Today, sequencing is done fast by chopping techniques, machines shine light on the **DNA** and based on the reaction they get from the **DNA** they can record different bases of the **DNA**. Others pass the **DNA** through a Nano-scale hole and different bases cause different perturbations so based on measuring the current they can detect the bases and sequence them. But, all of them are not perfect, so they chop DNA into smaller portions and try to work on them and sequence them. To have a vision a human DNA has 3.2 billion base pairs, but some of these just chop it into pieces containing at maximum thousands of pairs. Also, as the number of pairs sequenced increases the errors increase and more computation is needed, which we count it as a downside. As a conclusion, the bottleneck comes from our computation paradigm which is based on a move data, compute, and store. We have to rethink memory.

Genome sequence analysis can be categorized in two main methods: (1) Read mapping: method of aligning the reads against a known reference genome to detect matches and variations. (2) De novo Assembly: method of merging the reads in order to construct the original sequence. Both of these methods are computationally heavy processes.

**Read Mapping**: Map many short DNA fragments (reads) to a known reference genome with some difference allowed.

**The Gist**: DNA read mapping is heavily bottlenecked by Data Movements. Processing-in-Memory can alleviate the bottleneck. However, we need to design mapping and filtering algorithms to fit processing-in-memory. In this way, crossing layers can be very rewarding by increasing the efficiency, by providing information from higher layers to lower layers in computational devices.

**Note**: The bigger memory capacity is always slower because of longer interconnects that slows down data movement.

**Run-ahead Execution Idea**: Most of the time, CPUs just are waiting for memory to get the data to do process on them. We can execute some instructions beforehand so we can improve our performance.

DRAM stores charge in a capacitor. The capacitor must be large enough for reliable sensing. Access transistor should be large enough for low leakage and high retention time. Scaling beyond 40-35 nm (2013) is challenging (By ITRS in 2009). Today, DRAM cells have scaled to 16-17 nm, while the logic has scaled to 5-7 nm.

Main Trend: Hybrid Main Memory. Hardware, software manage data allocation and movement to achieve the best of multiple technologies.

**Hybrid MM** = **DRAM** (fast, durable, small, leaky, volatile, high-cost) + **PCM** (Large, non-volatile, low-cost, slow, wears out, high active energy)

In a multi-core system, cores interfere with each other when accessing shared main memory. Uncontrolled interference leads to many problems like QoS and performance degradations.

As the systems get larger and more heterogeneous, DRAM is looking like a bus in the whole system. Like past that we detoured bus topologies for some architectures, we have to rethink it too.