Using the SRA Data Block Descriptor

Draft B 22 Oct 2009

National Center for Biotechnology Information – National Library of Medicine

The SRA schema version 1.1 release candidate I (SRA_1-1i) supports the following constructs to help describe the assignment of run file objects to SRA run data blocks.

This content will eventually join the SRA XML Writer's Guide.

Multiple Data Blocks

The XML schema allows you to specify multiple data blocks in sequence order, but you are not guaranteed to emit the blocks in any order. Use the new **DATA_BLOCK.serial** attribute to impose a total ordering on the data blocks so that they will get loaded in the order specified.

Example: One SOLiD run broken into 95 pieces for ease of transmission:

```
    <DATA_BLOCK name="VAB_Florence_20080709_1_1000G_10" serial="1">
        <FILES>
            <FILE filename="Florence_20080709_1_1000G_10.0001.0001_0025.srf"
filetype="srf"></FILE>
        </FILES>
    </DATA_BLOCK>
    <DATA_BLOCK name="VAB_Florence_20080709_1_1000G_10" serial="26">
        <FILES>
            <FILE filename="Florence_20080709_1_1000G_10.0002.0026_0050.srf"
filetype="srf"></FILE>
        </FILES>
    </DATA_BLOCK>
```

and so on.

Multiple Samples, User De-multiplexed

The XML schema now allows you to specify multiple data blocks per run each of which is assigned to a subset of the sequencing that is associated with a particular sample. In this case the submitter has de-multiplexed the sequencing run and submitted separate files. A default file may be used to contain the reads that did not get assigned to a particular sample. The **DATA_BLOCK.member** attribute records the pool member name that the reads should be assigned to.

```
    <DATA_BLOCK
      serial = "1"
      name = "FMSX0OV"
      region = "1"
      member_name = "default"
    >
      <FILES>
```

```
        <FILE filename="default.sff" filetype="sff" checksum_method="MD5"
checksum="4026fc6b91ed2ffbef374a665e02802b" />
      </FILES>
    </DATA_BLOCK>
    <DATA_BLOCK
      serial = "2"
      name = "FMSX0OV"
      region = "1"
      member_name = "R27Cecum"
    >
     <FILES>
        <FILE filename="R27Cecum.sff" filetype="sff" checksum_method="MD5"
checksum="7f7ba170dbc6a25409a5eb6d845da88f" />
      </FILES>
    </DATA_BLOCK>
```

Multiple Segments

The submitter may present different parts of the spot sequence in distinct files. The records must exist in both files and be in the same order. The **DATA_BLOCK. FILES.FILE.read_label** connects the file with the named read in a spot descriptor.

For a certain spot descriptor:

```
        <SPOT_DESCRIPTOR>
            <SPOT_DECODE_SPEC>
                <NUMBER_OF_READS_PER_SPOT>2</NUMBER_OF_READS_PER_SPOT>
                <READ_SPEC>
                    <READ_INDEX>1</READ_INDEX>
                    <READ_LABEL>forward</READ_LABEL>
                    <READ_CLASS>Application Read</READ_CLASS>
                    <READ_TYPE>Forward</READ_TYPE>
                    <BASE_COORD>1</BASE_COORD>
                </READ_SPEC>
                <READ_SPEC>
                    <READ_INDEX>2</READ_INDEX>
                    <READ_LABEL>reverse</READ_LABEL>
                    <READ_CLASS>Application Read</READ_CLASS>
                    <READ_TYPE>Reverse</READ_TYPE>
                    <BASE_COORD>37</BASE_COORD>
                </READ_SPEC>
            </SPOT_DECODE_SPEC>
        </SPOT_DESCRIPTOR>
```

can have the associated RUN code:

```
    <DATA_BLOCK name = "HWX170-FC8080_1000" sector="1">
      <FILES>
        <FILE filename="HWX170-FC8080_1000_1.fastq" filetype="fastq"
checksum_method="MD5" checksum="d41d8cd98f00b204e9800998ecf8427e"
read_label=”forward”/>
      </FILES>
        <FILE filename="HWX170-FC8080_1000_2.fastq" filetype="fastq"
checksum_method="MD5" checksum="204e9800998ecf8427ed41d8cd98f00b"
read_label=”reverse”/>
      </FILES>
    </DATA_BLOCK>
```

Multiple Data Series

A native format submission may consist of a single data block containing multiple data series (columns) each represented by a distinct file. The **DATA_BLOCK. FILES.FILE.data_series_label** can be used to define a precise mapping between components and columns.

```
    <DATA_BLOCK>
       <FILES>
         <FILE filename='Solid0044_20081126_2_F3.csfasta' filetype="SOLiD_native"
checksum_method="MD5" checksum="d41d8cd98f00b204e9800998ecf8427e"
data_series_label=”INSDC:csbases”/>
       </FILES>
    </DATA_BLOCK>

    <DATA_BLOCK>
       <FILES>
         <FILE filename='Solid0044_20081126_2_F3_QV.qual' filetype="SOLiD_native"
checksum_method="MD5" checksum="9800998ecf8427ed41d8cd98f00b204e"
data_series_label=”INSDC:phred”/>
       </FILES>
    </DATA_BLOCK>
```

Combining Segments and Data Series

The two parameters **DATA_BLOCK.FILES.FILE.read_label** and **DATA_BLOCK.FILES.FILE.data_series_label** can be combined into a two dimensional specification of files to segments and columns.

```
    <DATA_BLOCK>
       <FILES>
         <FILE filename='Solid0044_20081126_2_F3.csfasta' filetype="SOLiD_native"
checksum_method="MD5" checksum="d41d8cd98f00b204e9800998ecf8427e" read_label=”forward”
data_series_label=”INSDC:csbases”/>
       </FILES>
    </DATA_BLOCK>

    <DATA_BLOCK>
       <FILES>
         <FILE filename='Solid0044_20081126_2_F3_QV.qual' filetype="SOLiD_native"
checksum_method="MD5" checksum="9800998ecf8427ed41d8cd98f00b204e" read_label=”forward”
data_series_label=”INSDC:phred”/>
       </FILES>
    </DATA_BLOCK>

    <DATA_BLOCK>
       <FILES>
         <FILE filename='Solid0044_20081126_2_R3.csfasta' filetype="SOLiD_native"
checksum_method="MD5" checksum="4d1d8cd98f00b204e9800998ecf8427e" read_label=”reverse”
data_series_label=”INSDC:csbases”/>
       </FILES>
    </DATA_BLOCK>

    <DATA_BLOCK>
       <FILES>
         <FILE filename='Solid0044_20081126_2_R3_QV.qual' filetype="SOLiD_native"
checksum_method="MD5" checksum="8900998ecf8427ed41d8cd98f00b204e" read_label=”reverse”
data_series_label=”INSDC:phred”/>
       </FILES>
    </DATA_BLOCK>
```

Specifying fastq

Fastq forms are particularly problematic as their formats are unconstrained. To better support this kind of submission during a transition period certain DATA_BLOCK parameters can be used to reduce the ambiguity of the format.

The **DATA_BLOCK.FILES.FILE.quality_scoring_system** parameter can be used to specify whether the quality scores encountered in the fastq file are phred scale or log-odds scale. The SRA will convert log-odds into phred, but to do this properly the loader must know whether the log-odds scale is being used. For example:

```
<RUN alias="KN-930_1" >
  <EXPERIMENT_REF accession="SRX002983"></EXPERIMENT_REF>
    <DATA_BLOCK name="KN-930" sector="1">
      <FILES>
        <FILE filename="KN-930_1.fastq" filetype="fastq" quality_scoring_system="log-odds"/>
      </FILES>
    </DATA_BLOCK>
</RUN>
```