NCBI Assembly Archive
RFC v.1.1

Martin Shumway[1], Vladimir Alexeyev[2], Deanna Church (communicating author) (e-mail: trace@ncbi.nlm.nih.gov)[2], Steven Salzberg[1]
([1]The Institute for Genomic Research (TIGR), [2]Center for Biotechnology Information (NCBI))

1. Overview

The new Assembly Archive at NCBI is a repository of fully and partially complete genomic assemblies that exists in association with sequence submissions in Genbank and trace submissions in the NCBI Trace Archive. This is a major addition to the existing archives of trace data and sequence data. The repository provides users with the ability to access and evaluate the assemblies from which finished genomic nucleotide sequence has been derived. Many benefits accrue to users of this data, including for example the ability to determine that a spurious frame shift has occurred, or that a putative SNP is not well supported by adequate coverage.

This Specification describes the information content of a submission object, its format, and the procedure by which submissions and updates should be made. It does not describe retrieval techniques and viewing options. It does not specify the manner in which the data is stored at NCBI.

The repository has the ability to store two types of information, assembly and alignment. The set of instructions detailing how a set of traces contribute to an assembly is called labeled 'assembly'. Information concerning the alignments of traces (that may or may not have been used in the assembly) is labeled 'alignment'. The distinction is that an alignment of a pool of traces to an assembly may give different results than the instructions given when the same set of traces are used to define an assembly. During the course of an assembly, heuristics are generally applied in specific regions in the assembly to produce a more biologically relevant answer. The rest of the field names and their definitions remain the same.

Information fields are denoted as **<major division>.<field>**, for example **contig.submitter_reference**. Data types are left unspecified in this draft but in all cases are assumed to be strings.

The impetus for writing this specification came from a meeting between TIGR and NLM held on 22 Aug 2003 hosted by David Lipman and Jim Ostell of NLM.

1.1. Goals

- Provide a central repository of genome assemblies.
- Augment public repositories of experimental data that supports scientific results from

genome sequencing.
- Allow anyone to evaluate the quality of a genome assembly.
- Provide links to sequence data held in the Trace Archive.
- Allow for submission of alternate assemblies of the same sequences.
- Allow for submission of alternate base callings of the same assembly.

1.2. Scope

IS
- Submit and store contigs
- Refers to traces
- Submission and storage of contig consensus
- Refers to taxonomic objects
- Refers to traces
- Supports circularized contigs
- Allows for the construction of a multiple alignment for the traces in a contig

IS NOT
- Submission and storage of scaffolds
- Storage of trace data
- Submission and storage of contig features
- Description of sequencing project or organism
- Submission and storage of lab data including clone insert mapping
- Description of secondary structure such as hairpins and hard stops
- Stores edits that would have to be applied to raw traces in order to construct a multiple alignment

2. Assembly Submission

The core of each Assembly Submission is the ASSEMBLY.xml XML document format of which is described in details in this section.

The submission .xml file contains the following sections:

```
Assembly Block
     Contigs
     [
       Contig
       Traces
       [
         Trace
       ]*
     ]+
```

where the '*' denotes the section which can be repeated 0 or more times, and the '+' - for the section which can be repeated 1 or more times.

2.1. Assembly Block

This block gives general information about the entry. Usually a submission corresponds to a genome assembly of an organism or a structure within the organism's genome. The assembly is uniquely identified by its ID (AI) which is assigned on submission.

The submission block consists of the following fields and/or attributes:

Assembly fields/elements:
center_name – Submitter's institution designation (required)

taxid – Genbank taxonomic reference (required)
date – Date that the submission was prepared (optional)
description – Freeform description of the assembly or the submission (required)
structure – Submitter's structural assignment, for example chromosome 3. For some genomes these designations may have been standardized (required)
ncontigs – Number of contigs in the assembly (required)
nconbases – Number of bases of consensus in the contigs of the assembly(optional for validation purposes) (required)
nbasecalls – Total number of base calls used in the assembly (required)
ntraces – Number of traces referred to in the assembly (required)
coverage – Ratio of (optional)
contig – The submission contains one or more contigs (required)
Assembly attributes:
submitter_reference – Submitter's free text reference attribute, submitter's internal reference id (optional)
type – Attribute of the submission type (NEW, UPDATE, REPLACE or REMOVE) (required)
ai – Assembly archive identifier (required if type == UPDATE or REPLACE or REMOVE)

2.2. Contig Set

The submission contains one or more contigs. This is the *contig set*.

2.2.1. Contig

The contig record is uniquely identified by its ID (CI) which is assigned on submission. The **contig.submitter_reference** attribute's value allows the contigs to be identified with a designator meaningful to the submitting institution.

The contig is composed of a *consensus sequence* **(contig.consensus)** of **contig.nconbases** base pairs and a list of gaps (**contig.congaps**) that must be inserted in order to align the consensus with its constituent tiling. The contig's consensus can also be accompanied by a sequence of quality scores, (**contig.conqualities**) which are generated during the assembly.

The consensus is a sequence of nucleotide base calls that can comprise any IUB code (A,C,G,T,M,R, #). In addition further codes that denote ambiguities with respect to gaps are supported:

```
    a = A or gap
    c = C or gap
    t = T or gap
    g = G or gap
```

At NCBI side these IUB extensions will be substituted with the appropriate standard IUB codes:

a with A, c with C, t with T, and g with G accordingly.

The contig consensus strand is not specified and its sense is always "forward".

The consensus also contains a set of gap that describes how to "stretch" the contig consensus in order to fit the tiling underneath. The *gapped consensus* so described is indexed from 1 and is inclusive (thus the trivial consensus of one base has a span of [1,1]). The gap set is denoted as a set of relative offsets that tells how many base calls to skip before inserting a gap. For example, the following ungapped consensus can be re-gapped as follows:

```
ACTTTCATGGTACTGATCCAGT  + (1,5,0,0,2)  =>  A-CTTTC---AT-GGTACTGATCCAGT
```

A gapped consensus may start with a gap run and may end with a gap run.

The sum of **contig.nconbases** and **|contig.ncongaps|** is the size of the gapped consensus, while **contig.nbasecalls** refers to the total number of basecalls within the valid ranges of the traces used in the assembly.

2.2.2. Alternate Consensus

Rather than supplying the consensus sequence as part of the Assembly Archive submission (either in-line or as a file), it is possible to simply refer to an entry in NCBI Genbank by accession or by GI. Set the **contig.source** value to ACCESSION.VERSION or GI depending on which reference is used. The **contig.congaps** field must still be set. Care must be taken to ensure that the assembly consensus indeed matches the Genbank sequence. In addition, it is important that the submitter maintain the consistency between the assembly and the Genbank entry if either is changed over time.

A major feature of the Assembly Archive Viewer is the ability to project the annotation of genomic information from Genbank onto the assembly contigs.

2.2.3. Contig Conformation

It is possible for a contig to represent a circularized structure.

The **contig.conformation** attribute denotes whether the contig is linear (default) or circular. In the case of circularity, the consensus should not wrap on itself. Rather, traces that begin at the end of the contig and wraparound to the beginning are denoted with their **trace.tiling_start** and **trace.consensus.start** having negative values equal to the number of bases (gapped/ungapped) by which they wraparound. For example, if a trace wraps around the end of a contig by two bases, then **trace.consensus_start** is -1. This scheme maintains the invariant that the end minus the start equals the length (gapped/ungapped). The trace ordering scheme defined above is also maintained.

2.2.4. Pseudo-molecules

It is possible for a contig to represent a pseudo-molecule. Such submissions may contain more information about the relationship of contigs to one another, and more faithfully reflect the associated Genbank sequence.

The preparation of such a contig is described as follows. The actual contigs in the original assembly are ordered and oriented. Gaps are filled in with the approximate number of Ns representing the each gap's size. Negative gaps (overlaps) between adjacent contigs are resolved by unifying their alignments. A single contig is produced that spans the pseudo-molecule and aligns exactly with it. The contig may have areas of zero trace coverage. This contig thus constitutes the assembly submission.

2.2.5. Description

In many cases it may be useful to provide descriptive information for a contig other than just the identifier. Such optional descriptive information can be added here.

2.2.6. Scaffolds

Many assemblies produce a rich set of relationships among contig and markers that together represent a scaffolding of a genomic structure (e.g. chromosome arm). The

Assembly Archive Specification does not presently address support for scaffolds, but will do so in a future revision, and will probably be based on the existing Genbank AGP format.

2.3. Trace Set

A consensus is composed of zero or more traces called a trace set. In the case where no traces are available, **contig.ntraces** is 0 or this entry can be omitted.

Traces are references to entries in the NCBI Trace Archive and must have either the **trace.ti** or the **trace.trace_name** attribute set. In later case the **trace.center_name** optional attribute maybe set, otherwise the center name assumed to be same as the **assembly.cetner_name**. Once a Trace Archive entry is inserted the entry never expires or changes, so its identifier uniquely identifies the entry. While the presence of the **trace.ti** is guaranteed, it may be marked as replaced or withdrawn. The Assembly Archive does not take responsibility for assessing whether a contig record is still valid given the state of its referenced traces. No rule is proposed to ensure the consistency of contig records with respect to its constituent traces. For example, if one trace is invalidated in the Trace Archive and it is used in an area of deep coverage in the Assembly Archive record, then the contig might still be considered valid and its result is not challenged.

The trace record is designed to be lightweight, referring to all content as stored in the trace record. The number of bases in the trace that are used in the assembly is **trace.nbasecalls**. The *valid range* **trace.valid's_start** and **stop** attributes denote the start and stop coordinates (1-based, inclusive) of the unclipped trace sequence as stored in the Trace record. In other words, **[trace.valid.start, trace.valid.stop]** describes the closed interval of the sequence that may be used in assembly. Because the trace record is considered "forward", **trace.valid.start** < **trace.valid.stop**. The value of **trace.nbasecalls** is **trace.valid.stop** - **trace.valid.start** + 1. The **trace.nbasecalls** parameter is optional. The trace *valid range* does not have to be contained within the trace's *clear range* (the region screened of vector and clear of bad quality, as stored in the Trace Archive). This is done in order to permit alternative vector clipping and low quality trimming, for example when an assembly is performed on traces submitted by another center.

The *tiling range* of **trace.tiling**'s start and stop attributes (1-based inclusive) denote the gapped coordinates that locate the trace within the contig's tiling. The **trace.tiling** direction attribute denotes the trace's orientation within the tiling with respect to the trace record. This attribute can be consulted for ease of validating mate pair orientations. Note that this is not the same as the direction of the trace with respect to its clone (a trace attribute). Unlike the contig case, a gap may not start a trace sequence nor end it.

The **trace.consensus's start** and **stop** attributes denote the *consensus range* of the trace in the ungapped consensus of the first and last base of the trace contributing to the consensus. In other words, the **trace.consensus.start**-th base position (from 1) is the first position covered by the trace. This is useful for quickly determining coverage depth of the contig. These attributes are optional and help to locate the trace with respect to the consensus of the genome rather than the tiling.

Traces are listed in order of their occurrence in the contig's tiling. The first order is by **trace.tiling.start**, then by length of the tiling range (shortest first), else by **trace.ti**, else by **trace.trace_name**.

The following invariants are observed:

- The number of bases in the valid range is equal to the number of base calls in the tiling range.
- The consensus range is contained within the tiling range.
- The start coordinate is always <= the stop coordinate in all ranges.
- The trace valid range does not exceed the length of the trace.

Each trace also comes with a gap list. The **trace.tracegaps** tells where to put the gaps into the trace as it is being laid down in the tiling. To do this the trace must first be reverse complemented in the case where its **trace.direction** is reverse before rendering the gaps. This is done for ease of computation. The gap list is also 1-based inclusive.

3. Caveats

3.1. Validation of Data

To make sure that the submitted data is consistent, the data is validated during the loading process, and can be rejected if one of the validation points fails.

Below are the validation steps:

a. Referenced trace is invalid if:

- it does not exist in the trace archive;
- its original length is lesser than **valid.stop-valid.start**+1;
- its original length is lesser than that specified in the alignment;
- its tiling coordinates are not within the consensus boundaries;
- its gaps are not consistent with the gaps in the consensus;
- it is misaligned, i.e. the number of differences between the trace and the consensus exceeds the allowed threshold;

b. If a particular trace is determined to be invalid and the contig's coverage in its range is insufficient then the failure of a single trace will fail the whole consensus. NCBI will compute the validity statistic and reserves the right to reject submissions based on the failure to meet the statistic's threshold value.

3.2. Constructing a Multiple Alignment

The Assembly Archive Specification contains enough information to reconstruct the multiple alignment of traces in a contig. However, if contig traces referenced in the Trace Archive include edits from their raw form, then the multiple alignment of the resulting contig may not reflect these edits when constructed from the trace records. To allow for such a construction, the submitter should upload the auxiliary BASE_FILE, QUAL_FILE, and PEAK_FILE files when making the associated Trace Submission so as to reflect the latest edited sequence version of the trace, if it differs from the raw entry.

3.3. Alignment Optimality

It is possible for a multiple alignment as represented by the contig record to be less than optimal. This standard does not require alignment optimality. Thus it is possible to have the following situation:

```
Consensus:     ACT---GACGTTACT
Trace1:        ACTT--GACGTTACT
Trace2:        ACT-T-GACGTTACT
Trace3:        ACT--TGACGTTACT
```

This could be considered an error in the submitting institution's assembly.

Also it is possible that a column is completely gapped including the consensus:

```
Consensus:   GA-GTTACT
Trace1:      GA-GTTACT
Trace2:      GA-GTTACT
Trace3:      GA-GTTACT
```

While non-optimal, this alignment is benign.

4. Assembly Loading Pipeline

This section describes format of Assembly Submission files and procedure of transferring and processing submissions in the NCBI Assembly Archive. The format of accepted Assembly files is described in the Assembly Submission section.

4.1. Format and contents of a submission

A submission is a single file archived with 'tar' or 'gtar' UNIX utility and possibly compressed with 'gzip' or 'bzip2' UNIX utilities. The submission file name must use only latin alphanumerical symbols and appropriate extension: 'tar', 'tar.gz', 'tgz', or 'tar.bz2'. Files included into the archive cannot be compressed. In the future other submission formats may be added.

The described TAR archive must not have leading '/' and the following two files must be on the top level of the archive hierarchy:

- ASSEMBLY.xml - assembly data file as described in the Assembly Submission section
- MD5 - listing each file in the submission along with MD5 checksum

The TAR archive may also contain accompanying data files and subdirectories spelled exactly how they are referred in the ASSEMBLY.xml file.

All other files are ignored.

4.2. Submitting procedure

4.2.1. NCBI FTP account

Assembly submissions can be accepted only by uploading to NCBI FTP server. To make arrangements for this please contact NCBI Trace team at trace@ncbi.nlm.nih.gov. Be ready to provide full center name, abbreviated center name, contact persons names and emails, as well as a test submission which complys the RFC.

An FTP account will be created on NCBI FTP server, login/password information emailed to the submitting center.

4.2.2. Data validation

Before uploading Assembly submissions to NCBI FTP server the data must be verified using either XML DTD or XML Schema. The XML Schema is more preferable as it is more comprehensive.

4.3. Tracking submission files

Once a submission file uploaded to FTP an unique Submission ID is assigned, and the status of the submission can be obtained using Submission Tracking System (http://www.ncbi.nlm.nih.gov/Traces/trackit.cgi?m=track&s=1).

Each Assembly Submission changes it's status in the following order:

- received
- processing
- loaded or updated

In some cases a submission could have the 'PENDING' state, which means it needs human intervention to resume processing.

4.4. Loading, assigning Assembly and Contig IDs

Upon successful loading of an assembly the following unique numerical identifiers will be assigned:

- Assembly ID (AI)
- Contig ID (CI)

These identifiers are very important -- these are public IDs which will never be re-used or expired. They will be used to update, delete and retrieve assembly data.

4.5. Reporting

Results of the Assembly loading can be always viewed on the Tracking Page (http://www.ncbi.nlm.nih.gov/Traces/trackit.cgi?m=track&s=1) and the Submission data on the Trace Assembly page (http://www.ncbi.nlm.nih.gov/Traces/assembly).

There also email report will be sent to the list of contact persons of a center.

4.6. Updating data in the Assembly Archive

The intent of the Assembly Archive is to serve as a repository of experimental data in support of scientific conclusions. Therefore it is expected that the Archive will be used for finished experiments whose conclusions have reached publication. Unfinished experimental data and provisional results should instead be submitted to the NCBI GSS and WGS divisions. This section reviews features of the Archive that support correction of existing submissions.

In all cases updates are only accepted if they are submitted by the original submitting center. Updates are only allowed on the latest version of an assembly.

4.6.1. Updating an Assembly's Metadata

The metadata of an assembly (**[assembly|alignment].[taxid, date, description, structure, submitter_reference]**) can be modified by setting the **[assembly|alignment].type** attribute to UPDATE. The **ai** field must reference an existing assembly record in the Archive.

4.6.2. Replacing an Assembly

An assembly submission can be replaced in its entirety by submitting a new assembly that references an existing submission by matching the **[assembly|alignment].ai** to an existing assembly record in the Assembly Archive, and setting the **[assembly|alignment].type** attribute to REPLACE. The existing assembly is not actually deleted but simply marked as superseded, so that the assembly can continue to be referenced in literature by the old **[assembly|alignment].ai** attribute. However, it is no longer displayed in the Archive.

4.6.3. Updating Contigs in an Assembly

An assembly submission can be updated on an individual contig basis using the above procedure and additionally supplying contig records to add, modify, or remove from the assembly.

To add a contig, set the **contig.type** attribute to NEW for the contig to add. To modify an existing contig in an assembly, set the **contig.type** attribute to REPLACE and supply the **contig.ci** value for the contig to update. To remove a contig from an assembly, set the **contig.type** attribute to REMOVE, set the **contig.ci** value for the contig to remove, and finally set the fields in the contig record to NULL. Contigs that are modified or removed from an assembly are also not actually deleted, but are marked as superseded so that they may continue to be referenced in literature. The **contig.ci** attribute is otherwise ignored.

Note that it is likely that fields in the **[assembly|alignment]** block that describe the sizing of the assembly will change as a result of any contig update operation. The REPLACE submission should provide new values that reflect new assembly attribute values following the update. These will be revalidated as part of the update procedure. NCBI may reject a submission if these values are incorrect following an update attempt.

4.6.4. Updating Traces

If a trace referenced in the Trace Archive has changed, it is assigned a new identifier (TI). Because of this, a trace update will not affect an existing Assembly Archive submission. However, to ensure that edits of a trace are propagated to an assembly record, it is necessary to update the assembly with a new assembly that references the new trace (changing the **trace.ti** field). This can be done with a contig update or by completely replacing the assembly entry.

5. Appendix A: Assembly XML DTD

An XML formatted file with the following proposed DTD. This DTD was prepared with ability to validate the submission's fields, and its values.

Note: Dates should be provided in the following format: "MM/DD/YYYY[hh:mm:ss]"

```
<!--DOCTYPE assembly [ -->
<!-- assembly is a sequence of elements -->
<!-- contig is a substructure with minimum one occurrence -->
<!-- optional elements are: coverage -->
<!ELEMENT assembly/alignment (center_name, taxid?, date?,
                    description, structure,
                    ncontigs, nconbases?, nbasecalls?, ntraces?,
                    coverage?, contig+)>
<!ATTLIST assembly/alignment
          submitter_reference CDATA #IMPLIED
          type (NEW | UPDATE | REPLACE | REMOVE) "NEW"
          ai ( CDATA )
>
<!ELEMENT center_name         ( #PCDATA )>
<!ELEMENT submitter_reference ( #PCDATA )>
<!ELEMENT taxid               ( #PCDATA )>
<!ELEMENT date                ( #PCDATA )>
<!ELEMENT description         ( #PCDATA )>
<!ELEMENT structure           ( #PCDATA )>
<!ELEMENT ncontigs            ( #PCDATA )>
<!ELEMENT nconbases           ( #PCDATA )>
<!ELEMENT nbasecalls          ( #PCDATA )>
<!ELEMENT ntraces             ( #PCDATA )>
<!ELEMENT coverage            ( #PCDATA )>
<!-- if contig does not have any gaps they can be omitted -->
<!-- ncongaps, congaps, conqualities and trace are optional elements
     of the contig structure -->
<!-- ngaps is always accompanied by gaps -->
<!ELEMENT contig
             (nconbases,nbasecalls,ntraces?,(ncongaps,congaps)?,
              consensus,conqualities?,trace*)>
<!-- contig's confirmation attribute conformation has default value
     LINEAR -->
<!ATTLIST contig
          submitter_reference CDATA #IMPLIED
          conformation        (LINEAR | CIRCULAR) "LINEAR"
          type (NEW | UPDATE | REMOVE) "NEW"
          ci ( CDATA )
>
<!-- ntraces and nbasecalls have been defined already -->
<!ELEMENT ncongaps        (#PCDATA ) >
<!ELEMENT congaps         (#PCDATA ) >
<!ELEMENT accession       ( #PCDATA )>
<!-- the actual gap values can be provided via external file or inline -->
<!--external file is the default attribute -->
<!ATTLIST congaps source (FILE | INLINE) "FILE">
<!-- the actual consensus values can be provided via external file or inline
-->
<!-- external file is the default attribute -->
<!ELEMENT consensus (#PCDATA ) >
<!ATTLIST consensus  source (FILE | INLINE | ACCESSION | GI) "FILE">
<!-- ntracegaps and tracegaps can be omitted if none are specified, or -->
<!-- while ntracegaps can be omitted if tracegaps is specified -->
<!ELEMENT trace (ti?, trace_name?, center_name?, nbasecalls, valid?, tiling,
traceconsensus,
                 (ntracegaps?, tracegaps)?)>
<!ELEMENT ti    (#PCDATA)>
<!ELEMENT trace_name (#PCDATA)>
<!ELEMENT ntracegaps (#PCDATA)>
<!ELEMENT tracegaps  (#PCDATA)>
<!ELEMENT valid      (start, stop)>
<!ELEMENT start      (#PCDATA)>
<!ELEMENT stop       (#PCDATA)>
<!ELEMENT tiling     (start, stop)>
<!ATTLIST tiling direction (FORWARD | REVERSE) "FORWARD">
<!ATTLIST tracegaps source (FILE | INLINE) "INLINE">
<!ELEMENT traceconsensus (start, stop) >
<!-- ] -->
```

6. Appendix B: Assembly XML Schema

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

 <xs:element name='assembly'>
  <xs:complexType>
   <xs:sequence>
    <xs:element ref='center_name'/>
    <xs:element ref='date' minOccurs='0' maxOccurs='1'/>
    <xs:element ref='taxid' minOccurs='0' maxOccurs='1'/>
    <xs:element ref='description'/>
    <xs:element ref='structure'/>
    <xs:element ref='ncontigs'/>
    <xs:element ref='nconbases' minOccurs='0' maxOccurs='1'/>
    <xs:element ref='ntraces' minOccurs='0' maxOccurs='1'/>
    <xs:element ref='nbasecalls' minOccurs='0' maxOccurs='1'/>
    <xs:element ref='coverage' minOccurs='0' maxOccurs='1'/>
    <xs:element ref='contig' maxOccurs='unbounded'/>
   </xs:sequence>
   <xs:attribute name='submitter_reference' type='xs:string' use='optional'/>

    <xs:attribute name="type" default="NEW">
        <xs:simpleType>
            <xs:restriction base='xs:string'>
                <xs:enumeration value='NEW'/>
                <xs:enumeration value='UPDATE'/>
                <xs:enumeration value='REPLACE'/>
                <xs:enumeration value='REMOVE'/>
            </xs:restriction>
        </xs:simpleType>
    </xs:attribute>

  </xs:complexType>
 </xs:element>

    <xs:element name="center_name">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:enumeration value="TIGR"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:element>

 <xs:element name='taxid' type="xs:integer"/>
 <xs:element name='date' type="xs:string"/>

 <xs:element name='description' type="xs:string"/>
 <xs:element name='structure' type="xs:string"/>

 <xs:element name='ncontigs' type="xs:integer"/>

 <xs:element name='nconbases' type="xs:integer"/>

 <xs:element name='nbasecalls' type="xs:integer"/>

 <xs:element name='ntraces' type="xs:integer"/>

 <xs:element name='coverage' type="xs:decimal"/>


 <xs:element name='contig'>
  <xs:complexType>
   <xs:sequence>
    <xs:element ref='ntraces' minOccurs='0' maxOccurs='1'/>
    <xs:element ref='nconbases'/>
                <xs:element ref='nbasecalls' minOccurs='0' maxOccurs='1'/>
                <xs:element ref='description' minOccurs='0' maxOccurs='1'/>
                <xs:sequence minOccurs='0' maxOccurs='1'>
     <xs:element ref='ncongaps' type="xs:integer"/>
     <xs:element ref='congaps' type="xs:integer"/>
    </xs:sequence>
    <xs:element ref='consensus'/>
    <xs:sequence minOccurs='0' maxOccurs='1'>
     <xs:element ref='conqualities' type="xs:integer"/>
    </xs:sequence>
    <xs:element ref='trace' minOccurs='0' maxOccurs='unbounded'/>
   </xs:sequence>
         <xs:attribute name='submitter_reference' type='xs:string'
use='optional'/>
         <xs:attribute name='tiling' type='xs:string' use='optional'/>

    <xs:attribute name="conformation" default="LINEAR">
        <xs:simpleType>
            <xs:restriction base='xs:string'>
                <xs:enumeration value='LINEAR'/>
                <xs:enumeration value='CIRCULAR'/>
            </xs:restriction>
        </xs:simpleType>
    </xs:attribute>

    <xs:attribute name="type" default="NEW">
        <xs:simpleType>
            <xs:restriction base='xs:string'>
                <xs:enumeration value='NEW'/>
                <xs:enumeration value='UPDATE'/>
                <xs:enumeration value='REPLACE'/>
                <xs:enumeration value='REMOVE'/>
            </xs:restriction>
        </xs:simpleType>
    </xs:attribute>

  </xs:complexType>
 </xs:element>

 <xs:element name='ncongaps' type="xs:integer"/>

 <xs:element name='congaps'>
  <xs:complexType>
   <xs:simpleContent>
      <xs:extension base="xs:string">
       <xs:attribute name="source" default="FILE">
        <xs:simpleType>
         <xs:restriction base='xs:string'>
          <xs:enumeration value='FILE'/>
          <xs:enumeration value='INLINE'/>
         </xs:restriction>
        </xs:simpleType>
       </xs:attribute>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
 </xs:element>

 <xs:element name='consensus'>
  <xs:complexType mixed='true'>
   <xs:simpleContent>
      <xs:extension base="xs:string">
       <xs:attribute name="source" default="FILE">
        <xs:simpleType>
         <xs:restriction base='xs:string'>
          <xs:enumeration value='FILE'/>
          <xs:enumeration value='INLINE'/>
          <xs:enumeration value='ACCESSION.VERSION'/>
          <xs:enumeration value='GI'/>
         </xs:restriction>
        </xs:simpleType>
       </xs:attribute>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
 </xs:element>

<xs:element name='conqualities'>
    <xs:complexType>
        <xs:simpleContent>
            <xs:extension base="xs:string">
                <xs:attribute name="source" default="FILE">
                    <xs:simpleType>
                        <xs:restriction base='xs:string'>
                            <xs:enumeration value='FILE'/>
                            <xs:enumeration value='INLINE'/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:attribute>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
</xs:element>


 <xs:element name='trace'>
  <xs:complexType>
   <xs:sequence>
    <xs:element ref='ti' minOccurs='0' maxOccurs='1'/>
    <xs:element ref='trace_name' minOccurs='0' maxOccurs='1'/>
    <xs:element ref='center_name' minOccurs='0' maxOccurs='1'/>
    <xs:element ref='nbasecalls'/>
    <xs:element ref='valid'/>
    <xs:element ref='tiling'/>
    <xs:element ref='traceconsensus'/>
    <xs:sequence minOccurs='0' maxOccurs='1'>
    <xs:element ref='ntracegaps' minOccurs='0' maxOccurs='1'/>
     <xs:element ref='tracegaps'/>
    </xs:sequence>
   </xs:sequence>
  </xs:complexType>
 </xs:element>

 <xs:element name='ti' type="xs:integer"/>

 <xs:element name='trace_name' type="xs:string"/>

 <xs:element name='ntracegaps' type="xs:integer"/>

 <xs:element name='tracegaps'>
  <xs:complexType>
   <xs:simpleContent>
      <xs:extension base="xs:string">
       <xs:attribute name="source" default="FILE">
        <xs:simpleType>
         <xs:restriction base='xs:string'>
          <xs:enumeration value='FILE'/>
          <xs:enumeration value='INLINE'/>
         </xs:restriction>
        </xs:simpleType>
       </xs:attribute>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
 </xs:element>

 <xs:element name='valid'>
  <xs:complexType>
   <xs:sequence>
    <xs:element ref='start'/>
    <xs:element ref='stop'/>
   </xs:sequence>
  </xs:complexType>
 </xs:element>
 <xs:element name='start' type="xs:integer"/>

 <xs:element name='stop' type="xs:integer"/>

 <xs:element name='tiling'>
  <xs:complexType>
   <xs:sequence>
    <xs:element ref='start'/>
    <xs:element ref='stop'/>
   </xs:sequence>
   <xs:attribute name='direction' default='FORWARD'>
    <xs:simpleType>
     <xs:restriction base='xs:string'>
      <xs:enumeration value='FORWARD'/>
      <xs:enumeration value='REVERSE'/>
     </xs:restriction>
    </xs:simpleType>
   </xs:attribute>
  </xs:complexType>
 </xs:element>

 <xs:element name='traceconsensus'>
  <xs:complexType>
   <xs:sequence>
    <xs:element ref='start'/>
    <xs:element ref='stop'/>
   </xs:sequence>
  </xs:complexType>
 </xs:element>
</xs:schema>
```