



GenarchBench

Porting a Genomics Suite to A64FX

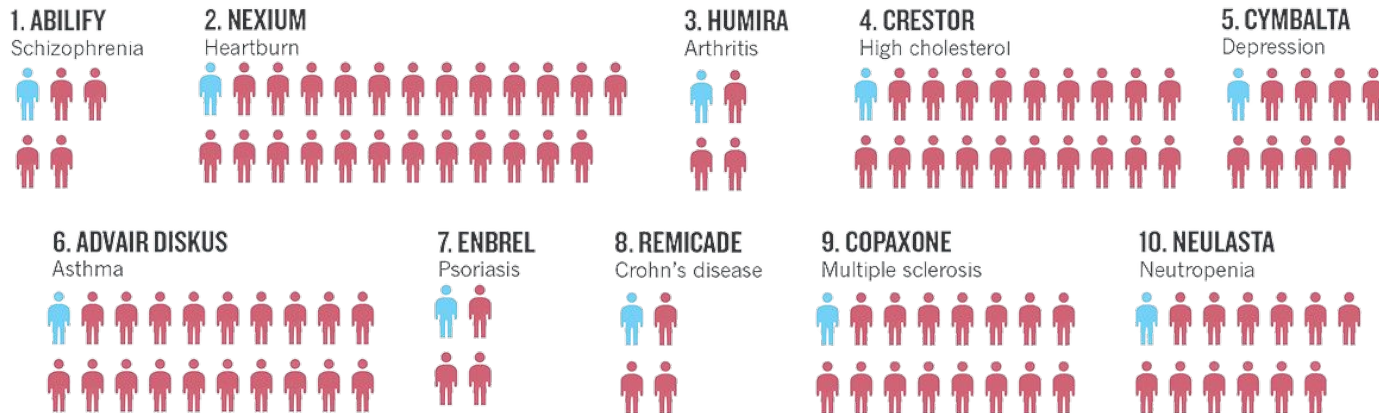
Lorién López-Villellas (lorien.lopez@bsc.es)
Santiago Marco-Sola (santiago.marco@bsc.es)
Jesús Alastruey-Benedé (jalastru@unizar.es)
Pablo Ibáñez (imarin@unizar.es)
Miquel Moretó (miquel.moreto@bsc.es)



**Barcelona
Supercomputing
Center**
Centro Nacional de Supercomputación

June 2, 2022, AHUG ISC22 Workshop

Motivation: Precision medicine



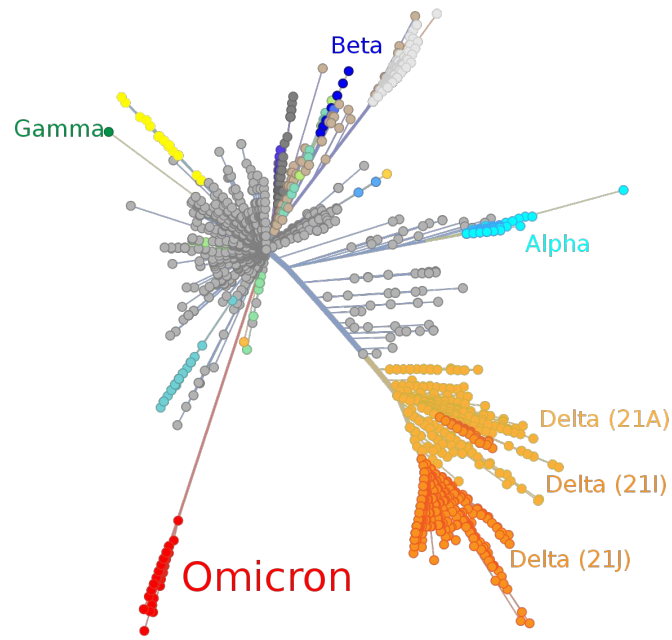
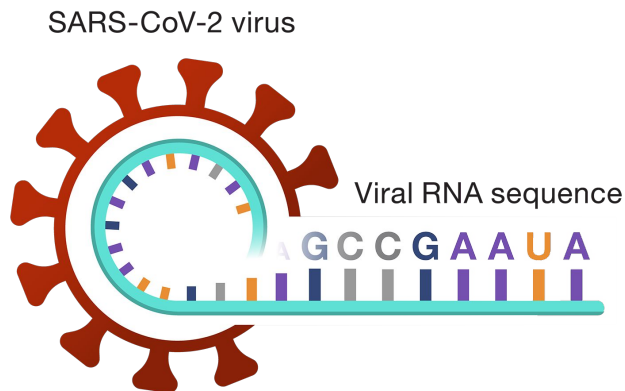
For every person they do help (blue), the **ten highest-grossing drugs in the USA** fail to improve the conditions of between 3 and 24 people (red).

Schork, Nicholas J.
"Personalized medicine: time for one-person trials." Nature.

Motivation: Covid

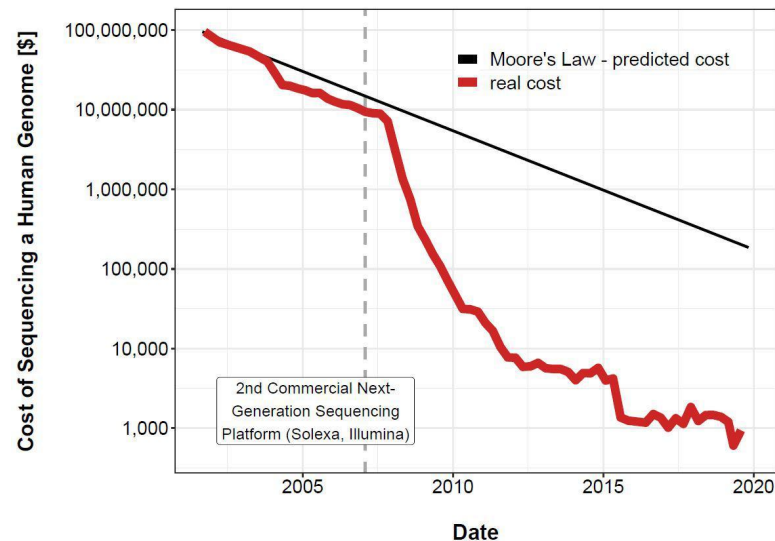
1.2 Motivation: Population-wide tracking of SARS-CoV-2 variants

Scientists worldwide are analyzing **SARS-CoV-2 genomes** to look for known variants of concern, detect emerging ones, and support public health needs. As of December 2021, just the Broad Institute of MIT and Harvard was processing up to **10,000 samples each week**.



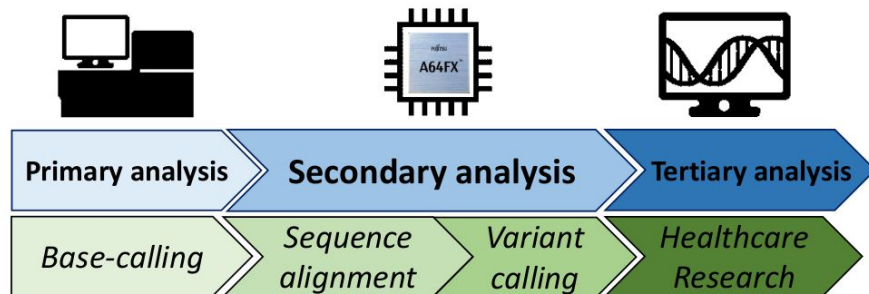
Sequencing Technologies

- Sequencing technologies have improved exponentially in the past decade
 - Reducing cost
 - Increasing yield
 - Simplifying methods
- We can sequence a complete individual (i.e., whole genome DNA) in less 72h for less than \$1000
- Current HPC cannot catch-up with the constant increase in genomic data to analyze



GenarchBench

- **13 kernels** selected from the most widely-used genomic tools:
 - Computationally-demanding
 - Covering all steps of the sequence-analysis pipeline
 - Based on the GenomicsBench[1] and complemented with additional kernels
 - Publicly available: <https://gitlab.bsc.es/smarco/genarch-bench>
- **Goal:** Produce an efficient porting of these kernels to the A64FX processor



GenarchBench

Selected Genomic Kernels

#	Benchmark	Description	Genome Analysis Step	Characterization
1	ABEA	Adaptive Banded Signal to Event Alignment	Base-calling	Compute-bound
2	BPM	Bit-Parallel Myers Alignment	Sequence Alignment	Compute-bound
3	BSW	Banded Smith-Waterman	Sequence Alignment	Compute-bound
4	CHAIN	Seed Chaining	Sequence Alignment	Compute-bound
5	FAST-CHAIN	Fast Seed Chaining	Sequence Alignment	Compute-bound
6	DBG	De-Bruijn Graph Construction	De-novo Genome Assembly	Compute-bound
7	FMI	FM-Index	Sequence Alignment	Memory-bound
8	KMER-CNT	K-mer Counting	Sequence Alignment / Assembly	Memory-bound
9	NN-BASE	Neural Network-based Base Calling	Variant Calling	Compute-bound
10	NN-VARIANT	Neural Network-based Variant Calling	Variant Calling	Compute-bound
11	PILEUP	Pileup Counting	Variant Calling	Compute-bound
12	POA	Partial-Order Alignment	Sequence Alignment	Compute-bound
13	WFA	Wavefront Alignment Algorithm	Sequence Alignment	Compute-bound

A64FX Porting Methodology

1. Code adaptation for compilation and execution on A64FX (ARMv8-a+SVE)
 - Different compilers: GCC and Fujitsu compiler (FCC)
2. Functional verification
 - Some bugs only appear when executing in A64FX
 - Different results than Intel → Find bugs/understand why
3. Performance profiling
 - Limited support for common profiling tools (e.g., perf) → Learn new ones (FIPP, FAPP)
4. Performance tuning
 - SVE Intrinsics, compiler pragmas, loop optimizations (e.g., manual fission), etc.

Performance Characterization

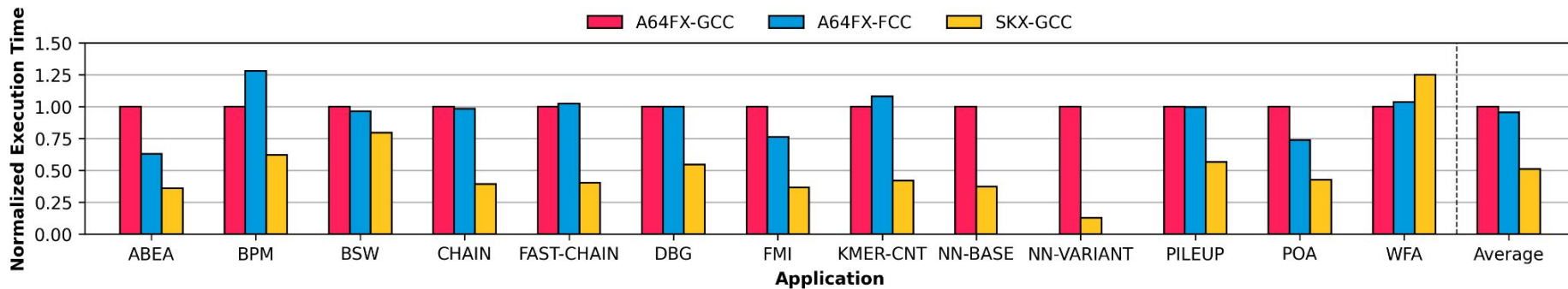
Experimental Setup

	ARM Fujitsu Computing Node	MareNostrum4 Computing Node
	A64FX	2 x Intel Xeon Platinum 8160 (SKX)
Cores	4 x 12	2 x 24
Frequency	2.1 GHz	2.2 GHz
Main memory	32GB HBM2	96GB DDR4 2667 Mhz
Peak bandwidth	4 x 256 GB/s	2 x 120 GB/s
L1	64 KB (4-way)	32 KB (4-way)
L2	N/A	1 MB
LLC	4 x 8 MB	2 x 33 MB
Issue rate	4	4+1
ROB (entries)	128	224
Application Inputs/Datasets		
Small		Large



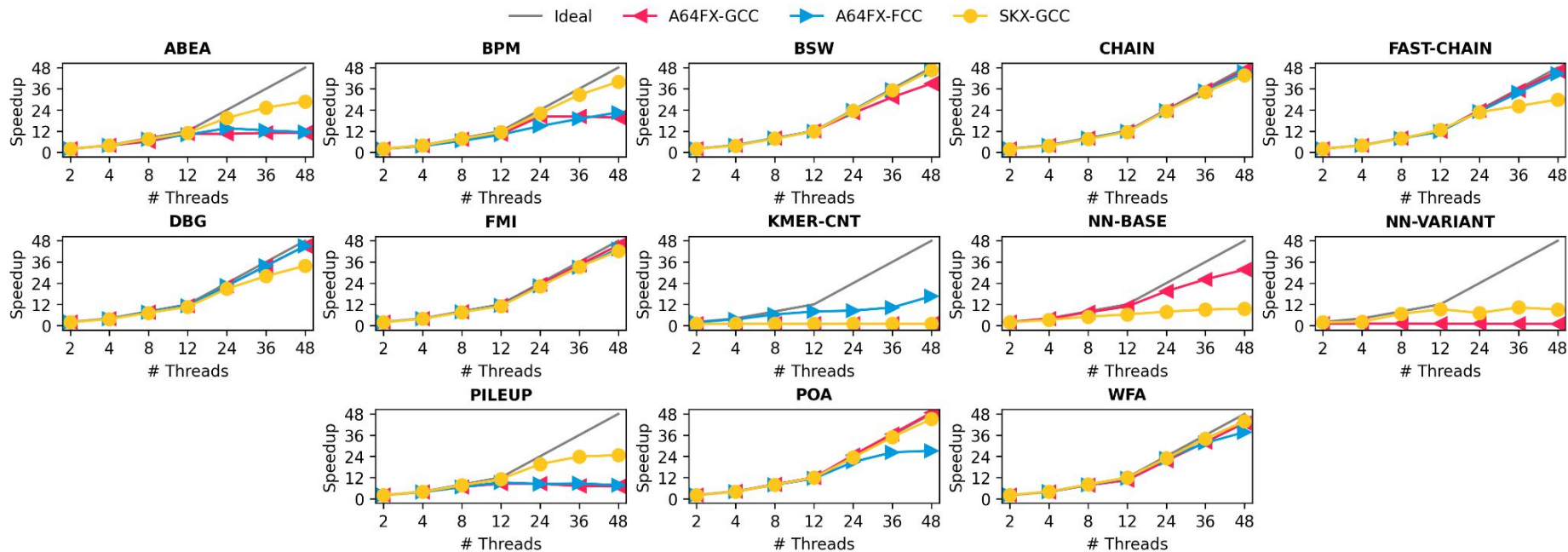
Performance Characterization

Single Thread Execution Time



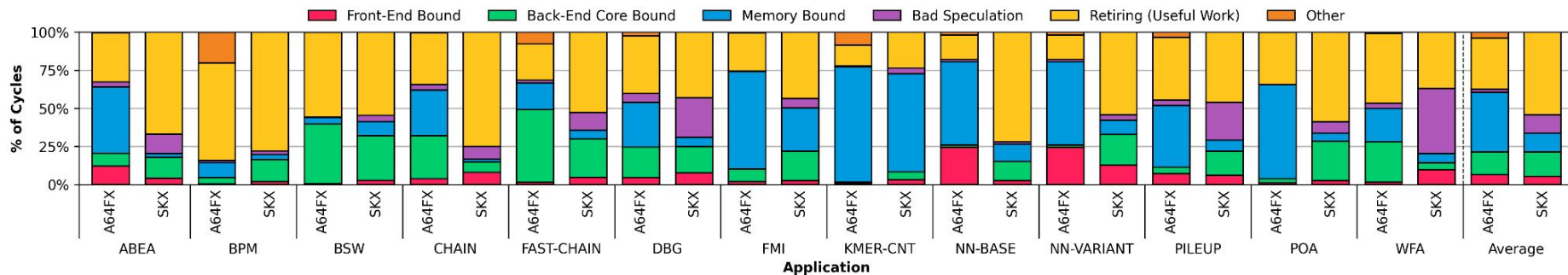
Performance Characterization

Scalability



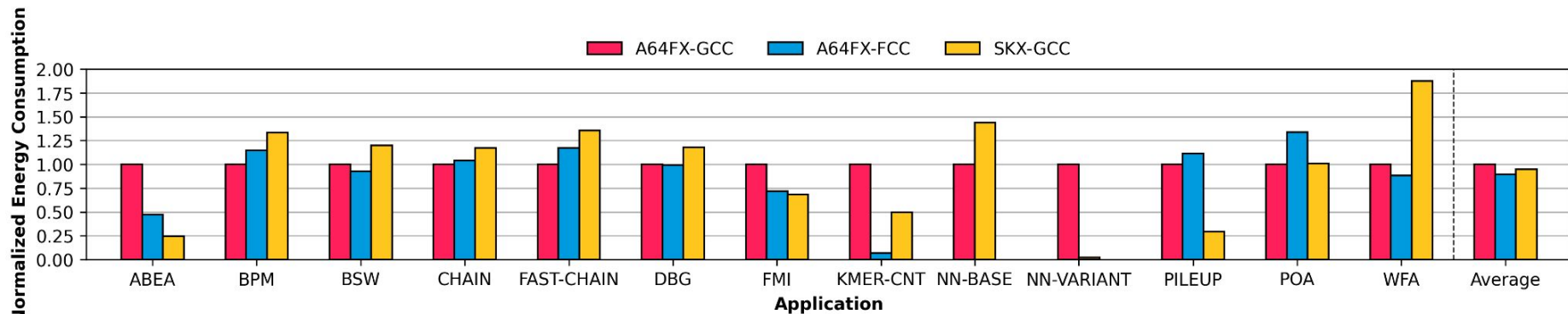
Performance Characterization

Cycle-Accounting



Performance Characterization

Energy Consumption with 48 Threads



Conclusions

- **A64FX requires code adaptations and tuning** efforts to exploit its potential
 - Code and compilation adaptation
 - Usage of different profiling tools
 - Required validation of ported tools and computation kernels
 - Aggressive code adaptations to obtain good performance
- A64FX demonstrates to be **energy efficient** (lower power consumption)
- Compute-bound applications are suitable to exploit A64FX resources
- Memory-intensive applications suffer on the A64FX
 - Limited memory hierarchy resources and high latencies (compared to other processors)
- **On-going work**
 - Obtain the instruction mix of the applications, Roofline model, further acceleration



GenarchBench

Porting a Genomics Suite to A64FX

Lorién López-Villellas (lorien.lopez@bsc.es)

Santiago Marco-Sola (santiago.marco@bsc.es)

Jesús Alastruey-Benedé (jalastru@unizar.es)

Pablo Ibáñez (imarin@unizar.es)

Miquel Moretó (miquel.moreto@bsc.es)



**Barcelona
Supercomputing
Center**

Centro Nacional de Supercomputación

June 2, 2022, AHUG ISC22 Workshop

References

- [1] Arun Subramaniyan, Yufeng Gu, Timothy Dunn, Somnath Paul, Md. Vasimuddin, Sanchit Misra, David Blaauw, Satish Narayanasamy, Reetuparna Das. GenomicsBench: A Benchmark Suite for Genomics. In IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS). 2021 (to appear). URL: www.github.com/arun-sub/genomicsbench