AIRR Formats WG

Syed Ahmad Chan, Brian Corrie, Chaim Schramm, Corey Watson, Duncan Ralph, Erick Matsen, Felix Breden, Jason Vander Heiden, Nishanth Marthandan, Susanna Marquez

Uri Laserson and Scott Christley, *co-chairs*

Goal

Define file formats that make it easy to analyze, share, and inter-operate with AIRR data

tl;dr

- Defined file format for VDJ assignment/rearrangement annotations.
- Defined machine-readable specification for MiAIRR schema objects and VDJ rearrangements.
  - Used in iReceptor API to integrate with VDJServer
- Setup GitHub repository with specification files, documentation, and software.
  - `github.com/airr-community/airr-standards`
- Python reference library to read/write/validate AIRR format files.
  - Tested with ChangeO and VDJServer.

General file format considerations

- Standard format that provides ease-of-use, non-programmer accessibility, machine-readable specification, and tool interoperability.

- Enables NIH FAIR (findable, accessible, interoperable, reusable) guidelines for a broad spectrum of tools, services and APIs.

- Design for the future to support Big Data and computationally intensive analysis pipelines.

Objects that need modeling

`study` (MiAIRR metadata)

`read` (fastq)

`germline` (IMGT)

`alignment` (read to germline)

`rearrangement` (e.g., heavy chain)

`clone`

Objects that need modeling

`study` (MiAIRR metadata)

~~`read` (fastq)~~

~~`germline` (IMGT)~~

`alignment` (read to germline)

`rearrangement` (e.g., heavy chain)

`clone`

Focus on analysis

AIRR Formats

Alignments

`rearrangement_id`
`sequence`
`segment` (e.g., `J`)
`call` (e.g., `IGHJ6*01`)
`score`
`cigar`
`...`

many-to-one →

Rearrangements

`rearrangement_id`
`sequence`
`v_call`
`d_call`
`j_call`
`junction_nt`
`...`
`(approx 70 fields)`

Tab-delimited text files

What is a dataset?

1. YAML/JSON metadata
2. Tab-delimited data

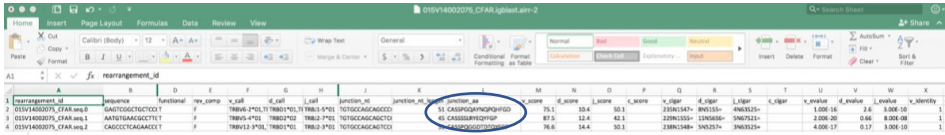
```
my_study/my_awesome_airr_study.yaml
my_study/my_awesome_airr_study.tsv
```

AIRR format example

	A	B	D	E	F	G	H	J	K	L
1	rearrangement_id	sequence	functional	rev_comp	v_call	d_call	j_call	junction_nt	junction_nt_length	junction_aa
2	015V14002075_CFAR.seq.0	GAGTCGGCTGCTCCC	T	F	TRBV6-2*01,TF	TRBD1*01,TF	TRBJ1-5*01	TGTGCCAGCAGCCC	51	CASSPGQAYNC
3	015V14002075_CFAR.seq.1	AATGTGAACGCCTTC	T	F	TRBV5-4*01	TRBD2*02	TRBJ2-7*01	TGTGCCAGCAGCTC	45	CASSSSLRYEQY
4	015V14002075_CFAR.seq.2	CAGCCCTCAGAACC	T	F	TRBV12-3*01,T	TRBD1*01	TRBJ2-3*01	TGTGCCAGCAGTCC	51	CASSPQGGDTI
5	015V14002075_CFAR.seq.3	TCGGCTGCTCCCTCC	T	F	TRBV6-1*01	TRBD1*01	TRBJ1-2*01	TGTGCCAGCAGTGA	54	CASSEDSIGGSY
6	015V14002075_CFAR.seq.4	GCACAGAGCGGGGC	F	F	TRBV7-3*01	TRBD1*01,TF	TRBJ2-7*01	TGTGCCAGCAGCTC	55	CASSS*PPGRFI
7	015V14002075_CFAR.seq.5	GCCTTGTTGCTGGGC	T	F	TRBV5-6*01	TRBD2*01	TRBJ1-2*01	TGTGCCAGCAGCTT	54	CASSFGRFGGK
8	015V14002075_CFAR.seq.6	AGGCTGCTGTCGGC	T	F	TRBV6-5*01	TRBD1*01	TRBJ1-2*01	TGTGCCAGCAGTTA	45	CASSYGSGNGV
9	015V14002075_CFAR.seq.7	ACCCTGGAGTCTGC	T	F	TRBV25-1*01	TRBD2*02	TRBJ1-4*01	TGTGCCAGCAGTGA	45	CASSDSPREKLF
10	015V14002075_CFAR.seq.8	GTGAGCAACATGAG	T	F	TRBV29-1*01,T	TRBD1*01	TRBJ2-7*01	TGCAGCGTGGTGAG	48	CSVVRTENAYE
11	015V14002075_CFAR.seq.9	GAGTCGGCTGCTCCC	T	F	TRBV6-1*01		TRBJ2-7*01	TGTGCCAGCAGTGA	51	CASSEYEKGTYI
12	015V14002075_CFAR.seq.10	CCTGCAGAACTGGA	T	F	TRBV14*01	TRBD1*01	TRBJ1-6*02	TGTGCCAGCAGCCA	54	CASSQETGKNN
13	015V14002075_CFAR.seq.11	CTGCTCCCTCCCAAA	F	F	TRBV6-2*01,TF	TRBD2*01	TRBJ2-3*01	TGTGCCAGCAGTTA	58	CASSYPPGLAG
14	015V14002075_CFAR.seq.12	CTGGAGTCCGCCAG	T	F	TRBV28*01	TRBD2*02	TRBJ2-1*01	TGTGCCAGCAGTTTT	48	CASSFTGGYNE
15	015V14002075_CFAR.seq.13	GAGTCGGCTGCTCCC	T	F	TRBV6-2*01,TF	TRBD2*02	TRBJ1-5*01	TGTGCCAGCAGTGA	51	CASSDGSSLNQ
16	015V14002075_CFAR.seq.14	ATCCAGCCTGCAAAC	T	F	TRBV11-2*01	TRBD1*01,TF	TRBJ1-2*01	TGTGCCAGCAGCAC	48	CASSTWGAGY
17	015V14002075_CFAR.seq.15	CCTGCAAAGCTTGAC	T	F	TRBV11-2*01	TRBD2*01	TRBJ2-7*01	TGTGCCAGCAGCTTC	54	CASSLTGLAGN
18	015V14002075_CFAR.seq.16	TCACCAGGCCTGGG	T	F	TRBV15*02	TRBD1*01	TRBJ1-5*01	TGTGCCACCGGAGA	54	CATGEGALKRN
19	015V14002075_CFAR.seq.17	CTACACACCCTGCAG	T	F	TRBV4-3*01	TRBD1*01,TF	TRBJ1-2*01	TGCGCCAGCAGCCC	48	CASSPDGAGYC
20	015V14002075_CFAR.seq.18	TGCAGCCAGAAGAC	F	F	TRBV4-1*01,TF	TRBD1*01	TRBJ2-5*01	GCTAGCGTGGGAGA	31	ASVGDPVLRAX

All AIRR rearrangement format mandatory fields

Name	Type	Mandatory	Description
rearrangement_id	string	mandatory	Read/sequence identifier; often identical to a read identifier, but not necessarily (especially where a rearrangement is derived from a multiple read consensus).
sequence	string	mandatory	Nucleotide sequence (e.g., the "read" sequence; revcomp'd if necessary)
sample_id	string	mandatory	The biological sample this read derives from (e.g., from BioSample database)
functional	boolean	mandatory	VDJ sequence is predicted to be functional
rev_comp	boolean	mandatory	Sequence is reverse complemented
v_call	string	mandatory	V allele assignment
d_call	string	mandatory	D allele assignment
j_call	string	mandatory	J allele assignment
c_call	string	mandatory	C gene assignment (e.g., IGHG4, IGHA2, IGHE, TRBC)
junction_nt	string	mandatory	Nucleotide sequence of the junction region (CDR3 plus conserved residues; i.e., IMGT's JUNCTION)
junction_nt_length	integer	mandatory	Number of junction nucleotides in sequence_vdj
junction_aa	string	mandatory	Amino acid sequence of the junction region (CDR3 plus conserved residues; i.e., IMGT's JUNCTION)
junction_aa_length	integer		Number of junction amino acids in sequence_vdj
v_score	number	mandatory	V alignment score
d_score	number	mandatory	D alignment score
j_score	number	mandatory	J alignment score
c_score	number	mandatory	C alignment score
v_cigar	string	mandatory	V alignment CIGAR string
d_cigar	string	mandatory	D alignment CIGAR string
j_cigar	string	mandatory	J alignment CIGAR string
c_cigar	string	mandatory	C alignment CIGAR string



IMGT: “AA Junction” →

junction_aa
CASSPGQAYNQPQHFGD
CASSSSLRYEQYFGP
CASSPQGGDTDTQYFGP

← MiXCR: “aaSeqCDR3”

← IgBlast: “CDR3”

AIRR: “junction_aa”

AIRR compliant Repository/Tool 1

AIRR: YAML/JASON/Tab-delimited via API

AIRR compliant Repository/Tool 2

Study/Sample Metadata
IMGT annotations
Change-O clones
PARTIS clonal lineage

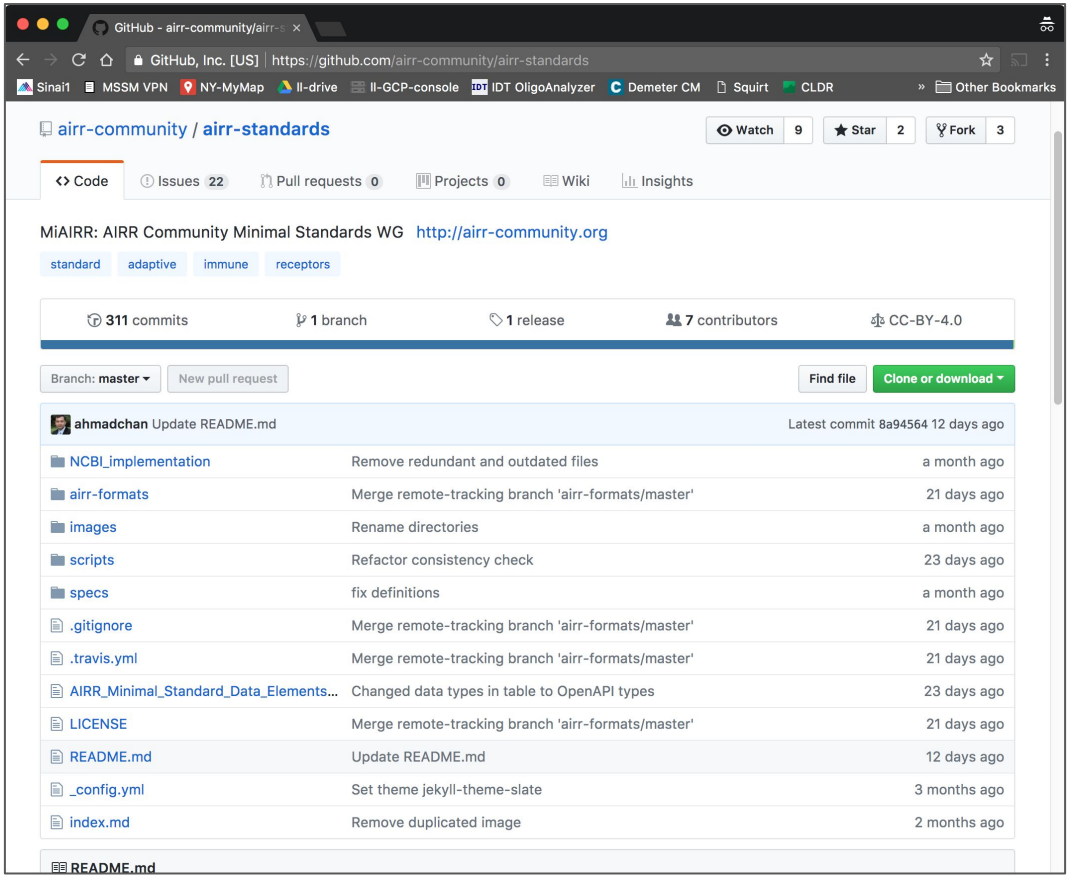
→ **AIRR rearrangements** ←

Study/Sample Metadata
IgBlast annotations
MiXCR clones
IgPhyML clonal lineage

AIRR clones format?

AIRR lineage format?

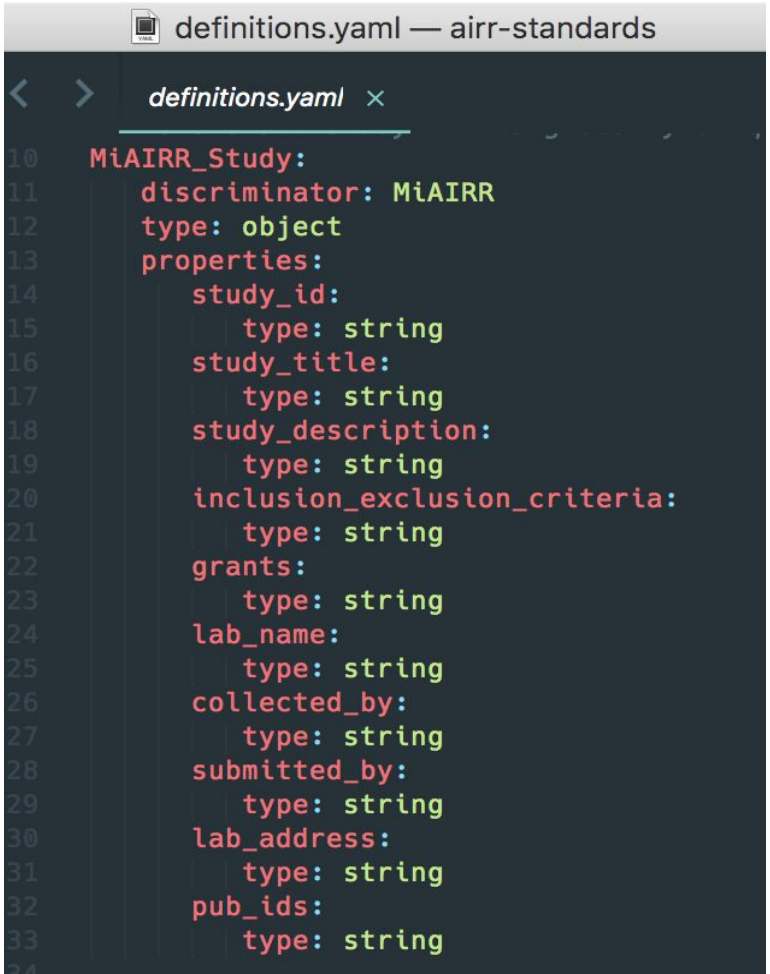
GitHub



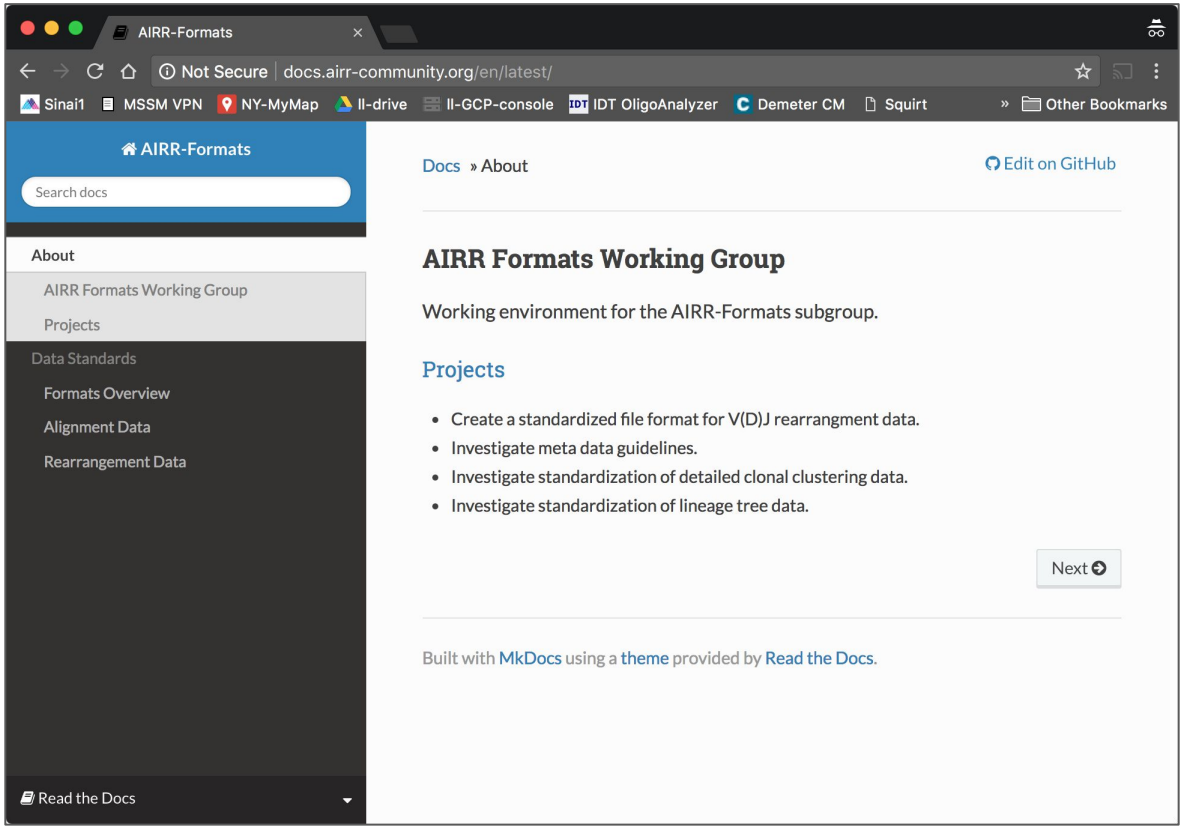
github.com/airr-community/airr-standards

Machine-readable specs

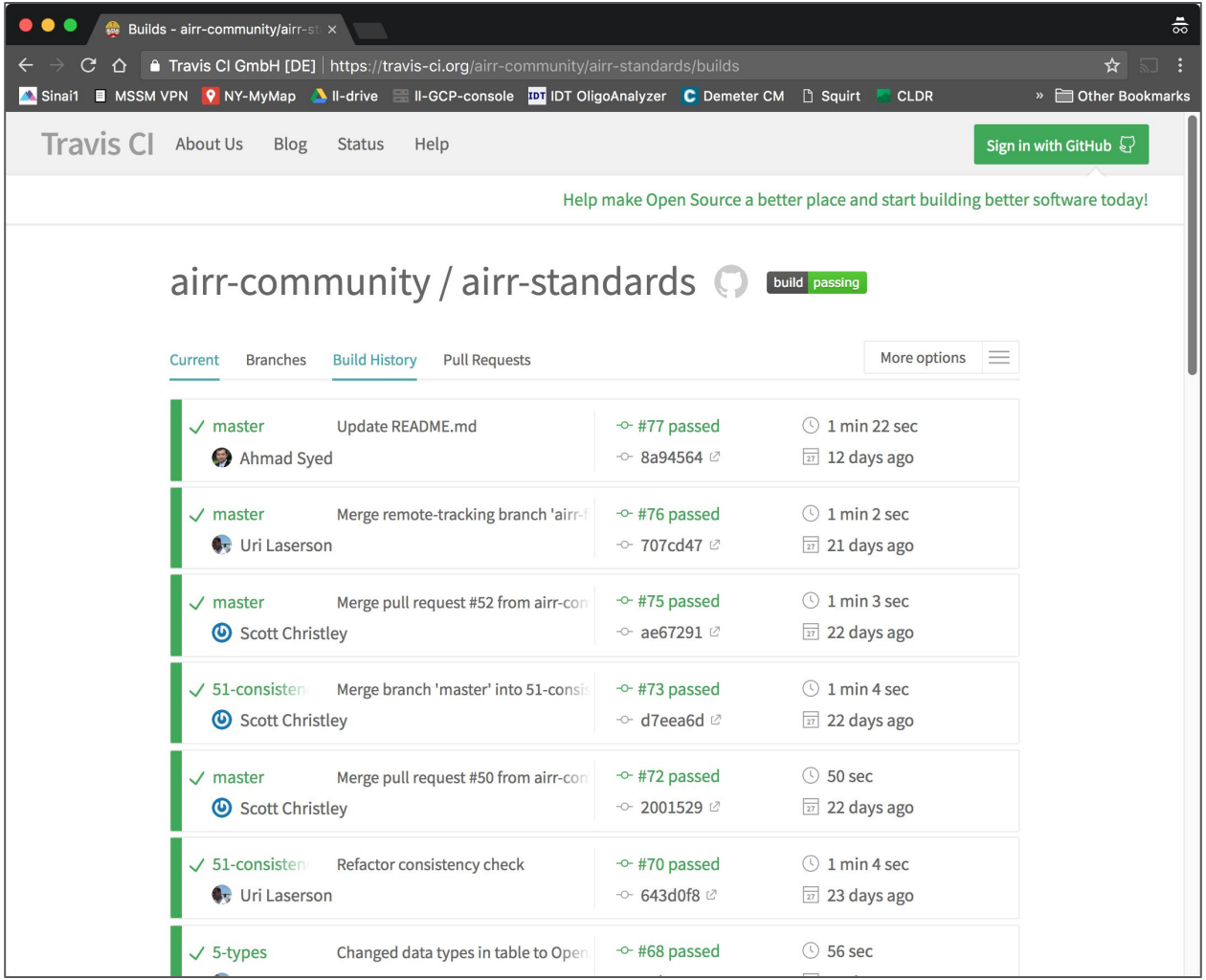
- Interoperability
- Continuous-integration
- Automation
- Minimizing ambiguity



Read the Docs



docs.airr-community.org


Builds - airr-community/airr-st
Travis CI GmbH [DE] | https://travis-ci.org/airr-community/airr-standards/builds
Sinai1
MSSM VPN
NY-MyMap
II-drive
II-GCP-console
IDT OligoAnalyzer
Demeter CM
Squirt
CLDR
Other Bookmarks
Travis CI
About Us
Blog
Status
Help
Sign in with GitHub
Help make Open Source a better place and start building better software today!
airr-community / airr-standards
build passing
Current
Branches
Build History
Pull Requests
More options
master Update README.md #77 passed 1 min 22 sec
Ahmad Syed 8a94564 12 days ago
master Merge remote-tracking branch 'airr- #76 passed 1 min 2 sec
Uri Laserson 707cd47 21 days ago
master Merge pull request #52 from airr-con #75 passed 1 min 3 sec
Scott Christley ae67291 22 days ago
51-consisten Merge branch 'master' into 51-consis #73 passed 1 min 4 sec
Scott Christley d7eea6d 22 days ago
master Merge pull request #50 from airr-con #72 passed 50 sec
Scott Christley 2001529 22 days ago
51-consisten Refactor consistency check #70 passed 1 min 4 sec
Uri Laserson 643d0f8 23 days ago
5-types Changed data types in table to Open #68 passed 56 sec

Goals for 2018

- Submit manuscript to publicize format
- Develop format for representing clones
- Finish integration of GitHub repository with MiAIRR
- Finish specifying metadata file format
- Public release of reference library to read/write/validate AIRR format files.
  - Initially targeting Python and R
- Releasing documentation incl. example output/use.

"Data Modeling and Representation WG"

- Multiple WGs are designing implementation standards and could use technical input on data representation.
- Coordination with AIRR Working Groups to specify data models, e.g.,
  - Common Repo defining minimal APIs for repositories and REST resources
  - MinStd choosing ontologies for their fields
  - Germline defining new germlines and annotations
- Ensure all AIRR groups are working in mutually compatible ways (in terms of data)
  - Ensure we have liaisons on all other relevant working groups
- Work on representation of provenance of data sets

Thanks to Formats WG!

New contributions and members are welcome!

docs.airr-community.org

General file format considerations

- Ease-of-use
  - Standard container files
    - Commonly available parsers
    - e.g., JSON, CSV, XML
  - Argue over schema, not representation
- Non-programmer accessibility
  - Tabular, non-nested data (Excel-compat)
  - Non-binary data
- Big data
  - Splittability
  - Operate on directories of files
- Metadata

AIRR format overview

- Format
  - Tab-delimited text for data
  - YAML/JSON for metadata
- Coordinates are 0-based with half-open intervals (like Python)
- Data types correspond to OpenAPI spec
- Columns are optional but use AIRR-spec'd names when relevant
  - Please suggest useful columns!
- CIGAR format for alignments