OAS_wrapper: A Python Package for Analyzing Observed Antibody Space Data

Viswanadham Sridhara, Ilya Mazo

Abstract

Observed Antibody Space (OAS) [1] database has a collection of over billion sequences from over 90 different studies. The immune repertories in OAS are annotated and tabulated for easy downstream processing. Here, we built a python package **OAS_wrapper** (https://pypi.org/project/OAS_wrapper) to parse, analyze, and visualize OAS data for improved annotation and reporting, including generating data for training antibody language models. By integrating key bioinformatics tools and reference databases, OAS_wrapper simplifies sequence annotation, comparison with germline, and includes other functionalities that provide metrics on different annotated regions (i.e., CDRs). The user-friendly output ensures the package can cater to researchers with varying level of computational expertise.

Introduction

Antibody sequences are central to immunology research, with applications spanning vaccine development, therapeutic antibody design, and immune repertoire profiling. Observed Antibody space (OAS) database is a repository that hosts both paired and unpaired antibody sequences, compiled from over 90 different studies. As of 21st Nov, 2024, there are over 2.4 billion sequences, most of which are unpaired data. At least 2 million sequences in OAS are paired (heavy/light) data coming from 12 different studies.

The studies in OAS are divided into data-units depending on specified parameters. More information is available at OAS database [2]. A typical user downloads a single or multiple data units. The downloaded table has metadata information (germline, isotype, B-cell source, disease/vaccine states and other fields) annotated by OAS. While most of the fields are useful, some fields lack supporting information (e.g., V, D and J gene sequences from IMGT [3]). In addition, a user has to look at multiple fields to confirm the annotation of critical regions such as CDRs. Here, we built a python package OAS_wrapper, to simplify the above mentioned tasks, with a goal to provide easy visualization, annotation, alignment (using Biopython [4] pairwise python module), along with some basic analysis of the OAS data. Some of the key functionalities of this package include, but not limited to:

1. Provide original IMGT sequences for V, D and J calls made in OAS data using IMGT reference database.

2. Align sequence and germline to highlight regions of mismatches, providing positional information.

3. Group data by germline, to infer sequences that originate from germline, including providing information on V, D and J annotations.

4.Annotate sequence with CDRs and FWRs for easy inference of regions of interest

Methods

The **OAS_wrapper** includes the following features to enhance antibody sequence data analysis:

Retrieval of Original IMGT Sequences

Using the IMGT reference database, **OAS_wrapper** retrieves the full-length V, D, and J germline sequences corresponding to the gene calls made in OAS data. This ensures that researchers can cross-reference their sequences with original IMGT reference sequence data.

Sequence-Germline Alignment with Positional Information

The package uses Biopython module to do a pairwise alignment of variable region of sequence to its germline counterpart, highlighting mismatches and providing indexes of mismatches. This feature enables researchers to pinpoint mutation hotspots and study their potential functional implications.

Grouping by Germline

The package has functionality that groups sequences by their inferred germline origin, facilitating population-level analyses. Each group includes essential details, such as V(D)J annotations. Such groupings are critical as similar antibody sequences generally tend to have similar function.

Annotation of Sequence Regions with CDRs

The package has functionality to annotate CDR and other important regions of the sequence. This functionality makes interpretation easy when combined with other functionalities within the package e.g., alignment mismatches occuring in a particular CDR is informative.

Additional functionality of OAS_wrapper

Additionally, the package offers interactive visualizations of sequence lengths, quality scores and frequency distributions of V(D)J calls, and tabulate other metrics typically useful for large-scale antibody sequence data analysis. The functions written in the OAS_wrapper are generic, and can be used for both unpaired and paired datasets. The **OAS_wrapper** relies on widely adopted Python libraries, including: 1. **Pandas**: For data manipulation and analysis 2. **Biopython**: For sequence alignment and handling biological data and 3. **Matplotlib**: For creating publication-quality visualizations

Results

The **OAS_wrapper** is tailored to immunologists, bioinformaticians, and data scientists engaged in antibody research. Since primarily, OAS database is divided into Unpaired and Paired data units, we provided tutorials for each of the paired and unpaired datasets. Here we briefly summarize some of

the findings analyzing a paired dataset from an individual, identified with SARS-COV-2 positive. The data is of PMBC cells, and specifically analyzing Naive B-cells (published dataset [5]). Link to original data unit file: https://opig.stats.ox.ac.uk/webapps/oas/dataunit_paired?unit=Jaffe_2022/csv/1287203_1_Paired_All.csv.gz

The metadata from the input dataset (`1287203_1_Paired_All.csv.gz`) can be extracted using the `OAS_wrapper` library. This metadata provides high-level information about the dataset, while the rest of the file include critical columns required for downstream analysis. This particular file has 198 columns:

```
sequence_id_heavy | sequence_heavy  | locus_heavy     | stop_codon_heavy | vj_in_frame_heavy | v_frameshift_heavy
productive_heavy | rev_comp_heavy  | complete_vdj_heavy | v_call_heavy    | d_call_heavy    | j_call_heavy
sequence_alignment_heavy | germline_alignment_heavy | sequence_alignment_aa_heavy | germline_alignment_aa_heavy | v_al
d_alignment_start_heavy | d_alignment_end_heavy | j_alignment_start_heavy | j_alignment_end_heavy | v_sequence_alignme
v_germline_alignment_heavy | v_germline_alignment_aa_heavy | d_sequence_alignment_heavy | d_sequence_alignment_aa_heav
j_sequence_alignment_heavy | j_sequence_alignment_aa_heavy | j_germline_alignment_heavy | j_germline_alignment_aa_heav
cdr1_heavy      | cdr1_aa_heavy   | fwr2_heavy      | fwr2_aa_heavy   | cdr2_heavy      | cdr2_aa_heavy
fwr3_heavy      | fwr3_aa_heavy   | fwr4_heavy      | fwr4_aa_heavy   | cdr3_heavy      | cdr3_aa_heavy
junction_heavy  | junction_length_heavy | junction_aa_heavy | junction_aa_length_heavy | v_score_heavy   | d_score_hea
j_score_heavy   | v_cigar_heavy   | d_cigar_heavy   | j_cigar_heavy   | v_support_heavy | d_support_heavy
j_support_heavy | v_identity_heavy | d_identity_heavy | j_identity_heavy | v_sequence_start_heavy | v_sequence_end_hea
v_germline_start_heavy | v_germline_end_heavy | d_sequence_start_heavy | d_sequence_end_heavy | d_germline_start_heavy
j_sequence_start_heavy | j_sequence_end_heavy | j_germline_start_heavy | j_germline_end_heavy | fwr1_start_heavy | fwr
cdr1_start_heavy | cdr1_end_heavy  | fwr2_start_heavy | fwr2_end_heavy  | cdr2_start_heavy | cdr2_end_heavy
fwr3_start_heavy | fwr3_end_heavy  | fwr4_start_heavy | fwr4_end_heavy  | cdr3_start_heavy | cdr3_end_heavy
np1_heavy       | np1_length_heavy | np2_heavy       | np2_length_heavy | c_region_heavy  | Isotype_heavy
Redundancy_heavy | ANARCI_numbering_heavy | ANARCI_status_heavy | sequence_id_light | sequence_light  | locus_light
stop_codon_light | vj_in_frame_light | v_frameshift_light | productive_light | rev_comp_light  | complete_vdj_light
v_call_light    | d_call_light    | j_call_light    | sequence_alignment_light | germline_alignment_light | sequence_a
germline_alignment_aa_light | v_alignment_start_light | v_alignment_end_light | d_alignment_start_light | d_alignment_
j_alignment_end_light | v_sequence_alignment_light | v_sequence_alignment_aa_light | v_germline_alignment_light | v_ge
d_sequence_alignment_aa_light | d_germline_alignment_light | d_germline_alignment_aa_light | j_sequence_alignment_ligh
j_germline_alignment_aa_light | fwr1_light      | fwr1_aa_light   | cdr1_light      | cdr1_aa_light   | fwr2_light
...
j_germline_end_light | fwr1_start_light | fwr1_end_light  | cdr1_start_light | cdr1_end_light  | fwr2_start_light
fwr2_end_light  | cdr2_start_light | cdr2_end_light  | fwr3_start_light | fwr3_end_light  | fwr4_start_light
fwr4_end_light  | cdr3_start_light | cdr3_end_light  | np1_light       | np1_length_light | np2_light
np2_length_light | c_region_light  | Isotype_light   | Redundancy_light | ANARCI_numbering_light | ANARCI_status_light
```

Figure 1: Columns present for a paired data unit file obtained from the OAS database.

A user can pick columns of interest and examine basic metrics/plots (e.g., lengths and distributions). The user can input any variable that is a string (e.g., sequence/germline or there subunits) and plot the metrics. The variables picked in the Figure 2 is just an example on few columns.

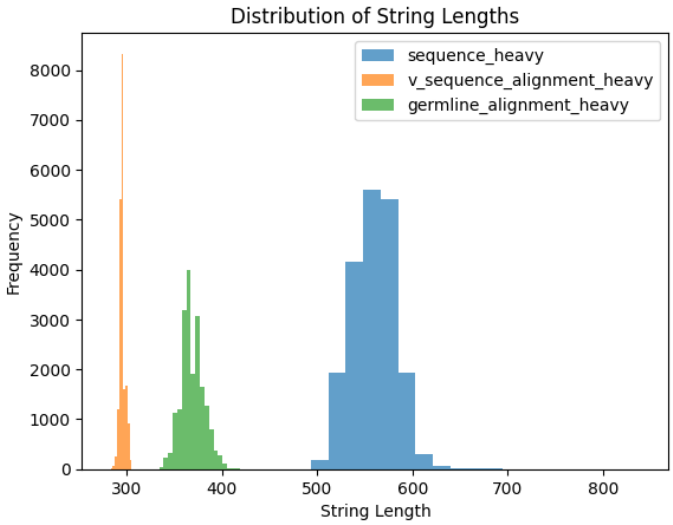

Figure 2: Distributions of sequence, germline and variable region of the sequence.

If a sequence is associated with a germline, the user is generally interested to understand how the sequence aligns with the germline, to understand the mismatches. We provided a functionality to identify these mismatches, including providing positional information.

```
Original Sequence 1: GAGGTGCAGCTGTTGGAGTCTGGGGGAGGCTTAGTACAGCCTGGGGGGTC
Original Sequence 2: GAGGTGCAGCTGTTGGAGTCTGGGGGAGGCTTGGTACAGCCTGGGGGGTC
Aligned and Highlighted Differences:
GAGGTGCAGCTGTTGGAGTCTGGGGGAGGCTTAGTACAGCCTGGGGGGTCCCTGAGACTCTCCTGTGTAGC
GAGGTGCAGCTGTTGGAGTCTGGGGGAGGCTTGGTACAGCCTGGGGGGTCCCTGAGACTCTCCTGTGCAGC
Indices of Differences: [32, 67, 148, 157, 166, 294, 295, 296, 297, 298
```

Figure 3: Mismatches are provided in red (e.g., A/G, T/C at positions 32 and 67 respectively)

A functionality to group sequences by germline is provided to identify hotspot regions. For example, in the above dataset, we see 16 sequences being identified with a particular germline.

germline_alignment_heavy	CAGGTGCAGCTGCAGGAGTCGGGCCCAGGACTGGTGAAGCCTTCAC...
number_of_sequences	16
sequence_heavy	GGGAGGGTCCTGCTCACATGGGAAATACTTTCTGAGAGTCCTGGAC...
v_call_heavy	IGHV4-31*03
j_call_heavy	IGHJ4*02
d_call_heavy	IGHD3-10*01

Figure 4: Sequence information, along with V, D and J identifiers provided for each of the germlines

OAS database only has information on the V, D and J identifiers and the user has to download the IMGT references database, to identify the original V-, D- and J- sequences. So, we built a functionality to map these sequences using the information from OAS and IMGT references.

Feature	Value
sequence_heavy	AGCTCTGAGAGAGGAGCCCAGCCCTGGGATTTTCAGGTGTTTTCAT...
v_call	IGHV3-23*01
v_sequence	gaggtgcagctgttggagtctggggga...ggcttggtacagcctg...
d_call	IGHD3-3*01
d_sequence	gtattacgatttttggagtggttattatacc
j_call	IGHJ3*02
j_sequence	tgatgcttttgatatctggggccaagggacaatggtcaccgtctct...
sequence_alignment_heavy	GAGGTGCAGCTGTTGGAGTCTGGGGGAGGCTTAGTACAGCCTGGGG...
germline_alignment_heavy	GAGGTGCAGCTGTTGGAGTCTGGGGGAGGCTTGGTACAGCCTGGGG...

Figure 5: Mapping IMGT sequences with identifiers and tabulating the data with sequence/alignment

We also provided functionality to annotated the query sequence with functional regions (e.g., CDRs and FWRs).

```
AGCCTGGGGGGTCCCTGAGA(cdr1 170-193)CTCTCCTGTGTAGCCTCTGGATTCACCTTTAGCAGCTATGCCATGAGCTGGGTCCGCCAG(cdr2 245-253)GCT
CAGAGACAATTCCAAGAACACGCTGTATCTGCAAATGAACAGCCTG(cdr3 362-394)AGAGCCGAGGACACGGCCGTATATTACTGTGCGAAAACCCCCAAATACGATGTTT
```

Figure 6: CDR1, CDR2 and CDR3 are annotated on the query sequence

Key Outcomes

- **Integration with IMGT Database**: Seamless extraction and alignment of V, D, and J germline sequences provided valuable insights into sequence-germline relationships.
- **Enhanced Data Understanding**: Summary statistics and visualizations allowed better comprehension of sequence distributions and quality.
- **Region-Specific Insights**: Annotation of CDR and FWR regions facilitated detailed analysis of functionally significant regions in antibody sequences.
- **Efficient Filtering**: Identification of germlines with the highest sequence mappings enabled prioritization of relevant sequences for downstream analysis.

The `OAS_wrapper` scripts proved robust and efficient for large-scale immunogenomics datasets, enabling comprehensive sequence characterization and annotation with minimal manual intervention.

Conclusion

We developed OAS_wrapper, a python package that parses and analyzes the antibody sequence data in OAS database, and provides information that allows researchers to extract maximum value from OAS data unit files. This open source solution promotes transparency and reproducibility, allowing the users to include the functionality described here within their own bioinformatics pipelines.

Code Availability

OAS_wrapper is an open-source Python package under an MIT license. Source code, documentation, and installation instructions can be downloaded from https://github.com/sridhara-omics/OAS_wrapper and https://pypi.org/project/OAS_wrapper. The package can run on any standard desktop computer or computing cluster.

1. Olsen, T. H., Boyles, F. & Deane, C. M. Observed Antibody Space: A diverse database of cleaned, annotated, and translated unpaired and paired antibody sequences. *Protein Science* **31**, 141–146 (2022).
2. Observed Antibody Space: https://opig.stats.ox.ac.uk/webapps/oas/documentation.
3. Manso, T. *et al.* IMGT® databases, related tools and web resources through three main axes of research and development. *Nucleic Acids Research* **50**, D1262–D1272 (2022).
4. Cock, P. J. A. *et al.* Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics* **25**, 1422–1423 (2009).
5. Jaffe, D. B. *et al.* Functional antibodies exhibit light chain coherence. *Nature* **611**, 352–357 (2022).