Accelerating bioinformatics tools for high-throughput DNA sequence analysis

Alexander Goesmann
Computational Genomics
Bioinformatics Resource Facility
Center for Biotechnology
Bielefeld University

Paderborn, 14.3.2013
Advances in genome & transcriptome research driven by novel sequencing methods
High-throughput Sequencing

Roche GS FLX+ - 2008
- ~700 bases / read
- 1 GB / run
- Run-time 10+ h
- High costs
- Homopolymers

Illumina GAIIx - 2009
- Up to 2 x 150 bases
- 95 GB / run
- Run-time up to 14 d
- Low costs
- No homopolymers

Ion Torrent PGM - 2011
- ~400 bases/read
- 1 GB / run
- Run-time 3 h
- Low costs
- Homopolymers

Illumina MiSeq - 2012
- Up to 2 x 250 bases
- 6 GB / run
- Run-time up to 27 h
- Low costs
- No homopolymers
Accelerating bioinformatics tools for high-throughput sequence analysis

Illumina HiSeq

Illumina HiSeq 1000 - 2013

- Up to 2 x 100 bases
- 300 GB / run
- 3 billion reads
- Run-time up to 10 days
- Very low costs
- No homopolymers
Accelerating bioinformatics tools for high-throughput sequence analysis

Genome Sequencing Ten Years Ago

- **Sinorhizobium meliloti:**
  - size: 6.7 Mb
  - duration: three years
  - costs: about 3 Mio Euro

- **Corynebacterium glutamicum:**
  - size: 3.3 Mb
  - duration: 2.5 years
  - costs: about 1.5 Mio Euro

- **Human Genome Project:**
  - size: 3,000 Mb
  - duration: 13 years
  - costs: about 300 Mio $
**Genome Sequencing Today**

- **Corynebacterium kroppenstedtii**
  - 2.44 Mb
  - one GS FLX standard sequencing run
  - two weeks
  - ~ 20,000 Euro (year 2007)

- **James Watson (454 Life Sciences, Roche)**
  - 3.000 Mb
  - 234 GS FLX standard sequencing runs (7.4 coverage)
  - two months
  - 2 Mio $ (year 2007)

- **MiSeq 2013**
  - 20 bacterial genomes with 3 Mb
  - One day
  - ~ 3.000 Euro
Accelerating bioinformatics tools for high-throughput sequence analysis

DNA – Nothing but 454 letters?

gtgagccgagaactcatcttttttgcacggaacctggcagcgcaagtgttgccgatctcacaaccttttgagccagcaagcgcagctggtgaccttcggaggggttcgtctggtggctgtttgtgtgctgtgctgtctggtcgacgacgcgatgcgcgagtcgacgttgctgtttgtgtgctgtggtctggttgctgtggttgtgttgttgtgttgtgtgtgtgtgtgtgtgtgtgtgtgtggtgtgtgtgtgtgtgtgtgtggtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtg
EMBL/GenBank/DDBJ growth statistics

- **EMBL-Bank Growth**
  - 25-Jun-2012
  - Emmbl Release 112 (31 May 2012):
    - 247,335,689 sequence entries / 429,512,389,024 nucleotides
    - Total: 201 GB compressed and 1.4 TB uncompressed data

  **Sequences (248.8 millions) — Bases (435.1 billions)**

- **Sequence Read Archive (SRA) Growth**
  - 25-Jun-2012
  - Emmbl Release 112 (31 May 2012):
    - 247,335,689 sequence entries / 429,512,389,024 nucleotides
    - Total: 201 GB compressed and 1.4 TB uncompressed data

  **Sequences (2.3 trillions) — Bases (257.4 trillions)**
Accelerating bioinformatics tools for high-throughput sequence analysis

The process of genome annotation

**Sequence**

Region Annotation

Identify functional Regions

Regions

- Promotor
- tRNA
- Terminator
- RBS
- CDS
- CDS

Function Annotation

Assign Function

Functions

- tRNA–ile
- Branched-chain amino acid aminotransferase
- Aspartokinase I
- ileV
- ilvE
- thrA

Promotor

- tRNA
- Terminator
- RBS
- CDS
- CDS
Measuring similarity by alignments

- Similarity is measured through **global or local alignments**
- Example of a local alignment (7 positions out of 9):

```
  B A N A N - E
 R H A B A R B E R
 1 2 1 2 1-1 2 -> Score=8
```

- Biological/bio-chemical information is put into **scoring** scheme
- **Gap costs** are beyond bio-chemical knowledge

Software History:

- **SSEARCH** (Pearson) Smith-Waterman algorithm for local optimal alignment
- **MPSEARCH** (EBI) faster, improved
- **FASTA** (Pearson, Lipman) heuristic for global alignment
- **BLAST(2)** (Altschul, Lipman...) heuristic for local (&gapped) alignment

Compute time for pairwise comparison is **proportional query and database size.**
Compared to the others, **BLAST2 does it the fastest, however least sensitive way.**
Recent development of bioinformatics tools for the analysis and visualization of RNAseq data

SARUMAN & VAMP
RNA sequencing (RNAseq)

**Goal:**
Utilize high-throughput sequencing of cDNA libraries to analyze and characterize entire transcriptomes including operon structures, transcript starts, new transcripts

**Advantages:**
- Single nucleotide resolution
- Very high dynamic range
- No cross-hybridization
- Catalogue all species of transcripts, including small RNAs

**Requirements:**
- Fast and reliable short read mapping
- Automated & standardized analysis
- Visualization

⇒ Joint effort with the technology platform “Genomics” (J. Kalinowski, C. Rückert)
⇒ Development of three RNAseq protocols by K. Pfeiffer
Short read matching

• Several algorithms / tools published & available
• Heuristic methods are fast
• Do not guarantee exact results
  • BWA
  • SOAP2
  • Bowtie
  • PASS
• Exact methods are much slower
  • SHRiMP
Mapping of massive amounts of short read data from 454, Solexa, Solid sequencing by using modern graphic cards (GPUs) to speed up read matching against reference genomes:

- SARUMAN – Semiglobal Alignment of short Reads using CUDA and Needleman-Wunsch
- Exact algorithm, no heuristic
- Find deletions, insertions and substitutions
- Report all matches

Download:
http://www.cebitec.uni-bielefeld.de/brf/saruman/saruman.html

Blom et al., Bioinformatics, 2011
Accelerating bioinformatics tools for high-throughput sequence analysis

SARUMAN Filter Algorithm

**Given:**
- a read $f$, $|f| = m$
- a genome sequence $g$, $|g| = n$
- an error threshold $e \geq 0$

**Wanted:**
All starting positions $i$ in $g$, such that there exists an alignment of $g[i...]$ and $f$ with at most $e$ errors (mismatches and/or indels). Show one such alignment.

**Theory**
A read $f$ that matches the reference genome MUST have at least two matching segments from the sets $s_i$ and/or $k_i$.

**Error location**
We know that $k_i$ overlaps $s_i$ on $q - R$ positions. These positions are free from errors ($s_i$ matched). So the error causing $k_i$ not to match must have been on the last $R$ positions of $k_i$.

One segment of set $(s_j...s_0)$ must match. ($e+1$ segments for $e$ errors)

If only one segment matches, there is exactly one error (mismatch or indel) in every remaining segment. Otherwise a second segment of this set would match.

**Iterative approach**
If $k_i$ does not match, we can conclude that the last $q - R$ positions of $s_{i+1}$ are free from errors, which are the first $q - R$ positions of $k_{i+1}$. So if $k_{i+1}$ does not match, the next error is in the first $R$ positions of $s_{i+2}$.

**Iterative approach**
Check all remaining segments of set $k$ until a match is found or last segment $K_e$ is reached. If $K_e$ does not match there has to be an error in the last $R$ bases of the read ($(e + 1)$-th error).

**Shifting segments**
Check every segment not only on one match-position, but on all possible match-positions by shifting the segment to the left or right by up to $e$ positions.
### SARUMAN Filter Algorithm

**Given:**
- a read \( f \), \( |f| = m \)
- a genome sequence \( g \), \( |g| = n \)
- an error threshold \( e \)

**Wanted:**
- All starting positions exist, an alignment exists, and \( e \) errors (mismatches or indels) are allowed.

**Theory**
- A read \( f \) that is longer than \( e \) must have at least one segment that matches the genome with at least \( e \) mismatches.
- We know that these positions must be matched, so the error must have been there.

**Error location**
- If only one segment matches, there is exactly one error (mismatch or indel) in every remaining segment.
- Otherwise a second segment of this set would match.

**Iterative approach**
- If \( k_i \) does not match, we can conclude that the last \( q - R \) positions of \( s_{i+1} \) are free from all of the \( q \) types of mismatches, with error rate \( 1) \)-th power.

---

![Clock Image](image)
SARUMAN Filter Algorithm

• All possible match positions are found => Multiple hits per read
• Algorithm works with Insertions/Deletions
• User-defined error ratio
• Potential matches have to be verified by alignment
• Alignments are small (short reads) & can be computed in parallel

➢ Large amounts of small independent jobs can be processed efficiently on graphics cards using CUDA (or OpenCL)
Accelerating bioinformatics tools for high-throughput sequence analysis

SARUMAN – Technical Overview

- Large number of reads is prepared for aligning (e.g. 200,000 36bp reads)
- Sequences and parameters are copied to the GPU
- GPU aligns many sequences in parallel
- During alignment the host already collects new hits
- Maximal number of parallel alignments depends on GPU memory
  ⇒ 3 x IBM iDataPlex servers with 78 GB RAM and 2 x Tesla M2070 GPUs
Evaluation using an artificial data set

Mapping of ~18 mio. artificial Solexa reads (75bp)

- *E. coli* K12 str. M1665 as source organism
- Max 2 errors included (mismatches/insertions/deletions)

### Table 1. Sensitivity evaluation with an artificial dataset of 17,980.142 reads (75bp) generated from *Escherichia coli K12 MG1655* with up to two errors. MCP (Mapped to Correct Position) denotes the number of mapped reads that had a match to their original position. BMCP (Best Match at Correct Position) denotes the number of reads where the best match was located at the correct position.

<table>
<thead>
<tr>
<th></th>
<th>SARUMAN</th>
<th>SOAP2</th>
<th>Bowtie</th>
<th>BWA</th>
<th>SHRIMP</th>
<th>PASS</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mapped</td>
<td>17,980,142</td>
<td>15,142,908</td>
<td>15,123,838</td>
<td>17,746,484</td>
<td>17,980,142</td>
<td>16,873,044</td>
<td>17,980,142</td>
</tr>
<tr>
<td>Not mapped</td>
<td>0</td>
<td>2,837,234</td>
<td>2,856,304</td>
<td>233,658</td>
<td>0</td>
<td>1,107,098</td>
<td>0</td>
</tr>
<tr>
<td>Perfect</td>
<td>4,999,944</td>
<td>4,999,942</td>
<td>4,999,944</td>
<td>4,999,944</td>
<td>4,999,944</td>
<td>4,999,944</td>
<td>4,999,944</td>
</tr>
<tr>
<td>With errors</td>
<td>12,980,198</td>
<td>10,142,966</td>
<td>10,123,894</td>
<td>12,746,540</td>
<td>12,980,198</td>
<td>11,873,109</td>
<td>12,980,198</td>
</tr>
<tr>
<td>1 mismatch</td>
<td>4,999,908</td>
<td>4,999,906</td>
<td>4,999,908</td>
<td>4,999,908</td>
<td>4,999,908</td>
<td>4,999,908</td>
<td>4,999,908</td>
</tr>
<tr>
<td>2 mismatches</td>
<td>4,999,938</td>
<td>4,999,936</td>
<td>4,999,936</td>
<td>4,999,936</td>
<td>4,999,936</td>
<td>4,999,936</td>
<td>4,999,936</td>
</tr>
<tr>
<td>1 insertion</td>
<td>496,092</td>
<td>34,648</td>
<td>29,000</td>
<td>489,788</td>
<td>499,092</td>
<td>478,754</td>
<td>499,092</td>
</tr>
<tr>
<td>2 insertions</td>
<td>499,998</td>
<td>1,243</td>
<td>496</td>
<td>496</td>
<td>499,998</td>
<td>499,998</td>
<td>499,998</td>
</tr>
<tr>
<td>1 deletion</td>
<td>493,569</td>
<td>46,740</td>
<td>46,740</td>
<td>491,454</td>
<td>493,560</td>
<td>475,916</td>
<td>493,560</td>
</tr>
<tr>
<td>2 deletion</td>
<td>493,354</td>
<td>4,828</td>
<td>4,828</td>
<td>433,003</td>
<td>493,354</td>
<td>452,714</td>
<td>493,354</td>
</tr>
<tr>
<td>1 ins. &amp; 1 mism.</td>
<td>500,000</td>
<td>21,374</td>
<td>12,308</td>
<td>458,957</td>
<td>500,000</td>
<td>18,334</td>
<td>500,000</td>
</tr>
<tr>
<td>1 Del. &amp; 1 Mism.</td>
<td>492,450</td>
<td>34,291</td>
<td>29,798</td>
<td>467,329</td>
<td>493,450</td>
<td>447,042</td>
<td>493,450</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>MCP</th>
<th>BMCP</th>
<th>Total Alignments</th>
<th>Run time</th>
<th>RAM usage</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>17,899,092</td>
<td>15,070,286</td>
<td>15,061,178</td>
<td>17,432,842</td>
<td>17,246,645</td>
</tr>
<tr>
<td></td>
<td>17,899,086</td>
<td>15,070,286</td>
<td>15,061,178</td>
<td>17,430,632</td>
<td>17,245,811</td>
</tr>
<tr>
<td></td>
<td>19,971,674</td>
<td>16,425,886</td>
<td>16,542,168</td>
<td>18,022,317</td>
<td>19,423,932</td>
</tr>
<tr>
<td>Runtime</td>
<td>12:05 min</td>
<td>06:40 min</td>
<td>18:56 min</td>
<td>15:09 min</td>
<td>95:06 min</td>
</tr>
<tr>
<td>RAM usage</td>
<td>3,375,236 kb</td>
<td>702,964 kb</td>
<td>14,420 kb</td>
<td>117,172 kb</td>
<td>1,315,644 kb</td>
</tr>
</tbody>
</table>
Accelerating bioinformatics tools for high-throughput sequence analysis

Evaluation

<table>
<thead>
<tr>
<th></th>
<th>SARUMAN</th>
<th>SOAP2</th>
<th>BOWTIE</th>
<th>BWA</th>
<th>SHRiMP</th>
<th>PASS</th>
<th>Referenz</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mapped</td>
<td>17,984,730</td>
<td>15,105,693</td>
<td>15,146,620</td>
<td>17,805,133</td>
<td>17,984,730</td>
<td>16,925,234</td>
<td>17,984,730</td>
</tr>
<tr>
<td>Not mapped</td>
<td>0</td>
<td>2,879,037</td>
<td>2,838,110</td>
<td>179,597</td>
<td>0</td>
<td>1,059,496</td>
<td>0</td>
</tr>
<tr>
<td>BMCP</td>
<td>17,899,086</td>
<td>15,070,286</td>
<td>15,061,178</td>
<td>17,430,632</td>
<td>17,245,811</td>
<td>16,577,173</td>
<td></td>
</tr>
<tr>
<td>Total Alignments</td>
<td>19,971,674</td>
<td>16,425,886</td>
<td>16,542,168</td>
<td>18,022,317</td>
<td>19,423,932</td>
<td>18,447,418</td>
<td></td>
</tr>
</tbody>
</table>

BMCF (Best Match at Correct Position) denotes the number of reads where the best match was located at the correct position.

- Nearly 100% of reads were mapped to correct position
- Reads with errors map to other positions just by chance
- Highest number of total alignments
- All valid alignments computed, not only best alignments
### Accelerating bioinformatics tools for high-throughput sequence analysis

#### SARUMAN – Native Output

- Converter to SAM output available
- Native SAM exporter in progress

<table>
<thead>
<tr>
<th>Sequence Name</th>
<th>Start</th>
<th>End</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>KN-1360_6_18_1125_518</td>
<td>84862</td>
<td>84897</td>
<td>36</td>
</tr>
<tr>
<td>KN-1360_6_18_1649_499</td>
<td>86536</td>
<td>86564</td>
<td>36</td>
</tr>
<tr>
<td>KN-1360_6_18_738_1097</td>
<td>288387</td>
<td>288413</td>
<td>36</td>
</tr>
<tr>
<td>KN-1360_6_18_738_1097</td>
<td>288638</td>
<td>288643</td>
<td>36</td>
</tr>
<tr>
<td>KN-1360_6_18_1837_1844</td>
<td>84962</td>
<td>84997</td>
<td>36</td>
</tr>
<tr>
<td>KN-1360_6_18_35_90</td>
<td>84971</td>
<td>85006</td>
<td>36</td>
</tr>
<tr>
<td>KN-1360_6_18_57_1295</td>
<td>280770</td>
<td>280775</td>
<td>36</td>
</tr>
<tr>
<td>KN-1360_6_18_57_1295</td>
<td>321059</td>
<td>321059</td>
<td>36</td>
</tr>
<tr>
<td>KN-1360_6_18_1719_1971</td>
<td>84708</td>
<td>84732</td>
<td>36</td>
</tr>
<tr>
<td>KN-1360_6_18_1147_460</td>
<td>281080</td>
<td>281085</td>
<td>36</td>
</tr>
<tr>
<td>KN-1360_6_18_1147_460</td>
<td>321374</td>
<td>321374</td>
<td>36</td>
</tr>
<tr>
<td>KN-1360_6_18_567_66</td>
<td>281139</td>
<td>281139</td>
<td>36</td>
</tr>
<tr>
<td>KN-1360_6_18_567_66</td>
<td>321413</td>
<td>321420</td>
<td>36</td>
</tr>
<tr>
<td>KN-1360_6_18_707_1670</td>
<td>280125</td>
<td>280129</td>
<td>36</td>
</tr>
<tr>
<td>KN-1360_6_18_1444_211</td>
<td>280813</td>
<td>280818</td>
<td>36</td>
</tr>
<tr>
<td>KN-1360_6_18_1444_211</td>
<td>321099</td>
<td>321103</td>
<td>36</td>
</tr>
<tr>
<td>KN-1360_6_18_46_786</td>
<td>280809</td>
<td>280812</td>
<td>36</td>
</tr>
<tr>
<td>KN-1360_6_18_702_982</td>
<td>82617</td>
<td>82652</td>
<td>36</td>
</tr>
<tr>
<td>KN-1360_6_18_380_357</td>
<td>81994</td>
<td>82029</td>
<td>36</td>
</tr>
<tr>
<td>KN-1360_6_18_1724_985</td>
<td>321110</td>
<td>321117</td>
<td>36</td>
</tr>
<tr>
<td>KN-1360_6_18_1724_985</td>
<td>85254</td>
<td>85289</td>
<td>36</td>
</tr>
<tr>
<td>KN-1360_6_18_1397_1173</td>
<td>85767</td>
<td>85802</td>
<td>36</td>
</tr>
<tr>
<td>KN-1360_6_18_1397_1173</td>
<td>84794</td>
<td>84829</td>
<td>36</td>
</tr>
<tr>
<td>KN-1360_6_18_1766_86</td>
<td>172850</td>
<td>172853</td>
<td>36</td>
</tr>
<tr>
<td>KN-1360_6_18_1279_508</td>
<td>84896</td>
<td>84931</td>
<td>36</td>
</tr>
<tr>
<td>KN-1360_6_18_1052_2043</td>
<td>281076</td>
<td>281080</td>
<td>36</td>
</tr>
<tr>
<td>KN-1360_6_18_1052_2043</td>
<td>321368</td>
<td>321368</td>
<td>36</td>
</tr>
<tr>
<td>KN-1360_6_18_443_1153</td>
<td>262450</td>
<td>262450</td>
<td>36</td>
</tr>
<tr>
<td>KN-1360_6_18_1580_480</td>
<td>82199</td>
<td>82234</td>
<td>36</td>
</tr>
<tr>
<td>KN-1360_6_18_639_1262</td>
<td>86745</td>
<td>86780</td>
<td>36</td>
</tr>
</tbody>
</table>
VAMP – A short read browser

- Green: Perfect match – error-free alignment
- Yellow: Best match – alignment with errors but no better mapping elsewhere
- Red: Common match – everything else
VAMP – Zoom in

Legend
- Perfect match cov.
- Best match cov.
- Complete cov.

Position: 326253
- Forward strand
  - Perfect match cov.: 77
  - Best match cov.: 98
  - Complete cov.: 98
- Reverse strand
  - Perfect match cov.: 127
  - Best match cov.: 138
  - Complete cov.: 138
Accelerating bioinformatics tools for high-throughput sequence analysis

RNAseq results – identification of operon structures

- Operon encoding four genes & alternative transcription start
- Annotated reference genome
- Mapped reads of a library enriched for primary 5' transcript ends
- Mapped reads of a whole transcriptome library
Accelerating bioinformatics tools for high-throughput sequence analysis

RNAseq results - promoters

![RNAseq result diagram with transcription start, -35 box, and -10 box annotations]
Accelerating bioinformatics tools for high-throughput sequence analysis

RNAseq results – Improved genome annotation

Identification of missing gene

Corrected start

Confirmed start

Alexander Goesmann – Computational Genomics/BRF/CeBiTec
Accelerating bioinformatics tools for high-throughput sequence analysis

RNAseq results – Identification of new functional regions

novel miRNA

Cis-antisense transcripts

Novel sRNA

Alexander Goesmann – Computational Genomics/BRF/CeBiTec
Accelerating bioinformatics tools for high-throughput sequence analysis

VAMP – SNP detection
Accelerating bioinformatics tools for high-throughput sequence analysis

VAMP: Histogram, alignment viewer and SNPs
The Bielefeld Technology Platform Bioinformatics

- Virtual bioinformatics desktop with disk-less thin clients (everywhere!)
- Access to more than 700 bioinformatics tools & sequence databases
- Application servers (up to 2 TB RAM)
- 4024 CPU-Core Compute-Cluster
- Special purpose GPU & FPGA hardware
- 700 TB disk storage & 1.4 PB tape
Accelerating bioinformatics tools for high-throughput sequence analysis

Special Purpose Hardware

- 12 x TimeLogic SeqCruncher H1
- 2 x TimeLogic SeqCruncher J1
- Nvidia GeForce GTX 580
- Convey HC-1ex
NGS Applications for Transcriptomics

• GPUs
  • Pairwise alignment, short read mapping, MUMmer, HMMer
  • Moderate speedup
  • Limited memory
  • Open framework & relatively easy to learn

• Timelogic DeCypher
  • Smith-Waterman alignment, BLAST, VelociMapper, HMMer
  • Huge speedup
  • Proprietary technology

• Convey hybrid-core
  • Genome assembly (Velvet), Short read mapping (bwa), k-mer counting
  • Significant speedup
  • Efficient memory access
  • Open framework but hard to learn
Accelerating bioinformatics tools for high-throughput sequence analysis

People

- Stefan Albaum
- Regina Bisdorf
- Jochen Blom
- Rolf Hilker
- Tobias Jakobi
- Sebastian Jaenicke
- Lukas Jelonek
- Sebastian Jünemann
- Niko Kessler
- Florian Kollin
- Daniel Langenkämper
- Burkhard Linke
- Roza Parol-Kryger
- David Riess
- Oliver Rupp
- Jessica Schneider
- Oliver Schwengers
- Dominik Vahrenhorst
- Jörn Winnebald
- Student Programmers

System Administrators:
Björn Fischer, Torsten Kasch, Achim Neumann, Ralf Nolte, Rainer Orth, Volker Tölle

Group Leader: Dr. Alexander Goesmann
Thanks for your attention!

- Contact:agoesman@CeBiTec.Uni-Bielefeld.DE
- Homepage:http://www.cebitec.uni-bielefeld.de/brf
- Software:http://www.cebitec.uni-bielefeld.de/comics