$	$100 \cdot (1 - 10^{-q/10})$ %

Quality Values

Idea: interpret like phred scores (Ewing & Green, 1998)

quality	error rate	percent ok
Q 10	1 / 10	90 %
Q 20	1 / 100	99 %
Q 30	1 / 1 000	99.9 %
Q 40	1 / 10 000	99.99 %
Q 50	1 / 100 000	99.999 %
Q q	$10^{-q/10}$	$100 \cdot (1 - 10^{-q/10})$ %

Quality values q often range from 4 to 60.
They are encoded as the ASCII character of $q + 33$ or $q + 64$.

FASTQ format

FASTQ: extension of FASTA to store encoded quality values

```
@EM7LVYS01C1LWG
TCAGGGGGGAGCTTAAATTTGAAACTAGAAAAATTTTGAACAAAATAATCATAATTGTT
+EM7LVYS01C1LWG
=;8GC91*#==<C=EA.EA/<B=(<<:=HC90'FB5&;B:<GC6(=D=<<==C=C==B<
@EM7LVYS01B2EMP
TCAGGGGGGGGTTACACGTGCAGATTTGTTACACGGGTGTACTGTGAGGTTTGGGGTAC
+EM7LVYS01B2EMP
=<8F@71-*&#D=<=<=<===<<:FB1=C=;<=<FA/==<<=<===<D=FB0FB4%<<=
```

Technology Summary

- Different chemistries, measurement technologies, ...
- Different read lengths, number of reads, thoughputs, ...
- Different "native" output
- Output often converted to FASTQ:
  sequence + qualities
- Convenient: uniform interface for analysis

Technology Summary

- Different chemistries, measurement technologies, . . .
- Different read lengths, number of reads, thoughputs, . . .
- Different "native" output
- Output often converted to FASTQ:
  sequence + qualities
- Convenient: uniform interface for analysis
- but **lossy**!

Challenge: Efficiently Use Sequencer State

- Each technology has its specific strengths and weaknesses
- "Primary" output is available:
  - 454, IonTorrent: flowgrams (intensities)
  - SOLiD: color sequence
  - PacBio SMRT (single molecule real time): kinetics

Challenge: Efficiently Use Sequencer State

- Each technology has its specific strengths and weaknesses
- "Primary" output is available:
  - 454, IonTorrent: flowgrams (intensities)
  - SOLiD: color sequence
  - PacBio SMRT (single molecule real time): kinetics
- Develop methods that start from primary data, but:
  - short-lived (technology change)
  - must translate reference genome to primary format
  - combinatorial explosion ?

One Idea: Intermediate Flowgram Format

- 454 outputs flowgrams for fixed flow sequence TACG:
  (105, 3, 99, 7, 0, 112, 9, 103, . . .) means probably TCAG. . .

One Idea: Intermediate Flowgram Format

- 454 outputs flowgrams for fixed flow sequence TACG:
  (105, 3, 99, 7, 0, 112, 9, 103, . . .) means probably TCAG. . .
- What about these flow values?
  - 42 (presence?)
  - 149 (number?)
  - 899 (saturation?)

One Idea: Intermediate Flowgram Format

- 454 outputs flowgrams for fixed flow sequence TACG:
  (105, 3, 99, 7, 0, 112, 9, 103, . . .) means probably TCAG. . .
- What about these flow values?
  - 42 (presence?)
  - 149 (number?)
  - 899 (saturation?)
- Invent additional characters:
  - ACGT: confirmed nucleotides
  - acgt: potentially existing nucleotides
  - +: additional copies of preceding nucleotide

One Idea: Intermediate Flowgram Format

- 454 outputs flowgrams for fixed flow sequence TACG:
  (105, 3, 99, 7, 0, 112, 9, 103, . . .) means probably TCAG. . .
- What about these flow values?
  - 42 (presence?)
  - 149 (number?)
  - 899 (saturation?)
- Invent additional characters:
  - ACGT: confirmed nucleotides
  - acgt: potentially existing nucleotides
  - +: additional copies of preceding nucleotide
- Ex: (105, 35, 157, 12, 999, 99) = TaCcTTTTTT+A

One Idea: Intermediate Flowgram Format

- 454 outputs flowgrams for fixed flow sequence TACG:
  (105, 3, 99, 7, 0, 112, 9, 103, . . .) means probably TCAG. . .
- What about these flow values?
  - 42 (presence?)
  - 149 (number?)
  - 899 (saturation?)
- Invent additional characters:
  - ACGT: confirmed nucleotides
  - acgt: potentially existing nucleotides
  - +: additional copies of preceding nucleotide
- Ex: (105, 35, 157, 12, 999, 99) = TaCcTTTTTT+A
- Allows to use sequence alignment against reference
  - insert before + is cheap
  - deletion of acgt+ is cheap

One Million Dollar Challenge

Double IonTorrent Personal Genomics Machine's accuracy[1]:

- Pure algorithm / software challenge
- Use raw / primary data
- Produce more accurate FASTQ file than PGM does now

[1] http://www.lifetechnologies.com/about-life-technologies/grand-challenges/accuracy.html

Part 2

Biomedical Questions Addressed by High-Throughput Sequencing Technologies

Genomics

- De novo genome sequencing
- Genome resequencing

Genomics

- De novo genome sequencing
- Genome resequencing
- Variation discovery: e.g. SNP and CNV discovery; selected parts only: exome sequencing

Genomics

- De novo genome sequencing
- Genome resequencing
- Variation discovery: e.g. SNP and CNV discovery; selected parts only: exome sequencing
- Description of the pan-genome of a species

Genomics

- De novo genome sequencing
- Genome resequencing
- Variation discovery: e.g. SNP and CNV discovery; selected parts only: exome sequencing
- Description of the pan-genome of a species
- Population sequencing (e.g., virus population in host)

Genomics

- De novo genome sequencing
- Genome resequencing
- Variation discovery: e.g. SNP and CNV discovery; selected parts only: exome sequencing
- Description of the pan-genome of a species
- Population sequencing (e.g., virus population in host)
- Amplicon sequencing (ultra deep; rare variants)

Genomics

- De novo genome sequencing
- Genome resequencing
- Variation discovery: e.g. SNP and CNV discovery;
  selected parts only: exome sequencing
- Description of the pan-genome of a species
- Population sequencing (e.g., virus population in host)
- Amplicon sequencing (ultra deep; rare variants)
- Metagenomics:
  Genomic composition of ecological communities;
  Implications for inter-species relations

Genomics

- De novo genome sequencing
- Genome resequencing
- Variation discovery: e.g. SNP and CNV discovery;
  selected parts only: exome sequencing
- Description of the pan-genome of a species
- Population sequencing (e.g., virus population in host)
- Amplicon sequencing (ultra deep; rare variants)
- Metagenomics:
  Genomic composition of ecological communities;
  Implications for inter-species relations

Aim for (at least) some long reads (454, Sanger).
Use paired end reads with long insert sizes.

Transcriptomics (RNA-seq)

. . . will, in the long run, replace DNA microarrays.

- Transcriptome discovery / description:
  - exons
  - all splice variants of a gene
  - (small) noncoding RNA

Transcriptomics (RNA-seq)

. . . will, in the long run, replace DNA microarrays.

- Transcriptome discovery / description:
  - exons
  - all splice variants of a gene
  - (small) noncoding RNA
- Gene/exon/transcript expression analysis (binary)
- Gene/exon/transcript expression analysis (quantitative)
- Noncoding RNA expression analysis

Transcriptomics (RNA-seq)

. . . will, in the long run, replace DNA microarrays.

- Transcriptome discovery / description:
  - exons
  - all splice variants of a gene
  - (small) noncoding RNA
- Gene/exon/transcript expression analysis (binary)
- Gene/exon/transcript expression analysis (quantitative)
- Noncoding RNA expression analysis

Number of reads more important than length.
Splicing: Use paired-end reads.

Approaches to "Regulomics"

- ChIP-seq:
  sequence DNA bound to a protein (transcription factor)
  $\Rightarrow$ transcription factor binding sites and motifs

Approaches to "Regulomics"

- ChIP-seq:
  sequence DNA bound to a protein (transcription factor)
  $\Rightarrow$ transcription factor binding sites and motifs
- Determination of epigenetic modifications,
  e.g. basepair-level DNA methylation state:
  Bisulfite sequencing / SMRT sequencing

Part 3

Basic Bioinformatics: Read Mapping

Read Mapping: A Fundamental Task

Read Mapping Problem

Given:

1. short DNA sequence read (string),
2. per-base quality values
3. reference sequence (string)
4. error rate threshold

Read Mapping: A Fundamental Task

Read Mapping Problem

Given:

1. short DNA sequence read (string),
2. per-base quality values
3. reference sequence (string)
4. error rate threshold

Sought:

- locations in reference where read occurs, (quality-weighted) error rate below threshold ($\approx$ classical approximate matching / alignment.)

Read Mapping: A Fundamental Task

Read Mapping Problem

Given:

1. short DNA sequence read (string),
2. per-base quality values
3. reference sequence (string)
4. error rate threshold

Sought:

- locations in reference where read occurs, (quality-weighted) error rate below threshold ($\approx$ classical approximate matching / alignment.)

Variations:

- local alignments for long reads
- read ends may be adapters,
- spliced alignment across exon boundaries

Issues with the State of the Art

- Speed of read mapping: index necessary
  Indexing helps to locate **exact matches** of read substrings.
  1. index (parts of) the reference
  2. index (parts of) the reads

Issues with the State of the Art

- Speed of read mapping: index necessary
  Indexing helps to locate **exact matches** of read substrings.
  1. index (parts of) the reference
  2. index (parts of) the reads
- Size of reference & data (and indexes): GBs!
  32 bits limit us to addressing 2 GB,
  suffix array of human genome needs 24 GB (uncompressed).
  Usage of 64 bits doubles space requirements.

Issues with the State of the Art

- Speed of read mapping: index necessary
  Indexing helps to locate **exact matches** of read substrings.
  1. index (parts of) the reference
  2. index (parts of) the reads
- Size of reference & data (and indexes): GBs!
  32 bits limit us to addressing 2 GB,
  suffix array of human genome needs 24 GB (uncompressed).
  Usage of 64 bits doubles space requirements.
- Guarantees of read mapping (exact vs. heuristic)

Issues with the State of the Art

- Speed of read mapping: index necessary
  Indexing helps to locate **exact matches** of read substrings.
  1. index (parts of) the reference
  2. index (parts of) the reads
- Size of reference & data (and indexes): GBs!
  32 bits limit us to addressing 2 GB,
  suffix array of human genome needs 24 GB (uncompressed).
  Usage of 64 bits doubles space requirements.
- Guarantees of read mapping (exact vs. heuristic)
- Dealing with multiple matching loci
  (one best, all best, all suboptimal):
  effects on downstream analysis; repeats.

Indexing

Indexing the Reference

- Reference does not change over time.
- Reference is smaller than sum of reads.
- Reads must be considered sequentially anyway.

Indexing

Indexing the Reference

- Reference does not change over time.
- Reference is smaller than sum of reads.
- Reads must be considered sequentially anyway.

Indexing the Reads

- Scan over reference (genome) once
- Large data (600 GB), even larger index (5 TB)
- Index might be constructed during sequencing!

Indexing Short Exact Matches

Index Data Structures

- q-gram index
- hash-based index
- (enhanced, extended) suffix array
- compressed self-index

Indexing Short Exact Matches

Index Data Structures

- q-gram index
- hash-based index
- (enhanced, extended) suffix array
- compressed self-index

Idea: Fast Filtration

Appropriate choice of q implies:

- No exact q-gram match $\Rightarrow$ no good alignment
- However: exact q-gram match $\not\Rightarrow$ good alignment
- Necessary to verify matches with full alignment.

Indexing Short Exact Matches

q-gram index

- Read a q-gram as a base-4 number with q digits
- Example: AGTTCA $\mapsto$
  $(023310)_4 = 0 \cdot 1 + 1 \cdot 4 + 3 \cdot 16 + 3 \cdot 64 + 2 \cdot 256 + 0 \cdot 1024 = 756$
- 1-to-1 correspondence: q-grams $\leftrightarrow$ integers $\{0, \ldots, 4^q - 1\}$
- For each number, record positions of q-gam in reference

Indexing Short Exact Matches

q-gram index

- Read a q-gram as a base-4 number with q digits
- Example: AGTTCA $\mapsto$
  $(023310)_4 = 0 \cdot 1 + 1 \cdot 4 + 3 \cdot 16 + 3 \cdot 64 + 2 \cdot 256 + 0 \cdot 1024 = 756$
- 1-to-1 correspondence: q-grams $\leftrightarrow$ integers $\{0, \ldots, 4^q - 1\}$
- For each number, record positions of q-gam in reference

Hashing

- Define a hash function h
  that assigns a number $h(x)$ to each q-gram x
- Different $x \neq y$ may have same hash value $h(x) = h(y)$
- Often: larger q, ignore some characters

Indexing Short Exact Matches

Suffix array

Given a text T of length n,
the suffix array $\mathtt{pos}$ is a permutation of $\{1, \ldots, n\}$,
such that the suffix starting at position $\mathtt{pos}[r]$ is the r-th lexicographically smallest one.

Example: AGTTCA\$ has $\mathtt{pos} = (7, 6, 1, 5, 2, 4, 3)$ with suffixes (\$, A\$, AGTTCA\$, CA\$, GTTCA\$, TCA\$, TTCA\$).

Indexing Short Exact Matches

Suffix array

Given a text T of length n,
the suffix array $\texttt{pos}$ is a permutation of $\{1, \ldots, n\}$,
such that the suffix starting at position $\texttt{pos}[r]$ is the r-th lexicographically smallest one.

Example: AGTTCA\$ has $\texttt{pos} = (7, 6, 1, 5, 2, 4, 3)$ with suffixes (\$, A\$, AGTTCA\$, CA\$, GTTCA\$, TCA\$, TTCA\$).

Matches correspond to intervals of suffix array.
Example: T is found at $r \in [6, 7]$ with positions 4, 3.

Indexing Short Exact Matches

Suffix array

Given a text T of length n,
the suffix array $\texttt{pos}$ is a permutation of $\{1, \ldots, n\}$,
such that the suffix starting at position $\texttt{pos}[r]$ is the r-th lexicographically smallest one.

Example: AGTTCA\$ has $\texttt{pos} = (7, 6, 1, 5, 2, 4, 3)$ with suffixes (\$, A\$, AGTTCA\$, CA\$, GTTCA\$, TCA\$, TTCA\$).

Matches correspond to intervals of suffix array.
Example: T is found at $r \in [6, 7]$ with positions 4, 3.

"BWT" + "Backward search" obtain interval with little memory.
Position list $\texttt{pos}$ must be scanned on disk.

Challenge

Mismatch- and Difference-tolerant Indexing

Develop an index that locates matches

- with Hamming distance ≤ 1 or edit distance ≤ 1
- rapidly
- without wasting memory
- with few false hash hits

Challenge

Mismatch- and Difference-tolerant Indexing

Develop an index that locates matches

- with Hamming distance ≤ 1 or edit distance ≤ 1
- rapidly
- without wasting memory
- with few false hash hits

Requires engineering appropriate hash functions.
Requires understanding statistics of hash functions.

Challenge

Technology-Dependent Indexing

- 454 & IonTorrent sequence a **run** of nucleotides at a time: TAAGTCCCA = (T, AA, G, T, CCC, A).
- Unable to determine exact length of a long run: AAAAAA $\approx$ AAAAAAA

Challenge

Technology-Dependent Indexing

- 454 & IonTorrent sequence a **run** of nucleotides at a time: TAAGTCCCA = (T, AA, G, T, CCC, A).
- Unable to determine exact length of a long run: AAAAAA $\approx$ AAAAAAA
- Idea: Ignore run length completely!
- Transform reference and reads by "forgetting": TAAGTCCCA = (T, AA, G, T, CCC, A) $\mapsto$ TAGTCA

Challenge

Technology-Dependent Indexing

- 454 & IonTorrent sequence a **run** of nucleotides at a time: TAAGTCCCA = (T, AA, G, T, CCC, A).
- Unable to determine exact length of a long run: AAAAAA $\approx$ AAAAAAA
- Idea: Ignore run length completely!
- Transform reference and reads by "forgetting": TAAGTCCCA = (T, AA, G, T, CCC, A) $\mapsto$ TAGTCA
- No two adjacent characters are equal.
- Build indexing / hash function on this property.
- Effective alphabet size: 3 instead of 4

Challenge

Genome Assembly and Overlap Detection

- Illumina: up to $6 \cdot 10^9$ reads
- $3.6 \cdot 10^{19}$ potentially overlapping pairs
- Finding all overlapping read pairs is time-consuming!

Challenge

Genome Assembly and Overlap Detection

- Illumina: up to $6 \cdot 10^9$ reads
- $3.6 \cdot 10^{19}$ potentially overlapping pairs
- Finding all overlapping read pairs is time-consuming!
- Use specialized hardware:
  - GPGPU (graphics card) programming
  - FPGAs
  - SSDs for fast index access

  Problems: data transfer, programming tools

Part 4

Specific Bioinformatics Problems: Case Studies

Bisulfite Sequencing of CpG Islands (454)

Goal

Determination of methlyation state in CpG islands

454-Technology: Pros and Cons

- relatively long reads
- bisulfite treatment (meth-C $\mapsto$ C, but C $\mapsto$ U=T) works
- errors primiarily run lengths: $\mathtt{TTTTTT} \approx \mathtt{TTTTTTT}$
- problem: (close to) 3-letter alphabet, long runs

Read Mapping

In parallel against two genomes: bisulfite-treated, untreated
Variant: Shorten runs to one character in genomes and in reads

Task: Library Optimization

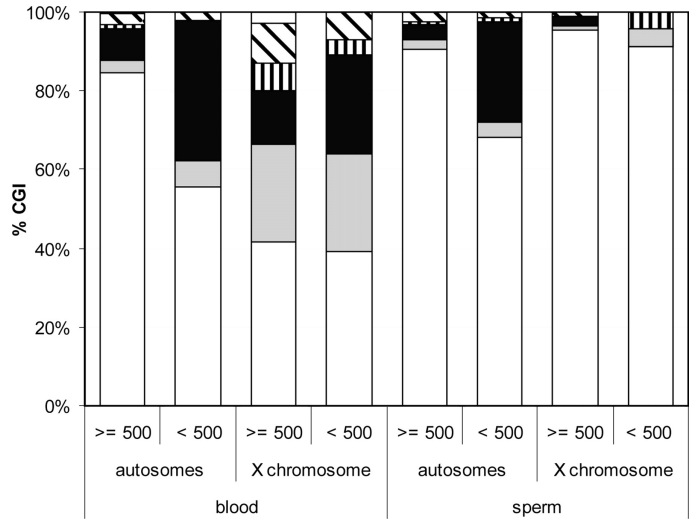
Goal: Sequencing of CpG islands.

Obtain them by restriction enzymes and length selection.

In-silico-optimization by simulation:

Experiment Number	Enzyme Combination			Fragment Length Min	Fragment Length Max	Number of Distinct Fragments	CGI Score	Alu Score	#Fragments in MinLength -5% to +5%	#Fragments in MaxLength -5% to +5%	Fragments in [Min, Max] hitting a CGI	CGIs hit by Fragments in [Min, Max]
0	MseI	Tsp509		613	800	99388	5473	21472	53501	25524	4344	4035
1	MseI	Tsp509	AluI	436	800	99890	15652	16442	44360	3868	10785	9204
2	MseI	Tsp509	NlaIII	463	800	99529	18384	15019	44658	5808	13264	10778
3	MseI	Tsp509	BfaI	532	800	99980	11400	18664	43833	12026	8349	7416
4	MseI	Tsp509	HpyCH4	436	800	99207	18142	12440	43068	4189	12537	10417
5	MseI	Tsp509	DpuI	536	800	99748	11740	11799	44704	13026	8636	7623
6	MseI	Tsp509	MboII	573	800	99255	9250	20326	48504	17360	7040	6343
7	MseI	Tsp509	MlyI	588	800	99642	8597	17305	51587	19675	6544	5921
8	MseI	Tsp509	BCCI	574	800	99162	9387	18362	47642	17366	7158	6461
9	MseI	Tsp509	AluI	NlaIII	337	800	99463	22451	15126	44353	1364	14368	11217
10	MseI	Tsp509	AluI	BfaI	381	800	99403	17455	17371	45858	2067	11490	9539
11	MseI	Tsp509	AluI	HpyCH4	332	800	99268	19141	14561	47417	1178	11975	9672
12	MseI	Tsp509	AluI	DpuI	378	800	99736	16823	11089	43809	2129	11005	9223
13	MseI	Tsp509	AluI	MboII	400	800	99865	16930	17280	46676	2655	11283	9479
14	MseI	Tsp509	AluI	MlyI	412	800	99568	15229	15229	45205	2862	10340	8851
15	MseI	Tsp509	AluI	BCCI	401	800	99173	16790	16466	47211	2362	11203	9450
16	MseI	Tsp509	NlaIII	BfaI	398	800	99090	22430	16298	40045	2690	15153	11901
17	MseI	Tsp509	NlaIII	HpyCH4	346	800	99187	25121	9605	45792	1595	16312	12482
18	MseI	Tsp509	NlaIII	DpuI	402	800	99102	21570	10010	44201	3039	14649	11599
19	MseI	Tsp509	NlaIII	MboII	425	800	99924	20966	16022	43109	3828	14623	11713
20	MseI	Tsp509	NlaIII	MlyI	437	800	99464	19648	12975	42603	4127	13774	11134
21	MseI	Tsp509	NlaIII	BCCI	424	800	99988	21247	15236	43892	3688	14814	11731
22	MseI	Tsp509	BfaI	HpyCH4	372	800	98930	21213	13141	43407	2043	13845	11133
23	MseI	Tsp509	BfaI	DpuI	456	800	99422	15632	10735	40167	5825	10894	9294
24	MseI	Tsp509	BfaI	MboII	490	800	99888	14261	19726	42657	7703	10075	8700
27	MseI	Tsp509	HpyCH4	DpuI	378	800	99084	19885	8165	42925	2093	13052	10639

Many CpG islands on chr X incompletely methylated



read start	pattern	read start	pattern	read start	pattern	read start	pattern
149279998	○●●	73672170	●●●○○●●	77245759	●○●●○○●●●○	48419335	○○●●●●○●○●●●●●
68030965	○●●	73672170	●●●○○●●	153339088	○●●○●○●○●○	48419335	○○●●●●○●○●●●●●
152900492	○●○●	144706855	●○●○○●○	47402759	●●●○●●●○○●	118485986	●●●○●●○●●●○○●●●
48569581	●○○○	95826086	●●●●○○●	73672170	●●○○○●○●●●	135160359	●●○●○○●●○●●○●●●
46190993	●●●○○	73672170	●●●○○●●	90576058	●●○●●●○○○○	48899399	○○○○○○○●●●●○●●●●
153397205	○○●●○●	100193070	○○○●●○●●	83328852	●●○○○○○●○○	31194225	●○○○○○○○○○○●○●●○
150096073	○○○●●●	12902828	●○○○●●○○	152804010	●●●●●○○○○○	39564884	○○○○○○○○●●○○●●●○
48797670	●●●○○○	39915562	○○○○○●●○	47363414	●○●○○●●●●●	135160359	●●●○●○○○●●○●●○●●●
153371648	●○○●○●	100193070	○○○●●○●●	83328852	●●○○○○○●○○	46317798	○○●●●○●●●●●●●●●○●○●
46190993	●●●○○○	39915562	○○○○○●●○	74060377	○○●●●○○○○○○	152528293	●●○○●●○●○●●○●○●○○●○○●●●
153371648	●○○●○●	100070333	○○●●○●●●	47267565	○○○●●○○●●●●	74659290	○○○●●●●○●●○●●○●○○○●○○○●
30236160	○○●○●●	138841328	○○○●●○●○○	133510640	○●●●○○○●●○○	124164746	●○●○○○●●○●●●●●●○●●●●●●●●●○●
153371648	●○○●○●	68640256	●●○●○○○○●	24940182	○○○○●●○●●○●○	68674716	○○○●●○●○○○○●●○○○●●●●○●●○○●○●●●
133965547	●●●○○○○	150096073	○○●●●○●○○	153176735	○○●●●○●○●●●○	118416991	●●●○○●●●●●●●●●●●○○○●○●●●○○●●●●
67829824	●○●●●●○	68640256	●●○●○○○○●	133506061	●○●●○●○●○○○○	48977071	●●●●●●○○○○○●●●●●●●○○○○○●○●●●○●●●
108754218	●○○○●○●	118889079	●○○●●●○●○	24940182	○○○○●●○●●○●○	48977071	●●●●●●○○○○○●●●●●●●○○○○○●●○●●●○●●●
36885226	●●●○○○●	102205057	●○●○○●○●●	139413333	○○○○●○●○●●○○●○	135676496	●●○●●●●●●○○●●●●○○○○○○○○○●●○○●●○●●●
73672170	●●●○○●●	102205057	●○●○○●○●●	39949487	○●○○○○●●●●●●●○		

miRNA Expression in Neuroblastoma (SOLiD)

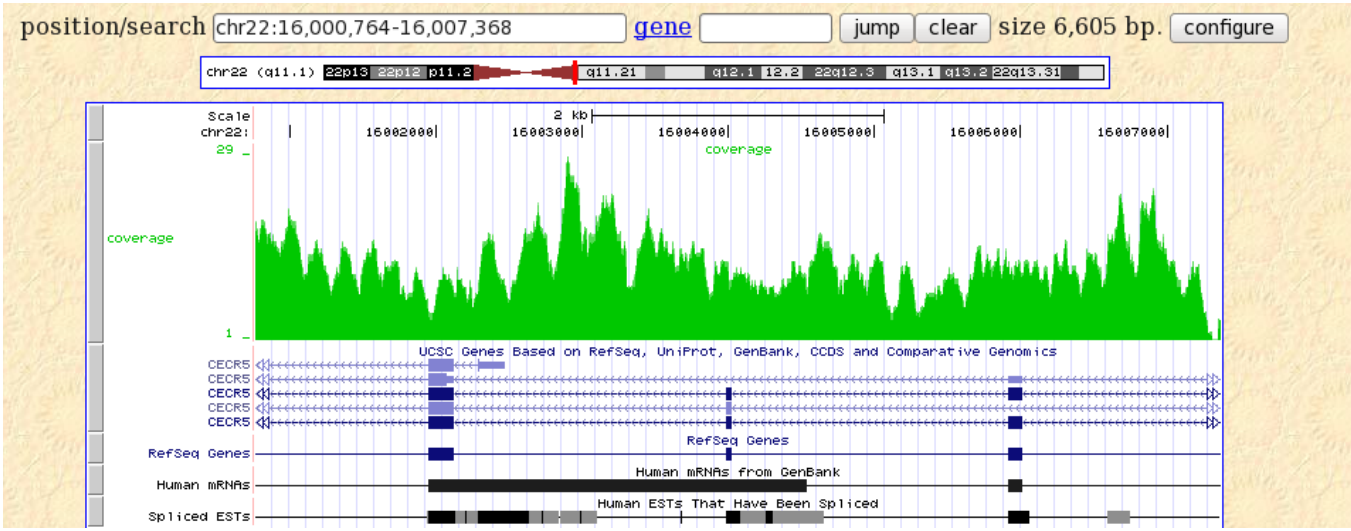
SOLiD: short reads (35 bp), ideal for short non-coding RNAs
dinucleotide color space

1. read mapping
2. classification of reads into RNA categories
3. quantification of miRNA expression:
   normalization method
4. differential expression between neuroblastoma subtypes?
   detection of weak differential expression
5. miRNA-Editing?
6. discovery of two new miRNAs: now in miRbase

Schulte, . . . , SR, Schramm; *Nucleic Acids Research*, 2010.

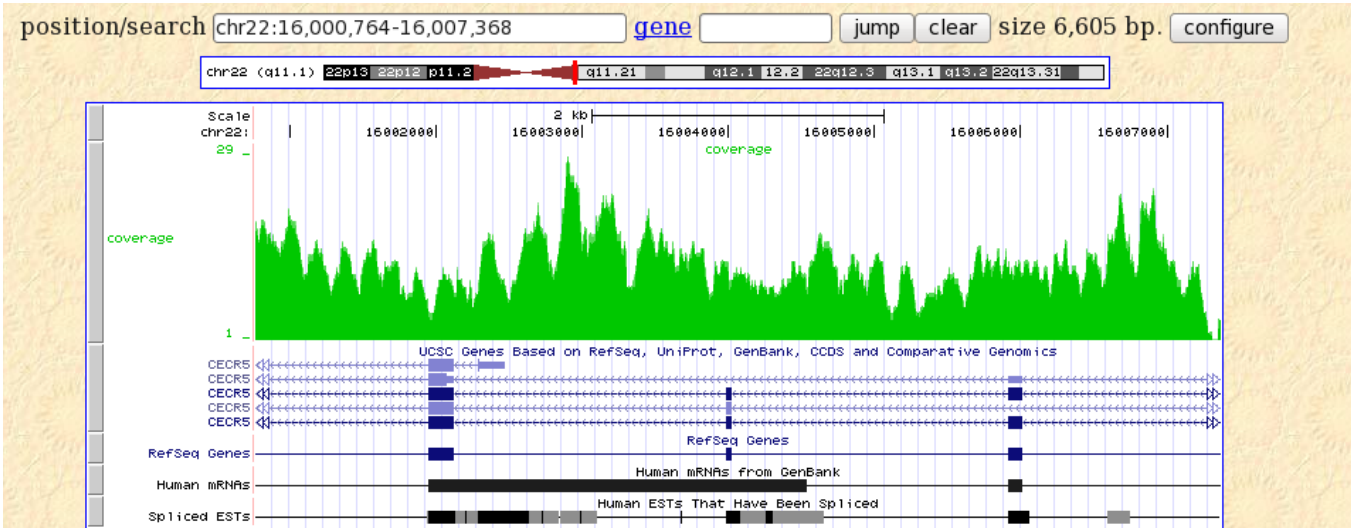
Statistical Challenges

- Uneven coverage in genome and transcriptome sequencing: Explain and correct library bias



Statistical Challenges

- Uneven coverage in genome and transcriptome sequencing: Explain and correct library bias



- Test for gene / exon / transcript presence
- Quantify gene / exon / transcript expression

Statistical Challenges

- Normalization between experiments
- Detection power of differential expression with few samples
- Statistical significance with multiple testing (e.g. exons)
- number of samples $n \ll p$ number of tests

Statistical Challenges

- Normalization between experiments
- Detection power of differential expression with few samples
- Statistical significance with multiple testing (e.g. exons)
- number of samples $n \ll p$ number of tests
- Solutions:
  - more samples!
  - independent confirmation
  - use external information (gene/protein networks)
  - exploit known dependencies (e.g. regulation)

Challenge: Data Structures for Pangenomes

Situation around 2000

Human genome almost done. Nothing left do do...

Challenge: Data Structures for Pangenomes

Situation around 2000

Human genome almost done. Nothing left do do...

Current "Genome" Projects

- 1000-Genomes-Project (human pangenome): over 3 Tbp sequences
- International Cancer Genome Consortium
- Human Gut Metagenome Initiative (100 bacteria per human cell, gene pool 100x bigger)

Image: M. Gerstenberg
Die ZEIT (12/2006)

Pangenome := entirety of genetic information of a species
Metagenome := $\sim$ of a community

Questions to a pangenome

- What's the genome of the 334-th sequenced person?
- How often and where does the motif TATAAW occur?
- Which variants of the dopamine D2 receptor gene exist?
- Which variables do these variants correlate with?

Questions to a pangenome

- What's the genome of the 334-th sequenced person?
- How often and where does the motif TATAAW occur?
- Which variants of the dopamine D2 receptor gene exist?
- Which variables do these variants correlate with?

Which data structures provide this information?

- lossless sequence representation
- fast search (index based), also approximate
- representation of consensus and variations (e.g., SNPs, CNVs)
- representation of rearrangements, repeats
- generalization ability
- integration of annotation and semantics

Questions to a pangenome

- What's the genome of the 334-th sequenced person?
- How often and where does the motif TATAAW occur?
- Which variants of the dopamine D2 receptor gene exist?
- Which variables do these variants correlate with?

Which data structures provide this information?

- lossless sequence representation
- fast search (index based), also approximate
- representation of consensus and variations (e.g., SNPs, CNVs)
- representation of rearrangements, repeats
- generalization ability
- integration of annotation and semantics

No existing ones!

Summary of Key Challenges

1. Use raw data / sequencer state information (instead of common FASTQ files)
   - +: no information loss
   - +: save conversion steps
   - –: too many combinations technology × application
   - –: technology short-lived

Summary of Key Challenges

1. Use raw data / sequencer state information (instead of common FASTQ files)
   - +: no information loss
   - +: save conversion steps
   - –: too many combinations technology × application
   - –: technology short-lived
2. New ideas in sequence indexing:
   - direct use of sequencer state information
   - error tolerance

Summary of Key Challenges

1. Use raw data / sequencer state information (instead of common FASTQ files)
   - +: no information loss
   - +: save conversion steps
   - –: too many combinations technology × application
   - –: technology short-lived
2. New ideas in sequence indexing:
   - direct use of sequencer state information
   - error tolerance
3. Special hardware for read mapping: GPGPUs, FPGAs

Summary of Key Challenges

1. Use raw data / sequencer state information (instead of common FASTQ files)
   - +: no information loss
   - +: save conversion steps
   - –: too many combinations technology × application
   - –: technology short-lived
2. New ideas in sequence indexing:
   - direct use of sequencer state information
   - error tolerance
3. Special hardware for read mapping: GPGPUs, FPGAs
4. Statistics:
   - bias correction
   - normalization
   - multiple testing, $n \ll p$

Summary of Key Challenges

1. Use raw data / sequencer state information (instead of common FASTQ files)
   - +: no information loss
   - +: save conversion steps
   - –: too many combinations technology × application
   - –: technology short-lived
2. New ideas in sequence indexing:
   - direct use of sequencer state information
   - error tolerance
3. Special hardware for read mapping: GPGPUs, FPGAs
4. Statistics:
   - bias correction
   - normalization
   - multiple testing, $n \ll p$
5. Data structures for variations / pangenomes