Secure Genome Processing in Public Cloud and HPC Environments

Andre Brinkmann, Jürgen Kaiser, **Lars Nagel**, Tim Süß
Zentrum für Datenverarbeitung

Martin Löwer, Ugur Sahin
TRON gGmbH

Johannes Gutenberg-Universität Mainz

Ci3 CLUSTER FÜR INDIVIDUALISIERTE IMMUNINTERVENTION

JOHANNES GUTENBERG UNIVERSITÄT MAINZ

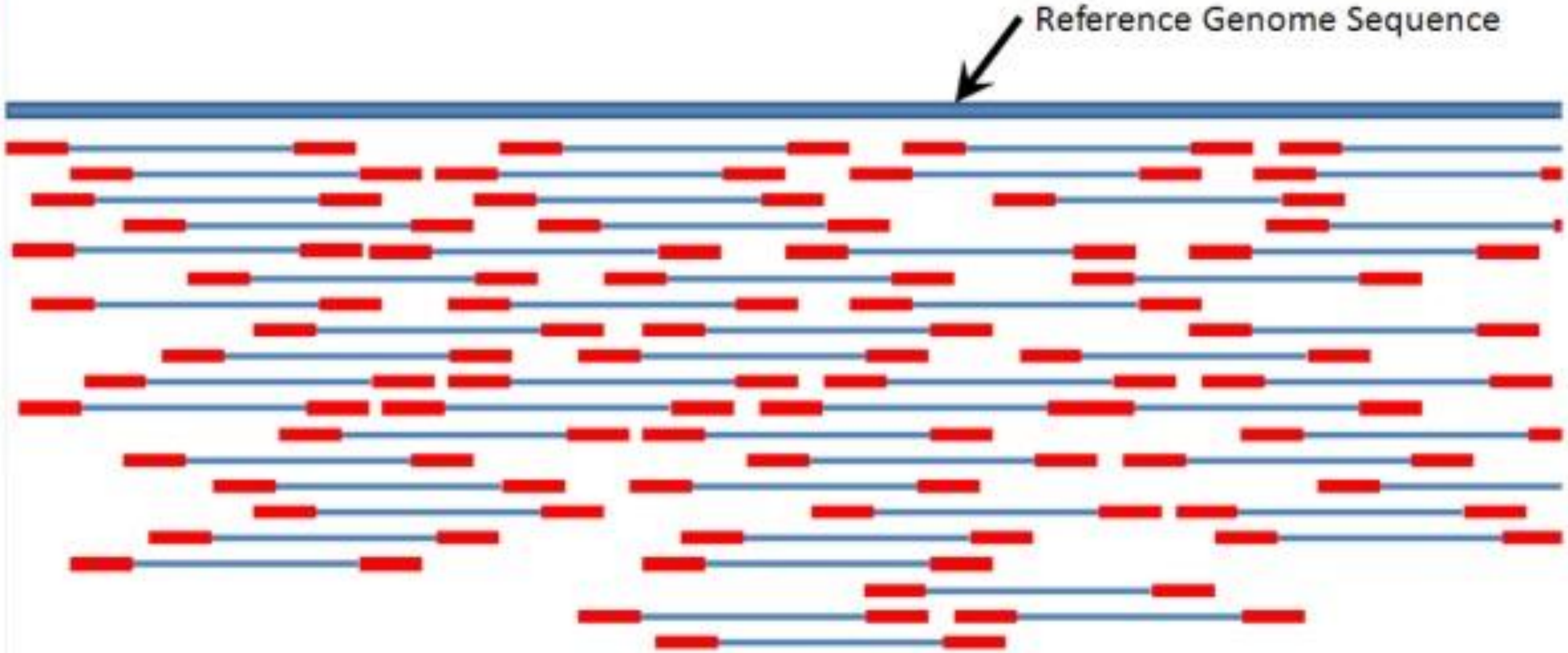
JG|U

Motivation

With high-throughput sequencing or next-generation sequencing it is possible to sequence the entire human genome in one day for less than $1000.

E.g., from: *"New Machines Can Sequence Human Genome in One Day"* (www.sci-tech-today.com)
by B. J. Fikes

NGS Principles - Coverage



Sequence same part many times:

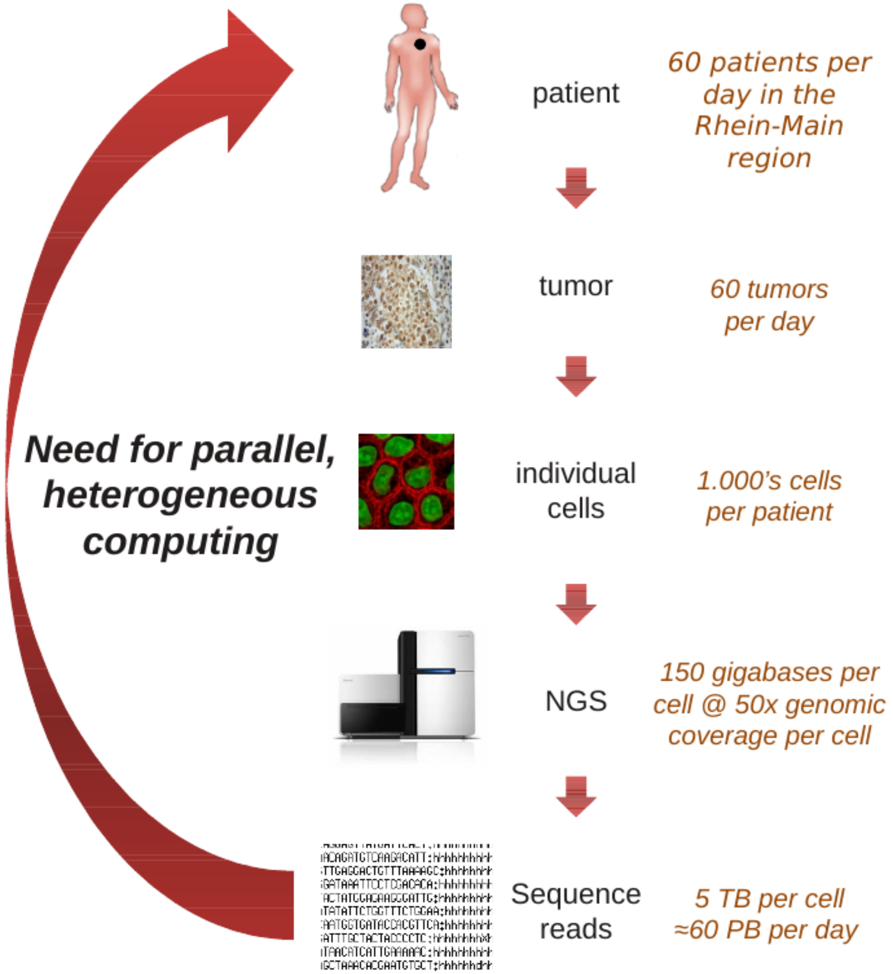
Coverage is number of times a base is covered by a read

Memorial Sloan Kettering Cancer Center

Motivation

- High-throughput sequencing (next-generation sequencing, NGS)
  - Sequencing of entire genomes in a matter of hours
  - Sequencing-by-synthesis requires a lot of computational power for aligning genome data (short sequences, 50-300 bases) to a reference genome and assembling them

- Biological / medical research
  - Treatment of diseases / personalized medicine requires analysis of human genomes at large scale (thousands of individuals per year)



Need for parallel, heterogeneous computing

patient — *60 patients per day in the Rhein-Main region*

tumor — *60 tumors per day*

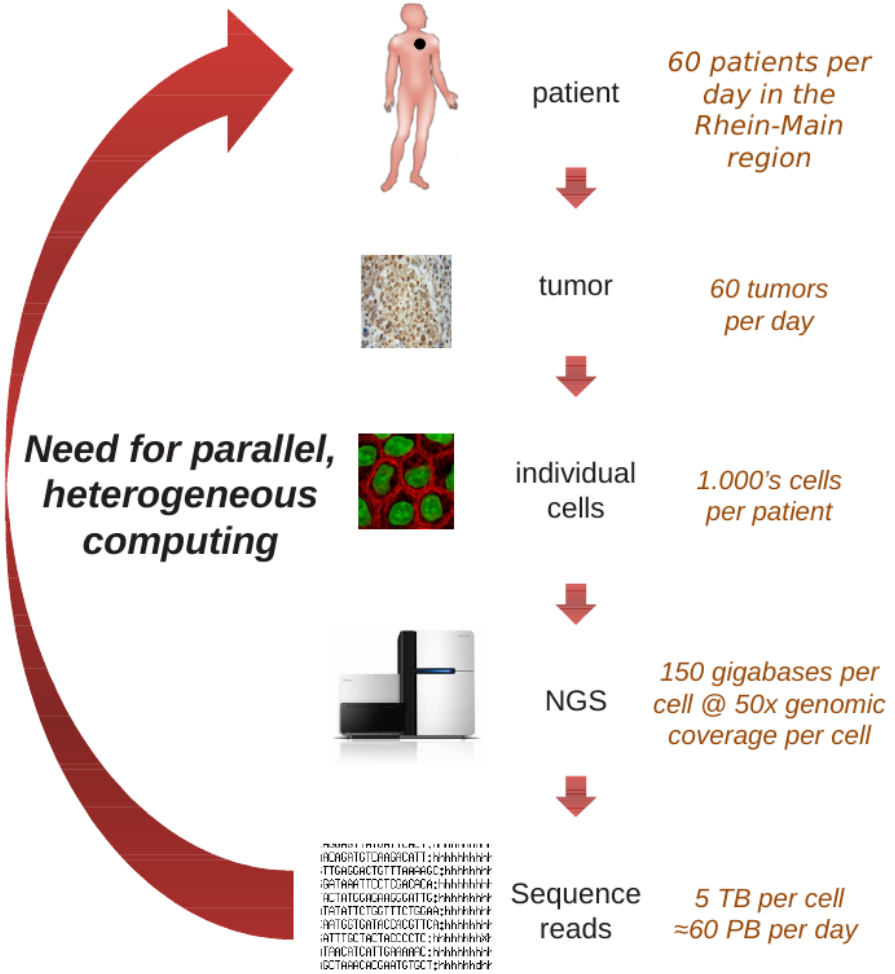
individual cells — *1.000's cells per patient*

NGS — *150 gigabases per cell @ 50x genomic coverage per cell*

Sequence reads — *5 TB per cell ≈60 PB per day*

Motivation

- Public / academic cloud or HPC systems would be inexpensive option, but due to data security expensive in-house facilities are used
  - HPC: shared by many trusted groups, and data security is low
  - Cloud: slightly better data security, but untrusted entities and companies
  - In-house: secure, but expensive to purchase and maintain
- What is actually necessary to secure data?
  - Patients can be identified by ID tags
  - But they can also be identified by their genome data


Need for parallel, heterogeneous computing
patient
60 patients per day in the Rhein-Main region
tumor
60 tumors per day
individual cells
1.000's cells per patient
NGS
150 gigabases per cell @ 50x genomic coverage per cell
Sequence reads
5 TB per cell
≈60 PB per day

Motivation

"[...] our findings show a clear path for identifying whether specific individuals are within a study based on summary-level statistics."

From *"Resolving Individuals Contributing Trace Amounts of DNA to Highly Complex Mixtures Using High-Density SNP Genotyping Microarrays"*
by Homer et al.

Requirements

1. Computationally expensive sequence alignment and assembly must be performed on a potentially vulnerable HPC cluster
2. Data should not be stored on disk in such a cluster and must be completely deleted after computation
3. Data must be secured during transfer and computation using strong encryption algorithms
4. Persons must not be identifiable based on the data (as data may be intercepted during computation)
5. Data security should not significantly degrade computational performance

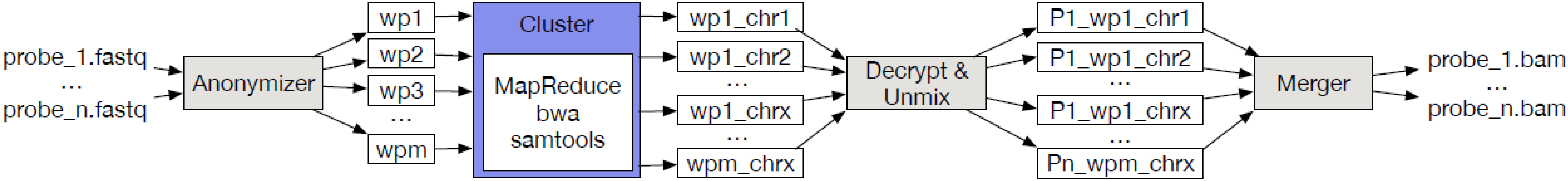
Techniques Applied

- Burrows-Wheeler Aligner

- Outsourcing computations to public HPC or cloud facilities
- Parallel processing using MapReduce approach

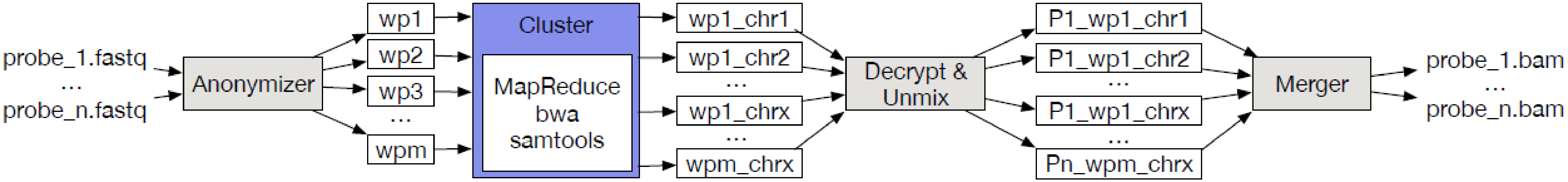
- Adding noise
- Cryptography

Pipeline



- Input:
  - After data acquisition by the sequencers the short reads are contained in FASTQ files
- Output:
  - Aligned reads are returned as BAM files
- Safe environment:
  - The environment of the user / customer is considered safe
  - The cluster and the data transfer to the cluster are considered unsafe

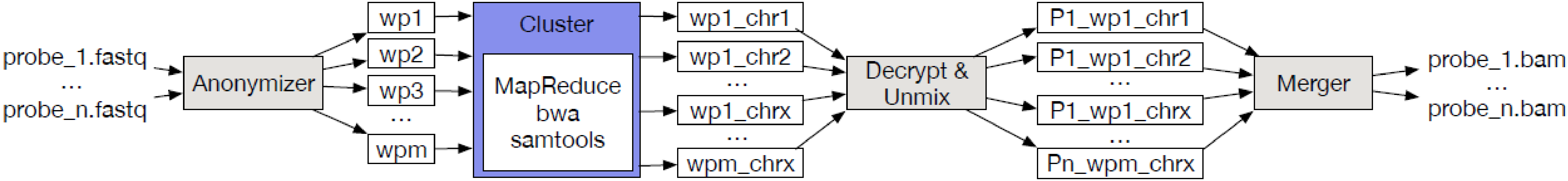
Pipeline



Four phases

- Anonymizer: Input anonymized and divided into work packages
- Cluster: Each work package processed in its own Hadoop job
- Decrypt & Unmix: De-anomymization and ordering
- Merger: Merging output files to one output file per probe

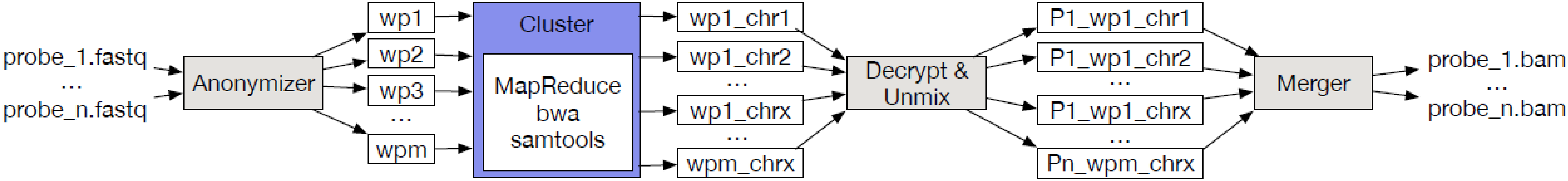
Pipeline: Anonymizer



Anonymizer:

- **Input**: Short reads (sequences) in FASTQ format from many patients read from local storage in safe area
- **Salting**: Fake reads with rare SNPs are added to the mix
- **Identifiers** added and encrypted using AES-128, AES-192 or AES-256
  - As encryption ordering cannot be maintained, electronic code book mode is used with random bits
- **Output**: Work packages (wp) of randomly mixed reads written back to local storage

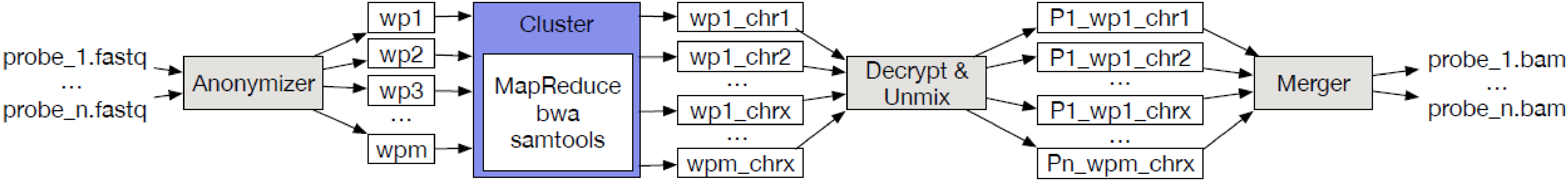
Pipeline: Cluster / MapReduce



Cluster / MapReduce:

- Only stage executed in shared and possibly unsafe computing center
    - Data transfer: OpenSSH
    - Identifier encrypted and data salted, randomly ordered (Anonymizer)
    - Data is never stored on disk in this stage
- Alignment of reads with respect to given reference genome ...

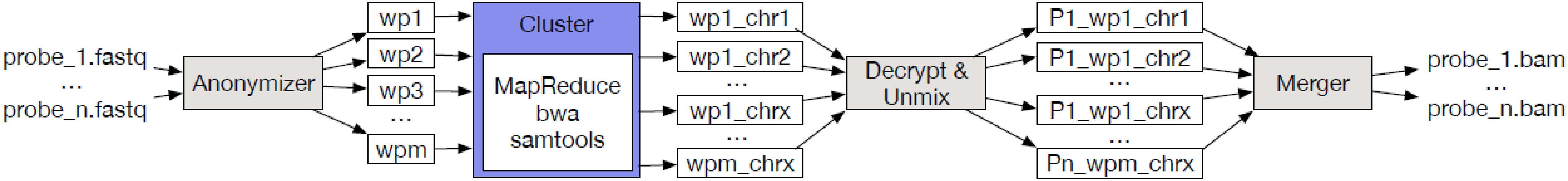
Pipeline: Cluster / MapReduce



Cluster / MapReduce:

- **Hadoop framework**: Each work package processed by a separate Hadoop job
- **Map**: alignment for all reads using the Burrows-Wheeler Aligner (aln); transformation to SAM format (samse, sampe); sorting reads by their position in the reference genome using SAMtools utilities
- **Hadoop partitioner**: Reads partitioned by chromosomes and sent to the reducers
- **Reduce**: Each reducer represents one chromosome; no further processing
- **Output**: One file for each chromosome

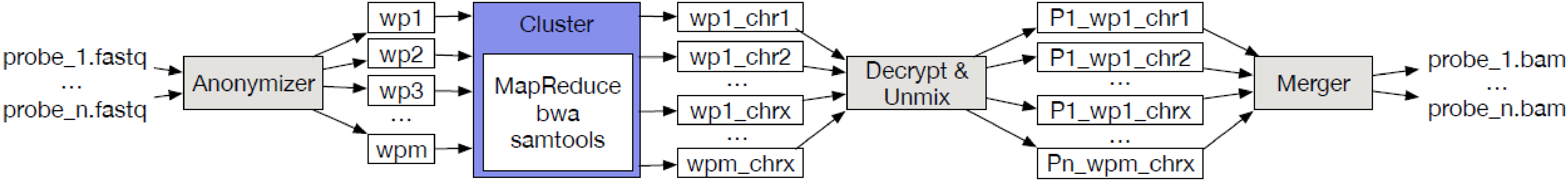
Pipeline: Decrypt & Unmix



Decrypt and Unmix:

- **Input**: Chromosome files of Hadoop jobs
- **Output**: SAM file for every patient / probe + work package + chromosome
  - File name: <probe>_<workpackage>_<chromosome>
  - Identifiers are decrypted
  - Reads are sorted into output files
  - Fake reads are discarded

Pipeline: Merger



Merger:

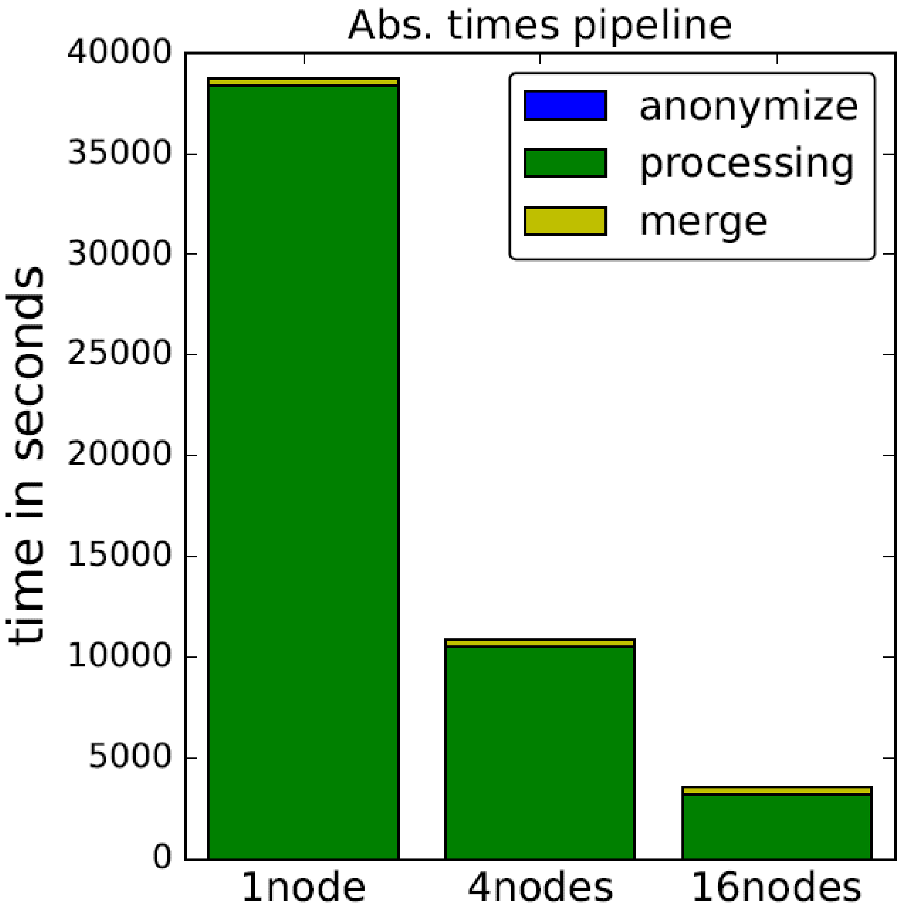
- **Input**: <probe>_<workpackage>_<chromosome> files from previous stage
- **Output**: <probe> file
  - Iteration over input files in chromosome order
  - In each file, iteration over reads in given order
  - Several chromosome files belonging to the same probe must be merged (like merge sort algorithm)
    - Order of reads determined by file header or the reference genome's annotation file / sequence dictionary

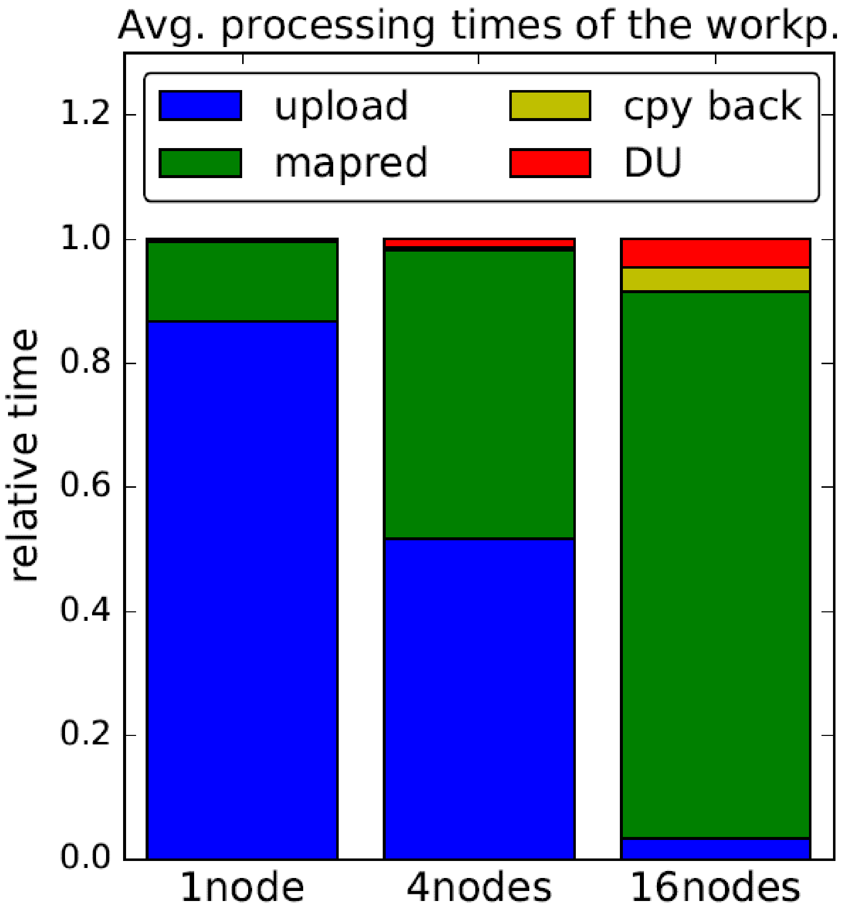
Usage

1. Define workspace (<workspace>) and input directory (<workspace>/input)
2. Copy configuration file to <workspace>/input and modify it if necessary
3. Call starter script: starter -i <workspace>/input -w <workspace>
4. Wait for the pipeline to finish.
5. Obtain results from <workspace>/results

- Once started, pipeline does not require any user interventions
- Configuration file defines cluster side and can be determined by provider
    - Pipeline / cluster settings (size of work packages, #nodes, #threads, #pseudo samples, host name, user group on host, batch queue on host etc.), tool settings (Hadoop, BWA, SAMtools and Java) and the location of additional data (reference genome, pseudo genome data)

Performance





- 100 GB of data
- Each node has 64 cores and 128 GB of memory
- High-speed network link (10 GB/s)

Conclusion

- System for processing sensitive genome data in a public environment without harming the privacy of patients
  - **Security**: Secure processing using cryptography and mixing data with fake data
  - **Performance**: Impact on processing performance is small
  - **Scalability**: System scales excellently
  - **User Interface**: Improvable
  - **Implementation**: Pipeline is used at University of Mainz processing data of cancer patients

- Work funded by the *German Federal Ministry of Education and Research* (BMBF) under grant 131A029D (Project "CI3").

Ci3 CLUSTER FÜR INDIVIDUALISIERTE IMMUNINTERVENTION

Bundesministerium
für Bildung
und Forschung