

Alignment Data Model

To describe alignments with additional information about the fragment and the read, we adopt the read alignment schema defined by the [Global Alliance for Genomics and Health](#) (GA4GH). A [ReadAlignment](#) object is equivalent to an alignment line in a [SAM](#) file. Note that we use a plain text file to store the header lines of a SAM file (i.e., lines with program and chromosomes information).

Data serialization systems such as Avro, Thrift and Google Protobuf can use the ReadAlignment schema and then read and write alignments in a compact, fast and binary data format. In addition, ReadAlignment objects can be stored in the Parquet format and then can be processed by a number of different systems: Spark, Hive, Impala and others.

Table of Contents:

- [Example](#)
- [More information and references](#)

Example

This section shows an alignment in SAM format and the equivalent ReadAlignment.

Example of alignment in SAM format:

```
1_229454865_229455276_0:0:0_0:0:0_0 83 1 229455177 60
100M = 229454865 -412
AGTGCTATTTGGATTTCATCCCATATGGGCCCATCTTGTGGTCTGAGGCCTGACAGGGCTCACCTGCAAGCTCGG
TTCTCTGCTGTCTTTGATATGGACT
????????????????????????????????????????????????????????????????????????
???????????????????????????? NM:i:0 AS:i:100 XS:i:0
```

ReadAlignment object for the previous alignment (JSON format):

```
{
  "id" : {
    "string" : "1_229454865_229455276_0:0:0_0:0:0_0"
  },
  "readGroupId" : "no-group",
  "fragmentName" : "1",
  "improperPlacement" : {
    "boolean" : true
  },
  "duplicateFragment" : {
    "boolean" : false
  },
  "numberReads" : {
    "int" : 2
  },
  "fragmentLength" : {
    "int" : -412
  },
  "readNumber" : {
    "int" : 0
  },
  "failedVendorQualityChecks" : {
    "boolean" : false
  },
  "alignment" : {
    "org.ga4gh.models.LinearAlignment" : {
      "position" : {
        "referenceName" : "1",
        "position" : 229455176,
        "strand" : "NEG_STRAND"
      },
      "mappingQuality" : {
        "int" : 60
      },
      "cigar" : [ {
        "operation" : "ALIGNMENT_MATCH",
        "operationLength" : 100,
        "referenceSequence" : null
      } ]
    }
  }
}
```

```

"secondaryAlignment" : {
  "boolean" : false
},
"supplementaryAlignment" : {
  "boolean" : false
},
"alignedSequence" : {
  "string" :
"AGTGCTATTTGGATTCCCATATGGGCCCCATCTTGTGGTCTGAGGCCTGACAGGGCTCACCTGCAAGCTCG
GTTCTCTGCTGTCTTTGATATGGACT"
},
"alignedQuality" : [ 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30,
30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30,
30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30,
30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30,
30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30 ],
"nextMatePosition" : {
  "org.ga4gh.models.Position" : {
    "referenceName" : "1",
    "position" : 229454865,
    "strand" : "POS_STRAND"
  }
},
"info" : {
  "AS" : [ "i", "100" ],
  "XS" : [ "i", "0" ],
  "NM" : [ "i", "0" ]
}
}

```

More information and references

- G4GH's schema documentation: <http://ga4gh-schemas.readthedocs.io/en/latest/schemas/reads.proto.html>
- Sequence Alignment/Map Format Specification: <https://samtools.github.io/hts-specs/SAMv1.pdf>
- Sequence Alignment/Map Optional Fields Specification: <https://samtools.github.io/hts-specs/SAMtags.pdf>