Erasmus MC
University Medical Center Rotterdam
Center for Bioinformatics

"Programming without Programming"
aka:
Use of Unix Tools for High Throughput Genome Sequencing

Michael J. Moorhouse,
Erasmus MC (Department of Bioinformatics*)

2009-05-12

** Until 31st May; then University of Sheffield / MRC*

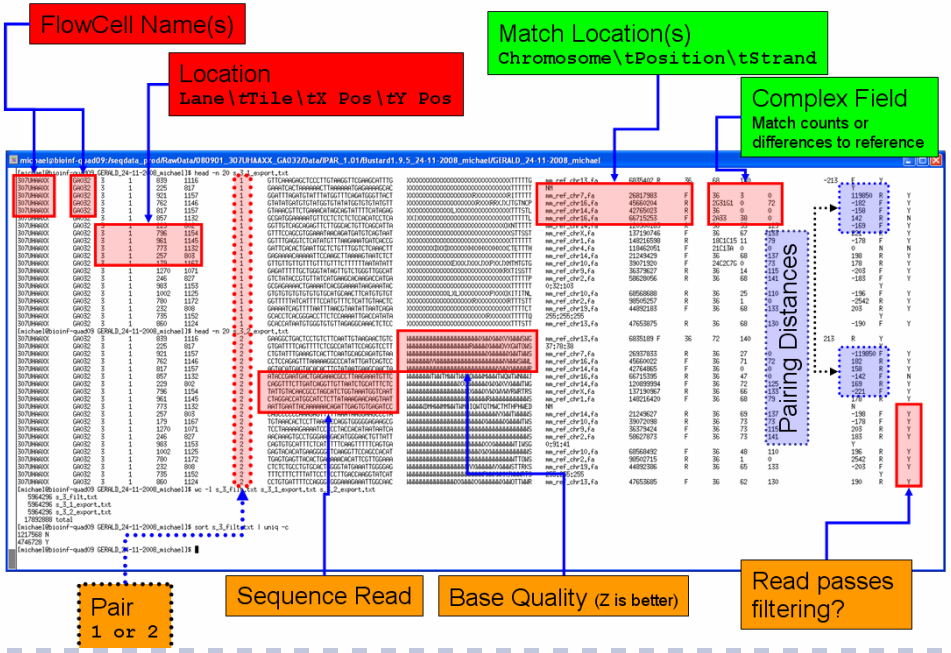
Example data kindly supplied by the Erasmus Centre for Biomics with thanks to Dr. Wilfred van Ijcken; additions / corrections by Jan van der Haarst

The Problem

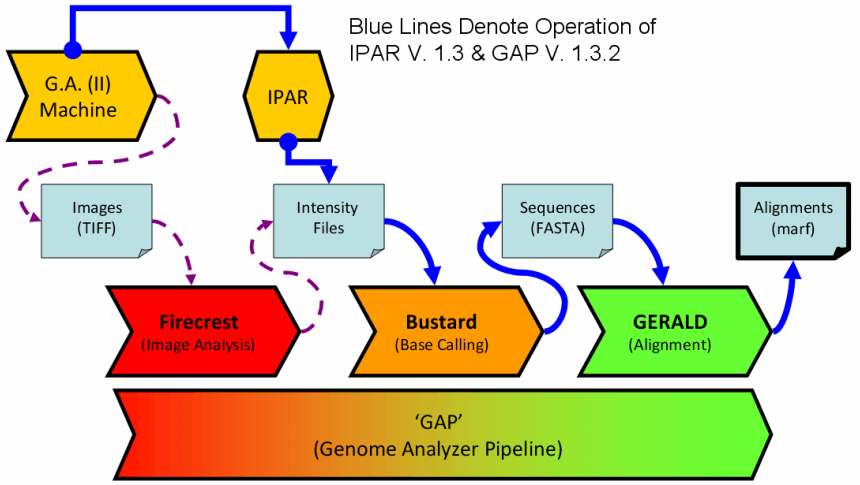
Erasmus MC
Center for Bioinformatics

- **Illumina Genome Analyzer 'GAP' software produces output in this tabular text format:**
  - 22 columns; ~14 Million Lines (~1.5Gb)
    - One line per read

Typical 'export.txt' file



Overview of GAP



Michael Moorhouse: May 2009

Tips & Tricks using Unix Command Line Tools

Erasmus MC
Center for Bioinformatics

- **Many unix tools supplied with modern Unix OSs; highly optimized and debugged. Demonstrated here:**
  - **grep** 'Select lines with a particular pattern'
  - **sort** 'Sorts line based on criterion'
  - **cut** 'Print column(s)'
  - **uniq** 'Report / count number of occurrences'
  - **gawk** A simple programming language
  - **wc** 'Count words / lines'
  - **head** 'Print the first (10) lines of the file
  - **time** 'Print how long command line took to execute'
- **Can be stitched together using a pipe '|' to manipulate STDIN / STDOUT:**

```
cut –f22 s_4_export.txt | grep Y | wc –l > myresult.count
```

Also, formally defined

Erasmus MC
Center for Bioinformatics

- **Copied from the Illumina GAP 1.3.2 Manual**
  - Original material copyrightedIllumina, Inc. San Diego; this is a derived work

Output File Formats

The sequences and base-specific quality scores are bundled by lane and come in several configurable text formats. The currently supported formats are fasta, fastq, and SCARF. For a description of each format, see *ANALYSIS Variables* on page 33.

Table 24 Final Output File Formats

Output File	Format
s_N_export.txt s_N_R_export.txt s_N_sorted.txt s_N_R_sorted.txt	Not all fields are relevant to a single-read analysis. 1. Machine (Parsed from Run Folder name) 2. Run Number (Parsed from Run Folder name) 3. Lane 4. Tile 5. X Coordinate of cluster 6. Y Coordinate of cluster 7. Index string (Bland for a non-indexed run) 8. Read number (1 or 2 for paired-read analysis, blank for a single-read analysis) 9. Read 10. Quality string—In symbolic ASCII format (ASCII character code = quality value + 64) 11. Match chromosome—Name of chromosome match OR code indicating why no match resulted 12. Match Contig—Gives the contig name if there is a match and the match chromosome is split into contigs (Blank if no match found) 13. Match Position—Always with respect to forward strand, numbering starts at 1 (Blank if no match found) 14. Match Strand—"F" for forward, "R" for reverse (Blank if no match found) 15. Match Descriptor—Concise description of alignment (Blank if no match found) • A numeral denotes a run of matching bases • A letter denotes substitution of a nucleotide: For a 35 base read, "35" denotes an exact match and "32C2" denotes substitution of a "C" at the 33rd position 16. Single-Read Alignment Score—Alignment score of a single-read match, or for a paired read, alignment score of a read if it were treated as a single read. Blank if no match found; any scores less than 4 should be considered as aligned to a repeat 17. Paired-Read Alignment Score—Alignment score of a paired read and its partner, taken as a pair. Blank if no match found; any scores less than 4 should be considered as aligned to a repeat 18. Partner Chromosome—Name of the chromosome if the read is paired and its partner aligns to another chromosome (Blank for single-read analysis) 19. Partner Contig—Not blank if read is paired and its partner aligns to another chromosome and that partner is split into contigs (Blank for single-read analysis) 20. Partner Offset—If a partner of a paired read aligns to the same chromosome and contig, this number, added to the Match Position, gives the alignment position of the partner (Blank for single-read analysis) 21. Partner Strand—To which strand did the partner of the paired read align? "F" for forward, "R" for reverse (Blank if no match found, blank for single-read analysis) 22. Filtering—Did the read pass quality filtering? "Y" for yes, "N" for no
s_N_sequence.txt s_N_R_sequence.txt	Filtered output User-specified: fasta, fastq, scarf (one sequence per line, not identifier)

Interesting One Liners Summary

Erasmus MC
Center for Bioinformatics

For Inspiration / discussion / demonstration here:

- How many reads are there?
- What are the sequences?
- How many reads passed filtering?
- Print out the reads that passed filtering in FASTA format
- Produce a count of all reads

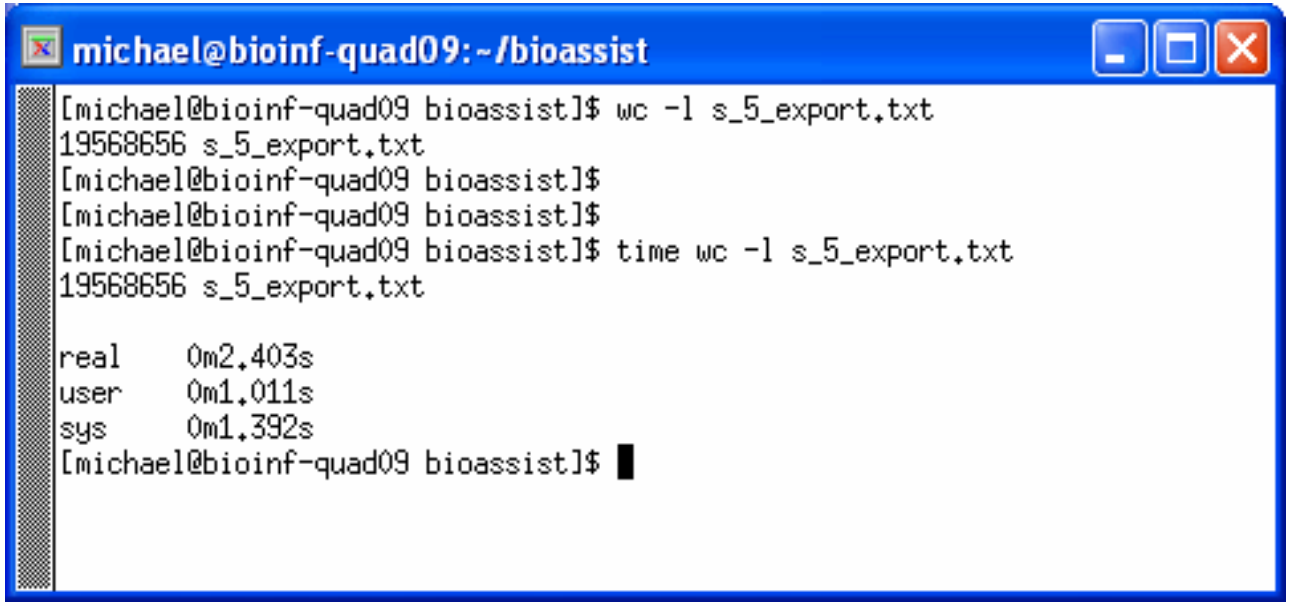
All using a ~1.5Gb file of HTGS Data

How many reads are there?

Erasmus MC
Center for Bioinformatics

- `wc -l s_5_export.txt`

Result:



Notes:

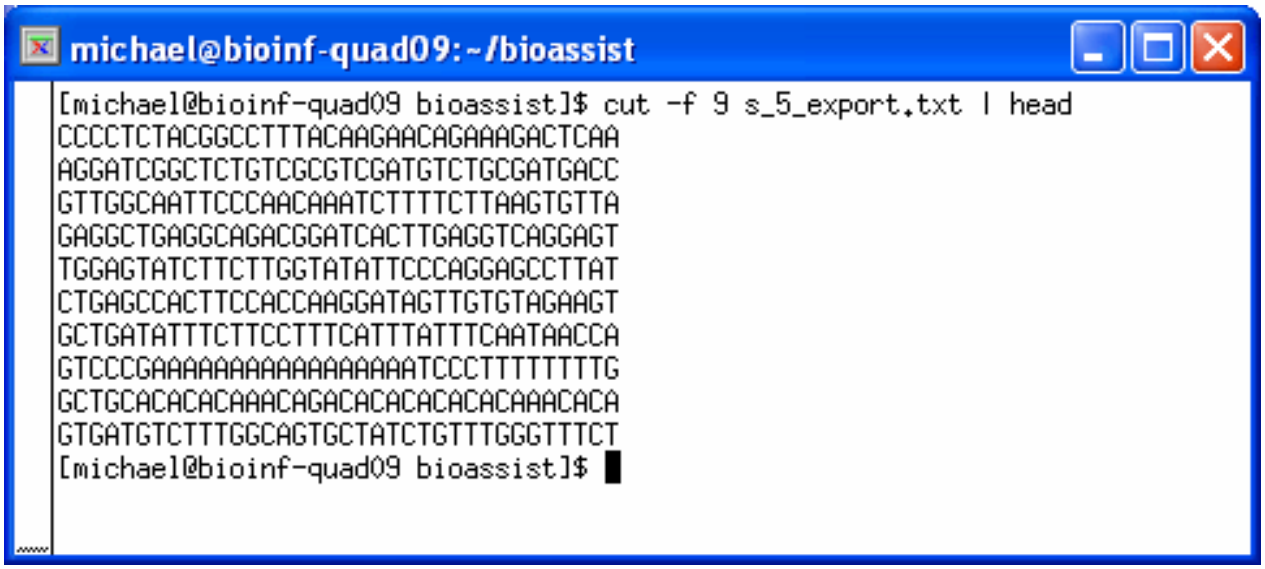
-l means 'line count'

What are the sequences?

Erasmus MC
Center for Bioinformatics

- `cut -f 9 s_5_export.txt | head`

Result:



Notes:

-f 9 means 'just the 9th field'

How many reads passed filtering?

Erasmus MC
Center for Bioinformatics

- `cut -f22 s_5_export.txt | grep Y | wc -l`

Result:

```
michael@bioinf-quad09:~/bioassist
[michael@bioinf-quad09 bioassist]$ cut -f22 s_5_export.txt | grep Y | wc -l
9883712
[michael@bioinf-quad09 bioassist]$ time cut -f22 s_5_export.txt | grep Y | wc -l
9883712

real    1m42.651s
user    1m42.502s
sys     0m1.698s
[michael@bioinf-quad09 bioassist]$
```

cf:

```
michael@bioinf-quad09:~/bioassist
[michael@bioinf-quad09 bioassist]$ wc -l s_5_export.txt
19568656 s_5_export.txt
[michael@bioinf-quad09 bioassist]$
[michael@bioinf-quad09 bioassist]$
[michael@bioinf-quad09 bioassist]$ time wc -l s_5_export.txt
19568656 s_5_export.txt

real    0m2.403s
user    0m1.011s
sys     0m1.392s
[michael@bioinf-quad09 bioassist]$
```

Notes:

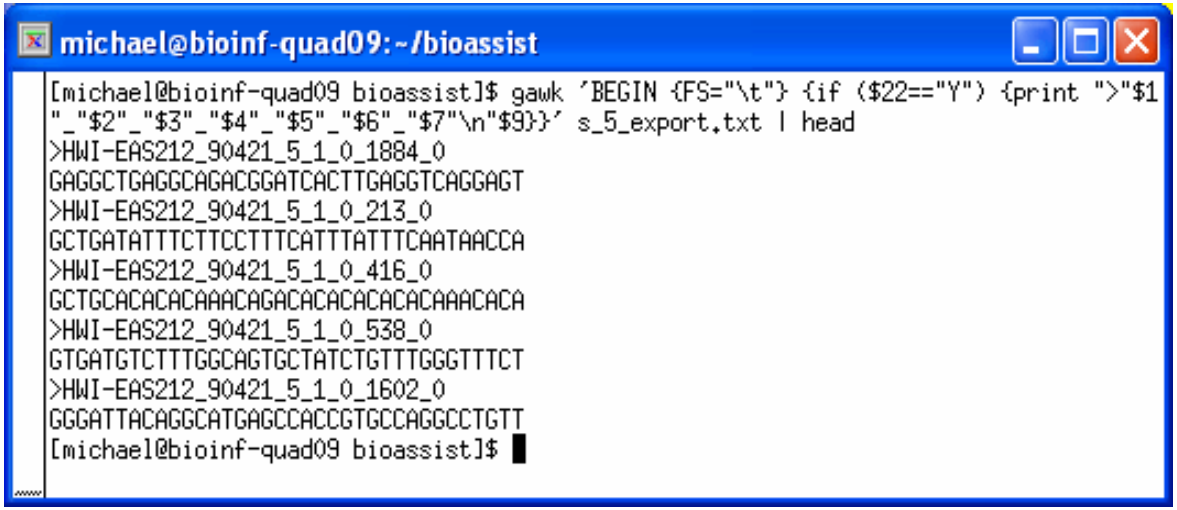
"-f 22" means 'just the 22nd field'; "grep Y" means only those lines containing 'Y'

Print out the reads that passed filtering in FASTA format

Erasmus MC
Center for Bioinformatics

- `gawk 'BEGIN {FS="\t"} {if ($22=="Y") {print ">"$1"_"$2"_"$3"_"$4"_"$5"_"$6"_"$7"\n"$9}}' s_5_export.txt | head`

Result:



Notes:

"FS="\t"" means 'set the Field Separator to 'tab''; "$1" = print 1st field etc.

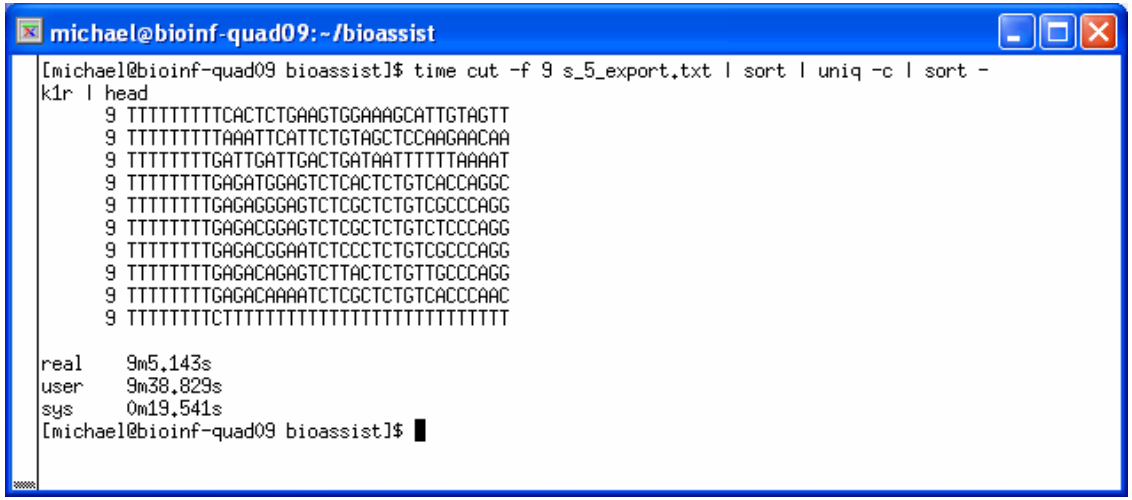
FASTA format is as above; first 7 fields used to make each sequence ID unique

Produce a count of all the reads

Erasmus MC
Center for Bioinformatics

- `cut -f9 s_4_export.txt | sort | uniq -c | sort -nk1`

Result:



Notes:

The construct '| sort | uniq –c' does the counting step; once counted the reads are sorted – most frequent to the top using '-nk1r' where '-n' = numeric order; '-r' = reverse order; '-k1' on the 1st column

This isn't a fast operation – 9 ½ minutes - possibly due to two sorting steps

Erasmus MC
University Medical Center Rotterdam
Center for Bioinformatics

"Spinning the Internal Web"
aka:
"How to use symbolic links to create subsets of files"

Michael J. Moorhouse,
Erasmus MC (Department of Bioinformatics*)

2009-05-12

Symbolic Links:

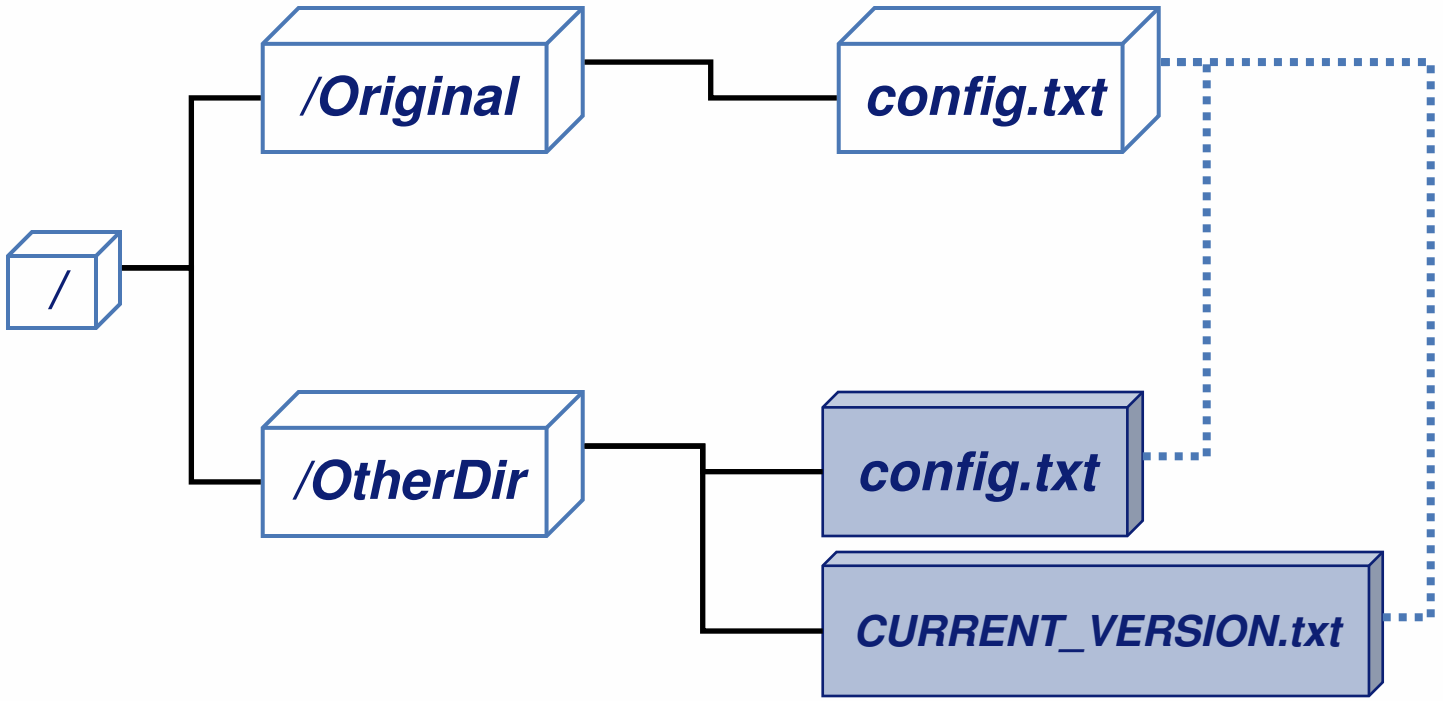
Erasmus MC
Center for Bioinformatics

- **Links in general:**
  - Allow the 'linking' (association, binding, mapping) of one location on a disk system to another
  - Created in Unix style OSs with the '`ln`' command/tool.
  - General concept:
    "Points" from '*There*' to '*Here*'
    - (omit the second name and you get a link with the same name)
  - Small: a simple 'pointer' is stored (~20 bytes*)
  - Fast to create and delete
    - **No data is copied until the link is accessed**
- **Soft or Hard?**
  - Soft: ln –s
    - The type I use most often
    - Deleting the link deletes the link; not the original file
    - Can operate across filesystems
    - Can be left 'dangling' (original file deleted or space not mapped)
    - Date is date when link is created
  - Hard: ln
    - I never use these…
    - Date is the date of the original file
    - Looks very much like the original file

* On my system might not be representative

Links - Diagrammatically

Erasmus MC
Center for Bioinformatics

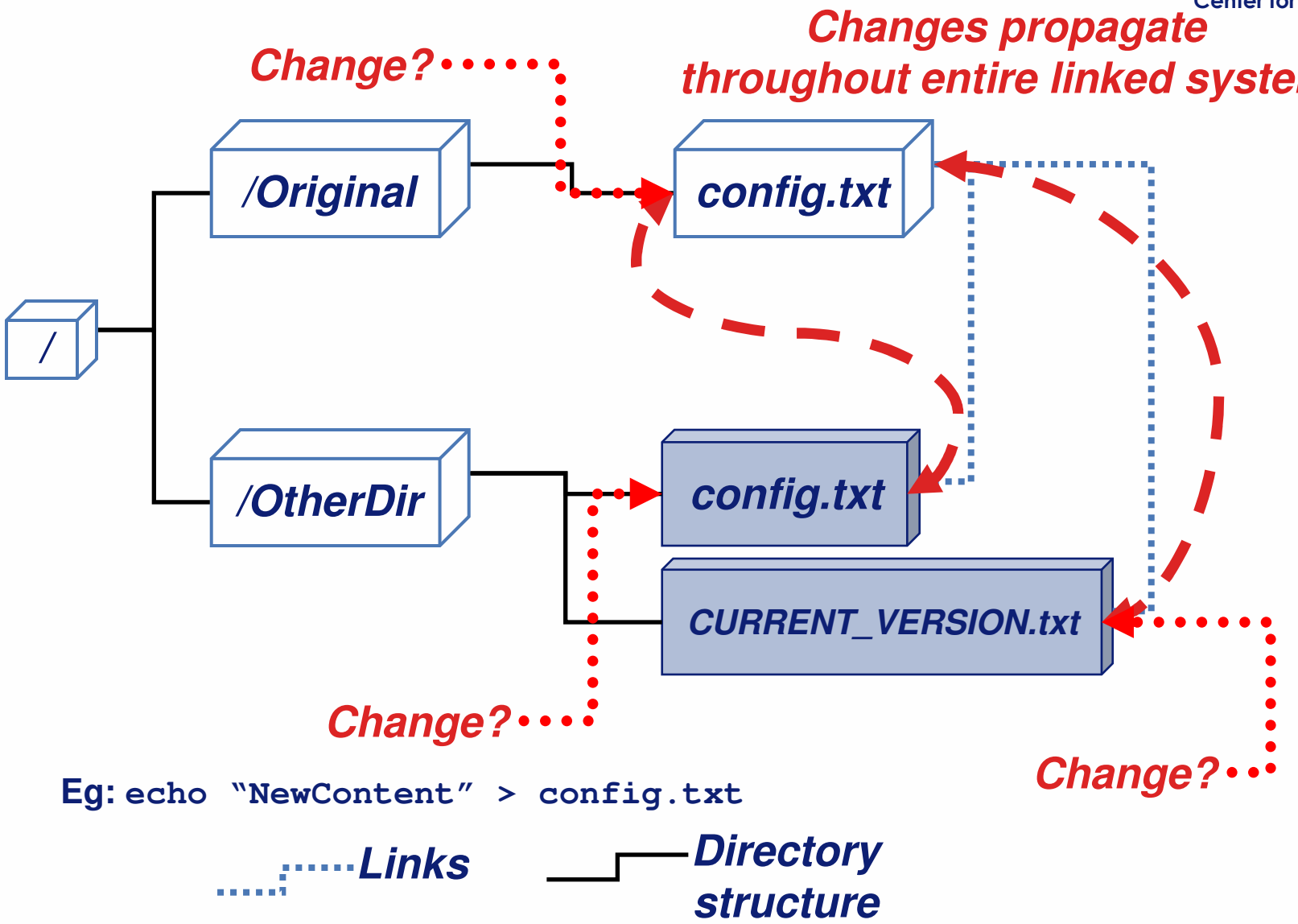


Links

Directory structure

Links – Prorogation of Changes

Erasmus MC
Center for Bioinformatics



Eg: `echo "NewContent" > config.txt`

Links

Directory structure

‘Dangling / Broken Links’ – Often a Problem

Erasmus MC
Center for Bioinformatics

- **Possibly to create *really* messy structures if overused**
  - Like all ‘links’ then…gets ‘Shambolic’?

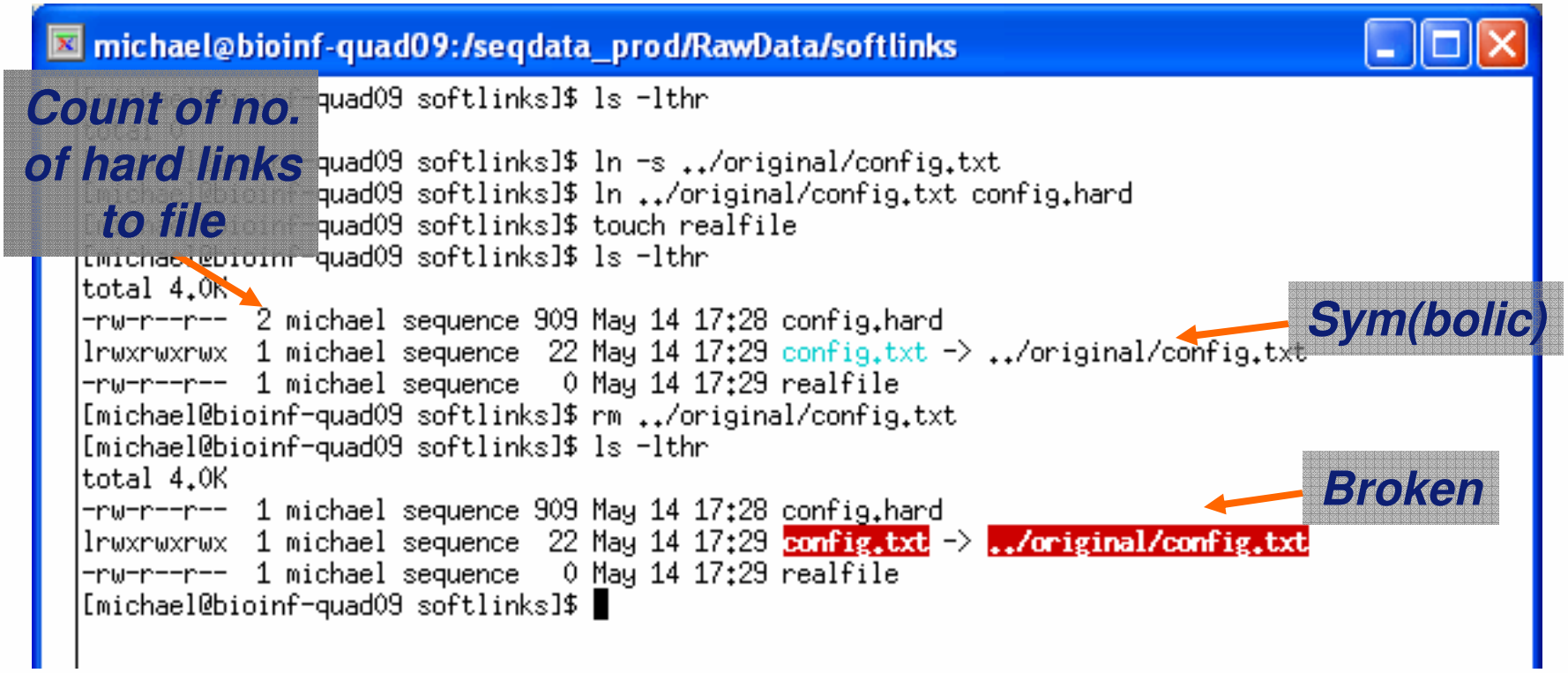
```
[michael@bioinf-quad09 softlinks]$ ls -lthr
total 0
lrwxrwxrwx  1 michael sequence 27 May 14 16:50 CURRENT_CONFIG.txt -> ../original/config.txt.soft
lrwxrwxrwx  1 michael sequence 27 May 14 17:01 config.txt.soft -> ../original/config.txt.soft
[michael@bioinf-quad09 softlinks]$ ln -s config.txt.soft internal_link
[michael@bioinf-quad09 softlinks]$ ls -lthr
total 0
lrwxrwxrwx  1 michael sequence 27 May 14 16:50 CURRENT_CONFIG.txt -> ../original/config.txt.soft
lrwxrwxrwx  1 michael sequence 27 May 14 17:01 config.txt.soft -> ../original/config.txt.soft
lrwxrwxrwx  1 michael sequence 15 May 14 17:02 internal_link -> config.txt.soft
[michael@bioinf-quad09 softlinks]$ rm config.txt.soft
[michael@bioinf-quad09 softlinks]$ ls -lthr
total 0
lrwxrwxrwx  1 michael sequence 27 May 14 16:50 CURRENT_CONFIG.txt -> ../original/config.txt.soft
lrwxrwxrwx  1 michael sequence 15 May 14 17:02 internal_link -> config.txt.soft
[michael@bioinf-quad09 softlinks]$
```

Broken Link

Coloring in Shell – for reference

Erasmus MC
Center for Bioinformatics

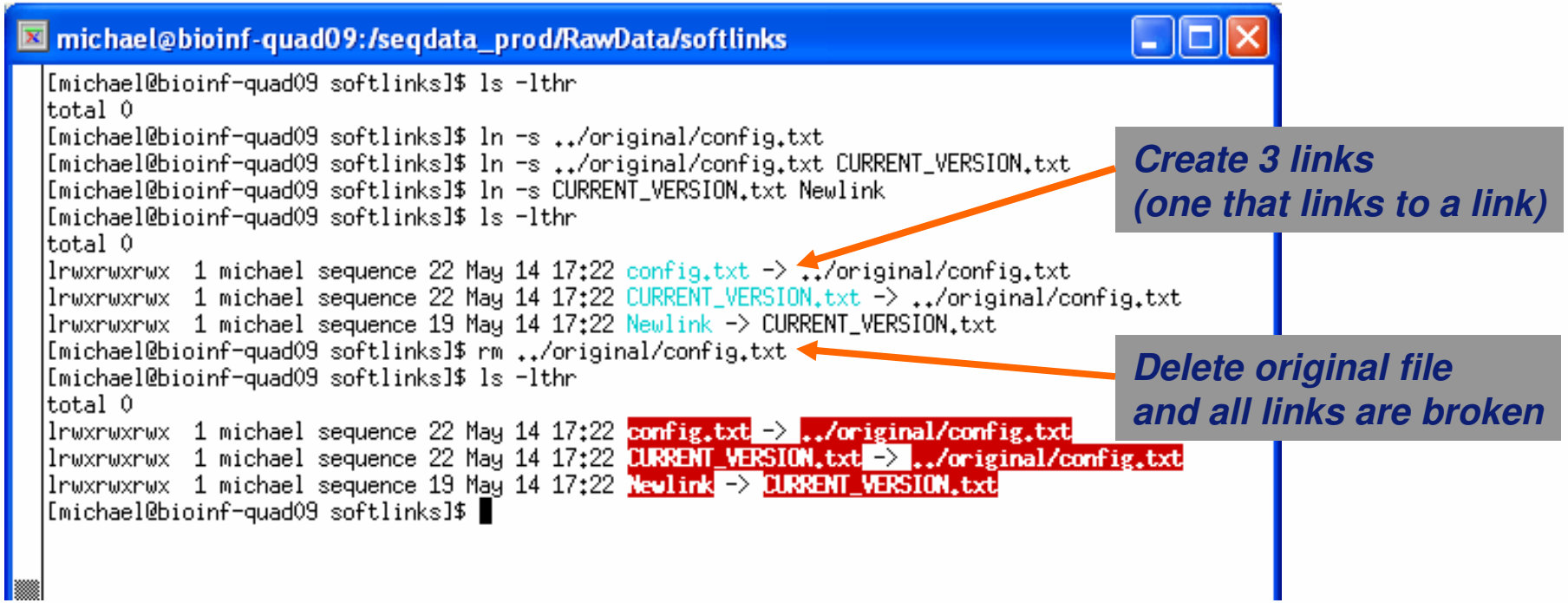
- Symlinks often colored to help distinguish them



Broken Links: Extreme Deconstruction

Erasmus MC
Center for Bioinformatics

- **Delete the original file and everything collapses:**



Use for 'Spinning' Large sets of files

Erasmus MC
Center for Bioinformatics

- **I have used this in two scenarios:**

1. Illumina Infinium II

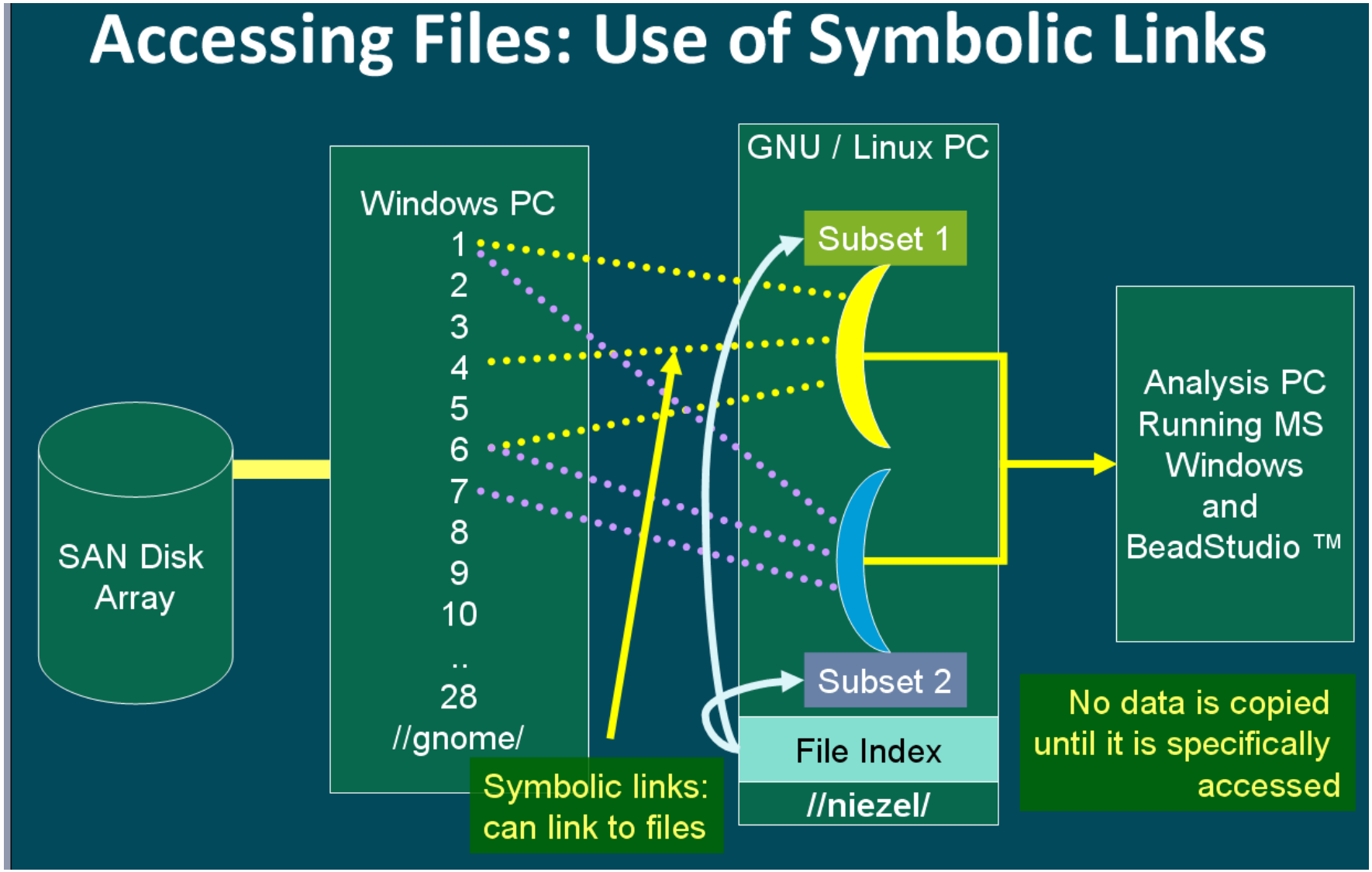
- For producing bespoke collections of the RAW Illumina BeadArray file stored across 28 different directories (actually separate RAID Groups on a SAN Array)
  - Yes – a hack round the 'professional' system's low flexibiility
- Often 3000 out of 10000 pairs of files needed to be presented to a Microsoft Windows .NET application coded/ supported by Illumina, Inc.

2. Genome Analyzer // GAP output

- Lots of both small files and a few big ones as generated by Illumina Inc. GAP (Genome Analyzer Pipeline)
- Need to be collected together for tarring / zipping to make them easy to transfer

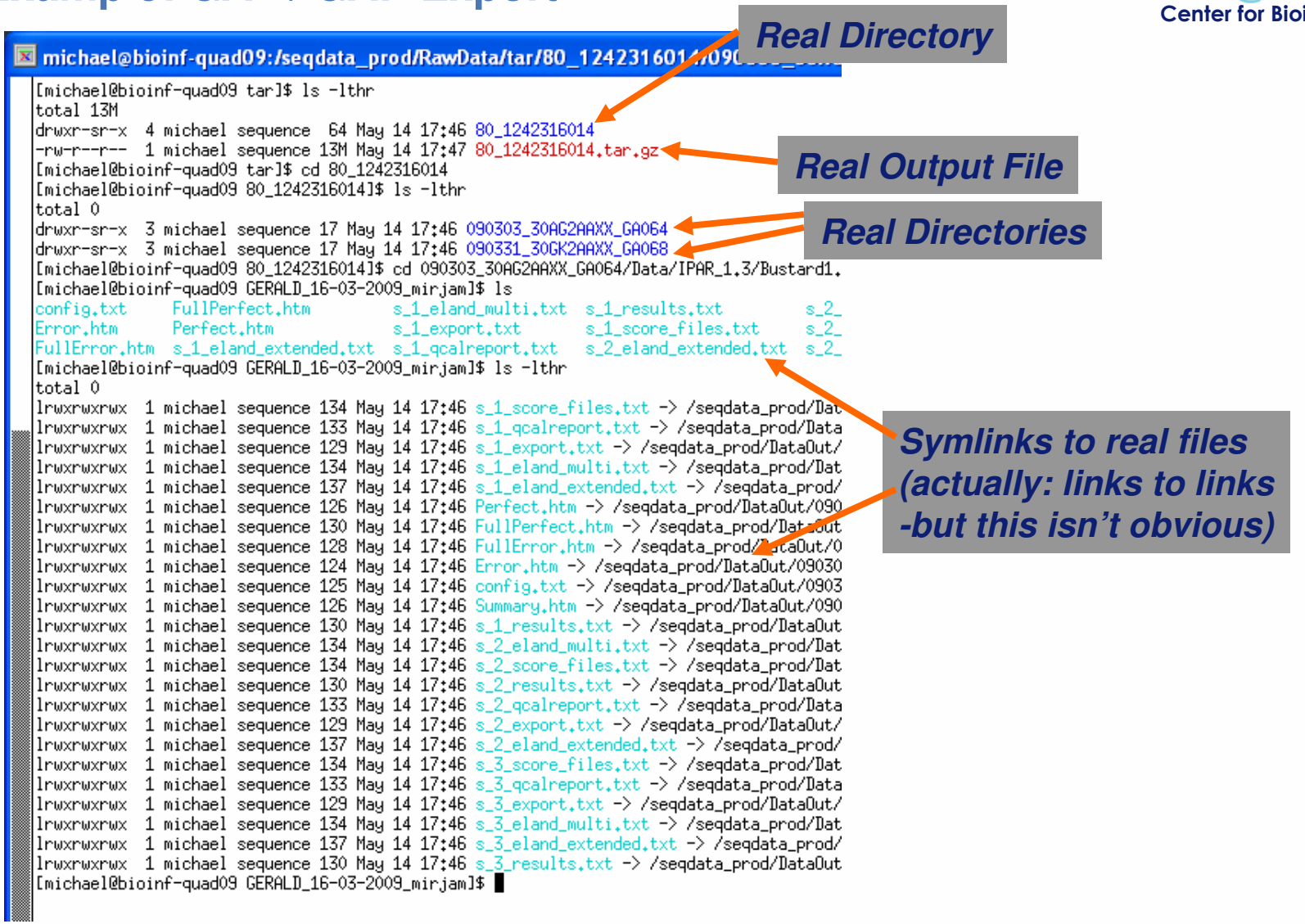
From 'Bridge Meeting' – 2007 (ERGO // Infinium II)

Erasmus MC
Center for Bioinformatics



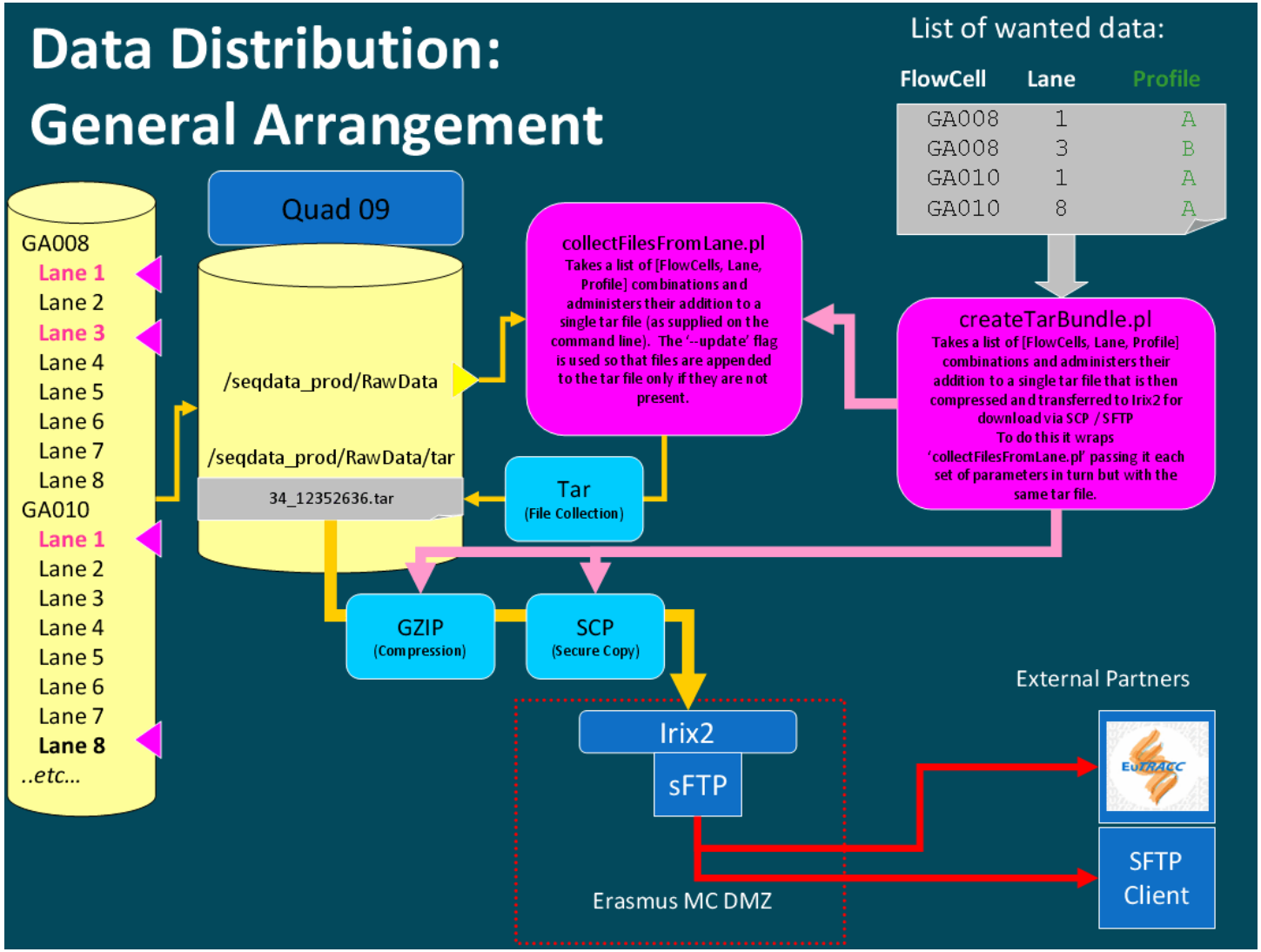
Example: GAII / GAP Export

Erasmus MC
Center for Bioinformatics



From Utrecht Meeting: 2009 (GAII / GAP)

Erasmus MC
Center for Bioinformatics



Michael Moorhouse: May 2009

General Comments

Erasmus MC
Center for Bioinformatics

- **Running 'ln –s' is easy; determining from / to is difficult**
- **Creation of links is very, very fast – especially on the same mount**
  - And deleting them is just as fast
- **I used Perl to build the set of directories – under control so I could adapt the structure if necessary**
- **Many file orientated command line tools use the '-l' parameter to do *special* processing of symbolic links**
  - i.e. cp –l will link to files, not actually copy them*
- **Again: directories & link are small: 32Kb representing 36 Gb**
  - Use –L with du command to 'follow links' and report real sizes

```
[michael@bioinf-quad09 tar]$ du -sh 80_1242316014
32K     80_1242316014
[michael@bioinf-quad09 tar]$ du -Lsh 80_1242316014
36G     80_1242316014
```

* Which replaces effective if verbose constructs like this:
  ls blahdieblah.* | xargs -i ln -s {}