# Illumina HiSeq-2000 Address Transform

*National Center for Biotechnology Information (NCBI) – National Library of Medicine*

*Draft D  Oct 15 2010*

## Problem Statement

The SRA Illumina Genome Analyzer loaders rely on the GA and GA II address field convention (*flowcell:lane:tile:x:y*) to determine the order of spots in the run data file and to detect duplicates. To constrain the size of these fields, the SRA loader allows up to a maximum of 24,576 for X and Y. With HiSeq-2000 the addressing that reflected use of a discrete camera field per tile has given way to a continuous camera field covering much more area, so Y values up to 200,000 are commonly seen. All HiSeq-2000 runs have this problem.

There is no simple way to fix the SRA loaders to adapt to HiSeq-2000 data. Instead the SRA loaders will adopt a more general strategy of address determination and duplicate detection. In the meantime, we propose that submitters transform their HiSeq-2000 data in such a way as to restore the GA addressing parameters without loss of data or information. This approach will be reversible if it later becomes possible to archive the native addresses. Runs receiving this treatment should be specially marked. A fix in the SRA loader code will make this solution redundant. The SRA addressing improvement is expected early 2011.

## Treatment

We have defined the following spot address transformation:

Spot_Tile* = (Spot_Tile x 100) + (Spot_Y / 20000)

Spot_Y* = Spot_Y % 20000

Along with this transformation, some of the transformed spots need to be relocated to later in the *qseq* files to ensure that tile addresses are contiguous, which is another requirement of the SRA loaders. For example, in a file that we received from a submitter s*_1_1_0001_qseq.txt*, is the following sequence of spots (line numbers indicated on the left):

| | | | | | | |
|---|---|---|---|---|---|---|
| 242328: | SL-HAG | 118 | 1 | 1 | 2204 | 19980 |
| 242329: | SL-HAG | 118 | 1 | 1 | 2213 | 19996 |
| **242330:** | **SL-HAG** | **118** | **1** | **1** | **2187** | **20000** |
| 242331: | SL-HAG | 118 | 1 | 1 | 2295 | 19756 |
| 242332: | SL-HAG | 118 | 1 | 1 | 2380 | 19756 |

The address transformation will have this result:

| | | | | | | |
|---|---|---|---|---|---|---|
| 242328: | SL-HAG | 118 | 1 | 100 | 2204 | 19980 |
| 242329: | SL-HAG | 118 | 1 | 100 | 2213 | 19996 |
| **242330:** | **SL-HAG** | **118** | **1** | **101** | **2187** | **0** |
| 242331: | SL-HAG | 118 | 1 | 100 | 2295 | 19756 |
| 242332: | SL-HAG | 118 | 1 | 100 | 2380 | 19756 |

In order for the transformed address on record 242330 to load successfully, it is relocated further into the file as record 245435:

| | | | | | | |
|---|---|---|---|---|---|---|
| 242328: | SL-HAG | 118 | 1 | 100 | 2204 | 19980 |
| 242329: | SL-HAG | 118 | 1 | 100 | 2213 | 19996 |
| 242330: | SL-HAG | 118 | 1 | 100 | 2295 | 19756 |
| 242331: | SL-HAG | 118 | 1 | 100 | 2380 | 19756 |
| … | | | | | | |
| **245435:** | **SL-HAG** | **118** | **1** | **101** | **2187** | **0** |

To accomplish this reordering, the transform must buffer spots that are assigned to a tile (101) not currently being input (100). Then, as the tile value changes on input (to 101), the currently buffered spots (tile 101) are written out and the cycle is repeated.

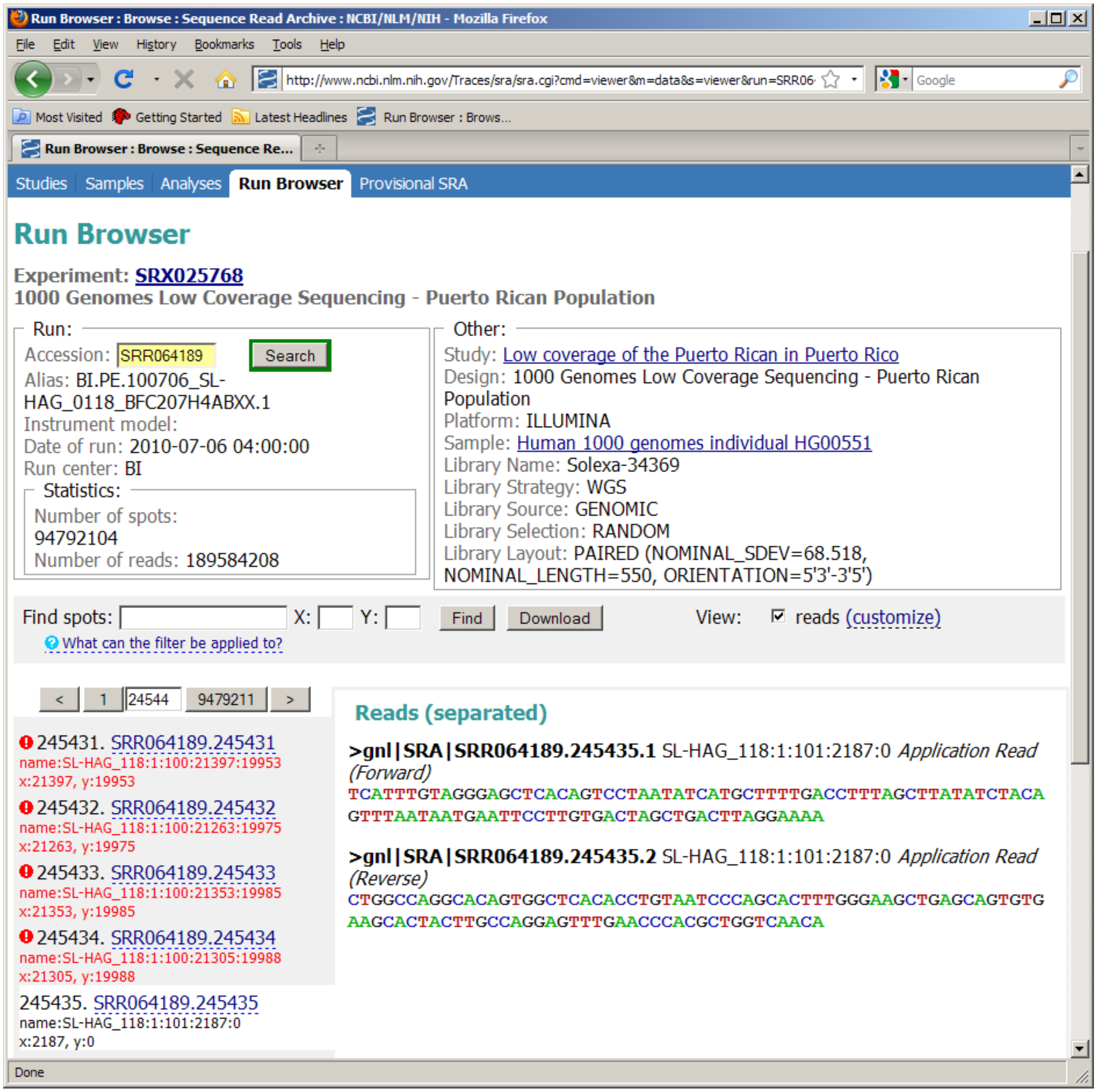
A perl script is provided below that performs this operation on qseq files that should be available in your Illumina run folder.

Please also add a RUN_ATTRIBUTE, *HISeq_address_transform*, with a value of *yes*, to the Run xml in order to record the application of the address transformation.

The SRA submission can contain either the modified *qseq* file(s) (filetype is fastq), or can be converted into SRF format using the illumina2srf utility from sequenceread-2.1.2 (http://sourceforge.net/projects/sequenceread) (be sure to use version 2.1.2 or later, as sequenceread-2.1.1 had problems with the modified tiles (e.g. a tile of 100 changes to 1).

## Example

A screen capture of the modified and loaded run SRR064189 is presented below.


Run Browser : Browse : Sequence Read Archive : NCBI/NLM/NIH - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://www.ncbi.nlm.nih.gov/Traces/sra/sra.cgi?cmd=viewer&m=data&s=viewer&run=SRR06
Google
Most Visited
Getting Started
Latest Headlines
Run Browser : Brows...
Run Browser : Browse : Sequence Re...
Studies
Samples
Analyses
Run Browser
Provisional SRA
Done

## Run Browser

**Experiment: SRX025768**
**1000 Genomes Low Coverage Sequencing - Puerto Rican Population**

Run:

Accession: SRR064189 [Search]
Alias: BI.PE.100706_SL-HAG_0118_BFC207H4ABXX.1
Instrument model:
Date of run: 2010-07-06 04:00:00
Run center: BI

Statistics:
Number of spots: 94792104
Number of reads: 189584208

Other:

Study: Low coverage of the Puerto Rican in Puerto Rico
Design: 1000 Genomes Low Coverage Sequencing - Puerto Rican Population
Platform: ILLUMINA
Sample: Human 1000 genomes individual HG00551
Library Name: Solexa-34369
Library Strategy: WGS
Library Source: GENOMIC
Library Selection: RANDOM
Library Layout: PAIRED (NOMINAL_SDEV=68.518, NOMINAL_LENGTH=550, ORIENTATION=5'3'-3'5')

Find spots: X: Y: [Find] [Download] View: ☑ reads (customize)
What can the filter be applied to?

[<] [1] 24544 [9479211] [>]

245431. SRR064189.245431
name:SL-HAG_118:1:100:21397:19953
x:21397, y:19953

245432. SRR064189.245432
name:SL-HAG_118:1:100:21263:19975
x:21263, y:19975

245433. SRR064189.245433
name:SL-HAG_118:1:100:21353:19985
x:21353, y:19985

245434. SRR064189.245434
name:SL-HAG_118:1:100:21305:19988
x:21305, y:19988

245435. SRR064189.245435
name:SL-HAG_118:1:101:2187:0
x:2187, y:0

### Reads (separated)

**>gnl|SRA|SRR064189.245435.1** SL-HAG_118:1:101:2187:0 *Application Read (Forward)*

```
TCATTTGTAGGGAGCTCACAGTCCTAATATCATGCTTTTGACCTTTAGCTTATATCTACA
GTTTAATAATGAATTCCTTGTGACTAGCTGACTTAGGAAAA
```

**>gnl|SRA|SRR064189.245435.2** SL-HAG_118:1:101:2187:0 *Application Read (Reverse)*

```
CTGGCCAGGCACAGTGGCTCACACCTGTAATCCCAGCACTTTGGGAAGCTGAGCAGTGTG
AAGCACTACTTGCCAGGAGTTTGAACCCACGCTGGTCAACA
```

To view this interactively, please visit :

http://www.ncbi.nlm.nih.gov/Traces/sra/sra.cgi?cmd=viewer&m=data&s=viewer&run=SRR064189

## Method

```
#!/opt/perl-5.8.8/bin/perl
# This is a code excerpt that performs address transformation on Illumina HiSeq-2000
reads in order to
# make them loadable by SRA loaders relying on GA and GA II spot addressing
convention.
#

use strict;

die "\nHiSeq2000_2_SRA.pl  < qseq file >\n\n"
    if ( scalar @ARGV eq 0 && -t STDIN );

my $TILE_INDEX = 3;
my $Y_INDEX = 5;

# Initialize using first line in qseq file

my $firstLine = <>;
my @spot = split(/\t/,$firstLine);
adjustSpot ( \@spot );
printSpot ( \@spot );
my $currTile = $spot[$TILE_INDEX];
my $nextTile = 0;
my @nextTileSpots = ();

# Continue to process qseq file

while (<>)
    {
        my @spot = split(/\t/,$_);
        adjustSpot ( \@spot );

        # Print spots in *$currTile*

        if ( $spot[$TILE_INDEX] eq $currTile )
            {
                printSpot ( \@spot );
            }

        # Determine first *$nextTile* value and start collecting
        # *$nextTile* spots into @nextTileSpots .

        elsif ( ! ( $nextTile) )
            {
                $nextTile = $spot[$TILE_INDEX];
                push @nextTileSpots,\@spot;
            }

        # After *$nextTile* is set, continue collecting *$nextTile*
        # spots into @nextTileSpots

        elsif ( $spot[$TILE_INDEX] eq $nextTile )
            {
                push @nextTileSpots,\@spot;
            }

        # If *$spot[$TILE_INDEX]* is not *$currTile* or *$nextTile*,
        # then set *$currTile* and *$nextTile* to new values.
        # Output spots collected in @nextTileSpots, and start
        # collecting a new set of *$nextTile* spots in @nextTileSpots.

        else
            {
                printTileSpots ( \@nextTileSpots );
                $currTile = $nextTile;
                $nextTile = $spot[$TILE_INDEX];
                @nextTileSpots = ();
                push @nextTileSpots,\@spot;
            }
    }

printTileSpots ( \@nextTileSpots );

#############################################################
sub printTileSpots {
    my $nextTileSpotsRef = shift;
    foreach my $spotRef ( @$nextTileSpotsRef )
        {
            printSpot ( $spotRef );
        }
}

#############################################################
sub adjustSpot {
    my $spotRef = shift;
    $$spotRef[$TILE_INDEX] = ( $$spotRef[$TILE_INDEX] * 100 ) + int (
$$spotRef[$Y_INDEX] / 20000 ) ;
    $$spotRef[$Y_INDEX] = $$spotRef[$Y_INDEX] % 20000 ;
}

#############################################################
sub printSpot {
    my $spotRef = shift;
    print join ( "\t", @$spotRef );
}
```