DOI 10.5281/zenodo.

Bader2025

This is the public code and data repository for Bader et al. 2025.

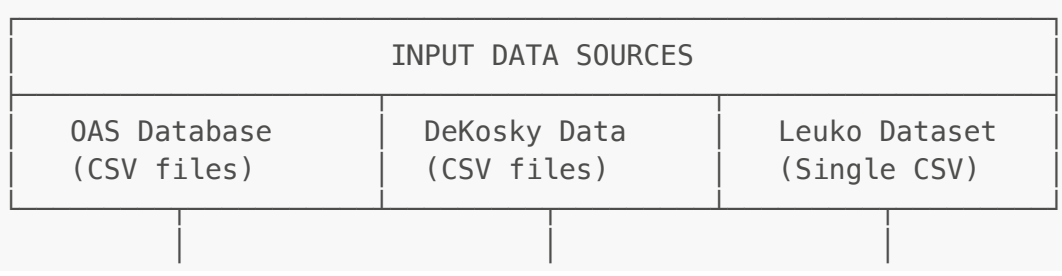
Abstract

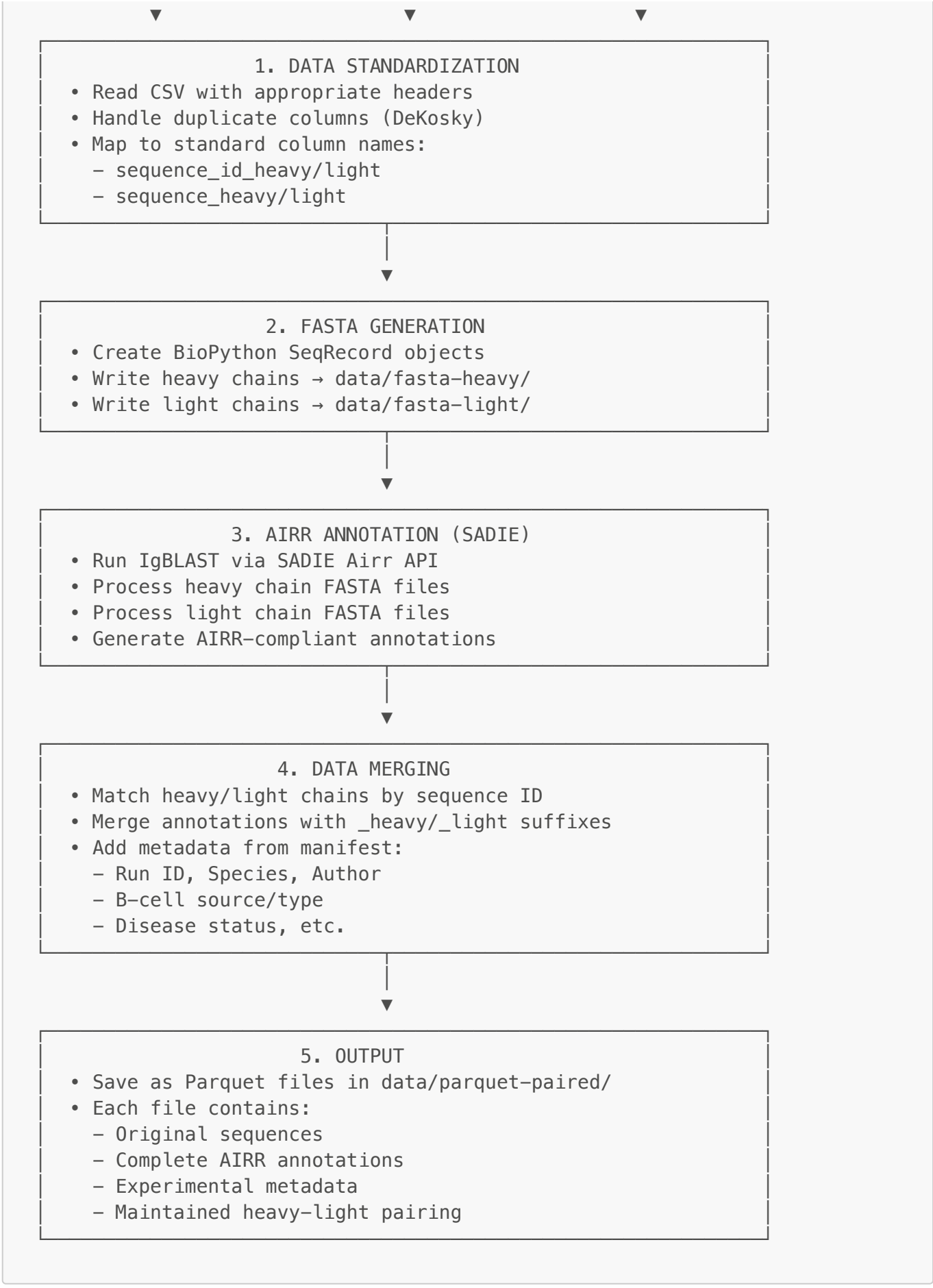
This pipeline processes paired heavy and light chain antibody sequences from the Observed Antibody Space (OAS) database, DeKosky and Leuko datasets. The sequences are annotated using SADIE's IgBLAST integration and saved as Parquet files with complete AIRR-compliant annotations and metadata for downstream analysis.

Directory Structure

```
data/
├── OAS_paired/            # OAS paired sequence CSV files
│   ├── ERR4082227_paired.csv
│   ├── ERR4082235_paired.csv
│   └── ...
├── DeKosky_paired/        # DeKosky dataset CSV files
│   ├── SRR1585248_joined_NoAlleles.csv
│   ├── SRR1585265_joined_NoAlleles.csv
│   └── ...
├── D326651_Leuko_human_naive.csv  # Leuko dataset
├── oas_manifest.csv       # OAS metadata manifest
├── fasta-heavy/           # Generated heavy chain FASTA files
│   ├── ERR4082227.fasta
│   └── ...
├── fasta-light/           # Generated light chain FASTA files
│   ├── ERR4082227.fasta
│   └── ...
├── parquet-paired/        # Final output Parquet files
│   ├── ERR4082227.parquet
│   ├── SRR1585248.parquet
│   ├── SRR1585265.parquet
│   └── ...
└── README.md
└── run-sadie.ipynb          # Main processing notebook
```

Processing Pipeline Diagram





Data Sources

1. OAS Database

- **Location**: `data/OAS_paired/`
- **Format**: CSV files with paired heavy/light sequences
- **Species**: Human antibody sequences
- **Manifest**: `data/oas_manifest.csv` contains metadata for each dataset

2. DeKosky Dataset

- **Location**: `data/DeKosky_paired/`
- **Format**: CSV files with custom column structure
- **Cell Type**: Naive B-cells from PBMC
- **Special Processing**: Requires column renaming due to duplicate headers

3. Leuko Dataset

- **Location**: `data/D326651_Leuko_human_naive.csv`
- **Cell Type**: Naive B-cells from PBMC
- **Author**: Jonathan Hurtado

Processing Pipeline

Step 1: Data Loading and Preparation

1. **Read CSV files** with appropriate headers (some files have JSON headers requiring special handling)
2. **Standardize column names**:
    - Map to `sequence_id_heavy`, `sequence_id_light`, `sequence_heavy`, `sequence_light`
    - Handle duplicate column names in DeKosky data

Step 2: FASTA Generation

1. **Create FASTA files** for heavy and light chains separately:
    - Heavy chains saved to `data/fasta-heavy/`
    - Light chains saved to `data/fasta-light/`
2. **Use BioPython** to properly format sequences with IDs

Step 3: AIRR Annotation with SADIE

1. **Run IgBLAST** via SADIE's Airr API on each FASTA file
2. **Generate AIRR-compliant annotations** including:
    - V(D)J gene assignments
    - CDR3 sequences
    - Framework regions
    - Junction analysis

Step 4: Paired Data Merging

1. **Match heavy and light chains** using sequence IDs
2. **Merge annotations** with suffixes `_heavy` and `_light`
3. **Add metadata** from manifest:

    - Run ID
    - Species
    - B-cell source (PBMC)
    - B-cell type (Naive B-cells)
    - Author information
    - Disease status
    - Other experimental metadata

Step 5: Output Generation

1. **Save as Parquet files** in `data/parquet-paired/`
2. **File naming**: Uses run ID or dataset identifier
3. **Format**: Apache Parquet for efficient storage and querying

Output Structure

Each Parquet file contains:

- **Sequence data**: Original nucleotide sequences for heavy and light chains
- **AIRR annotations**: Complete IgBLAST results for both chains
- **Metadata**: Experimental and sample information
- **Pairing information**: Maintained heavy-light chain relationships

Technical Details

Dependencies

- pandas
- BioPython (Bio.Seq, Bio.SeqRecord, Bio.SeqIO)
- SADIE (for AIRR annotation via IgBLAST)

Performance

- Processing time varies by dataset size
- Example: SRR datasets process in ~20-30 seconds each
- DeKosky datasets: ~4.5 minutes for complete processing

Error Handling

- Checks for existing files to avoid overwriting
- Handles mixed data types in columns
- Manages memory by deleting dataframes after processing

Usage Example

To run the pipeline:

1. Ensure all dependencies are installed
2. Place raw data in appropriate directories
3. Run the notebook cells sequentially

4. Output will be generated in `data/parquet-paired/`

Example Processing Flow

```
# Process a single OAS file
filename = "ERR4082227"
df = pd.read_csv(f"data/OAS_paired/{filename}_paired.csv")

# Standardize columns
df['sequence_id_heavy'] = df['sequence_id_heavy'].astype(str)
df['sequence_id_light'] = df['sequence_id_light'].astype(str)

# Create FASTA and run SADIE
heavy_df = airr_api.run_fasta(f"data/fasta-heavy/{filename}.fasta")
light_df = airr_api.run_fasta(f"data/fasta-light/{filename}.fasta")

# Merge and save
paired_df = pd.merge(heavy_df, light_df, on='tmp_id', suffixes=('_heavy',
'_light'))
paired_df.to_parquet(f'data/parquet-paired/{filename}.parquet')
```

Output Example

Each Parquet file contains ~100+ columns including:

```
Heavy Chain Columns:
- sequence_id_heavy
- sequence_heavy
- v_call_heavy
- d_call_heavy
- j_call_heavy
- cdr3_aa_heavy
- junction_heavy
- ...

Light Chain Columns:
- sequence_id_light
- sequence_light
- v_call_light
- j_call_light
- cdr3_aa_light
- junction_light
- ...

Metadata:
- run
- species
- bsource
- btype
- author
```

```
  - disease
  - file_name
```

Manifest Generation

A tailored manifest (`data/oas_manifest_human_paired.csv`) is created containing:

- Only human paired sequences
- Unique author entries
- Sorted by run ID
- Ready for downstream analysis