Ion Torrent in a Little Detail

Sebastian Magierowski

***Abstract*—Some background and details on the IonTorrent sequencing machine**

I. Ion Torrent and Semiconductors

What is special about Ion Torrent's sequencing technology? They are using a CMOS sensor array chip to "perform all the data collection necessary for a simple, on-chip, sequencing chemistry." [1]. As such they note, "this is **unlike all other sequencing instruments** currently commercially available, which rely on extensive "off-chip" instrumentation to collect data, typically via sophisticated optical stems and specialized scientific grade CCD cameras." [1].

Boldly, IonTorrent claims that theirs are the "**first commercial biosensor arrays** for any purpose (e.g. DNA sequencing or DNA microarrays) ever fully embodied as a CMOS device" [1]. Thus, they see themselves as **the only company** presently capable of taking true and direct advantage of Moore's Law. It is also interesting to note that Ion Torrent's sequencing chemistry is "**the only** commercially available chemistry that uses entirely natural nucleotides." [1]. The other techniques likely rely on the addition of light-sensitive markers/dyes. They also claim that their "chip sensor arrays in development are **the most high-throughput sensor array chip devices ever produced for any purpose**" [1].

II. Ion Torrent Technology Overview

The Ion Torrent technology exploits several fundamental discoveries [1]:

1) Introduction of a viable ion-sensitive transistor sensor **device** (pHFET) by Bergveld in 1970.
2) Sakurai and Husimi's demonstration that a pHFET could **sense polymerase** incorporation through H^+ release in 1992.
3) Bausells *et al.* demonstration of Bergveld's **device in CMOS** in 1999.
4) Bergveld's idea of a multi-sensor "pH Camera" which led to Milgrew's proposal for a **scalable array** architecture in 2003.
5) Rothberg's invention of the 454 massively parallel sequencer over 1999–2005.

Some critical challenges that were overcome include [1]:

1) Trapped charge in devices.
2) Creating a well on top of array without damage.
3) **Devising an architecture that could move the signals off-chip at the requisite extremely high rates.**

And the new technical challenges remaining [1]:

1) Processing large amounts of data produced in short periods of time.

Many thanks to EMIL's friends.

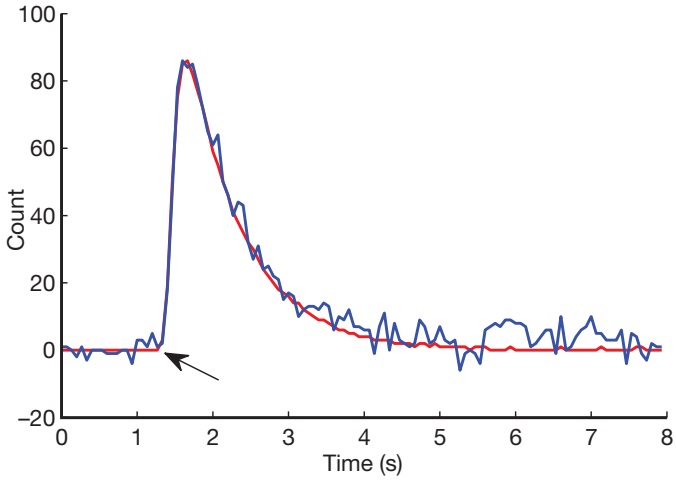

Fig. 1. Characteristic signal pulse sensed by Ion Torrent ISFET sensor with the background signal removed (an important point if you refer to Fig. 2). [2]. The signal pictured seems sampled at roughly 15 Hz.

III. Sensor Notes

An incorporation event causes the release of protons which are registered as a characteristic pulse as pictured in Fig. 1. If multiple nucleotides are incorporated within a single even, this pulse would simply be higher and stretch further in time. It is up to pattern recognition circuitry employing learned knowledge of baseline patterns to correctly discern pulses associated with single and multiple incorporations. This discernment is (almost certainly) done by a digital detector, which works on samples of the reaction pulse shown in Fig. 1. These pulses are sampled at roughly 100 Hz [1], although the example pulse given in Fig. 1 is 15 Hz.

The problem of pulse identification is just one challenge. As a chemical is flowed across the sensor wells of an IonTorrent (IT) chip, it causes changes in pH regardless of an incorporation or not. Thus, it is up to the detection circuitry to erase this background signal. A comparison of the signals, in terms of pH, encountered during incorporation (blue) and no incorporation (red) is shown in Fig. 2.

I don't know exactly the reason for the signal dynamics pictured in Fig. 2. Clearly, the background contribution is substantial. The rise in the background signal in Fig. 2 (red) is potentially due to the flow of reagent into the well resulting in a rise in charge, which peaks and finally drops off once the majority of ion-releasing reactions have completed. Before a follow-up nucleotide wash can be initiated the system must either wait for the previous background signal to drop to zero, or perhaps use another non-reacting reagent to help accelerate the removal of the background signal. In fact, this is indeed what happens, a **wash cycle is used to clear up remnants** of

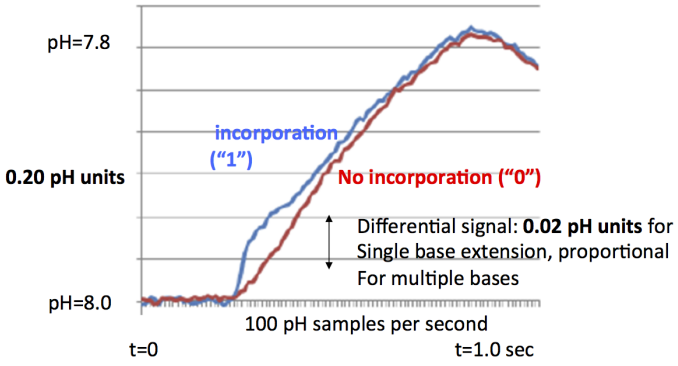

Fig. 2. The actual IT signal with background contribution present [1]. Clearly, the background makes a substantial contribution to the overall signal.

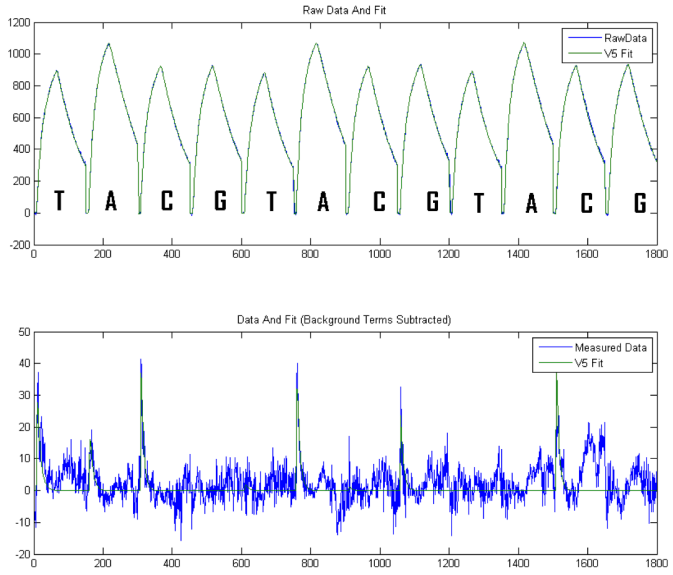

Fig. 3. A series of raw signals (top) obtained as a result of a repetitive IT flow strategy (ACGT) and the pulse signals with the background removed (bottom) [1].

a nucleotide flow in preparation for a following nucleotide.

Very interestingly to deal with background elimination IT is keen on not loading too many wells with beads (up to 90% well loading is ok) [1]. Having empty wells distributed across the chip helps the signal processing software keep track of local variations in the background signal that it can then correct for (on a local basis).

Another look at the signals gathered as part of the sequencing process are shown in Fig. 3. Once again, the pulses shown at the top figure include the substantial background signal contribution. The y-axis denotes the recorded signal in mV. So it looks like we get net signals (i.e. with background) of between about 0 and 1 V with actual incorporation pulses (i.e. with the background removed) **peaking at only 40 mV**.

After their 1-V peak (top curve in Fig. 3), the net signals (i.e. incorporation and background) slowly receded as the reaction is used up. Their sudden drop to zero is caused by the wash cycle introduced in between nucleotide flows.

I am not certain what exactly the y-axis refers to, but my guess is sample number. In the figure it seems that each

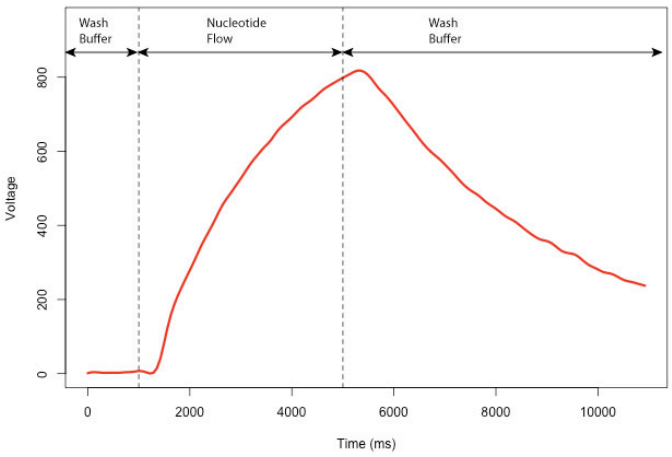

Fig. 4. Another capture of a measured raw signal (i.e. with background present), with important flow cycles labelled [3].

flow cycle lasts about 175 samples. If my 15-Hz sampling approximation regarding Fig. 1 is correct than this amounts to $175/15 \approx 11$ s to flow each chemical.

The nature of the signal collected by IT sensors (i.e. small bump atop a background signal as shown in Fig. 2) would seem to **preclude a simple detector at the sensor front-end**. This coupled with the variation in signal encountered between each well seems to require the need to send samples of the raw signals for post-processing in the base calling logic. However **if a means of reliable and compact base calling could be devised right at the sensor location** it would seem to potentially have good impact on the behaviour of the system.

IV. Pre-Processing

We now discuss the process of turning raw sensor measurements into single numbers (i.e. one whole pulse associated with an incorporation become one number). Forgive me as this section will slightly repetitive of the material presented in § III.

A. Raw Signals

An IT `DAT` file contains the samples of a pulse captured from a single well during a single flow. An example pulse is again shown in Fig. 4. As there are typically 260 flows then for each well 260 `DAT` files will be generated.

As indicated in Fig. 4 a flow in fact consists of 3 phases [3]

1) Wash buffer (well specific baseline measurements)
2) Nucleotide flow (acidic)
3) Less acidic wash

Another example of the type of signals that are collected with incorporation occur is shown in [3].

B. Background

As already noted in order to extract the **signal incorporation profile** (i.e. a pulse such as that in Fig. 1) the background contribution must be removed. It seems that IT uses some **multi-parameter model** of this background signal [3] that it obtains by looking at measurements from the most proximate

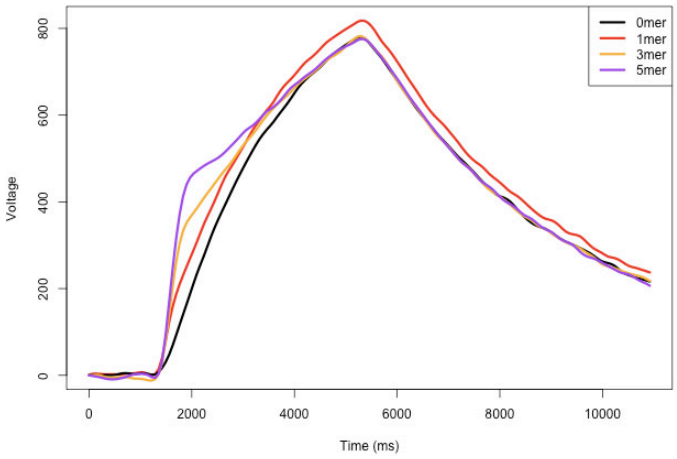

Fig. 5. Another capture of a measured signal (i.e. with background) at different numbers of incorporation for a T nucleotide [3].

empty wells (i.e. no bead inside) to the one who's profile you want to extract (see the 90% note above).

This is very interesting: *The parameter fitting, which uses linear algebra is the most computationally expensive task in the whole Analysis pipeline. The use of* **NVIDIA Telsa GPU** *has greatly reduced processing time in Torrent Suite v1.4.* [3].

Can we compete at this level at all? Perhaps as a solution better integrated with bioinformatics? Or do we not stand a chance against NVIDIA?

V. Base Calling

How is a raw IT incorporation signal (see bottom Fig. 3) converted into a predicted nucleotide sequence? An iterative procedure is applied to accomplish this after a sequence of some length ($\sim$ 200 base pairs) has been synthesized.

A. Initial Key Normalization

The first problem is to normalize your pulse amplitudes, how do you know what voltage corresponds to no incorporation (0-mer) and which to a single (1-mer) or more (n-mer) incorporation? This is where **normalization** comes in.

IT claims that each of its bead-attached sequences starts with a 5-mer adapter sequence also called a **key sequence** [1]. In another source I have seen this described as a 4-mer test fragment ATCG [4]. This sequence is used as a normalization key, when subjected to a 7-step flow sequence consisting of TACGTAC (numbered 0 to 7) I expect a 0-mer signal at flows 0,2,3,5 (i.e. zeros for TCGA flows) and a 1-mer signal at flows 1,4,6 (i.e. 1's fro ATC flows). Why isn't a G 1-mer included? Since the G is the last nucleotide in our key we are unsure what follows it. If it is a G then the signal we get could correspond to a 2-mer rather than a 1-mer. I suppose a simple way to handle this would be extend the key to a 5-mer adapter sequence (as IT claims).

The method of signal normalization then applies the following steps [4]:

1) Find the average 0-mer signal (from the 4 keys you collected).
2) Subtract this average 0-mer signal from all the flows ($\sim$ 200) that you collected. If a subtraction goes below zero, set it to zero.
3) Find the average 1-mer signal (from the 3 keys you collected).
4) Divide each of your 0-adjusted flows by dividing all of them by your 1-mer average (this assumes a linear relationship between 1-mer signal amplitude and n-mer signal amplitude).
5) With each flow we can expect a slight drop in the recorded signal level (perhaps 0.05%) [4]. Thus, to correct for this (**"signal droop"**) we should introduce a "decay correction") and slightly boost each flow i by the expected drop. For example by a factor of $1+i\cdot 5\times 10^{-4}$.

B. Iterative Normalization

After the initial correction above we can use a sliding window of measurements (perhaps over 50–60 flows) to establish updated averages for 1-mer values that are used to further normalize ensuing measures.

C. Phase Correction

Besides droop, two other key errors are present in the sequencing-by-synthesis process that collectively lead to so called **phase errors**. One is **incomplete extension** (IE), a situation where a flowed nucleotide was expected to attach to an available position in the beaded template but for some reason did not. This amounts to a **false negative** and is a significant problem in sequencing-by-synthesis. Another, relatively negligible issue is the **carry forward** (CF) problem where for some reason a flowed nucleotide bonds when it was no supposed to. This amounts to a **false positive**.

The phase errors themselves refer to the occurrence of IE and/or CF events which shift the sequence suffering them out of sequence with its neighbours. For instance in the case of IE with the sequence AGTCTA present (on the bead) and a flow introduced with the intention of binding to A, some sequences will miss A essentially delaying their synthesis until the next flow intended to bond with A. From this point on this sequence is out of phase with its correctly operating neighbours (on the bead).

The IE problem can be quite severe, at an 0.01 IE probability only 60% of the sequences will be in phase by the 50th base. If IE is improved to 0.001 80% of the reads will be in phase by the 200th base.

To battle the phase correction problem IT uses a **redundant flow cycle** with a period of 32 nt's (unlike the simple 4 nt period implied in Fig. 3.

D. Thresholding

At one time it seems that IT took a **greedy algorithm** approach to base-calling [3]. This involved a hard thresholding scheme where for example a signal of 2.49 was called a 2-mer and a signal of 2.51 was called a 3-mer. This is a distorting decision as difference of 0.02 is amplified to 1.

TABLE I
ION TORRENT'S SENSOR ARRAY SERIES

Chip	Well Size [μm]	Well Pitch [μm]	Pixel Tran.	Chip Area [mm^2]	Sensor Count M	Tech. Node [nm]	Year
314	3	5.1	3T	10.6×11	1.2 (1.5)	350	Q1 2011
316	3	5.1	3T	16.9×17.1	6.3 (7.2)	350	Q3 2011
318	3	4.1	2T	16.9×17.1	11 (13)	350	Q1 2012
PI	1.25	1.68	2T	23.7×20	154 (165)	110	Q3 2012
PII			2T	23.7×20	(660)	110	Q1 2013

VI. CHIP SCALES

Ion Torrent's chips have taken extensive advantage of CMOS scaling opportunities. Table I summarizes some of the characteristics of Ion Torrent's 3-series and their Proton-series chips.

VII. SEQUENCING ABILITY

A tradeoff to consider in sequencing is that:

- The longer the segment read the less accurate the segment base-call
- The shorter the segment read the less accurate the (ensuing) whole genome reconstruction

IT keeps this in mind when presenting their sequencing accuracy metric which, as I explain shortly, is given in the form [L]AQ[N] and AQ[N].

The [L]AQ[N] measure denotes the accuracy of the first L bases of some segment read. The accuracy is measured by noting how closely it matches some reference genome when the read is correctly aligned to the corresponding genome locus (hence AQ – "aligned quality"). The actual read length may be longer than L.

For example, 100AQ20 notes that of the first 100 bases called there was one error percentage of $10^{-20/10} = 1\%$ error rate. So, expect one error in the first 100 bases of your read. As you might have noticed the quality number is on a logarithmic scale. Think of it as a value in dB. If its 20 your "signal-to-noise" ratio is 20-dB, which is 100, a number to be interpreted as 1 error in 100. A value of 30 implies an quality of 99.9%.

VIII. NEEDS

What are some IT needs that ASICs could help with?

1) Pre-filtering to produce "obviously low-quality" reads [1].

REFERENCES

[1] B. Merriman and J. M. Rothberg, "Progress in Ion Torrent semiconductor chip based sequencing," *Electrophor.*, vol. 33, pp. 3397–3417, Dec. 2012.
[2] J. M. Rothberg *et al.*, "An integrated semiconductor device enabling non-optical genome sequencing," *Nature*, vol. 475, no. 7356, pp. 348–352, July 2011.
[3] "Challenges in improving ion torrent raw accuracy (Part 3)," http://biolektures.wordpress.com/2011/08/22/challenges-in-improving-ion-torrent-raw-accuracy-part-3/, accessed: 2014-07-05.
[4] "Fundamentals of base calling (Part 1)," http://biolektures.wordpress.com/2011/08/10/fundamentals-of-base-calling-part-1/, accessed: 2014-07-05.