

---

# hybkit

*Release v0.3.4*

**Daniel Stribling**

**Nov 28, 2023**

## DOCUMENTATION CONTENTS:

<b>1</b>	<b>hybkit Hyb File Specification</b>	<b>5</b>
1.1	Columns . . . . .	6
1.2	Flags . . . . .	6
1.3	Other Details . . . . .	7
1.4	Example . . . . .	7
<b>2</b>	<b>hybkit API</b>	<b>8</b>
2.1	hybkit (module) . . . . .	8
2.2	hybkit.type_finder . . . . .	34
2.3	hybkit.analysis . . . . .	40
2.4	hybkit.plot . . . . .	47
2.5	hybkit.settings . . . . .	51
2.6	hybkit.util . . . . .	56
2.7	hybkit.errors . . . . .	60
<b>3</b>	<b>hybkit Toolkit</b>	<b>61</b>
3.1	hyb_check . . . . .	61
3.2	hyb_filter . . . . .	67
3.3	hyb_eval . . . . .	77
3.4	hyb_analyze . . . . .	86
<b>4</b>	<b>Example Analyses</b>	<b>94</b>
4.1	Example Type-miRNA Analysis . . . . .	94
4.2	Example Target Analysis . . . . .	97
4.3	Example Grouped Target Analysis . . . . .	98
4.4	Example Fold Analysis . . . . .	99
<b>5</b>	<b>About</b>	<b>103</b>
5.1	Renne Lab . . . . .	103
5.2	Lead Developer . . . . .	103
5.3	Changelog . . . . .	103
<b>6</b>	<b>References</b>	<b>106</b>
	<b>Bibliography</b>	<b>107</b>
	<b>Python Module Index</b>	<b>108</b>
	<b>Index</b>	<b>109</b>

1 2 3 4 5 6 7 8 9

The hybkit python API, an extendable documented codebase for creation of custom analyses of hyb-format data.

Integrated analysis of predicted secondary structure (fold) information for the API and command-line utilities.

Example analyses for publicly available qCLASH hybrid sequence data implemented in each of the command-line scripts and hybkit Python API.

The hybkit toolkit includes several command-line utilities for manipulation of hyb-format data:

Utility	Description
hyb_check	Parse hyb (and fold) files and check for errors
hyb_eval	Evaluate hyb (and fold) records to identify / assign segment types and miRNAs using custom criteria
hyb_filter	Filter hyb (and fold) records to a specific custom subset
hyb_analyze	Perform an energy, type, miRNA, target, or fold analysis on hyb (and fold) files and plot results

These scripts are used on the command line with hyb (and associated "vienna" or "CT") files. For example, to filter a hyb and corresponding vienna file to contain only hybrids with a sequence identifier containing the string "kshv":

Example:

```
$ hyb_filter -i my_hyb_file.hyb -f my_hyb_file.vienna --filter any_seg_
↳ contains kshv
```

Further detail on the usage of each script is provided in the [hybkit Toolkit](#) (page 61) section of *this documentation* (page ??).

Hybkit provides a Python3 module with a documented API for interacting with records in hyb files and associated vienna or CT files. This capability was inspired by the [BioPython Project](#)<sup>15</sup>. The primary utility is provided by a class for hyb records (HybRecord), a class for fold records (FoldRecord), and file-iterator classes (HybFile, ViennaFile, CTFile, HybFoldIter). Record attributes can be analyzed, set, and evaluated using included class methods.

For example, a workflow to print the identifiers of only sequences within a ".hyb" file that contain a miRNA can be performed as such:

```
#!/usr/bin/env python3
import hybkit
in_file = '/path/to/my_hyb_file.hyb'

# Open a hyb file as a HybFile Object:
with hybkit.HybFile.open(in_file, 'r') as hyb_file:

    # Return each line in a hyb file as a HybRecord object
    for hyb_record in hyb_file:
```

(continues on next page)

<sup>1</sup> <https://github.com/RenneLab/hybkit/releases>

<sup>2</sup> <https://pypi.org/project/hybkit/>

<sup>3</sup> <http://bioconda.github.io/recipes/hybkit/README.html>

<sup>4</sup> <http://bioconda.github.io/recipes/hybkit/README.html>

<sup>5</sup> <https://quay.io/repository/biocontainers/hybkit?tab=tags>

<sup>6</sup> <https://app.circleci.com/pipelines/github/RenneLab/hybkit>

<sup>7</sup> <https://hybkit.readthedocs.io/en/latest/?badge=latest>

<sup>8</sup> <https://coveralls.io/github/RenneLab/hybkit>

<sup>9</sup> <https://pypi.org/project/hybkit/>

<sup>15</sup> <https://biopython.org/>

(continued from previous page)

```
# Analyze each record to assign segment types
hyb_record.eval_types()

# If the record contains a long noncoding RNA type, print the record identifier.
if hyb_record.has_prop('any_seg_type_contains', 'lncRNA')
    print(hyb_record.id)
```

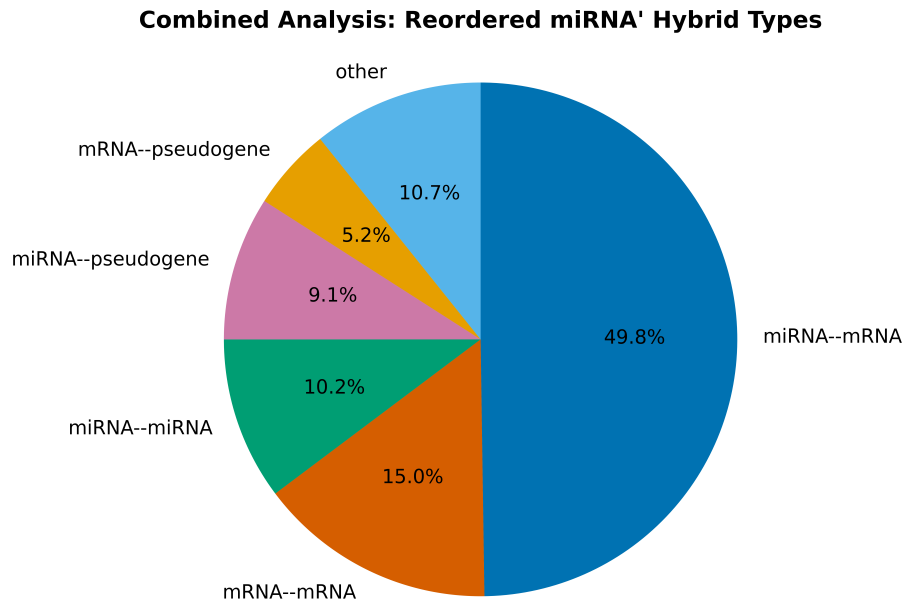
Further documentation on the hybkit API can be found in the *hybkit API* (page 8) section of *this documentation* (page ??).

Hybkit provides several example analyses for hyb data using the utilities provided in the toolkit. These include:

Analysis	Description
Type/miRNA Analysis	Quantify sequence types and miRNA types in a hyb file
Target Analysis	Analyze targets of a set of miRNAs from a single experimental replicate
Grouped Target Analysis	Analyze and plot targets of a set of miRNAs from pooled experimental replicates
Fold Analysis	Analyze and plot predicted miRNA folding patterns in miRNA-containing hybrids

These analyses provide analysis results in both tabular and graph form. As an illustration, the example summary analysis includes the return of the contained hybrid sequence types as both a csv table and as a pie chart:

CSV Output<sup>16</sup>



Further detail on each provided analysis can be found in the *Example Analyses* (page 94) section of *this documentation* (page ??).

<sup>16</sup> [https://raw.githubusercontent.com/RenneLab/hybkit/master/example\\_01\\_type\\_miRNA\\_analysis/example\\_output/combined\\_analysis\\_type\\_hybrid\\_types.csv](https://raw.githubusercontent.com/RenneLab/hybkit/master/example_01_type_miRNA_analysis/example_output/combined_analysis_type_hybrid_types.csv)

**Dependencies:**

- Python3.8+
- matplotlib<sup>17</sup> >= 3.7.1 (*Hunter JD. (Computing in Science & Engineering 2007)*<sup>18</sup>)
- BioPython<sup>19</sup> >= 1.79 (*Cock et al. (Bioinformatics 2009)*<sup>20</sup>)
- typing\_extensions <<https://pypi.org/project/typing-extensions/>> >= 4.8.0

**Via PyPI / Python PIP:**<sup>21</sup>

The recommended installation method is via hybkit's [PyPI Package Index](#)<sup>22</sup> using `python3 pip`<sup>23</sup>, which will automatically handle version control and dependency installation:

```
$ python3 -m pip install hybkit
```

**Via Conda:**<sup>24 25</sup>

For users of conda, the hybkit package and dependencies are hosted on the the [Bioconda](#)<sup>26</sup> channel, and can be installed using conda:

```
$ conda install -c bioconda hybkit
```

**Via Docker/Singularity:**<sup>27</sup>

The hybkit package is also available as a [Docker](#)<sup>28</sup> image and [Singularity](#)<sup>29</sup> container, hosted via the [BioContainers](#)<sup>30</sup> project on [quay.io](#)<sup>31</sup>. To pull the image via docker:

```
$ docker pull quay.io/biocontainers/hybkit:0.3.3--pyhdfd78af_0
```

To pull the image via singularity:

```
$ singularity pull docker://quay.io/biocontainers/hybkit:0.3.3--pyhdfd78af_0
```

**Manually Download and Install:**<sup>32</sup>

Use git to clone the project's Github repository:

```
$ git clone git://github.com/RenneLab/hybkit
```

OR download the zipped package:

<sup>17</sup> <https://matplotlib.org/><sup>18</sup> <https://doi.org/10.1109/MCSE.2007.55><sup>19</sup> <https://biopython.org/><sup>20</sup> <https://doi.org/10.1093/bioinformatics/btp163><sup>21</sup> <https://pypi.org/project/hybkit/><sup>22</sup> <https://pypi.org/project/hybkit/><sup>23</sup> <https://pip.pypa.io/en/stable/><sup>24</sup> <http://bioconda.github.io/recipes/hybkit/README.html><sup>25</sup> <http://bioconda.github.io/recipes/hybkit/README.html><sup>26</sup> <https://bioconda.github.io/><sup>27</sup> <https://quay.io/repository/biocontainers/hybkit?tab=tags><sup>28</sup> <https://www.docker.com/><sup>29</sup> <https://sylabs.io/singularity/><sup>30</sup> <https://biocontainers.pro/><sup>31</sup> <https://quay.io/repository/biocontainers/hybkit?tab=tags><sup>32</sup> <https://github.com/RenneLab/hybkit/releases>

```
$ curl -OL https://github.com/RenneLab/hybkit/archive/master.zip  
$ unzip master.zip
```

Then install using python setuptools:

```
$ python setup.py install
```

Further documentation on hybkit usage can be found in *this documentation* (page ??).

Hybkit provides a suite of unit tests to maintain stability of the API and script functionalities. To run the API test suite, install pytest and run the tests from the root directory of the hybkit package:

```
$ pip install pytest  
$ pytest
```

Command-line scripts can be tested by running the `auto_test.sh` script in the `auto_tests` directory:

```
$ ./auto_tests/auto_test.sh
```

hybkit is a free, sharable, open-source project.

All source code and executable scripts contained within this package are considered part of the "hybkit" project and are distributed without any warranty or implied warranty under the GNU General Public License v3.0 or any later version, described in the "LICENSE" file.

## HYBKIT HYB FILE SPECIFICATION

Version: v0.3.4

The ".hyb" (hyb file) format is described by [Travis2014] along with the Hyb software package as a "gff-related format that contains sequence identifiers, read sequences, 1-based mapping coordinates, and annotation information for each chimera".

Each line in a hyb file (a hyb "record") contains information about an RNA sequence read identified as a chimera by an RNA hybridization analysis. Each line contains 15 or 16 columns separated by tab characters ("\t") and provides information on each of the two aligned segments identified within the sequence read. The columns are described as follows by [Travis2014]:

Column 1, unique sequence identifier.

Column 2, read sequence [...].

Column 3, predicted binding energy in kcal/mol.

Columns 4–9, mapping information for first fragment of read: name of matched transcript, coordinates in read, coordinates in transcript, mapping score.

Columns 10–15, mapping information for second fragment of read.

Column 16 (optional, [...]), list of annotations in the format: "feature1=value1; feature2=value2;..."

The hybkit project uses an extended version of this description, including assigning columns reference names, and defining allowed flags.

## 1.1 Columns

#	Name	Description
1	id	Hybrid Read Identifier
2	seq	Read Nucleotide Sequence
3	energy	Predicted Gibbs Free-Energy of Intra-Hybrid Folding
4	seg1_ref_name	Segment 1 Mapping Reference Identity
5	seg1_read_start	Segment 1 Mapping Start on Read
6	seg1_read_end	Segment 1 Mapping End on Read
7	seg1_ref_start	Segment 1 Mapping Start on Reference
8	seg1_ref_end	Segment 1 Mapping End on Reference
9	seg1_score	Segment 1 Mapping Score
10	seg2_ref_name	Segment 2 Mapping Reference Identity
11	seg2_read_start	Segment 2 Mapping Start on Read
12	seg2_read_end	Segment 2 Mapping End on Read
13	seg2_ref_start	Segment 2 Mapping Start on Reference
14	seg2_ref_end	Segment 2 Mapping End on Reference
15	seg2_score	Segment 2 Mapping Score
16	flags	Hybrid Read Analysis Details

## 1.2 Flags

### Hyb Flags:

The following four flags are used by the Hyb software package ([Travis2014]). The definitions provided describe how these flags are used in the hybkit package.

**count\_total** - Integer: Total represented hybrid records, if records have been combined.

**count\_last\_clustering** - Integer: Total represented hybrid records at last clustering.

**two\_way\_merged** - {"0" or "1"} Boolean representation of whether entries with mirrored 5' and 3' hybrids were merged if the record is a combined record.

**seq\_IDs\_in\_cluster** - String: Comma-separated list of all record IDs of hybrids merged into this hybrid entry.

### hybkit Flags:

The following flags are used by hybkit.

**read\_count** - Integer: Number of sequence reads represented by this record. If the record is combined, this represents the total read count for all merged entries.

**orient** - String: Orientation of strand. Options: "F" (Forward), "IF" (Inferred Forward), "R" (Reverse), "IR" (Inferred Reverse), "U" (Unknown), or "IC" (Inferred Conflicting).

**seg1\_type** - String: Assigned segment type of segment 1, ex: "miRNA" or "mRNA".

**seg2\_type** - String: Assigned segment type of segment 2, ex: "miRNA" or "mRNA".

**seg1\_det** - String: Arbitrary detail about segment 1.

**seg2\_det** - String: Arbitrary detail about segment 2.

**miRNA\_seg** - String: Indicates which (if any) segment mapping is a miRNA. Options are "N" (none), "5p" (seg1), "3p" (seg2), "B" (both), or "U" (unknown).



**target\_reg** - String: Assigned region of the miRNA target. Options are "5pUTR", "C" ([C]oding), "3pUTR", "NON" ([NON]coding), "N" ([N]one), or "U" ([U]nknown).

**ext** - Integer: "0" or "1", Boolean representation of whether record sequences were bioinformatically extended as is performed by the Hyb software package.

**dataset** - String: Label for sequence dataset id (eg. source file), when combining records from different datasets.

## 1.3 Other Details

Item	Role
"\t" (tab)	Column Delimiter
"."	Missing Data Placeholder (equivalent to None)
".hyb"	File Suffix
".hyb.gz"	gzipped File Suffix
Disallowed	Header Lines
Disallowed	In-file Comments

## 1.4 Example

An example .hyb format line (courtesy of [Gay2018]):

```
2407_718    ATCACATTGCCAGGGATTCCAATCCCAACAATGTGAAAACGGCTGTC    .    MIMAT00000078_
↪MirBase_miR-23a_microRNA    1    21    1    21    0.0027    ENSG00000188229_
↪ENST00000340384_TUBB2C_mRNA    23    49    1181    1207    1.2e-06
```

## HYBKIT API

The hybkit API provides a Python3 module with classes allowing parsing and manipulation of hyb-format data as python objects, including built-in analysis and plotting functionality for common tasks in hybrid sequence analysis.

<a href="#">hybkit</a> (page 8)	Data classes for storing, evaluating, and iterating over records
<a href="#">settings</a> (page 51)	Constants and settings information for hybkit classes and toolkit scripts
<a href="#">type_finder</a> (page 34)	Class for customizable identification of segment type from reference identifiers
<a href="#">analysis</a> (page 40)	Classes for predefined analyses of hyb records
<a href="#">plot</a> (page 47)	Plotting methods for analysis results
<a href="#">util</a> (page 56)	Support methods for executable scripts
<a href="#">errors</a> (page 60)	Error classes for the hybkit package

### 2.1 hybkit (module)

Module storing primary hybkit classes and hybkit API.

This module contains classes and methods for reading, writing, and manipulating data in the hyb genomic sequence format ([Travis2014]). For more information, see the [hybkit Hyb File Specification](#) (page 5).

An example string of a hyb-format line from [Gay2018] is:

```
2407_718\tATCACATTGCCAGGGATTCCAATCCCCAACAATGTGAAAACGGCTGTC\t.\tMIMAT0000078_MirBase_miR-
↪23a_microRNA\t1\t21\t1\t21\t0.0027\tENSG00000188229_ENST00000340384_TUBB2C_mRNA\t23\t
↪t49\t1181\t1207\t1.2e-06
```

Hybkit functionality is primarily based on classes for storage and evaluation of chimeric genomic sequences and associated fold-information:

<a href="#">HybRecord</a> (page 9)	Class to store a single hyb (hybrid) sequence record
<a href="#">FoldRecord</a> (page 23)	Class to store predicted RNA secondary structure information for hybrid reads

Also included are classes for reading, writing, and iterating over files containing hybrid information:

<a href="#">HybFile</a> (page 21)	Class for reading and writing hyb-format files [Travis2014] containing chimeric RNA sequence information as <a href="#">HybRecord</a> (page 9) objects
<a href="#">ViennaFile</a> (page 29)	Class for reading and writing Vienna-format files [ViennaFormat] containing RNA secondary structure information in dot-bracket format as <a href="#">FoldRecord</a> (page 23) objects
<a href="#">CtFile</a> (page 31)	-BETA- Class for reading Connectivity Table (.ct)-format files [CTFormat] containing predicted RNA secondary-structure information as used by UNAFold <sup>33</sup> as <a href="#">FoldRecord</a> (page 23) objects
<a href="#">HybFoldIter</a> (page 33)	Class for concurrent iteration over a <a href="#">HybFile</a> (page 21) and a <a href="#">ViennaFile</a> (page 29) or <a href="#">CtFile</a> (page 31)

## 2.1.1 HybRecord Class

```
class hybkit.HybRecord(id: str34, seq: str35, energy: Optional36[Union37[float38, int39, str40]] = None,
    seg1_props: Optional41[Dict42[str43, Union44[float45, int46, str47]]] = None,
    seg2_props: Optional48[Dict49[str50, Union51[float52, int53, str54]]] = None, flags:
    Optional55[Dict56[str57, Any58]] = None, read_count: Optional59[int60] = None,
    allow_undefined_flags: Optional61[bool62] = None)
```

Class for storing and analyzing chimeric (hybrid) RNA-seq reads in hyb format.

Hyb file (hyb) format entries are a GFF-related file format described by [Travis2014] that contain information about a genomic sequence read identified to be a hybrid by a chimeric read caller. Each line contains 15 or 16 columns separated by tabs ("\t") and provides annotations on each component. An example hyb-format line from [Gay2018]:

```
2407_718\tATCACATTGCCAGGGATTTCGAATCCCAACAATGTGAAAACGGCTGTC\t.\tMIMAT0000078_
→MirBase_miR-23a_microRNA\t1\t21\t1\t21\t0.0027\tENSG00000188229_ENST00000340384_
→TUBB2C_mRNA\t23\t49\t1181\t1207\t1.2e-06
```

The columns are respectively described in hybkit as:

```
id, seq, energy, seg1_ref_name, seg1_read_start, seg1_read_end, seg1_ref_start,
seg1_ref_end, seg1_score, seg2_ref_name, seg2_read_start, seg2_read_end,
seg2_ref_start, seg2_ref_end, seg2_score, flags
```

(For more information, see the [hybkit Hyb File Specification](#) (page 5))

The preferred method for reading hyb records from lines is with the [HybRecord.from\\_line\(\)](#) (page 17) constructor:

```
# line = "2407_718\tATC..."
hyb_record = hybkit.HybRecord.from_line(line)
```

This is the constructor used by the [HybFile](#) (page 21) class to parse hyb files. For example, to print all hybrid identifiers in a hyb file:

```
with hybkit.HybFile('path/to/file.hyb', 'r') as hyb_file:
    # performs "hyb_record = hybkit.HybRecord.from_line(line)" for each line in file
    for hyb_record in hyb_file:
        print(hyb_record.id)
```

HybRecord objects can also be constructed directly. A minimum amount of data necessary for a HybRecord object is the genomic sequence and its corresponding identifier.

<sup>33</sup> <http://www.unafold.org/>

## Examples

```
hyb_record_1 = hybkit.HybRecord('1_100', 'ACTG')
hyb_record_2 = hybkit.HybRecord('2_107', 'CTAG', '-7.3')
hyb_record_3 = hybkit.HybRecord('3_295', 'CTTG', energy='-10.3')
```

Details about segments are provided via python dictionaries with [keys](#) (page 11) specific to each segment. Data can be provided either as strings or as floats/integers (where appropriate). For example, to create a HybRecord object representing the example line given above:

```
seg1_props = {'ref_name': 'MIMAT0000078_MirBase_miR-23a_microRNA',
              'read_start': '1',
              'read_end': '21',
              'ref_start': '1',
              'ref_end': '21',
              'score': '0.0027'}
seg2_props = {'ref_name': 'ENSG00000188229_ENST00000340384_TUBB2C_mRNA',
              'read_start': 23,
              'read_end': 49,
              'ref_start': 1181,
              'ref_end': 1207,
              'score': 1.2e-06}
seq_id = '2407_718'
seq = 'ATCACATTGCCAGGGATTTC CAATCCCCAACAATGTGAAACGGCTGTC'
energy = None

hyb_record = hybkit.HybRecord(seq_id, seq, energy, seg1_props, seg2_props)
# OR
hyb_record = hybkit.HybRecord(seq_id, seq, seg1_props=seg1_props, seg2_props=seg2_
↪ props)
```

### Parameters

- **id** ([str](#)<sup>63</sup>) -- Identifier for the hyb record
- **seq** ([str](#)<sup>64</sup>) -- Nucleotide sequence of the hyb record
- **energy** ([str](#)<sup>65</sup> or [float](#)<sup>66</sup>, optional) -- Predicted energy of sequence folding in kcal/mol
- **seg1\_props** ([dict](#)<sup>67</sup>, optional) -- Properties of segment 1 of the record, containing possible [segment column](#) (page 11) keys: (ref\_name, read\_start, read\_end, ref\_start, ref\_end, score)
- **seg2\_props** ([dict](#)<sup>68</sup>, optional) -- Properties of segment 2 of the record, containing possible: [segment column](#) (page 11) keys: (ref\_name, read\_start, read\_end, ref\_start, ref\_end, score)
- **flags** ([dict](#)<sup>69</sup>, optional) -- Dict with keys of flags for the record and their associated values. By default flags must be defined in [ALL\\_FLAGS](#) (page 11) but custom flags can be supplied by changing [HybRecord.settings\['custom\\_flags'\]](#) (page 11). This setting can also be disabled by setting 'allow\_undefined\_flags' to [True](#)<sup>70</sup> in [HybRecord.settings](#) (page 11).
- **allow\_undefined\_flags** ([bool](#)<sup>71</sup>, optional) -- If [True](#)<sup>72</sup>, allows flags not defined in [ALL\\_FLAGS](#) (page 11) or [HybRecord.settings\['custom\\_flags'\]](#) (page 11) to be added to the record. If not provided, defaults to the value in [HybRecord.settings\['allow\\_undefined\\_flags'\]](#) (page 11).

## Variables

- **id** ([str](#)<sup>73</sup>) -- Identifier for the hyb record (Hyb format: <read-num>\_<read-count>)
- **seq** ([str](#)<sup>74</sup>) -- Nucleotide sequence of the hyb record
- **energy** ([str](#)<sup>75</sup>) -- Predicted energy of folding
- **seg1\_props** ([dict](#)<sup>76</sup>) -- Information on chimeric segment 1, contains [segment column](#) (page 11) keys: [ref\\_name](#) ([str](#)<sup>77</sup>), [read\\_start](#) ([int](#)<sup>78</sup>), [read\\_end](#) ([int](#)<sup>79</sup>), [ref\\_start](#) ([int](#)<sup>80</sup>), [ref\\_end](#) ([int](#)<sup>81</sup>), and [score](#) ([float](#)<sup>82</sup>).
- **seg2\_props** ([dict](#)<sup>83</sup>) -- Information on segment 2, contains [segment column](#) (page 11) keys: [ref\\_name](#) ([str](#)<sup>84</sup>), [read\\_start](#) ([int](#)<sup>85</sup>), [read\\_end](#) ([int](#)<sup>86</sup>), [ref\\_start](#) ([int](#)<sup>87</sup>), [ref\\_end](#) ([int](#)<sup>88</sup>), and [score](#) ([float](#)<sup>89</sup>).
- **flags** ([dict](#)<sup>90</sup>) -- Dict of flags with possible [flag keys](#) (page 11) and values as defined in the [Flags](#) (page 6) section of the [hybkit Hyb File Specification](#) (page 5).
- **fold\_record** ([FoldRecord](#) (page 23)) -- Information on the predicted secondary structure of the sequence set by [set\\_fold\\_record\(\)](#) (page 15).
- **allow\_undefined\_flags** ([bool](#)<sup>91</sup>) -- Whether to allow undefined flags to be set.

**HYBRID\_COLUMNS** = ('id', 'seq', 'energy')

Record columns 1-3 defining parameters of the overall hybrid, defined by the Hyb format

**SEGMENT\_COLUMNS** = ('ref\_name', 'read\_start', 'read\_end', 'ref\_start', 'ref\_end', 'score')

Record columns 4-9 and 10-15, respectively, defining annotated parameters of seg1 and seg2 respectively, defined by the Hyb format

**ALL\_FLAGS** = ('count\_total', 'count\_last\_clustering', 'two\_way\_merged', 'seq\_ids\_in\_cluster', 'read\_count', 'orient', 'det', 'seg1\_type', 'seg2\_type', 'seg1\_det', 'seg2\_det', 'miRNA\_seg', 'target\_reg', 'ext', 'dataset')

Flags defined by the hybkit package. Flags 1-4 are utilized by the Hyb software package. For information on flags, see the [Flags](#) (page 6) portion of the [hybkit Hyb File Specification](#) (page 5).

**settings** = {'allow\_undefined\_flags': False, 'allow\_unknown\_seg\_types': False, 'custom\_flags': [], 'hyb\_placeholder': '.', 'mirna\_types': ['miRNA', 'microRNA'], 'reorder\_flags': True}

Class-level settings. See [settings.HybRecord\\_settings\\_info](#) (page 52) for descriptions.

## TypeFinder

Link to [type\\_finder.TypeFinder](#) (page 34) class for parsing sequence identifiers in assigning segment types by [eval\\_types\(\)](#) (page 14).

**SET\_PROPS** = ('energy', 'full\_seg\_props', 'fold\_record', 'eval\_types', 'eval\_mirna', 'eval\_target')

Properties for the [is\\_set\(\)](#) (page 16) method.

- **energy** : energy is not None
- **full\_seg\_props** : Each seg key is in [segN\\_props](#) (page 10) dict and is not None
- **fold\_record** : [fold\\_record](#) (page 10) has been set
- **eval\_types** : [seg1\\_type](#) (page 10) and [seg2\\_type](#) (page 10) flags have been set
- **eval\_mirna** : [miRNA\\_seg](#) (page 10) flag has been set

GEN\_PROPS = ('has\_indels',)

General record properties for the `prop()` (page 16) method.

- `has_indels` : either `seg1` or `seg2` alignments has insertions/deletions, shown by differing read/reference length for the same alignment

```
STR_PROPS = ('id_is', 'id_prefix', 'id_suffix', 'id_contains', 'seq_is',
'seq_prefix', 'seq_suffix', 'seq_contains', 'seg1_is', 'seg1_prefix', 'seg1_suffix',
'seg1_contains', 'seg2_is', 'seg2_prefix', 'seg2_suffix', 'seg2_contains',
'any_seg_is', 'any_seg_prefix', 'any_seg_suffix', 'any_seg_contains',
'seg1_type_is', 'seg1_type_prefix', 'seg1_type_suffix', 'seg1_type_contains',
'seg2_type_is', 'seg2_type_prefix', 'seg2_type_suffix', 'seg2_type_contains',
'any_seg_type_is', 'any_seg_type_prefix', 'any_seg_type_suffix',
'any_seg_type_contains')
```

String-comparison properties for the `prop()` (page 16) method.

- **Field Types:**

- `id` : *record.id* (page 10)
- `seq` : *record.seq* (page 10)
- `seg1` : *seg1\_props['ref\_name']* (page 10)
- `seg2` : *seg2\_props['ref\_name']* (page 10)
- `any_seg` : *seg1\_props['ref\_name']* (page 10) OR *seg2\_props['ref\_name']* (page 10)
- `seg1_type` : *seg1\_type* (page 6) flag
- `seg2_type` : *seg2\_type* (page 6) flag
- `any_seg_type` : *seg1\_type* (page 6) OR *seg2\_type* (page 6) flags

- **Comparisons:**

- `is` : Comparison string matches field exactly
- `prefix` : Comparison string matches beginning of field
- `suffix` : Comparison string matches end of field
- `contains` : Comparison string is contained within field

MIRNA\_PROPS = ('has\_mirna', 'no\_mirna', 'mirna\_dimer', 'mirna\_not\_dimer',  
'5p\_mirna', '3p\_mirna')

miRNA-evaluation-related properties for the `prop()` (page 16) method. Requires *miRNA\_seg* (page 6) flag to be set by *eval\_mirna()* (page 15) method.

- `has_mirna` : **Either or Both** `Seg1` or `seg2` hve been **identified as** a miRNA
- `no_mirna` : **Both** `Seg1` and `seg2` have been identified as **Not** a miRNA
- `mirna_dimer` : **Both** `seg1` and `seg2` have been **identified as** a miRNA
- `mirna_not_dimer` : **One and Only One** of `seg1` or `seg2` has been **identified as** a miRNA
- `5p_mirna` : `Seg1` (5p) has been identified as a miRNA
- `3p_mirna` : `Seg2` (3p) has been identified as a miRNA

```
MIRNA_STR_PROPS = ('mirna_is', 'mirna_prefix', 'mirna_suffix', 'mirna_contains',
'target_is', 'target_prefix', 'target_suffix', 'target_contains',
'mirna_seg_type_is', 'mirna_seg_type_prefix', 'mirna_seg_type_suffix',
'mirna_seg_type_contains', 'target_seg_type_is', 'target_seg_type_prefix',
'target_seg_type_suffix', 'target_seg_type_contains')
```

- Comparisons:
- `is` : Comparison string matches field exactly
- `prefix` : Comparison string matches beginning of field
- `suffix` : Comparison string matches end of field
- `contains` : Comparison string is contained within field

```
HAS_PROPS = ('has_indels', 'id_is', 'id_prefix', 'id_suffix', 'id_contains',
'seq_is', 'seq_prefix', 'seq_suffix', 'seq_contains', 'seg1_is', 'seg1_prefix',
'seg1_suffix', 'seg1_contains', 'seg2_is', 'seg2_prefix', 'seg2_suffix',
'seg2_contains', 'any_seg_is', 'any_seg_prefix', 'any_seg_suffix',
'any_seg_contains', 'seg1_type_is', 'seg1_type_prefix', 'seg1_type_suffix',
'seg1_type_contains', 'seg2_type_is', 'seg2_type_prefix', 'seg2_type_suffix',
'seg2_type_contains', 'any_seg_type_is', 'any_seg_type_prefix',
'any_seg_type_suffix', 'any_seg_type_contains', 'has_mirna', 'no_mirna',
'mirna_dimer', 'mirna_not_dimer', '5p_mirna', '3p_mirna', 'mirna_is',
'mirna_prefix', 'mirna_suffix', 'mirna_contains', 'target_is', 'target_prefix',
'target_suffix', 'target_contains', 'mirna_seg_type_is', 'mirna_seg_type_prefix',
'mirna_seg_type_suffix', 'mirna_seg_type_contains', 'target_seg_type_is',
'target_seg_type_prefix', 'target_seg_type_suffix', 'target_seg_type_contains')
```

All allowed properties for the `prop()` (page 16) method. See [GEN\\_PROPS](#) (page 11), [STR\\_PROPS](#) (page 12), [MIRNA\\_PROPS](#) (page 12), and [MIRNA\\_STR\\_PROPS](#) (page 12)

**set\_flag**(*flag\_key*: [str](#)<sup>92</sup>, *flag\_val*: [Optional](#)<sup>93</sup>[[Union](#)<sup>94</sup>[[float](#)<sup>95</sup>, [int](#)<sup>96</sup>, [str](#)<sup>97</sup>, [bool](#)<sup>98</sup>]], *allow\_undefined\_flags*: [Optional](#)<sup>99</sup>[[bool](#)<sup>100</sup>] = `None`) → `None`<sup>101</sup>

Set the value of record `flag_key` to `flag_val`.

#### Parameters

- **flag\_key** ([str](#)<sup>102</sup>) -- Key for flag to set.
- **flag\_val** -- Value for flag to set.
- **allow\_undefined\_flags** ([bool](#)<sup>103</sup>, optional) -- Allow inclusion of flags not defined in [ALL\\_FLAGS](#) (page 11) or in `settings['custom_flags']` (page 11). If not provided, uses setting in 'HybRecord.allow\_undefined\_flags' (Defaults to value in: `settings['allow_undefined_flags']` (page 11)).

**get\_seg1\_type**(*require*: [bool](#)<sup>104</sup> = `False`) → [Optional](#)<sup>105</sup>[[str](#)<sup>106</sup>]

Return the [seg1\\_type](#) (page 6) flag if defined, or return `None`.

#### Parameters

**require** -- If True, raise an error if `seg1_type` is not defined.

**get\_seg2\_type**(*require*: [bool](#)<sup>107</sup> = `False`) → [Optional](#)<sup>108</sup>[[str](#)<sup>109</sup>]

Return the [seg2\\_type](#) (page 6) flag if defined, or return `None`.

#### Parameters

**require** ([bool](#)<sup>110</sup>, optional) -- If True, raise an error if `seg2_type` is not defined.

**get\_seg\_types**(*require*: [bool](#)<sup>111</sup> = `False`) → [Tuple](#)<sup>112</sup>[[Optional](#)<sup>113</sup>[[str](#)<sup>114</sup>], [Optional](#)<sup>115</sup>[[str](#)<sup>116</sup>]]

Return "seg1\_type" (or `None`), "seg2\_type" (or `None`) flags.

Return a tuple of the [seg1\\_type](#) (page 6) and [seg2\\_type](#) (page 6) flags for each respective flag that is defined, or `None` for each flag that is not.

**Parameters**

**require** ([bool](#)<sup>117</sup>, optional) -- If True, raise an error if either flag is not defined.

**get\_read\_count** (*require: bool*<sup>118</sup> = False) → [Optional](#)<sup>119</sup>[[int](#)<sup>120</sup>]

Return the [read\\_count](#) (page 6) flag if defined, otherwise return None.

**Parameters**

**require** ([bool](#)<sup>121</sup>, optional) -- If True, raise an error if the "read\_count" flag is not defined.

**get\_record\_count** (*require: bool*<sup>122</sup> = False) → [int](#)<sup>123</sup>

Return [count\\_total](#) (page 6) flag if defined, or return 1 (this record).

**Parameters**

**require** ([bool](#)<sup>124</sup>, optional) -- If True, raise an error if the "count\_total" flag is not defined.

**get\_mirna\_props** (*allow\_mirna\_dimers: bool*<sup>125</sup> = False, *require: bool*<sup>126</sup> = True) → [Optional](#)<sup>127</sup>[[Dict](#)<sup>128</sup>]

Return the seg\_props dict corresponding to the miRNA segment, if set.

If [eval\\_mirna\(\)](#) (page 15) has been run, return the seg\_props dict corresponding to the miRNA segment type as determined by checking the [miRNA\\_seg](#) (page 6) flag, or [None](#)<sup>129</sup> if the record does not contain a miRNA.

**Parameters**

- **allow\_mirna\_dimers** ([bool](#)<sup>130</sup>, optional) -- If True, consider miRNA dimers as a miRNA/target pair and return the 5p miRNA segment properties.
- **require** ([bool](#)<sup>131</sup>, optional) -- If True, raise an error if the read does not contain a miRNA-annotated segment (Default: True).

**get\_target\_props** (*allow\_mirna\_dimers: bool*<sup>132</sup> = False, *require: bool*<sup>133</sup> = True) → [Optional](#)<sup>134</sup>[[Dict](#)<sup>135</sup>]

Return the seg\_props dict corresponding to the target segment, if set.

If [eval\\_mirna\(\)](#) (page 15) has been run, return the seg\_props dict corresponding to the target segment type as determined by checking the [miRNA\\_seg](#) (page 6) flag, (and returning the other segment), or [None](#)<sup>136</sup> if the record does not contain a miRNA or contains two miRNAs.

**Parameters**

- **allow\_mirna\_dimers** ([bool](#)<sup>137</sup>, optional) -- If True, consider miRNA dimers as a miRNA/target pair and return the 3p miRNA segment properties as the arbitrarily-selected "target" of the dimer pair.
- **require** ([bool](#)<sup>138</sup>, optional) -- If True, raise an error if the read does not contain a single target-annotated segment (Default: True).

**eval\_types** (*allow\_unknown: Optional*<sup>139</sup>[[bool](#)<sup>140</sup>] = None) → [None](#)<sup>141</sup>

Find the types of each segment using the the [TypeFinder](#) (page 11) class.

This method provides [HybRecord.seg1\\_props](#) and [HybRecord.seg2\\_props](#) to the [TypeFinder](#) (page 11) class, linked as attribute [HybRecord.TypeFinder](#) (page 11). This uses the method: [TypeFinder.find](#) (page 35) set by [TypeFinder.set\\_method](#) (page 35) or [TypeFinder.set\\_custom\\_method](#) (page 36) to set the [seg1\\_type](#) (page 6), [seg2\\_type](#) (page 6) flags if not already set.

To use a type-finding method other than the default, prepare the [TypeFinder](#) (page 34) class by preparing and setting [TypeFinder.params](#) (page 34) and using [TypeFinder.set\\_method](#) (page 35).

**Parameters**

**allow\_unknown** ([bool](#)<sup>142</sup>, optional) -- If True, allow segment types that cannot be



identified and set them as "unknown". Otherwise raise an error. If not provided uses setting in `settings['allow_unknown_seg_types']` (page 11).

**set\_fold\_record**(*fold\_record*: [Union](#)<sup>143</sup>[[FoldRecord](#) (page 23), [Tuple](#)<sup>144</sup>[[FoldRecord](#) (page 23), [Any](#)<sup>145</sup>]], *allow\_energy\_mismatch*: [bool](#)<sup>146</sup> = `False`) → [None](#)<sup>147</sup>

Check and set provided *fold\_record* ([FoldRecord](#) (page 23)) as attribute *fold\_record*.

Ensures that *fold\_record* argument is an instance of [FoldRecord](#) and has a matching sequence to this [HybRecord](#), then set as [HybRecord.fold\\_record](#) (page 10).

#### Parameters

- **fold\_record** ([FoldRecord](#) (page 23)) -- [FoldRecord](#) (page 23) instance to set as [HybRecord.fold\\_record](#) (page 10).
- **allow\_energy\_mismatch** ([bool](#)<sup>148</sup>, optional) -- If `True`, allow mismatched *fold\_record* and [HybRecord](#) energy. Otherwise, raise an error.

**eval\_mirna**(*override*: [bool](#)<sup>149</sup> = `False`, *mirna\_types*: [Optional](#)<sup>150</sup>[[bool](#)<sup>151</sup>] = `None`) → [None](#)<sup>152</sup>

Analyze and set miRNA properties from type properties in the *hyb* record.

If not already done, determine whether a miRNA exists within this record and set the *miRNA\_seg* (page 6) flag. This evaluation requires the *seg1\_type* (page 6) and *seg2\_type* (page 6) flags to be populated, which can be performed by the [eval\\_types\(\)](#) (page 14) method.

#### Parameters

- **override** ([bool](#)<sup>153</sup>, optional) -- If `True`, override existing *miRNA\_seg* (page 6) flag if present.
- **mirna\_types** ([list](#)<sup>154</sup>, [tuple](#)<sup>155</sup>, or [set](#)<sup>156</sup>, optional) -- Iterable of string representing sequence types considered as miRNA. Otherwise, the types are used from `settings['mirna_types']` (page 11) (it is suggested that this be provided as a [set](#)<sup>157</sup> for fastest checking).

**mirna\_details**(*detail*: [Literal](#)<sup>158</sup>['all', 'mirna\_ref', 'target\_ref', 'mirna\_seg\_type', 'target\_seg\_type', 'mirna\_seq', 'target\_seq', 'mirna\_fold', 'target\_fold'] = 'all', *allow\_mirna\_dimers*: [bool](#)<sup>159</sup> = `False`) → [Optional](#)<sup>160</sup>[[Union](#)<sup>161</sup>[[Dict](#)<sup>162</sup>, [str](#)<sup>163</sup>]]

Provide a detail about the miRNA or target following [eval\\_mirna\(\)](#) (page 15).

Analyze miRNA properties within the sequence record and provide a detail as output. Unless *allow\_mirna\_dimers* is `True`, this method requires record to contain a non-dimer miRNA, otherwise an error will be raised.

#### Parameters

- **detail** ([str](#)<sup>164</sup>) --  
Type of detail to return. Options include:  
all : Dict of all properties (default)  
mirna\_ref : Identifier for Assigned miRNA  
target\_ref : Identifier for Assigned Target  
mirna\_seg\_type : Assigned *seg\_type* of miRNA  
target\_seg\_type : Assigned *seg\_type* of target  
mirna\_seq : Annotated subsequence of miRNA  
target\_seq : Annotated subsequence of target  
mirna\_fold : Annotated fold substring of miRNA (requires *fold\_record* set)  
target\_fold : Annotated fold substring of target (requires *fold\_record* set)

- **allow\_mirna\_dimers** ([bool](#)<sup>165</sup>, optional) -- Allow miRNA/miRNA dimers. The 5p-position will be assigned as the "miRNA", and the 3p-position will be assigned as the "target".

**mirna\_detail**(\*args, \*\*kwargs)

Deprecate, alias for [mirna\\_details\(\)](#) (page 15).

Deprecated since version v0.3.0.

**is\_set**(prop: [str](#)<sup>166</sup>) → [bool](#)<sup>167</sup>

Return True if HybRecord property "prop" is set (if relevant) and is not None.

Options described in [SET\\_PROPS](#) (page 11).

#### Parameters

**prop** ([str](#)<sup>168</sup>) -- Property / Analysis to check

**not\_set**(prop: [str](#)<sup>169</sup>) → [bool](#)<sup>170</sup>

Return False if HybRecord property "prop" is set (if relevant) and is not None.

( returns not `is_set(prop)` )

#### Parameters

**prop** ([str](#)<sup>171</sup>) -- Property / Analysis to check

**prop**(prop: [str](#)<sup>172</sup>, prop\_compare: [Optional](#)<sup>173</sup>[[str](#)<sup>174</sup>] = None) → [bool](#)<sup>175</sup>

Return True if HybRecord has property: prop.

Check property against list of allowed properties in [HAS\\_PROPS](#) (page 13). If query property has a string comparator, provide this in prop\_compare. Raises an error if a prerequisite field is not set (use [is\\_set\(\)](#) (page 16) to check whether properties are set).

Specific properties available to check are described in attributes:

<a href="#">GEN_PROPS</a> (page 11)	General Record Properties
<a href="#">STR_PROPS</a> (page 12)	Field String Comparison Properties
<a href="#">MIRNA_PROPS</a> (page 12)	miRNA-Associated Record Properties
<a href="#">MIRNA_STR_PROPS</a> (page 12)	miRNA-Associated String Comparison Properties

#### Parameters

- **prop** ([str](#)<sup>176</sup>) -- Property to check
- **prop\_compare** ([str](#)<sup>177</sup>, optional) -- Comparator to check.

**has\_prop**(\*args, \*\*kwargs)

Return True if HybRecord has property: prop.

Deprecated since version v0.3.0: Use [prop\(\)](#) (page 16) instead.

**to\_line**(newline: [bool](#)<sup>178</sup> = True, sep: [str](#)<sup>179</sup> = '\n') → [str](#)<sup>180</sup>

Return a hyb-format string representation of the record.

#### Parameters

- **newline** ([bool](#)<sup>181</sup>, optional) -- Terminate returned string with a newline (default: True)
- **sep** ([str](#)<sup>182</sup>, optional) -- Separator between fields (Default: "\t")

**to\_csv**(*newline*: *bool*<sup>183</sup> = False) → *str*<sup>184</sup>

Return a comma-separated hyb-format string representation of the record.

#### Parameters

**newline** (*bool*<sup>185</sup>, optional) -- If True, end the returned string with a newline.

**to\_fields**(*missing\_obj*: *Optional*<sup>186</sup>[*Union*<sup>187</sup>[*float*<sup>188</sup>, *int*<sup>189</sup>, *str*<sup>190</sup>, *bool*<sup>191</sup>]] = None) → *dict*<sup>192</sup>

Return a python dictionary representation of the record.

Returns a dictionary with keys corresponding to the fields in the hyb-format file, and values corresponding to the values in the record. Output can be used with the pandas DataFrame constructor or csv.DictWriter.

#### Parameters

**missing\_obj** (*optional*) -- Object to use for missing values. Default = *None*<sup>193</sup>.

**to\_fasta\_record**(*mode*: *Literal*<sup>194</sup>['hybrid', 'seg1', 'seg2', 'mirna', 'target'] = 'hybrid', *annotate*: *bool*<sup>195</sup> = True, *allow\_mirna\_dimers*: *bool*<sup>196</sup> = False) → *None*<sup>197</sup>

Return nucleotide sequence as BioPython SeqRecord object.

#### Parameters

- **mode** (*str*<sup>198</sup>, optional) --

Determines which sequence component to return. Options:

hybrid: Entire hybrid sequence (default)

seg1: Sequence 1 (if defined)

seg2: Sequence 2 (if defined)

mirna: miRNA sequence of miRNA/target pair (if defined, else None)

target: Target sequence of miRNA/target pair (if defined, else None)

- **annotate** (*bool*<sup>199</sup>, optional) -- Add name of components to fasta sequence identifier if present.

- **allow\_mirna\_dimers** (*bool*<sup>200</sup>, optional) --

If True, allow miRNA dimers to be

returned as miRNA sequence (the 5p segment

will be selected as the "miRNA").

**to\_fasta\_str**(*mode*: *Literal*<sup>201</sup>['hybrid', 'seg1', 'seg2', 'mirna', 'target'] = 'hybrid', *annotate*: *bool*<sup>202</sup> = True) → *str*<sup>203</sup>

Return nucleotide sequence as a fasta string.

#### Parameters

- **mode** (*str*<sup>204</sup>, optional) --

as with *to\_fasta\_record()* (page 17) method.

- **annotate** (*bool*<sup>205</sup>, optional) -- Add name of components to fasta sequence identifier if present.

**classmethod from\_line**(*line*: *str*<sup>206</sup>, *hybformat\_id*: *bool*<sup>207</sup> = False, *hybformat\_ref*: *bool*<sup>208</sup> = False) → Self

Construct a HybRecord instance from a single-line hyb-format string.

The Hyb software package ([Travis2014]) records read-count information in the "id" field of the record, which can be read by setting *hybformat\_id*=True. Additionally, the Hyb hOH7 database contains the segment type in the identifier of each reference in the 4th field, which can be read by setting *hybformat\_ref*=True.

**Parameters**

- **line** (*str*<sup>209</sup>) -- hyb-format string containing record information.
- **hybformat\_id** (*bool*<sup>210</sup>, optional) -- If True, read count information from identifier in <read\_number>\_<read\_count> format.
- **hybformat\_ref** (*bool*<sup>211</sup>, optional) -- If True, read additional record information from identifier in <gene\_id>\_<transcript\_id>\_<gene\_name>\_<seg\_type> format.

**Returns**

*HybRecord* (page 9) instance containing record information.

```
classmethod from_fasta_records(seg1_record: None212, seg2_record: None213, hyb_id: Optional214[str215] = None, energy: Optional216[Union217[float218, int219, str220]] = None, flags: Optional221[Dict222[str223, Any224]] = None) → Self
```

Construct a HybRecord instance from two BioPython SeqRecord Objects.

Create artificial HybRecord from two SeqRecord Objects For the hybrid:

```
id: [seg1_record.id]--[seg2_record.id] (overwritten by "id" parameter if provided)
seq: seg1_record.seq + seg2_record
```

For each segment:

```
FASTA_Sequence_ID -> segN_ref_name
```

```
FASTA_Description -> Flags: segN_det (Overwritten if segN_det flag is provided directly)
```

Optional fields to add via function arguments:

```
hyb_id
energy
flags
```

**Parameters**

- **seg1\_record** (*SeqRecord*) -- Biopython SeqRecord object containing information on the left/first/5p hybrid segment (seg1)
- **seg2\_record** (*SeqRecord*) -- Biopython SeqRecord object containing information on the right/second/3p hybrid segment (seg2)
- **hyb\_id** (*str*<sup>225</sup>, optional) -- Identifier for the hyb record (overwrites generated id if provided)
- **energy** (*str*<sup>226</sup> or *float*<sup>227</sup>, optional) -- Predicted energy of sequence folding in kcal/mol
- **flags** (*dict*<sup>228</sup>, optional) -- Dict with keys of flags for the record and their associated values. Any flags provided overwrite default-generated flags.

**Returns**

*HybRecord* (page 9) instance containing record information.

```
classmethod to_fields_header() → Literal229['id', 'seq', 'energy', 'seg1_ref_name', 'seg1_read_start', 'seg1_read_end', 'seg1_ref_start', 'seg1_ref_end', 'seg1_score', 'seg2_ref_name', 'seg2_read_start', 'seg2_read_end', 'seg2_ref_start', 'seg2_ref_end', 'seg2_score', 'flags']
```

Return a list of the fields in a *HybRecord* (page 9) object.

For use with the *to\_fields()* (page 17) method.

**classmethod** *to\_csv\_header*(*newline*: *bool*<sup>230</sup> = *False*) → *Lit-*  
*eral*<sup>231</sup>['id,seq,energy,seg1\_ref\_name,seg1\_read\_start,seg1\_read\_end,seg1\_ref\_start,seg1\_ref\_e

Return a comma-separated string representation of the fields in the record.

For use with the *to\_csv()* (page 16) method.

**Parameters**

**newline** (*bool*<sup>232</sup>, optional) -- If True, end the returned string with a newline.

---

```

34 https://docs.python.org/3/library/stdtypes.html#str
35 https://docs.python.org/3/library/stdtypes.html#str
36 https://docs.python.org/3/library/typing.html#typing.Optional
37 https://docs.python.org/3/library/typing.html#typing.Union
38 https://docs.python.org/3/library/functions.html#float
39 https://docs.python.org/3/library/functions.html#int
40 https://docs.python.org/3/library/stdtypes.html#str
41 https://docs.python.org/3/library/typing.html#typing.Optional
42 https://docs.python.org/3/library/typing.html#typing.Dict
43 https://docs.python.org/3/library/stdtypes.html#str
44 https://docs.python.org/3/library/typing.html#typing.Union
45 https://docs.python.org/3/library/functions.html#float
46 https://docs.python.org/3/library/functions.html#int
47 https://docs.python.org/3/library/stdtypes.html#str
48 https://docs.python.org/3/library/typing.html#typing.Optional
49 https://docs.python.org/3/library/typing.html#typing.Dict
50 https://docs.python.org/3/library/stdtypes.html#str
51 https://docs.python.org/3/library/typing.html#typing.Union
52 https://docs.python.org/3/library/functions.html#float
53 https://docs.python.org/3/library/functions.html#int
54 https://docs.python.org/3/library/stdtypes.html#str
55 https://docs.python.org/3/library/typing.html#typing.Optional
56 https://docs.python.org/3/library/typing.html#typing.Dict
57 https://docs.python.org/3/library/stdtypes.html#str
58 https://docs.python.org/3/library/typing.html#typing.Any
59 https://docs.python.org/3/library/typing.html#typing.Optional
60 https://docs.python.org/3/library/functions.html#int
61 https://docs.python.org/3/library/typing.html#typing.Optional
62 https://docs.python.org/3/library/functions.html#bool
63 https://docs.python.org/3/library/stdtypes.html#str
64 https://docs.python.org/3/library/stdtypes.html#str
65 https://docs.python.org/3/library/stdtypes.html#str
66 https://docs.python.org/3/library/functions.html#float
67 https://docs.python.org/3/library/stdtypes.html#dict
68 https://docs.python.org/3/library/stdtypes.html#dict
69 https://docs.python.org/3/library/stdtypes.html#dict
70 https://docs.python.org/3/library/constants.html#True
71 https://docs.python.org/3/library/functions.html#bool
72 https://docs.python.org/3/library/constants.html#True
73 https://docs.python.org/3/library/stdtypes.html#str
74 https://docs.python.org/3/library/stdtypes.html#str
75 https://docs.python.org/3/library/stdtypes.html#str
76 https://docs.python.org/3/library/stdtypes.html#dict
77 https://docs.python.org/3/library/stdtypes.html#str
78 https://docs.python.org/3/library/functions.html#int
79 https://docs.python.org/3/library/functions.html#int
80 https://docs.python.org/3/library/functions.html#int
81 https://docs.python.org/3/library/functions.html#int
82 https://docs.python.org/3/library/functions.html#float
83 https://docs.python.org/3/library/stdtypes.html#dict
84 https://docs.python.org/3/library/stdtypes.html#str
85 https://docs.python.org/3/library/functions.html#int
86 https://docs.python.org/3/library/functions.html#int
87 https://docs.python.org/3/library/functions.html#int
88 https://docs.python.org/3/library/functions.html#int
89 https://docs.python.org/3/library/functions.html#float
90 https://docs.python.org/3/library/stdtypes.html#dict
91 https://docs.python.org/3/library/functions.html#bool
92 https://docs.python.org/3/library/stdtypes.html#str
93 https://docs.python.org/3/library/typing.html#typing.Optional
94 https://docs.python.org/3/library/typing.html#typing.Union
95 https://docs.python.org/3/library/functions.html#float
96 https://docs.python.org/3/library/functions.html#int
97 https://docs.python.org/3/library/stdtypes.html#str
98 https://docs.python.org/3/library/functions.html#bool
99 https://docs.python.org/3/library/typing.html#typing.Optional
100 https://docs.python.org/3/library/functions.html#bool
101 https://docs.python.org/3/library/constants.html#None

```

---

```

102 https://docs.python.org/3/library/stdtypes.html#str
103 https://docs.python.org/3/library/functions.html#bool
104 https://docs.python.org/3/library/functions.html#bool
105 https://docs.python.org/3/library/typing.html#typing.Optional
106 https://docs.python.org/3/library/stdtypes.html#str
107 https://docs.python.org/3/library/functions.html#bool

```

## 2.1.2 HybFile Class

**class** `hybkit.HybFile`(*path*: [str](#)<sup>233</sup>, *\*args*: [Any](#)<sup>234</sup>, *hybformat\_id*: [Optional](#)<sup>235</sup> [[bool](#)<sup>236</sup>] = `None`, *hybformat\_ref*: [Optional](#)<sup>237</sup> [[bool](#)<sup>238</sup>] = `None`, *from\_file\_like*: [bool](#)<sup>239</sup> = `False`, *\*\*kwargs*: [Any](#)<sup>240</sup>)

Wrapper for a hyb-format text file which returns entries (lines) as `HybRecord` objects.

### Parameters

- **path** ([str](#)<sup>241</sup>) -- Path to text file to open as hyb-format file.
- **\*args** -- Arguments passed to [open\(\)](#) (page 22) function to open a text file for reading/writing.
- **hybformat\_id** ([bool](#)<sup>242</sup>, optional) -- If `True`, during parsing of lines read count information from identifier in `<read_number>_<read_count>` format. Defaults to value in [settings\['hybformat\\_id'\]](#) (page 21).
- **hybformat\_ref** ([bool](#)<sup>243</sup>, optional) -- If `True`, during parsing of lines read additional record information from identifier in `<gene_id>_<transcript_id>_<gene_name>_<seg_type>` format. Defaults to value in [settings\['hybformat\\_ref'\]](#) (page 21).
- **from\_file\_like** ([bool](#)<sup>244</sup>, optional) -- If `True`, the first argument is treated as a file-like object (such as `io.StringIO` or `gzip.GzipFile`) and the remaining positional arguments are ignored. (Default `False`)
- **\*\*kwargs** -- Keyword arguments passed to [open\(\)](#) (page 22) function to open a text file for reading/writing.

### Variables

- **hybformat\_id** ([bool](#)<sup>245</sup>) -- Read count information from identifier during line parsing
- **hybformat\_ref** ([bool](#)<sup>246</sup>) -- Read type information from reference name during line parsing
- **fh** (*file*) -- Underlying file handle for the `HybFile` object.

**settings** = {'hybformat\_id': `False`, 'hybformat\_ref': `False`}

Class-level settings. See [hybkit.settings.HybFile\\_settings\\_info](#) (page 52) for descriptions.

**close()** → [None](#)<sup>247</sup>

Close the file.

**read\_record()** → [str](#)<sup>248</sup>

Return next line of hyb file as `HybRecord` object.

**read\_records()** → [List](#)<sup>249</sup> [[str](#)<sup>250</sup>]

Return list of all (remaining) records in hyb file as `HybRecord` objects.

**write\_record**(*write\_record*: [HybRecord](#) (page 9)) → [None](#)<sup>251</sup>

Write a `HybRecord` object to file as a Hyb-format string.

Unlike the `file.write()` method, this method will add a newline to the end of each written record line.

### Parameters

**write\_record** ([HybRecord](#) (page 9)) -- Record to write.

**write\_records**(*write\_records*: [Iterable](#)<sup>252</sup> [[HybRecord](#) (page 9)]) → [None](#)<sup>253</sup>

Write a sequence of `HybRecord` objects as hyb-format lines to the Hyb file.

Unlike the `file.writelines()` method, this method will add a newline to the end of each written record line.

**Parameters**

**write\_records** (*list*<sup>254</sup>) -- List of *HybRecord* (page 9) objects to write.

**write\_fh** (\*args, \*\*kwargs) → *None*<sup>255</sup>

Write directly to the underlying file handle.

**write** (\*\_args, \*\*\_kwargs) → *None*<sup>256</sup>

Implement no-op / error for "write" method to catch errors.

Use *write\_record()* (page 21) or *write\_fh()* (page 22) instead.

**classmethod open** (path: *str*<sup>257</sup>, \*args: *Any*<sup>258</sup>, hybformat\_id: *Optional*<sup>259</sup>[*bool*<sup>260</sup>] = *None*, hybformat\_ref: *Optional*<sup>261</sup>[*bool*<sup>262</sup>] = *None*, \*\*kwargs: *Any*<sup>263</sup>) → *Self*

Open a path to a text file using *open()* (page 22) and return a *HybFile* object.

Arguments match those of the Python3 built-in *open()* (page 22) function and are passed directly to it.

This method is provided as a convenience function for drop-in replacement of the built-in *open()* (page 22) function.

Specific keyword arguments are provided for *HybFile*-specific settings:

**Parameters**

- **path** (*str*<sup>264</sup>) -- Path to file to open.
- **hybformat\_id** (*bool*<sup>265</sup>, optional) -- If True, during parsing of lines read count information from identifier in <read\_number>\_<read\_count> format. Defaults to value in *settings*[*hybformat\_id*] (page 21).
- **hybformat\_ref** (*bool*<sup>266</sup>, optional) -- If True, during parsing of lines read additional record information from identifier in <gene\_id>\_<transcript\_id>\_<gene\_name>\_<seg\_type> format. Defaults to value in *settings*[*hybformat\_ref*] (page 21).

**Example usage:**

```
with HybFile.open('path/to/file.hyb', 'r') as hyb_file:
    for record in hyb_file:
        print(record)
```

**Parameters**

- **\*args** -- Passed directly to *open()* (page 22).
- **\*\*kwargs** -- Passed directly to *open()* (page 22).

**Returns**

*HybFile* (page 21) object.



### 2.1.3 FoldRecord Class

**class** `hybkit.FoldRecord`(*id*: [str](#)<sup>267</sup>, *seq*: [str](#)<sup>268</sup>, *fold*: [str](#)<sup>269</sup>, *energy*: [Optional](#)<sup>270</sup>[[Union](#)<sup>271</sup>[[float](#)<sup>272</sup>, [int](#)<sup>273</sup>, [str](#)<sup>274</sup>]] = `None`, *seq\_type*: [Optional](#)<sup>275</sup>[[Literal](#)<sup>276</sup>['static', 'dynamic']] = `None`)

Class for storing secondary structure (folding) information for a nucleotide sequence.

This class supports the following file types: (Data courtesy of [[Gay2018](#)])

- The ".vienna" file format used by the [ViennaRNA](#)<sup>277</sup> package ([[ViennaFormat](#)]; [[Lorenz2011](#)]):

**Example:**

```
34_151138_MIMAT0000076_MirBase_miR-21_microRNA_1_19-...
TAGCTTATCAGACTGATGTTAGCTTATCAGACTGATG
.....(((((((.((((((.....)))))).)))))) (-11.1)
```

- The ".ct" file format used by [UNAFold](#)<sup>278</sup> and other packages ([[CTFormat](#)], [[Zuker2003](#)]):

**Example:**

```
41          dG = -8 dH = -93.9          seq1_name-seq2_name
1 A          0          2          0          1          0          0
2 G          1          3          0          2          0          0
...
...
```

(continues on next page)

233 <https://docs.python.org/3/library/stdtypes.html#str>  
234 <https://docs.python.org/3/library/typing.html#typing.Any>  
235 <https://docs.python.org/3/library/typing.html#typing.Optional>  
236 <https://docs.python.org/3/library/functions.html#bool>  
237 <https://docs.python.org/3/library/typing.html#typing.Optional>  
238 <https://docs.python.org/3/library/functions.html#bool>  
239 <https://docs.python.org/3/library/functions.html#bool>  
240 <https://docs.python.org/3/library/typing.html#typing.Any>  
241 <https://docs.python.org/3/library/stdtypes.html#str>  
242 <https://docs.python.org/3/library/functions.html#bool>  
243 <https://docs.python.org/3/library/functions.html#bool>  
244 <https://docs.python.org/3/library/functions.html#bool>  
245 <https://docs.python.org/3/library/functions.html#bool>  
246 <https://docs.python.org/3/library/functions.html#bool>  
247 <https://docs.python.org/3/library/constants.html#None>  
248 <https://docs.python.org/3/library/stdtypes.html#str>  
249 <https://docs.python.org/3/library/typing.html#typing.List>  
250 <https://docs.python.org/3/library/stdtypes.html#str>  
251 <https://docs.python.org/3/library/constants.html#None>  
252 <https://docs.python.org/3/library/typing.html#typing.Iterable>  
253 <https://docs.python.org/3/library/constants.html#None>  
254 <https://docs.python.org/3/library/stdtypes.html#list>  
255 <https://docs.python.org/3/library/constants.html#None>  
256 <https://docs.python.org/3/library/constants.html#None>  
257 <https://docs.python.org/3/library/stdtypes.html#str>  
258 <https://docs.python.org/3/library/typing.html#typing.Any>  
259 <https://docs.python.org/3/library/typing.html#typing.Optional>  
260 <https://docs.python.org/3/library/functions.html#bool>  
261 <https://docs.python.org/3/library/typing.html#typing.Optional>  
262 <https://docs.python.org/3/library/functions.html#bool>  
263 <https://docs.python.org/3/library/typing.html#typing.Any>  
264 <https://docs.python.org/3/library/stdtypes.html#str>  
265 <https://docs.python.org/3/library/functions.html#bool>  
266 <https://docs.python.org/3/library/functions.html#bool>

(continued from previous page)

...							
40	G	39	41	11	17	39	41
41	T	40	0	10	18	40	0

A minimum amount of data necessary for a FoldRecord object is a sequence identifier, a genomic sequence, and its fold representation.

Two types of FoldRecord objects are supported, 'static' and 'dynamic'. Static FoldRecord objects are those where the 'seq' attribute matches exactly to the corresponding HybRecord.seq attribute (where applicable). Dynamic FoldRecord objects are those where FoldRecord.seq is reconstructed from aligned regions of a HybRecord.seq chimeric read: Longer for chimeras with overlapping alignments, shorter for chimeras with gapped alignments.

Overlapping Alignment Example:

```
Static:
seg1: 11111111111111111111
seg2:                22222222222222222222
seq:  TAGCTTATCAGACTGATGTTTTAGCTTATCAGACTGATG

Dynamic:
seg1: 11111111111111111111
seg2:                22222222222222222222
seq:  TAGCTTATCAGACTGATGTTTTTTTAGCTTATCAGACTGATG
```

Gapped Alignment Example:

```
Static:
seg1: 1111111111111111
seg2:                22222222222222222222
seq:  TTAGCTTATCAGACTGATGTTAGCTTATCAGACTGATG

Dynamic:
seg1: 1111111111111111
seg2:                22222222222222222222
seq:  AGCTTATCAGACTGATTAGCTTATCAGACTGATG
```

Dynamic sequences are found in the Hyb program \*\_hybrids\_ua.hyb file type. This is primarily relevant in error-checking when setting the *HybRecord.set\_fold\_record()* (page 15) method.

When the 'static' FoldRecord type is used, the following methods are used for HybRecord.fold\_record error-checking:

- *static\_count\_hyb\_record\_mismatches()* (page 25)

When the 'dynamic' FoldRecord type is used, the following methods are used for HybRecord.fold\_record error-checking:

- *dynamic\_count\_hyb\_record\_mismatches()* (page 25)

#### Parameters

- **id** (*str*<sup>279</sup>) -- Identifier for record
- **seq** (*str*<sup>280</sup>) -- Nucleotide sequence of record.
- **fold** (*str*<sup>281</sup>) -- Fold representation of record.
- **energy** (*str*<sup>282</sup> or *float*<sup>283</sup>, optional) -- Energy of folding for record.

- **seq\_type** ([str<sup>284</sup>](#), optional) -- Expect sequence to be 'static' (match exactly to corresponding *HybRecord.seq* (page 10)), or 'dynamic' (construct from pieces of *HybRecord.seq* (page 10)). if not provided, defaults to `~settings['seq_type']` (page 25) setting. See *hybkit.settings.FoldRecord\_settings\_info* (page 53) for descriptions.

#### Variables

- **id** ([str<sup>285</sup>](#)) -- Sequence Identifier (often seg1name-seg2name)
- **seq** ([str<sup>286</sup>](#)) -- Genomic Sequence
- **fold** ([str<sup>287</sup>](#)) -- Dot-bracket Fold Representation, '(', '.', and ')' characters
- **energy** ([str<sup>288</sup>](#)) -- Predicted energy of folding
- **seq\_type** ([str<sup>289</sup>](#)) -- Whether sequence is 'static' or 'dynamic' (Default: 'static'; see Args for details)

**settings** = {'allowed\_mismatches': 0, 'error\_mode': 'raise', 'fold\_placeholder': '.', 'seq\_type': 'static'}

Class-level settings. See *hybkit.settings.FoldRecord\_settings\_info* (page 53) for descriptions.

**to\_vienna\_lines**(*newline*: [bool<sup>290</sup>](#) = True) → [List<sup>291</sup>\[str<sup>292</sup>\]](#)

Return a list of lines for the record in vienna format.

See (*Vienna File Format* (page 23)).

#### Parameters

**newline** ([bool<sup>293</sup>](#), optional) -- Add newline character to the end of each returned line. (Default: True)

**to\_vienna\_string**(*newline*: [bool<sup>294</sup>](#) = True) → [str<sup>295</sup>](#)

Return a 3-line string for the record in vienna format.

See (*Vienna File Format* (page 23)).

#### Parameters

**newline** ([bool<sup>296</sup>](#), optional) -- Terminate the returned string with a newline character. (Default: True)

**count\_hyb\_record\_mismatches**(*hyb\_record*: [HybRecord](#) (page 9)) → [int<sup>297</sup>](#)

Count mismatches between *hyb\_record.seq* and *fold\_record.seq*.

Uses *static\_count\_hyb\_record\_mismatches()* (page 25) if *seq\_type* is static, or *dynamic\_count\_hyb\_record\_mismatches()* (page 25) if *seq\_type* is dynamic.

#### Parameters

**hyb\_record** ([HybRecord](#) (page 9)) -- *hyb\_record* for comparison.

**static\_count\_hyb\_record\_mismatches**(*hyb\_record*: [HybRecord](#) (page 9)) → [int<sup>298</sup>](#)

Count mismatches between *hyb\_record.seq* and *fold\_record.seq*.

#### Parameters

**hyb\_record** ([HybRecord](#) (page 9)) -- *hyb\_record* for comparison.

**dynamic\_count\_hyb\_record\_mismatches**(*hyb\_record*: [HybRecord](#) (page 9)) → [int<sup>299</sup>](#)

Count mismatches between *hyb\_record.seq* and *dynamic fold\_record.seq*.

#### Parameters

**hyb\_record** ([HybRecord](#) (page 9)) -- *hyb\_record* for comparison

**matches\_hyb\_record**(*hyb\_record*: [HybRecord](#) (page 9), *allowed\_mismatches*: *Optional*<sup>300</sup>[*int*<sup>301</sup>] = None) → *bool*<sup>302</sup>

Return True if self.seq and hyb\_record.seq mismatches are <= allowed\_mismatches.

#### Parameters

- **hyb\_record** ([HybRecord](#) (page 9)) -- hyb\_record to compare.
- **allowed\_mismatches** (*int*<sup>303</sup>, optional) -- Number of mismatches allowed for a match. If not provided, defaults to the option in [settings\['allowed\\_mismatches'\]](#) (page 25).

**ensure\_matches\_hyb\_record**(*hyb\_record*: [HybRecord](#) (page 9), *allowed\_mismatches*: *Optional*<sup>304</sup>[*int*<sup>305</sup>] = None) → None<sup>306</sup>

Ensure self.seq matches hyb\_record.seq, else raise an error.

#### Parameters

- **hyb\_record** ([HybRecord](#) (page 9)) -- hyb\_record to compare.
- **allowed\_mismatches** (*int*<sup>307</sup>, optional) -- Number of mismatches allowed for a match. If not provided, defaults to the option in [settings\['allowed\\_mismatches'\]](#) (page 25).

**classmethod from\_vienna\_lines**(*record\_lines*: *List*<sup>308</sup>[*str*<sup>309</sup>], *error\_mode*: *Optional*<sup>310</sup>[*Literal*<sup>311</sup>['raise', 'warn\_return', 'return']] = None, *seq\_type*: *Optional*<sup>312</sup>[*Literal*<sup>313</sup>['static', 'dynamic']] = None) → *Union*<sup>314</sup>[*Tuple*<sup>315</sup>[None<sup>316</sup>, *str*<sup>317</sup>], *Tuple*<sup>318</sup>[*Literal*<sup>319</sup>['NOFOLD'], *str*<sup>320</sup>], *Tuple*<sup>321</sup>[*Literal*<sup>322</sup>['NOENERGY'], *str*<sup>323</sup>], Self]

Construct instance from a list of 3 strings of vienna-format ([\[ViennaFormat\]](#)) lines.

See [Vienna File Format](#) (page 23) for more details.

#### Parameters

**record\_lines** (*list*<sup>324</sup> or *tuple*<sup>325</sup>) -- Iterable of 3 strings corresponding to lines of a vienna-format record.

**classmethod from\_vienna\_string**(*record\_string*: *str*<sup>326</sup>, *error\_mode*: *Optional*<sup>327</sup>[*Literal*<sup>328</sup>['raise', 'warn\_return', 'return']] = None, *seq\_type*: *Optional*<sup>329</sup>[*Literal*<sup>330</sup>['static', 'dynamic']] = None) → *Union*<sup>331</sup>[*Tuple*<sup>332</sup>[None<sup>333</sup>, *str*<sup>334</sup>], *Tuple*<sup>335</sup>[*Literal*<sup>336</sup>['NOFOLD'], *str*<sup>337</sup>], *Tuple*<sup>338</sup>[*Literal*<sup>339</sup>['NOENERGY'], *str*<sup>340</sup>], Self]

Construct instance from a string representing 3 vienna-format ([\[ViennaFormat\]](#)) lines.

See [Vienna File Format](#) (page 23) for more details.

#### Parameters

**record\_string** (*str*<sup>341</sup> or *tuple*<sup>342</sup>) -- 3-line string containing a vienna-format record

**classmethod from\_ct\_lines**(*record\_lines*: *List*<sup>343</sup>[*str*<sup>344</sup>], *error\_mode*: *Optional*<sup>345</sup>[*Literal*<sup>346</sup>['raise', 'warn\_return', 'return']] = None, *seq\_type*: *Optional*<sup>347</sup>[*Literal*<sup>348</sup>['static', 'dynamic']] = None) → *Union*<sup>349</sup>[*Tuple*<sup>350</sup>[None<sup>351</sup>, *str*<sup>352</sup>], *Tuple*<sup>353</sup>[*Literal*<sup>354</sup>['NOFOLD'], *str*<sup>355</sup>], *Tuple*<sup>356</sup>[*Literal*<sup>357</sup>['NOENERGY'], *str*<sup>358</sup>], Self]

Create a FoldRecord from a list of record lines in ".ct" format ([\[CTFormat\]](#)).

See [CT File Format](#) (page 23) for more details.

**Warning:** This method is in beta stage, and is not well-tested.

**Parameters**

**record\_lines** (*list*<sup>359</sup> or *tuple*<sup>360</sup>) -- list containing lines of ct record

**classmethod from\_ct\_string**(*record\_string*: *str*<sup>361</sup>, *error\_mode*: *Optional*<sup>362</sup>[*Literal*<sup>363</sup>['raise', 'warn\_return', 'return']] = *None*, *seq\_type*: *Optional*<sup>364</sup>[*Literal*<sup>365</sup>['static', 'dynamic']] = *None*) → *Union*<sup>366</sup>[*Tuple*<sup>367</sup>[*None*<sup>368</sup>, *str*<sup>369</sup>], *Tuple*<sup>370</sup>[*Literal*<sup>371</sup>['NOFOLD'], *str*<sup>372</sup>], *Tuple*<sup>373</sup>[*Literal*<sup>374</sup>['NOENERGY'], *str*<sup>375</sup>], *Self*]

Create a FoldRecord entry from a multi-line string from ".ct" format ([[CTFormat](#)]).

See *CT File Format* (page 23) for more details.

**Warning:** This method is in beta stage, and is not well-tested.

**Parameters**

**record\_string** (*str*<sup>376</sup>) -- String containing lines of ct record

267 <https://docs.python.org/3/library/stdtypes.html#str>  
 268 <https://docs.python.org/3/library/stdtypes.html#str>  
 269 <https://docs.python.org/3/library/stdtypes.html#str>  
 270 <https://docs.python.org/3/library/typing.html#typing.Optional>  
 271 <https://docs.python.org/3/library/typing.html#typing.Union>  
 272 <https://docs.python.org/3/library/functions.html#float>  
 273 <https://docs.python.org/3/library/functions.html#int>  
 274 <https://docs.python.org/3/library/stdtypes.html#str>  
 275 <https://docs.python.org/3/library/typing.html#typing.Optional>  
 276 <https://docs.python.org/3/library/typing.html#typing.Literal>  
 277 <https://www.tbi.univie.ac.at/RNA/>  
 278 <http://www.unafold.org/>  
 279 <https://docs.python.org/3/library/stdtypes.html#str>  
 280 <https://docs.python.org/3/library/stdtypes.html#str>  
 281 <https://docs.python.org/3/library/stdtypes.html#str>  
 282 <https://docs.python.org/3/library/stdtypes.html#str>  
 283 <https://docs.python.org/3/library/functions.html#float>  
 284 <https://docs.python.org/3/library/stdtypes.html#str>  
 285 <https://docs.python.org/3/library/stdtypes.html#str>  
 286 <https://docs.python.org/3/library/stdtypes.html#str>  
 287 <https://docs.python.org/3/library/stdtypes.html#str>  
 288 <https://docs.python.org/3/library/stdtypes.html#str>  
 289 <https://docs.python.org/3/library/stdtypes.html#str>  
 290 <https://docs.python.org/3/library/functions.html#bool>  
 291 <https://docs.python.org/3/library/typing.html#typing.List>  
 292 <https://docs.python.org/3/library/stdtypes.html#str>  
 293 <https://docs.python.org/3/library/functions.html#bool>  
 294 <https://docs.python.org/3/library/functions.html#bool>  
 295 <https://docs.python.org/3/library/stdtypes.html#str>  
 296 <https://docs.python.org/3/library/functions.html#bool>  
 297 <https://docs.python.org/3/library/functions.html#int>  
 298 <https://docs.python.org/3/library/functions.html#int>  
 299 <https://docs.python.org/3/library/functions.html#int>  
 300 <https://docs.python.org/3/library/typing.html#typing.Optional>  
 301 <https://docs.python.org/3/library/functions.html#int>  
 302 <https://docs.python.org/3/library/functions.html#bool>  
 303 <https://docs.python.org/3/library/functions.html#int>  
 304 <https://docs.python.org/3/library/typing.html#typing.Optional>  
 305 <https://docs.python.org/3/library/functions.html#int>  
 306 <https://docs.python.org/3/library/constants.html#None>  
 307 <https://docs.python.org/3/library/functions.html#int>  
 308 <https://docs.python.org/3/library/typing.html#typing.List>  
 309 <https://docs.python.org/3/library/stdtypes.html#str>  
 310 <https://docs.python.org/3/library/typing.html#typing.Optional>  
 311 <https://docs.python.org/3/library/typing.html#typing.Literal>  
 312 <https://docs.python.org/3/library/typing.html#typing.Optional>  
 313 <https://docs.python.org/3/library/typing.html#typing.Literal>  
 314 <https://docs.python.org/3/library/typing.html#typing.Union>  
 315 <https://docs.python.org/3/library/typing.html#typing.Tuple>  
 316 <https://docs.python.org/3/library/constants.html#None>  
 317 <https://docs.python.org/3/library/stdtypes.html#str>  
 318 <https://docs.python.org/3/library/typing.html#typing.Tuple>  
 319 <https://docs.python.org/3/library/typing.html#typing.Literal>  
 320 <https://docs.python.org/3/library/stdtypes.html#str>  
 321 <https://docs.python.org/3/library/typing.html#typing.Tuple>  
 322 <https://docs.python.org/3/library/typing.html#typing.Literal>  
 323 <https://docs.python.org/3/library/stdtypes.html#str>  
 324 <https://docs.python.org/3/library/stdtypes.html#list>  
 325 <https://docs.python.org/3/library/stdtypes.html#tuple>  
 326 <https://docs.python.org/3/library/stdtypes.html#str>  
 327 <https://docs.python.org/3/library/typing.html#typing.Optional>  
 328 <https://docs.python.org/3/library/typing.html#typing.Literal>  
 329 <https://docs.python.org/3/library/typing.html#typing.Optional>  
 330 <https://docs.python.org/3/library/typing.html#typing.Literal>  
 331 <https://docs.python.org/3/library/typing.html#typing.Union>  
 332 <https://docs.python.org/3/library/typing.html#typing.Tuple>  
 333 <https://docs.python.org/3/library/constants.html#None>  
 334 <https://docs.python.org/3/library/stdtypes.html#str>  
 335 <https://docs.python.org/3/library/typing.html#typing.Tuple>  
 336 <https://docs.python.org/3/library/typing.html#typing.Literal>  
 337 <https://docs.python.org/3/library/stdtypes.html#str>  
 338 <https://docs.python.org/3/library/typing.html#typing.Tuple>  
 339 <https://docs.python.org/3/library/typing.html#typing.Literal>  
 340 <https://docs.python.org/3/library/stdtypes.html#str>

## 2.1.4 ViennaFile Class

```
class hybkit.ViennaFile(*args: Any377, seq_type: Optional378[Literal379['static', 'dynamic']] = None,
                        error_mode: Optional380[Literal381['raise', 'warn_return', 'return']] = None,
                        from_file_like: bool382 = False, **kwargs: Any383)
```

Vienna file wrapper that returns vienna-format file lines as FoldRecord objects.

See *Vienna File Format* (page 23) for more information.

### Parameters

- **seq\_type** (str<sup>384</sup>, optional) -- Type of FoldRecord to return: static, or dynamic (if not provided, uses *FoldRecord.settings['seq\_type']* (page 25)).
- **error\_mode** (str<sup>385</sup>, optional) -- String representing the error mode. If None, defaults to the value set in *settings['error\_mode']* (page 25). Options: "raise": Raise an error when encountered and exit program; "warn\_return": Print a warning and return the error\_value; "return": Return the error value with no warnings.
- **from\_file\_like** (bool<sup>386</sup>, optional) -- If True, treat the first argument as a file-like object (such as io.StringIO or gzip.GzipFile) and the remaining positional arguments are ignored (Default False).
- **\*args** -- Passed to *open()* (page 29).
- **\*\*kwargs** -- Passed to *open()* (page 29).

### Variables

- **fh** (file) -- File handle for the file being wrapped.
- **foldrecord\_seq\_type** (str<sup>387</sup>) -- Type of FoldRecord to return (see Args)
- **error\_mode** (str<sup>388</sup>) -- Mode for error catching (see Args)

**Warning:** Occasionally fold files can be poorly-formatted. In that case, this iterator attempts error-catching but this is not always successful so verbose error modes are encouraged.

```
read_record(override_error_mode: Optional389[Literal390['raise', 'warn_return', 'return']] = None) →
    Union391[Tuple392[None393, str394], Tuple395[Literal396['NOFOLD'], str397],
    Tuple398[Literal399['NOENERGY'], str400], Self]
```

Read next three lines and return output as FoldRecord object.

### Parameters

**override\_error\_mode** (str<sup>401</sup>) -- Override the error\_mode set in the *ViennaFile* (page 29) object. See the *ViennaFile Constructor* (page 29) for more information on allowed error modes.

```
close() → None402
```

Close the file handle.

```
classmethod open(path: str403, *args: Any404, **kwargs: Any405) → Self
```

Open a path to a text file using *open()* (page 29) and return relevant file object.

Arguments match those of the Python3 built-in *open()* (page 29) function and are passed directly to it.

This method is provided as a convenience function for drop-in replacement of the built-in *open()* (page 29) function.

Specific keyword arguments are provided for fold-file-specific settings:

**Parameters**

- **path** ([str](#)<sup>406</sup>) -- Path to file to open.
- **seq\_type** ([str](#)<sup>407</sup>, optional) -- Type of *FoldRecord* to return: static, or dynamic (if not provided, uses *FoldRecord.settings['seq\_type']* (page 25)).
- **error\_mode** ([str](#)<sup>408</sup>, optional) -- String representing the error mode. If None, defaults to the value set in *settings['error\_mode']* (page 25). Options: "raise": Raise an error when encountered and exit program; "warn\_return": Print a warning and return the error\_value; "return": Return the error value with no warnings.
- **\*args** -- Passed directly to *open()* (page 29).
- **\*\*kwargs** -- Passed directly to *open()* (page 29).

**Returns**

*HybFile* (page 21) object.

**read\_records()** → [List](#)<sup>409</sup>[*FoldRecord* (page 23)]

Return list of all *FoldRecord* (page 23) objects for this file type.

**settings** = {}

Class-level settings. See *hybkit.settings.FoldFile\_settings\_info* (page 54) for descriptions.

**write\_fh**(\*args: [Any](#)<sup>410</sup>, \*\*kwargs: [Any](#)<sup>411</sup>) → [None](#)<sup>412</sup>

Write directly to the underlying file handle.

**write\_record**(write\_record: *FoldRecord* (page 23)) → [None](#)<sup>413</sup>

Write a *FoldRecord* object for this file type.

Unlike the file.write() method, this method will add a newline to the end of each written record line.

**Parameters**

**write\_record** (*FoldRecord* (page 23)) -- *FoldRecord* (page 23) objects to write.

**write\_records**(write\_records: [Iterable](#)<sup>414</sup>[*FoldRecord* (page 23)]) → [None](#)<sup>415</sup>

Write a sequence of *FoldRecord* objects for this file type.

Unlike the file.writelines() method, this method will add a newline to the end of each written record line.

**Parameters**

**write\_records** ([list](#)<sup>416</sup>) -- List of *FoldRecord* (page 23) objects to write.



## 2.1.5 CtFile Class

```
class hybkit.CtFile(*args: Any417, seq_type: Optional418[Literal419['static', 'dynamic']] = None, error_mode:
    Optional420[Literal421['raise', 'warn_return', 'return']] = None, from_file_like: bool422 =
    False, **kwargs: Any423)
```

Ct file wrapper that returns ".ct" file lines as FoldRecord objects.

See *CT File Format* (page 23) for more information.

**Warning:** This class is in beta stage, and is not well-tested.

### Parameters

- **seq\_type** (str<sup>424</sup>, optional) -- Type of FoldRecord to return: static, or dynamic (if not provided, uses `FoldRecord.settings['seq_type']` (page 25)).
- **error\_mode** (str<sup>425</sup>, optional) -- String representing the error mode. If None, defaults to the value set in `settings['error_mode']` (page 25). Options: "raise": Raise an error when encountered and exit program; "warn\_return": Print a warning and return the error\_value; "return": Return the error value with no warnings.
- **from\_file\_like** (bool<sup>426</sup>, optional) -- If True, treat the first argument as a file-like

377 <https://docs.python.org/3/library/typing.html#typing.Any>  
378 <https://docs.python.org/3/library/typing.html#typing.Optional>  
379 <https://docs.python.org/3/library/typing.html#typing.Literal>  
380 <https://docs.python.org/3/library/typing.html#typing.Optional>  
381 <https://docs.python.org/3/library/typing.html#typing.Literal>  
382 <https://docs.python.org/3/library/functions.html#bool>  
383 <https://docs.python.org/3/library/typing.html#typing.Any>  
384 <https://docs.python.org/3/library/stdtypes.html#str>  
385 <https://docs.python.org/3/library/stdtypes.html#str>  
386 <https://docs.python.org/3/library/functions.html#bool>  
387 <https://docs.python.org/3/library/stdtypes.html#str>  
388 <https://docs.python.org/3/library/stdtypes.html#str>  
389 <https://docs.python.org/3/library/typing.html#typing.Optional>  
390 <https://docs.python.org/3/library/typing.html#typing.Literal>  
391 <https://docs.python.org/3/library/typing.html#typing.Union>  
392 <https://docs.python.org/3/library/typing.html#typing.Tuple>  
393 <https://docs.python.org/3/library/constants.html#None>  
394 <https://docs.python.org/3/library/stdtypes.html#str>  
395 <https://docs.python.org/3/library/typing.html#typing.Tuple>  
396 <https://docs.python.org/3/library/typing.html#typing.Literal>  
397 <https://docs.python.org/3/library/stdtypes.html#str>  
398 <https://docs.python.org/3/library/typing.html#typing.Tuple>  
399 <https://docs.python.org/3/library/typing.html#typing.Literal>  
400 <https://docs.python.org/3/library/stdtypes.html#str>  
401 <https://docs.python.org/3/library/stdtypes.html#str>  
402 <https://docs.python.org/3/library/constants.html#None>  
403 <https://docs.python.org/3/library/stdtypes.html#str>  
404 <https://docs.python.org/3/library/typing.html#typing.Any>  
405 <https://docs.python.org/3/library/typing.html#typing.Any>  
406 <https://docs.python.org/3/library/stdtypes.html#str>  
407 <https://docs.python.org/3/library/stdtypes.html#str>  
408 <https://docs.python.org/3/library/stdtypes.html#str>  
409 <https://docs.python.org/3/library/typing.html#typing.List>  
410 <https://docs.python.org/3/library/typing.html#typing.Any>  
411 <https://docs.python.org/3/library/typing.html#typing.Any>  
412 <https://docs.python.org/3/library/constants.html#None>  
413 <https://docs.python.org/3/library/constants.html#None>  
414 <https://docs.python.org/3/library/typing.html#typing.Iterable>  
415 <https://docs.python.org/3/library/constants.html#None>  
416 <https://docs.python.org/3/library/stdtypes.html#list>

object (such as `io.StringIO` or `gzip.GzipFile`) and the remaining positional arguments are ignored (Default `False`).

- **\*args** -- Passed to `open()` (page 32).
- **\*\*kwargs** -- Passed to `open()` (page 32).

#### Variables

- **fh** (file) -- File handle for the file being wrapped.
- **foldrecord\_seq\_type** (`str`<sup>427</sup>) -- Type of `FoldRecord` to return (see Args)
- **error\_mode** (`str`<sup>428</sup>) -- Mode for error catching (see Args)

**Warning:** Occasionally fold files can be poorly-formatted. In that case, this iterator attempts error-catching but this is not always successful so verbose error modes are encouraged.

**read\_record()** → `Union`<sup>429</sup>[`Tuple`<sup>430</sup>[`None`<sup>431</sup>, `str`<sup>432</sup>], `Tuple`<sup>433</sup>[`Literal`<sup>434</sup>['NOFOLD'], `str`<sup>435</sup>], `Tuple`<sup>436</sup>[`Literal`<sup>437</sup>['NOENERGY'], `str`<sup>438</sup>], `Self`]

Return the next CT record as a `FoldRecord` (page 23) object.

Call `next(self.fh)` to return the first line of the next entry. Determine the expected number of following lines in the entry, and read that number of lines further. Return lines as a `FoldRecord` object.

**write\_record = None**

CtFile Record Writing Not Implemented

**write\_records = None**

CtFile Record Writing Not Implemented

**close()** → `None`<sup>439</sup>

Close the file handle.

**classmethod open**(*path*: `str`<sup>440</sup>, \**args*: `Any`<sup>441</sup>, \*\**kwargs*: `Any`<sup>442</sup>) → `Self`

Open a path to a text file using `open()` (page 32) and return relevant file object.

Arguments match those of the Python3 built-in `open()` (page 32) function and are passed directly to it.

This method is provided as a convenience function for drop-in replacement of the built-in `open()` (page 32) function.

Specific keyword arguments are provided for fold-file-specific settings:

#### Parameters

- **path** (`str`<sup>443</sup>) -- Path to file to open.
- **seq\_type** (`str`<sup>444</sup>, optional) -- Type of `FoldRecord` to return: `static`, or `dynamic` (if not provided, uses `FoldRecord.settings['seq_type']` (page 25)).
- **error\_mode** (`str`<sup>445</sup>, optional) -- String representing the error mode. If `None`, defaults to the value set in `settings['error_mode']` (page 25). Options: `"raise"`: Raise an error when encountered and exit program; `"warn_return"`: Print a warning and return the error\_value; `"return"`: Return the error value with no warnings.
- **\*args** -- Passed directly to `open()` (page 32).
- **\*\*kwargs** -- Passed directly to `open()` (page 32).

#### Returns

`HybFile` (page 21) object.

**read\_records()** → [List](#)<sup>446</sup>[[FoldRecord](#) (page 23)]

Return list of all [FoldRecord](#) (page 23) objects for this file type.

**settings** = {}

Class-level settings. See [hybkit.settings.FoldFile\\_settings\\_info](#) (page 54) for descriptions.

**write\_fh**(\*args: [Any](#)<sup>447</sup>, \*\*kwargs: [Any](#)<sup>448</sup>) → [None](#)<sup>449</sup>

Write directly to the underlying file handle.

## 2.1.6 HybFoldIter Class

**class** [hybkit.HybFoldIter](#)([hybfile\\_handle](#): [HybFile](#) (page 21), [foldfile\\_handle](#): [FoldFile](#), [combine](#): [bool](#)<sup>450</sup> = [False](#), [iter\\_error\\_mode](#): [Optional](#)<sup>451</sup>[[Literal](#)<sup>452</sup>['raise', 'warn\_return', 'warn\_skip', 'skip', 'return']] = [None](#))

Iterator for simultaneous iteration over a [HybFile](#) (page 21) and [FoldFile](#) object.

This class provides an iterator to iterate through a [HybFile](#) (page 21) and one of a [ViennaFile](#) (page 29), or [CtFile](#) (page 31) simultaneously to return a [HybRecord](#) (page 9) and [FoldRecord](#) (page 23).

Basic error checking / catching is performed based on the value of the [~settings\['error\\_mode'\]](#) (page 34) setting.

### Parameters

- **hybfile\_handle** ([HybFile](#) (page 21)) -- [HybFile](#) object for iteration
- **foldfile\_handle** ([ViennaFile](#) (page 29) or [CtFile](#) (page 31)) -- [ViennaFile](#) (page 29) or [CtFile](#) (page 31) object for iteration

<sup>417</sup> <https://docs.python.org/3/library/typing.html#typing.Any>  
<sup>418</sup> <https://docs.python.org/3/library/typing.html#typing.Optional>  
<sup>419</sup> <https://docs.python.org/3/library/typing.html#typing.Literal>  
<sup>420</sup> <https://docs.python.org/3/library/typing.html#typing.Optional>  
<sup>421</sup> <https://docs.python.org/3/library/typing.html#typing.Literal>  
<sup>422</sup> <https://docs.python.org/3/library/functions.html#bool>  
<sup>423</sup> <https://docs.python.org/3/library/typing.html#typing.Any>  
<sup>424</sup> <https://docs.python.org/3/library/stdtypes.html#str>  
<sup>425</sup> <https://docs.python.org/3/library/stdtypes.html#str>  
<sup>426</sup> <https://docs.python.org/3/library/functions.html#bool>  
<sup>427</sup> <https://docs.python.org/3/library/stdtypes.html#str>  
<sup>428</sup> <https://docs.python.org/3/library/stdtypes.html#str>  
<sup>429</sup> <https://docs.python.org/3/library/typing.html#typing.Union>  
<sup>430</sup> <https://docs.python.org/3/library/typing.html#typing.Tuple>  
<sup>431</sup> <https://docs.python.org/3/library/constants.html#None>  
<sup>432</sup> <https://docs.python.org/3/library/stdtypes.html#str>  
<sup>433</sup> <https://docs.python.org/3/library/typing.html#typing.Tuple>  
<sup>434</sup> <https://docs.python.org/3/library/typing.html#typing.Literal>  
<sup>435</sup> <https://docs.python.org/3/library/stdtypes.html#str>  
<sup>436</sup> <https://docs.python.org/3/library/typing.html#typing.Tuple>  
<sup>437</sup> <https://docs.python.org/3/library/typing.html#typing.Literal>  
<sup>438</sup> <https://docs.python.org/3/library/stdtypes.html#str>  
<sup>439</sup> <https://docs.python.org/3/library/constants.html#None>  
<sup>440</sup> <https://docs.python.org/3/library/stdtypes.html#str>  
<sup>441</sup> <https://docs.python.org/3/library/typing.html#typing.Any>  
<sup>442</sup> <https://docs.python.org/3/library/typing.html#typing.Any>  
<sup>443</sup> <https://docs.python.org/3/library/stdtypes.html#str>  
<sup>444</sup> <https://docs.python.org/3/library/stdtypes.html#str>  
<sup>445</sup> <https://docs.python.org/3/library/stdtypes.html#str>  
<sup>446</sup> <https://docs.python.org/3/library/typing.html#typing.List>  
<sup>447</sup> <https://docs.python.org/3/library/typing.html#typing.Any>  
<sup>448</sup> <https://docs.python.org/3/library/typing.html#typing.Any>  
<sup>449</sup> <https://docs.python.org/3/library/constants.html#None>

- **combine** ([bool](#)<sup>453</sup>, optional) -- Use `HybRecord.set_fold_record(FoldRecord)` and return only the `HybRecord`.
- **iter\_error\_mode** ([str](#)<sup>454</sup>, optional) -- Error mode to use for reading `FoldRecord` (page 23) objects. If not set, defaults to the value in `settings['iter_error_mode']` (page 34).

#### Returns

([HybRecord](#) (page 9), [FoldRecord](#) (page 23))

```
settings = {'error_checks': ['hybrecord_indel', 'foldrecord_nofold',
                             'max_mismatch', 'energy_mismatch'], 'iter_error_mode': 'warn_skip',
            'max_sequential_skips': 100}
```

Class-level settings. See `settings.HybFoldIter_settings_info` (page 54) for descriptions.

**report()** → [List](#)<sup>455</sup>[[str](#)<sup>456</sup>]

Return a report of information from iteration.

**print\_report()** → [None](#)<sup>457</sup>

Print a report of information from iteration.

## 2.2 hybkit.type\_finder

hybkit TypeFinder Class.

This module contains the `TypeFinder` class to work with `HybRecord` to parse sequence identifiers to identify sequence type.

**class** `hybkit.type_finder.TypeFinder`

Class for parsing identifiers to identify sequence 'type'.

Designed to be used by the [hybkit.HybRecord](#) (page 9)

#### Variables

**params** ([dict](#)<sup>458</sup>) -- Stored parameters for string parsing, where applicable.

**find\_with\_params** = `None`

Placeholder for storing active method, set with `set_method()` (page 35) (see `set_method()` (page 35) for details).

**params** = `None`

Placeholder for parameters for active method, set with `set_method()` (page 35) (see `set_method()` (page 35) for details).

**default\_method** = `'hybformat'`

Default method assigned using `check_set_method()` (page 35)

**methods** = {'hybformat': 'method\_hybformat', 'id\_map': 'method\_id\_map',  
 'string\_match': 'method\_string\_match'}

<sup>450</sup> <https://docs.python.org/3/library/functions.html#bool>

<sup>451</sup> <https://docs.python.org/3/library/typing.html#typing.Optional>

<sup>452</sup> <https://docs.python.org/3/library/typing.html#typing.Literal>

<sup>453</sup> <https://docs.python.org/3/library/functions.html#bool>

<sup>454</sup> <https://docs.python.org/3/library/stdtypes.html#str>

<sup>455</sup> <https://docs.python.org/3/library/typing.html#typing.List>

<sup>456</sup> <https://docs.python.org/3/library/stdtypes.html#str>

<sup>457</sup> <https://docs.python.org/3/library/constants.html#None>

Dict of provided methods available to assign segment types

'hybformat'	<a href="#">method_hybformat()</a> (page 36)
'string_match'	<a href="#">method_string_match()</a> (page 36)
'id_map'	<a href="#">method_id_map()</a> (page 37)

```
param_methods = {'hybformat': None, 'id_map': 'make_id_map_params',
'string_match': 'make_string_match_params'}
```

Dict of param generation methods for type finding methods

'hybformat'	'N/A'
'string_match'	<a href="#">make_string_match_params()</a> (page 37)
'id_map'	<a href="#">make_id_map_params()</a> (page 38)

```
param_methods_needs_file = {'hybformat': False, 'id_map': True, 'string_match':
True}
```

Dict of whether parameter generation methods need an input file

'hybformat'	False <sup>459</sup>
'string_match'	True <sup>460</sup>
'id_map'	True <sup>461</sup>

```
classmethod set_method(method: str462, params: Optional463[Dict464[str465, Any466]] = None) →
None467
```

Select method to use when finding types.

Available methods are listed in [methods](#) (page 34).

#### Parameters

- **method** (str<sup>468</sup>) -- Method option from [methods](#) (page 34) to set for use as [find\(\)](#) (page 35).
- **params** (dict<sup>469</sup>, optional) -- Dict object of parameters to use by set method.

```
classmethod method_is_set() → bool470
```

Return whether a TypeFinder method has been set.

Methods should be set with [set\\_method\(\)](#) (page 35).

#### Returns

True if a method has been set, False otherwise.

#### Return type

bool<sup>471</sup>

```
classmethod check_set_method() → None472
```

If no TypeFinder method set, set as [default\\_method](#) (page 34).

```
classmethod find(seg_props: Dict473[str474, Union475[float476, int477, str478]]) → Optional479[str480]
```

Find type of segment using `TypeFinder.find_custom_method()`.

If a TypeFinder method has been set with [set\\_method\(\)](#) (page 35). use the current parameters set in [params](#) (page 34) to find the type of the provided segment by calling:

```
seg_type = :meth:`TypeFinder.find_custom_method`(seg_props, :attr`TypeFinder.  
→params`)
```

**Parameters**

**seg\_props** (*dict*<sup>481</sup>) -- seg\_props from *hybkit.HybRecord* (page 9)

**Returns**

Type of the provided segment, or None if a type cannot be identified.

**Return type**

*str*<sup>482</sup>

**classmethod set\_custom\_method**(*method*: *Callable*<sup>483</sup>, *params*: *Optional*<sup>484</sup>[*dict*<sup>485</sup>] = None) → *None*<sup>486</sup>

Set the method for use to find seg types.

This method is for providing a custom function. To use the included functions, use *set\_method()* (page 35). Custom functions provided must have the signature:

```
seg_type = custom_method(self, seg_props, params)
```

This function should return the string of the assigned segment type if found, or a None object if the type cannot be found. It can also take a dictionary in the "params" argument that specifies additional or dynamic search properties, as desired.

**Parameters**

- **method** (*method*) -- Method to set for use.
- **params** (*dict*<sup>487</sup>, *optional*) -- dict of custom parameters to set for use.

**static method\_hybformat**(*seg\_props*: *Dict*<sup>488</sup>[*str*<sup>489</sup>, *Union*<sup>490</sup>[*float*<sup>491</sup>, *int*<sup>492</sup>, *str*<sup>493</sup>]], *params*: *Optional*<sup>494</sup>[*dict*<sup>495</sup>] = None) → *Optional*<sup>496</sup>[*str*<sup>497</sup>]

Return the type of the provided segment, or None if segment cannot be identified.

This method works with sequence / alignment mapping identifiers in the format of the reference database provided by the Hyb Software Package, specifically identifiers of the format:

```
<gene_id>_<transcript_id>_<gene_name>_<seg_type>
```

This method returns the last component of the identifier, split by "\_", as the identified sequence type. (returns *None*<sup>498</sup> if the segment identifier does not contain "\_").

**Example**

```
"MIMAT00000076_MirBase_miR-21_microRNA" ---> "microRNA".
```

**Parameters**

- **seg\_props** (*dict*<sup>499</sup>) -- seg\_props from *hybkit.HybRecord* (page 9)
- **params** (*dict*<sup>500</sup>, *optional*) -- Unused in this method.

**static method\_string\_match**(*seg\_props*: *Dict*<sup>501</sup>[*str*<sup>502</sup>, *Union*<sup>503</sup>[*float*<sup>504</sup>, *int*<sup>505</sup>, *str*<sup>506</sup>]], *params*: *Optional*<sup>507</sup>[*dict*<sup>508</sup>] = None) → *Optional*<sup>509</sup>[*str*<sup>510</sup>]

Return the type of the provided segment, or None if unidentified.

This method attempts to find a string matching a specific pattern within the identifier of the aligned segment. Search options include "startswith", "contains", "endswith", and "matches", and returns the first type matching the criteria. The required params dict should contain a key for each desired search type, with a list of 2-tuples for each search-string with assigned-type.

### Example

```
params = {'endswith': [('_miR', 'microRNA'),
                      ('_trans', 'mRNA') ]}
```

This dict can be generated with the associated `make_string_match_params()` (page 37) method and an associated csv legend file with format:

```
#comment line
#search_type,search_string,seg_type
endswith,_miR,microRNA
endswith,_trans,mRNA
```

### Parameters

- **seg\_props** (*dict*<sup>511</sup>) -- *HybRecord* (page 9) segment properties dict to evaluate.
- **params** (*dict*<sup>512</sup>, *optional*) -- Dict with search parameters as described above.

**static** `make_string_match_params(legend_file: str513) → dict514`

Read csv and return a dict of search parameters for `method_string_match()` (page 36).

The my\_legend.csv file should have the format:

```
#comment line
#search_type,search_string,seg_type
endswith,_miR,microRNA
endswith,_trans,mRNA
```

Search\_type options include "startswith", "contains", "endswith", and "matches" The produced dict object contains a key for each search type, with a list of 2-tuples for each search-string and associated segment-type.

For example:

```
{'endswith': [('_miR', 'microRNA'),
              ('_trans', 'mRNA') ]}
```

**static** `method_id_map(seg_props: Dict515[str516, Union517[float518, int519, str520]], params: Optional521[dict522] = None) → Optional523[str524]`

Return the type of the provided segment or None if it cannot be identified.

This method checks to see if the identifier of the segment is present in the params dict. params should be formatted as a dict with keys as sequence identifier names, and the corresponding type as the respective values.

### Example

```
params = {'MIMAT0000076_MirBase_miR-21_microRNA': 'microRNA',
          'ENSG00000XXXXXX_NR003287-2_RN28S1_rRNA': 'rRNA'}
```

This dict can be generated with the associated `make_id_map_params()` (page 38) method.

#### Parameters

- **seg\_props** (`dict`<sup>525</sup>) -- *HybRecord* (page 9) segment properties dict to evaluate.
- **params** (`dict`<sup>526</sup>) -- Dict of mapping of sequence identifiers to sequence types.

#### Returns

Identified sequence type, or None if it cannot be found.

#### Return type

`str`<sup>527</sup>

**static** `make_id_map_params(mapped_id_files: List528[str529])` → `dict`<sup>530</sup>

Read file(s) into a mapping of sequence identifiers.

This method reads one or more files into a dict for use with the `method_id_map()` (page 37) method. The method requires passing a file path (or list/tuple of file paths) of `mapped_id_files`. Files listed in the `mapped_id_files` argument should have the format:

```
#comment line
#seg_id,seg_type
segA_unique_id,segA_type
segB_unique_id,segB_type
```

#### Parameters

**mapped\_id\_files** (`str`<sup>531</sup>, `list`<sup>532</sup>, or `tuple`<sup>533</sup>) -- Iterable object containing strings of paths to files containing id/type mapping information.



```

458 https://docs.python.org/3/library/stdtypes.html#dict
459 https://docs.python.org/3/library/constants.html#False
460 https://docs.python.org/3/library/constants.html#True
461 https://docs.python.org/3/library/constants.html#True
462 https://docs.python.org/3/library/stdtypes.html#str
463 https://docs.python.org/3/library/typing.html#typing.Optional
464 https://docs.python.org/3/library/typing.html#typing.Dict
465 https://docs.python.org/3/library/stdtypes.html#str
466 https://docs.python.org/3/library/typing.html#typing.Any
467 https://docs.python.org/3/library/constants.html#None
468 https://docs.python.org/3/library/stdtypes.html#str
469 https://docs.python.org/3/library/stdtypes.html#dict
470 https://docs.python.org/3/library/functions.html#bool
471 https://docs.python.org/3/library/functions.html#bool
472 https://docs.python.org/3/library/constants.html#None
473 https://docs.python.org/3/library/typing.html#typing.Dict
474 https://docs.python.org/3/library/stdtypes.html#str
475 https://docs.python.org/3/library/typing.html#typing.Union
476 https://docs.python.org/3/library/functions.html#float
477 https://docs.python.org/3/library/functions.html#int
478 https://docs.python.org/3/library/stdtypes.html#str
479 https://docs.python.org/3/library/typing.html#typing.Optional
480 https://docs.python.org/3/library/stdtypes.html#str
481 https://docs.python.org/3/library/stdtypes.html#dict
482 https://docs.python.org/3/library/stdtypes.html#str
483 https://docs.python.org/3/library/typing.html#typing.Callable
484 https://docs.python.org/3/library/typing.html#typing.Optional
485 https://docs.python.org/3/library/stdtypes.html#dict
486 https://docs.python.org/3/library/constants.html#None
487 https://docs.python.org/3/library/stdtypes.html#dict
488 https://docs.python.org/3/library/typing.html#typing.Dict
489 https://docs.python.org/3/library/stdtypes.html#str
490 https://docs.python.org/3/library/typing.html#typing.Union
491 https://docs.python.org/3/library/functions.html#float
492 https://docs.python.org/3/library/functions.html#int
493 https://docs.python.org/3/library/stdtypes.html#str
494 https://docs.python.org/3/library/typing.html#typing.Optional
495 https://docs.python.org/3/library/stdtypes.html#dict
496 https://docs.python.org/3/library/typing.html#typing.Optional
497 https://docs.python.org/3/library/stdtypes.html#str
498 https://docs.python.org/3/library/constants.html#None
499 https://docs.python.org/3/library/stdtypes.html#dict
500 https://docs.python.org/3/library/stdtypes.html#dict
501 https://docs.python.org/3/library/typing.html#typing.Dict
502 https://docs.python.org/3/library/stdtypes.html#str
503 https://docs.python.org/3/library/typing.html#typing.Union
504 https://docs.python.org/3/library/functions.html#float
505 https://docs.python.org/3/library/functions.html#int
506 https://docs.python.org/3/library/stdtypes.html#str
507 https://docs.python.org/3/library/typing.html#typing.Optional
508 https://docs.python.org/3/library/stdtypes.html#dict
509 https://docs.python.org/3/library/typing.html#typing.Optional
510 https://docs.python.org/3/library/stdtypes.html#str
511 https://docs.python.org/3/library/stdtypes.html#dict
512 https://docs.python.org/3/library/stdtypes.html#dict
513 https://docs.python.org/3/library/stdtypes.html#str
514 https://docs.python.org/3/library/stdtypes.html#dict
515 https://docs.python.org/3/library/typing.html#typing.Dict
516 https://docs.python.org/3/library/stdtypes.html#str
517 https://docs.python.org/3/library/typing.html#typing.Union
518 https://docs.python.org/3/library/functions.html#float
519 https://docs.python.org/3/library/functions.html#int
520 https://docs.python.org/3/library/stdtypes.html#str
521 https://docs.python.org/3/library/typing.html#typing.Optional
522 https://docs.python.org/3/library/stdtypes.html#dict
523 https://docs.python.org/3/library/typing.html#typing.Optional
524 https://docs.python.org/3/library/stdtypes.html#str
525 https://docs.python.org/3/library/stdtypes.html#dict
526 https://docs.python.org/3/library/stdtypes.html#dict
527 https://docs.python.org/3/library/stdtypes.html#str
528 https://docs.python.org/3/library/typing.html#typing.List
529 https://docs.python.org/3/library/stdtypes.html#str
530 https://docs.python.org/3/library/stdtypes.html#dict
531 https://docs.python.org/3/library/stdtypes.html#str

```

## 2.3 hybkit.analysis

Functions for analysis of HybRecord and FoldRecord objects.

### 2.3.1 Analysis

```
class hybkit.analysis.Analysis(analysis_types: Union534[Literal535['energy', 'type', 'mirna', 'target'],
List536[Literal537['energy', 'type', 'mirna', 'target']]], name:
Optional538[str539] = None, quant_mode: Optional540[Literal541['single',
'reads', 'records']] = None)
```

Class for analysis of hybkit HybRecord and FoldRecord objects.

This class contains multiple conceptual analyses for HybRecord/FoldRecord Data:

*Energy* (page 40): Analysis of values of predicted intra-hybrid folding energy

*Type* (page 40): Analysis of segment types

*miRNA* (page 41): Analysis of miRNA segments distributions

*Target* (page 41): Analysis of mirna target segment names and types

*Fold* (page 42): Analysis of folding data included in the analyzed hyb\_records.

This class used by selecting the desired analysis types on object initialization. Analyses are performed either by using either the `add_record()` or the `add_all_records()` methods. The results of the analysis are then available through the `get_all_results()` (page 43), `get_analysis_results()` (page 43), `get_specific_result()` (page 43), and `plot_analysis_results()` (page 44) methods, which can return (or plot) the results of all analyses or of a specific subset of analyses.

Details for each respective analysis are provided here:

#### Energy Analysis:

This analysis evaluates the energy of each *HybRecord* (page 9) object and provides a binned-histogram of all energy values represented.

#### Output Results:

`energy_analysis_count` (`int`<sup>542</sup>): Count of energy values evaluated

`has_energy_val` (`int`<sup>543</sup>): Count of hyb\_records with an energy value

`no_energy_val` (`int`<sup>544</sup>): Count of hyb\_records without an energy value

`energy_min` (`float`<sup>545</sup>): Minimum energy value

`energy_max` (`float`<sup>546</sup>): Maximum energy value

`energy_mean` (`float`<sup>547</sup>): Mean energy value

`energy_std` (`float`<sup>548</sup>): Standard deviation of energy values

`binned_energy_vals` (`Counter`<sup>549</sup>): Counter with integer keys of energy values from `energy_min` to `energy_max` storing the count of any hyb\_records with energy values that fall within that range (rounded to the next highest integer (e.g. -12.5 -> -12)).

#### Type Analysis:

This analysis evaluates the counts of each type of segment included in the *HybRecord* (page 9) objects. The types of segments are determined by the `seg1_type` (page 6) and `seg2_type` (page 6) flags, which are set by the `hybkit.HybRecord.eval_types()` (page 14) method.

Requirements:

`seg1_type` (page 6) and `seg2_type` (page 6) flags must be set for each HybRecord, (can be done by `hybkit.HybRecord.eval_types()` (page 14)).

**Output Results:**

`types_analysis_count` ([int<sup>550</sup>](#)): Count of hybrid types analyzed  
`hybrid_types` ([Counter<sup>551</sup>](#)): Counter containing annotated types of seg1 and seg (in original 5P (5-Prime) / 3P (3-Prime) order)  
`reordered_hybrid_types` ([Counter<sup>552</sup>](#)): Counter containing annotated types of seg1 and seg2. This is provided in "sorted" order, where types are sorted alphabetically (independent of 5P / 3P position).  
`mirna_hybrid_types` ([Counter<sup>553</sup>](#)): Counter containing annotated types of seg1 and seg2. This is provided in "miRNA-prime" order, where a miRNA type is always listed before other types, and then remaining types are sorted alphabetically (independent of 5P / 3P position).  
`seg1_types` ([Counter<sup>554</sup>](#)): Counter containing annotated type of segment in position seg1  
`seg2_types` ([Counter<sup>555</sup>](#)): Counter containing annotated type of segment in position seg2  
`all_seg_types` ([Counter<sup>556</sup>](#)): Counter containing position-independent annotated types

**miRNA Analysis:**

Analysis of miRNA segments in hybrids.

The `mirna_analysis` provides an analysis of what miRNA types are present in the `hyb` records. If a miRNA dimer is present in a hybrid, this is counted in `mirna_dimers`. If a single miRNA is present in a hybrid, this is counted in `mirnas_5p` or `mirnas_3p` depending on the miRNA location.

**Requirements:**

`mirna_seg` (page 6) flag must be set for each `HybRecord` (can be done by `hybkit.HybRecord.eval_mirna()` (page 15)).

**Output Results:**

`mirna_analysis_count` ([int<sup>557</sup>](#)): Count of miRNA hybrids analyzed  
`mirnas_5p` ([int<sup>558</sup>](#)): Count of 5P miRNAs detected  
`mirnas_3p` ([int<sup>559</sup>](#)): Count of 3P miRNAs detected  
`mirna_dimers` ([int<sup>560</sup>](#)): Count of miRNA dimers (5P + 3P) detected  
`non_mirna` ([int<sup>561</sup>](#)): Count of non-miRNA hybrids detected  
`has_mirna` ([int<sup>562</sup>](#)): Hybrids with 5P, 3P, or both as miRNA

**Target Analysis:**

Analysis of targets in miRNA-containing hybrids.

The target analysis provides an analysis of what annotated sequences and sequence types are targeted by any miRNA within the `hyb` records. If a miRNA is not present in a hybrid, the hybrid is not included in the analysis. If a miRNA dimer is present in a hybrid, the 5P miRNA is used for the analysis, and the 3P miRNA is considered the "target."

**Requirements:**

`mirna_seg` (page 6) flag must be set for each `HybRecord` (can be done by `hybkit.HybRecord.eval_mirna()` (page 15)).

**Output Results:**

`target_analysis_count` ([int<sup>563</sup>](#)): Count of hybrids analyzed  
`target_evals` ([int<sup>564</sup>](#)): Count of target evaluations performed  
`target_names` ([Counter<sup>565</sup>](#)): Counter containing names of miRNA targets detected.

`target_types` ([Counter](#)<sup>566</sup>): Counter containing types of miRNA targets detected.

#### Fold Analysis:

This analysis evaluates the predicted binding of miRNA within hyb records that contain a miRNA and have an associated [FoldRecord](#) (page 23) object as the attribute `fold_record`. This includes an analysis and plotting of the predicted binding by position among the provided miRNA.

#### Requirements:

The [mirna\\_seg](#) (page 6) flag must be set for each HybRecord (can be done by [hybkit.HybRecord.eval\\_mirna\(\)](#) (page 15)).

The [fold\\_record](#) (page 10) attribute must be set for each HybRecord with a corresponding [FoldRecord](#) (page 23) object. This can be done using the [hybkit.HybRecord.set\\_fold\\_record\(\)](#) (page 15) method.

#### Output Results:

`fold_analysis_count` ([int](#)<sup>567</sup>): Count of miRNA fold predictions analyzed

`folds_recorded` ([int](#)<sup>568</sup>): Count of fold predictions with a mirna fold

`mirna_nt_fold_counts` ([Counter](#)<sup>569</sup>): Counter with keys of miRNA position index and values of number of miRNAs with a predicted bound state at that index.

`mirna_nt_fold_props` ([Counter](#)<sup>570</sup>): Counter with keys of miRNA position index and values of proportion (0.0 - 1.0) of miRNAs with a predicted bound state at that index.

`fold_match_counts` ([Counter](#)<sup>571</sup>): Counter with keys of count of predicted matches between miRNA and target with values of count of miRNAs with that number of predicted matches.

#### Parameters

- `analysis_types` ([str](#)<sup>572</sup> or [list](#)<sup>573</sup> of [str](#)<sup>574</sup>) -- Analysis types to perform
- `name` ([str](#)<sup>575</sup>, optional) -- Name of the analysis
- `quant_mode` ([str](#)<sup>576</sup>, optional) -- Mode to use for record quantification. Options are "single": One count per record; "reads": If "read\_count" flag is set, count all reads in record (else count 1); "records": if the "record\_count" flag is set, count all individual records within combined record (else count 1). If not provided, defaults to the value in `Analysis.settings['quant_mode']`.

#### Variables

- `name` ([str](#)<sup>577</sup>) -- Name of the analysis
- `analysis_types` ([list](#)<sup>578</sup> of [str](#)<sup>579</sup>) -- List of analysis types to perform
- `quant_mode` ([str](#)<sup>580</sup>) -- Mode to use for record quantification.

```
settings = {'out_delim': ',', 'quant_mode': 'single'}
```

Class-level settings. See [hybkit.settings.Analysis\\_settings](#) (page 56) for descriptions.

```
analysis_options = ['energy', 'type', 'mirna', 'target', 'fold']
```

```
add_hyb_record(hyb_record: HybRecord (page 9)) → None581
```

Add a HybRecord object to the analysis.

#### Parameters

`hyb_record` ([HybRecord](#) (page 9)) -- HybRecord object to be added to the analysis.

**add\_hyb\_records**(*hyb\_records*: *List*<sup>582</sup>[*HybRecord* (page 9)], *eval\_types*: *bool*<sup>583</sup> = *False*, *eval\_mirna*: *bool*<sup>584</sup> = *False*) → *None*<sup>585</sup>

Add a list of *HybRecord* objects to the analysis.

#### Parameters

- **hyb\_records** (*HybFile* (page 21) or *list*<sup>586</sup> of *HybRecord* (page 9)) -- *HybFile* to iterate over, or iterable of *HybRecord* objects to be added to the analysis.
- **eval\_types** (*bool*<sup>587</sup>) -- If *True*, evaluate the hybrid type of the *HybRecord* before adding it to the analysis using *hybkit.HybRecord.eval\_types()* (page 14).
- **eval\_mirna** (*bool*<sup>588</sup>) -- If *True*, evaluate the miRNA segment of the *HybRecord* before adding it to the analysis using *hybkit.HybRecord.eval\_mirna()* (page 15).

**get\_all\_results**() → *dict*<sup>589</sup>

Return a dictionary with all results for all active analyses.

See *Analyses* (page 40) for details on the results for each analysis type.

#### Returns

**Dictionary with keys of analysis type and values of**  
dictionaries with results for that analysis type.

**Return type**  
*dict*<sup>590</sup>

**get\_analysis\_results**(*analysis*: *Literal*<sup>591</sup>['energy', 'type', 'mirna', 'target']) → *Dict*<sup>592</sup>

Return a dictionary with all results for a specific analysis.

See *Analyses* (page 40) for details on the results for each analysis type.

#### Parameters

**analysis** (*str*<sup>593</sup>) -- Analysis type to return results for.

#### Returns

**Dictionary with results for the specified analysis type.**  
see :ref:Analyses for details.

**Return type**  
*dict*<sup>594</sup>

**get\_specific\_result**(*result\_key*: *str*<sup>595</sup>) → *Any*<sup>596</sup>

Get a specific result from the analysis.

See *Analyses* (page 40) for details on the results for each analysis type.

#### Parameters

**result\_key** (*str*<sup>597</sup>) -- Result key to return from one of the enabled analyses.

#### Returns

Result value for the specified result key.

**get\_analysis\_delim\_str**(*analysis*: *Optional*<sup>598</sup>[*Literal*<sup>599</sup>['energy', 'type', 'mirna', 'target']] = *None*, *out\_delim*: *Optional*<sup>600</sup>[*str*<sup>601</sup>] = *None*) → *str*<sup>602</sup>

Return a delimited string containing the results of the analysis.

See *Analyses* (page 40) for details on the results for each analysis type.

#### Parameters

- **analysis** ([str](#)<sup>603</sup> or [list](#)<sup>604</sup> of [str](#)<sup>605</sup>) -- Analysis type for return results. If not provided, return the results for all active analyses.
- **out\_delim** ([str](#)<sup>606</sup>) -- Delimiter to use for output. If not provided, defaults to the value in [settings\['out\\_delim'\]](#) (page 42).

**write\_analysis\_delim\_str**(*out\_file\_name: Optional*<sup>607</sup>[\[str](#)<sup>608</sup>*]* = None, *analysis: Optional*<sup>609</sup>[\[Union](#)<sup>610</sup>[\[Literal](#)<sup>611</sup>[\['energy', 'type', 'mirna', 'target'\], List](#)<sup>612</sup>[\[Literal](#)<sup>613</sup>[\['energy', 'type', 'mirna', 'target'\]\]\]\]](#) = None, *out\_delim: Optional*<sup>614</sup>[\[str](#)<sup>615</sup>*]* = None) → [None](#)<sup>616</sup>

Write the results of the analysis to a delimited text file.

See [Analyses](#) (page 40) for details on the results for each analysis type.

#### Parameters

- **out\_file\_name** ([str](#)<sup>617</sup>) -- Path to output file. If not provided, defaults to: `./<analysis_name>_<analysis>.csv` if analysis/analyses provided, or `./<analysis_name>_multi_analysis.csv` if no analysis/analyses provided.
- **analysis** ([str](#)<sup>618</sup> or [list](#)<sup>619</sup> of [str](#)<sup>620</sup>) -- Analysis type for return results. If not provided, return the results for all active analyses.
- **out\_delim** ([str](#)<sup>621</sup>) -- Delimiter to use for output. If not provided, defaults to the value in [settings\['out\\_delim'\]](#) (page 42).

**write\_analysis\_results\_special**(*out\_basename: Optional*<sup>622</sup>[\[str](#)<sup>623</sup>*]* = None, *analysis: Optional*<sup>624</sup>[\[Union](#)<sup>625</sup>[\[Literal](#)<sup>626</sup>[\['energy', 'type', 'mirna', 'target'\], List](#)<sup>627</sup>[\[Literal](#)<sup>628</sup>[\['energy', 'type', 'mirna', 'target'\]\]\]\]](#) = None, *out\_delim: Optional*<sup>629</sup>[\[str](#)<sup>630</sup>*]* = None) → [List](#)<sup>631</sup>[\[str](#)<sup>632</sup>*]*

Write the results of the analyses to specialized text files.

See [Analyses](#) (page 40) for details on the results for each analysis type.

#### Parameters

- **out\_basename** ([str](#)<sup>633</sup>) -- Path for basename of output file. Files will be renamed using the provided path as the base name. If not provided, defaults to: `./<analysis_name>_<analysis>` if name is set, or `./Analysis_multi_<analysis>` if name not set.
- **analysis** ([str](#)<sup>634</sup> or [list](#)<sup>635</sup> of [str](#)<sup>636</sup>) -- Analysis type to write results files for. If not provided, write results files for all active analyses.
- **out\_delim** ([str](#)<sup>637</sup>) -- Delimiter to use for output where applicable. If not provided, defaults to the value in [settings\['out\\_delim'\]](#) (page 42).

**plot\_analysis\_results**(*out\_basename: Optional*<sup>638</sup>[\[str](#)<sup>639</sup>*]* = None, *analysis: Optional*<sup>640</sup>[\[Union](#)<sup>641</sup>[\[Literal](#)<sup>642</sup>[\['energy', 'type', 'mirna', 'target'\], List](#)<sup>643</sup>[\[Literal](#)<sup>644</sup>[\['energy', 'type', 'mirna', 'target'\]\]\]\]](#) = None) → [List](#)<sup>645</sup>[\[str](#)<sup>646</sup>*]*

Plot the results of the analyses.

See [Analyses](#) (page 40) for details on the results for each analysis type.

#### Parameters

- **analysis** ([str](#)<sup>647</sup> or [list](#)<sup>648</sup> of [str](#)<sup>649</sup>) -- Analysis type to plot results for. If not provided, plot results for all active analyses.
- **out\_basename** ([str](#)<sup>650</sup>) -- Path to output file. If not provided, defaults to: `./<analysis_name>` if name provided or `./analysis` if no name provided.

```
key = 'fold'
```

---

```

534 https://docs.python.org/3/library/typing.html#typing.Union
535 https://docs.python.org/3/library/typing.html#typing.Literal
536 https://docs.python.org/3/library/typing.html#typing.List
537 https://docs.python.org/3/library/typing.html#typing.Literal
538 https://docs.python.org/3/library/typing.html#typing.Optional
539 https://docs.python.org/3/library/stdtypes.html#str
540 https://docs.python.org/3/library/typing.html#typing.Optional
541 https://docs.python.org/3/library/typing.html#typing.Literal
542 https://docs.python.org/3/library/functions.html#int
543 https://docs.python.org/3/library/functions.html#int
544 https://docs.python.org/3/library/functions.html#int
545 https://docs.python.org/3/library/functions.html#float
546 https://docs.python.org/3/library/functions.html#float
547 https://docs.python.org/3/library/functions.html#float
548 https://docs.python.org/3/library/functions.html#float
549 https://docs.python.org/3/library/collections.html#collections.Counter
550 https://docs.python.org/3/library/functions.html#int
551 https://docs.python.org/3/library/collections.html#collections.Counter
552 https://docs.python.org/3/library/collections.html#collections.Counter
553 https://docs.python.org/3/library/collections.html#collections.Counter
554 https://docs.python.org/3/library/collections.html#collections.Counter
555 https://docs.python.org/3/library/collections.html#collections.Counter
556 https://docs.python.org/3/library/collections.html#collections.Counter
557 https://docs.python.org/3/library/functions.html#int
558 https://docs.python.org/3/library/functions.html#int
559 https://docs.python.org/3/library/functions.html#int
560 https://docs.python.org/3/library/functions.html#int
561 https://docs.python.org/3/library/functions.html#int
562 https://docs.python.org/3/library/functions.html#int
563 https://docs.python.org/3/library/functions.html#int
564 https://docs.python.org/3/library/functions.html#int
565 https://docs.python.org/3/library/collections.html#collections.Counter
566 https://docs.python.org/3/library/collections.html#collections.Counter
567 https://docs.python.org/3/library/functions.html#int
568 https://docs.python.org/3/library/functions.html#int
569 https://docs.python.org/3/library/collections.html#collections.Counter
570 https://docs.python.org/3/library/collections.html#collections.Counter
571 https://docs.python.org/3/library/collections.html#collections.Counter
572 https://docs.python.org/3/library/stdtypes.html#str
573 https://docs.python.org/3/library/stdtypes.html#list
574 https://docs.python.org/3/library/stdtypes.html#str
575 https://docs.python.org/3/library/stdtypes.html#str
576 https://docs.python.org/3/library/stdtypes.html#str
577 https://docs.python.org/3/library/stdtypes.html#str
578 https://docs.python.org/3/library/stdtypes.html#list
579 https://docs.python.org/3/library/stdtypes.html#str
580 https://docs.python.org/3/library/stdtypes.html#str
581 https://docs.python.org/3/library/constants.html#None
582 https://docs.python.org/3/library/typing.html#typing.List
583 https://docs.python.org/3/library/functions.html#bool
584 https://docs.python.org/3/library/functions.html#bool
585 https://docs.python.org/3/library/constants.html#None
586 https://docs.python.org/3/library/stdtypes.html#list
587 https://docs.python.org/3/library/functions.html#bool
588 https://docs.python.org/3/library/functions.html#bool
589 https://docs.python.org/3/library/stdtypes.html#dict
590 https://docs.python.org/3/library/stdtypes.html#dict
591 https://docs.python.org/3/library/typing.html#typing.Literal
592 https://docs.python.org/3/library/typing.html#typing.Dict
593 https://docs.python.org/3/library/stdtypes.html#str
594 https://docs.python.org/3/library/stdtypes.html#dict
595 https://docs.python.org/3/library/stdtypes.html#str
596 https://docs.python.org/3/library/typing.html#typing.Any
597 https://docs.python.org/3/library/stdtypes.html#str
598 https://docs.python.org/3/library/typing.html#typing.Optional
599 https://docs.python.org/3/library/typing.html#typing.Literal
600 https://docs.python.org/3/library/typing.html#typing.Optional
601 https://docs.python.org/3/library/stdtypes.html#str
602 https://docs.python.org/3/library/stdtypes.html#str
603 https://docs.python.org/3/library/stdtypes.html#str
604 https://docs.python.org/3/library/stdtypes.html#list
605 https://docs.python.org/3/library/stdtypes.html#str
606 https://docs.python.org/3/library/stdtypes.html#str
607 https://docs.python.org/3/library/typing.html#typing.Optional

```

---



## 2.4 hybkit.plot

Methods for plotting analyses of HybRecord and FoldRecord objects.

```
hybkit.plot.COLOR_DICT = {'Blue': '#0072B2', 'Bluish Green': '#009E73', 'Orange': '#E69F00', 'Reddish Purple': '#CC79A7', 'Sky Blue': '#56B4E9', 'Vermilion': '#D55E00', 'Yellow': '#F0E442'}
```

Default Colors for colored plots: Colors selected based on "Points of view: Color blindness" by Bang Wong, Nature Methods, 2011. Colors in RGB nomenclature (1-255): Black (0,0,0), Orange (230,159,0), Sky Blue (86,180,233), Bluish Green (0,158,115), Yellow (240,228,66), Blue (0,114,178), Vermilion (213,94,0), Reddish Purple (204,121,167)

```
hybkit.plot.COLOR_LIST = ['#0072B2', '#D55E00', '#009E73', '#CC79A7', '#E69F00', '#56B4E9', '#F0E442']
```

List of default colors for colored plots.

```
hybkit.plot.ENERGY_HIST_RC_PARAMS = {'axes.labelweight': 'bold', 'axes.titlepad': 15, 'axes.titlesize': 'large', 'axes.titleweight': 'bold', 'figure.dpi': 1200, 'figure.figsize': [6.4, 4.8]}
```

Default mpl rcParams for energy analysis histograms.

```
hybkit.plot.TYPE_PIE_SINGLE_RC_PARAMS = {'axes.labelweight': 'bold', 'axes.titlepad': 15, 'axes.titleweight': 'bold', 'figure.dpi': 1200, 'figure.figsize': [6.4, 4.8]}
```

Default mpl rcParams for type analysis pie charts.

```
hybkit.plot.TYPE_PIE_DUAL_RC_PARAMS = {'axes.labelweight': 'bold', 'axes.titlepad': 15, 'axes.titleweight': 'bold', 'figure.dpi': 1200, 'figure.figsize': (8, 4.8)}
```

Default mpl rcParams for type analysis pie charts.

```
hybkit.plot.TARGET_PIE_RC_PARAMS = {'axes.labelweight': 'bold', 'axes.titlepad': 15, 'axes.titleweight': 'bold', 'figure.dpi': 1200, 'figure.figsize': (9.6, 4.8)}
```

Default mpl rcParams for target analysis pie charts.

```
hybkit.plot.FOLD_MATCH_HIST_RC_PARAMS = {'axes.labelweight': 'bold', 'axes.titlepad': 15, 'axes.titlesize': 'large', 'axes.titleweight': 'bold', 'figure.dpi': 1200, 'figure.figsize': [6.4, 4.8]}
```

Default mpl rcParams for fold match analysis histograms.

```
hybkit.plot.FOLD_NT_COUNTS_HIST_RC_PARAMS = {'axes.labelweight': 'bold', 'axes.titlepad': 15, 'axes.titlesize': 'large', 'axes.titleweight': 'bold', 'figure.dpi': 1200, 'figure.figsize': [6.4, 4.8]}
```

Default mpl rcParams for fold nt counts analysis histograms.

```
hybkit.plot.PIE_DEFAULTS = {'COLORS': ['#0072B2', '#D55E00', '#009E73', '#CC79A7', '#E69F00', '#56B4E9', '#F0E442'], 'MIN_WEDGE_SIZE': 0.04, 'OTHER_THRESHOLD': 0.05, 'SETTINGS': {'autopct': '%1.1f%', 'counterclock': False, 'shadow': False, 'startangle': 90}}
```

Default Pie Chart Plot Settings.

```
hybkit.plot.BAR_DEFAULTS = {'BAR_ALIGN': 'edge', 'BAR_EDGE_COLOR': None, 'BAR_WIDTH': 0.9}
```

Default Bar Chart Plot Settings.

```
hybkit.plot.BAR_INT_DEFAULTS = {'BAR_ALIGN': 'center', 'BAR_EDGE_COLOR': None, 'BAR_WIDTH': 0.9}
```

Default Bar Chart of Integer Plot Settings.

```
hybkit.plot.ENERGY_DEFAULTS = {'MIN_COUNT': 0, 'MIN_DENSITY': 0.0, 'XLABEL': 'Hybrid  
Gibbs Free Energy (kcal/mol)', 'YLABEL': 'Hybrid Count'}
```

Default Bar Chart Plot Settings for Energy Histograms.

```
hybkit.plot.energy_histogram(results: Dict651[str652, Any653], plot_file_name: str654, title: str655, name:  
Optional656[str657] = None, rc_params: Dict658[str659, Any660] =  
{'axes.labelweight': 'bold', 'axes.titlepad': 15, 'axes.titlesize': 'large',  
'axes.titleweight': 'bold', 'figure.dpi': 1200, 'figure.figsize': [6.4, 4.8]},  
bar_params: Dict661[str662, Any663] = {'BAR_ALIGN': 'edge',  
'BAR_EDGE_COLOR': None, 'BAR_WIDTH': 0.9}) → None664
```

Plot histogram of hybrid energies from an [Analysis](#) (page 40) fold analysis.

#### Parameters

- **results** (dict<sup>665</sup>) -- Dictionary of energy counts from an [Analysis](#) (page 40) fold analysis (Key: binned\_energy\_vals).
- **plot\_file\_name** (str<sup>666</sup>) -- Name of output file.
- **title** (str<sup>667</sup>) -- Title of plot.
- **name** (str<sup>668</sup>, optional) -- Name of analysis to be included in plot title.
- **rc\_params** (dict<sup>669</sup>, optional) -- Dictionary of mpl rcParams. Defaults to [ENERGY\\_HIST\\_RC\\_PARAMS](#) (page 47).
- **bar\_params** (dict<sup>670</sup>, optional) -- Dictionary of bar plot parameters. Defaults to [BAR\\_DEFAULTS](#) (page 47).

```
hybkit.plot.type_count(results: Counter671, plot_file_name: str672, title: str673, name: Optional674[str675] =  
None, join_entries: bool676 = False, rc_params: Dict677[str678, Any679] =  
{'axes.labelweight': 'bold', 'axes.titlepad': 15, 'axes.titleweight': 'bold', 'figure.dpi':  
1200, 'figure.figsize': [6.4, 4.8]}) → None680
```

Plot pie chart of hybrid type counts from an [Analysis](#) (page 40) type analysis.

#### Parameters

- **results** (Counter<sup>681</sup>) -- Counter Object of type counts from an [Analysis](#) (page 40) type analysis.
- **plot\_file\_name** (str<sup>682</sup>) -- Name of output file.
- **title** (str<sup>683</sup>) -- Title of plot.

<sup>651</sup> <https://docs.python.org/3/library/typing.html#typing.Dict>

<sup>652</sup> <https://docs.python.org/3/library/stdtypes.html#str>

<sup>653</sup> <https://docs.python.org/3/library/typing.html#typing.Any>

<sup>654</sup> <https://docs.python.org/3/library/stdtypes.html#str>

<sup>655</sup> <https://docs.python.org/3/library/stdtypes.html#str>

<sup>656</sup> <https://docs.python.org/3/library/typing.html#typing.Optional>

<sup>657</sup> <https://docs.python.org/3/library/stdtypes.html#str>

<sup>658</sup> <https://docs.python.org/3/library/typing.html#typing.Dict>

<sup>659</sup> <https://docs.python.org/3/library/stdtypes.html#str>

<sup>660</sup> <https://docs.python.org/3/library/typing.html#typing.Any>

<sup>661</sup> <https://docs.python.org/3/library/typing.html#typing.Dict>

<sup>662</sup> <https://docs.python.org/3/library/stdtypes.html#str>

<sup>663</sup> <https://docs.python.org/3/library/typing.html#typing.Any>

<sup>664</sup> <https://docs.python.org/3/library/constants.html#None>

<sup>665</sup> <https://docs.python.org/3/library/stdtypes.html#dict>

<sup>666</sup> <https://docs.python.org/3/library/stdtypes.html#str>

<sup>667</sup> <https://docs.python.org/3/library/stdtypes.html#str>

<sup>668</sup> <https://docs.python.org/3/library/stdtypes.html#str>

<sup>669</sup> <https://docs.python.org/3/library/stdtypes.html#dict>

<sup>670</sup> <https://docs.python.org/3/library/stdtypes.html#dict>

- **name** ([str](#)<sup>684</sup>, optional) -- Name of analysis to be included in plot title.
- **join\_entries** ([bool](#)<sup>685</sup>, optional) -- If True, join two-tuple pairs into a single string for plot labels.
- **rc\_params** ([dict](#)<sup>686</sup>, optional) -- Dictionary of mpl rcParams. Defaults to TYPE\_PIE\_RC\_PARAMS.

`hybkit.plot.type_count_dual(results: Counter687, plot_file_name: str688, title: str689, name: Optional690[str691] = None, join_entries: bool692 = False, rc_params: Dict693[str694, Any695] = {'axes.labelweight': 'bold', 'axes.titlepad': 15, 'axes.titleweight': 'bold', 'figure.dpi': 1200, 'figure.figsize': (8, 4.8)}) → None696`

Plot pie chart of hybrid type counts from an [Analysis](#) (page 40) type analysis.

#### Parameters

- **results** ([Counter](#)<sup>697</sup>) -- Counter Object of type counts from an [Analysis](#) (page 40) type analysis.
- **plot\_file\_name** ([str](#)<sup>698</sup>) -- Name of output file.
- **title** ([str](#)<sup>699</sup>) -- Title of plot.
- **name** ([str](#)<sup>700</sup>, optional) -- Name of analysis to be included in plot title.
- **join\_entries** ([bool](#)<sup>701</sup>, optional) -- If True, join two-tuple pairs into a single string for plot labels.
- **rc\_params** ([dict](#)<sup>702</sup>, optional) -- Dictionary of mpl rcParams. Defaults to TYPE\_PIE\_RC\_PARAMS.

`hybkit.plot.target_count(*args, **kwargs) → None703`

Plot pie chart of target counts from an [Analysis](#) (page 40) type analysis.

<sup>671</sup> <https://docs.python.org/3/library/collections.html#collections.Counter>  
<sup>672</sup> <https://docs.python.org/3/library/stdtypes.html#str>  
<sup>673</sup> <https://docs.python.org/3/library/stdtypes.html#str>  
<sup>674</sup> <https://docs.python.org/3/library/typing.html#typing.Optional>  
<sup>675</sup> <https://docs.python.org/3/library/stdtypes.html#str>  
<sup>676</sup> <https://docs.python.org/3/library/functions.html#bool>  
<sup>677</sup> <https://docs.python.org/3/library/typing.html#typing.Dict>  
<sup>678</sup> <https://docs.python.org/3/library/stdtypes.html#str>  
<sup>679</sup> <https://docs.python.org/3/library/typing.html#typing.Any>  
<sup>680</sup> <https://docs.python.org/3/library/constants.html#None>  
<sup>681</sup> <https://docs.python.org/3/library/collections.html#collections.Counter>  
<sup>682</sup> <https://docs.python.org/3/library/stdtypes.html#str>  
<sup>683</sup> <https://docs.python.org/3/library/stdtypes.html#str>  
<sup>684</sup> <https://docs.python.org/3/library/stdtypes.html#str>  
<sup>685</sup> <https://docs.python.org/3/library/functions.html#bool>  
<sup>686</sup> <https://docs.python.org/3/library/stdtypes.html#dict>  
<sup>687</sup> <https://docs.python.org/3/library/collections.html#collections.Counter>  
<sup>688</sup> <https://docs.python.org/3/library/stdtypes.html#str>  
<sup>689</sup> <https://docs.python.org/3/library/stdtypes.html#str>  
<sup>690</sup> <https://docs.python.org/3/library/typing.html#typing.Optional>  
<sup>691</sup> <https://docs.python.org/3/library/stdtypes.html#str>  
<sup>692</sup> <https://docs.python.org/3/library/functions.html#bool>  
<sup>693</sup> <https://docs.python.org/3/library/typing.html#typing.Dict>  
<sup>694</sup> <https://docs.python.org/3/library/stdtypes.html#str>  
<sup>695</sup> <https://docs.python.org/3/library/typing.html#typing.Any>  
<sup>696</sup> <https://docs.python.org/3/library/constants.html#None>  
<sup>697</sup> <https://docs.python.org/3/library/collections.html#collections.Counter>  
<sup>698</sup> <https://docs.python.org/3/library/stdtypes.html#str>  
<sup>699</sup> <https://docs.python.org/3/library/stdtypes.html#str>  
<sup>700</sup> <https://docs.python.org/3/library/stdtypes.html#str>  
<sup>701</sup> <https://docs.python.org/3/library/functions.html#bool>  
<sup>702</sup> <https://docs.python.org/3/library/stdtypes.html#dict>

### Parameters

- **results** ([Counter](#)<sup>704</sup>) -- Counter Object of type counts from an [Analysis](#) (page 40) type analysis.
- **plot\_file\_name** ([str](#)<sup>705</sup>) -- Name of output file.
- **title** ([str](#)<sup>706</sup>) -- Title of plot.
- **name** ([str](#)<sup>707</sup>, optional) -- Name of analysis to be included in plot title.
- **join\_entries** ([bool](#)<sup>708</sup>, optional) -- If True, join two-tuple pairs into a single string for plot labels.
- **rc\_params** ([dict](#)<sup>709</sup>, optional) -- Dictionary of mpl rcParams. Defaults to [TARGET\\_PIE\\_RC\\_PARAMS](#) (page 47).

`hybkit.plot.fold_match_counts_histogram(results: Counter710, plot_file_name: str711, title: str712, name: Optional713[str714] = None, is_prop: bool715 = False, rc_params: Dict716[str717, Any718] = {'axes.labelweight': 'bold', 'axes.titlepad': 15, 'axes.titlesize': 'large', 'axes.titleweight': 'bold', 'figure.dpi': 1200, 'figure.figsize': [6.4, 4.8]}, bar_params: Dict719[str720, Any721] = {'BAR_ALIGN': 'center', 'BAR_EDGE_COLOR': None, 'BAR_WIDTH': 0.9}) → None722`

Plot histogram of predicted miRNA/target match count.

### Parameters

- **results** ([Counter](#)<sup>723</sup>) -- Counter Object of match counts from an [Analysis](#) (page 40) type analysis.
- **plot\_file\_name** ([str](#)<sup>724</sup>) -- Name of output file.
- **title** ([str](#)<sup>725</sup>) -- Title of plot.
- **is\_prop** ([bool](#)<sup>726</sup>, optional) -- If True, y axis is proportion.
- **name** ([str](#)<sup>727</sup>, optional) -- Name of analysis to be included in plot title.
- **rc\_params** ([dict](#)<sup>728</sup>, optional) -- Dictionary of mpl rcParams. Defaults to [FOLD\\_MATCH\\_HIST\\_RC\\_PARAMS](#) (page 47).
- **bar\_params** ([dict](#)<sup>729</sup>, optional) -- Dictionary of bar plot parameters. Defaults to [BAR\\_INT\\_DEFAULTS](#) (page 47).

<sup>703</sup> <https://docs.python.org/3/library/constants.html#None>

<sup>704</sup> <https://docs.python.org/3/library/collections.html#collections.Counter>

<sup>705</sup> <https://docs.python.org/3/library/stdtypes.html#str>

<sup>706</sup> <https://docs.python.org/3/library/stdtypes.html#str>

<sup>707</sup> <https://docs.python.org/3/library/stdtypes.html#str>

<sup>708</sup> <https://docs.python.org/3/library/functions.html#bool>

<sup>709</sup> <https://docs.python.org/3/library/stdtypes.html#dict>

`hybkit.plot.fold_mirna_nt_counts_histogram(*args, **kwargs) → None730`

Plot histogram of predicted miRNA/target match count.

#### Parameters

- **results** ([Counter<sup>731</sup>](#)) -- Counter Object of match counts from an [Analysis](#) (page 40) type analysis.
- **plot\_file\_name** ([str<sup>732</sup>](#)) -- Name of output file.
- **title** ([str<sup>733</sup>](#)) -- Title of plot.
- **is\_prop** ([bool<sup>734</sup>](#), optional) -- If True, y axis is proportion.
- **name** ([str<sup>735</sup>](#), optional) -- Name of analysis to be included in plot title.
- **rc\_params** ([dict<sup>736</sup>](#), optional) -- Dictionary of mpl rcParams. Defaults to [FOLD\\_NT\\_COUNTS\\_HIST\\_RC\\_PARAMS](#) (page 47).
- **bar\_params** ([dict<sup>737</sup>](#), optional) -- Dictionary of bar plot parameters. Defaults to [BAR\\_INT\\_DEFAULTS](#) (page 47).

## 2.5 hybkit.settings

This module contains settings information for hybkit classes and methods.

`hybkit.settings.HYB_SUFFIXES = ['.hyb', '.Hyb', '.HYB']`

Allowed suffixes for "Hyb" files.

`hybkit.settings.VIENNA_SUFFIXES = ['.vienna', '.Vienna', '.VIENNA']`

Allowed suffixes for "Vienna" files.

`hybkit.settings.CT_SUFFIXES = ['.ct', '.Ct', '.CT']`

Allowed suffixes for "Connection-Table" files.

<sup>710</sup> <https://docs.python.org/3/library/collections.html#collections.Counter>

<sup>711</sup> <https://docs.python.org/3/library/stdtypes.html#str>

<sup>712</sup> <https://docs.python.org/3/library/stdtypes.html#str>

<sup>713</sup> <https://docs.python.org/3/library/typing.html#typing.Optional>

<sup>714</sup> <https://docs.python.org/3/library/stdtypes.html#str>

<sup>715</sup> <https://docs.python.org/3/library/functions.html#bool>

<sup>716</sup> <https://docs.python.org/3/library/typing.html#typing.Dict>

<sup>717</sup> <https://docs.python.org/3/library/stdtypes.html#str>

<sup>718</sup> <https://docs.python.org/3/library/typing.html#typing.Any>

<sup>719</sup> <https://docs.python.org/3/library/typing.html#typing.Dict>

<sup>720</sup> <https://docs.python.org/3/library/stdtypes.html#str>

<sup>721</sup> <https://docs.python.org/3/library/typing.html#typing.Any>

<sup>722</sup> <https://docs.python.org/3/library/constants.html#None>

<sup>723</sup> <https://docs.python.org/3/library/collections.html#collections.Counter>

<sup>724</sup> <https://docs.python.org/3/library/stdtypes.html#str>

<sup>725</sup> <https://docs.python.org/3/library/stdtypes.html#str>

<sup>726</sup> <https://docs.python.org/3/library/functions.html#bool>

<sup>727</sup> <https://docs.python.org/3/library/stdtypes.html#str>

<sup>728</sup> <https://docs.python.org/3/library/stdtypes.html#dict>

<sup>729</sup> <https://docs.python.org/3/library/stdtypes.html#dict>

<sup>730</sup> <https://docs.python.org/3/library/constants.html#None>

<sup>731</sup> <https://docs.python.org/3/library/collections.html#collections.Counter>

<sup>732</sup> <https://docs.python.org/3/library/stdtypes.html#str>

<sup>733</sup> <https://docs.python.org/3/library/stdtypes.html#str>

<sup>734</sup> <https://docs.python.org/3/library/functions.html#bool>

<sup>735</sup> <https://docs.python.org/3/library/stdtypes.html#str>

<sup>736</sup> <https://docs.python.org/3/library/stdtypes.html#dict>

<sup>737</sup> <https://docs.python.org/3/library/stdtypes.html#dict>

hybkit.settings.FOLD\_SUFFIXES = ['.vienna', '.Vienna', '.VIENNA', '.ct', '.Ct', '.CT']

Allowed suffixes for "Vienna" and "Connection-Table" files.

hybkit.settings.MIRNA\_TYPES = ['miRNA', 'microRNA']

Default miRNA types for use in mirna\_analysis().

hybkit.settings.HybRecord\_settings\_info

Information for settings of *HybRecord* (page 9) class. Copied into *HybRecord\_settings* (page 55) for use at runtime.

```
hybkit.settings.HybRecord_settings_info = {
'allow_undefined_flags': {'Argp-Flag': None,
                          'Argp-Opts': {'const': True, 'nargs': '?'},
                          'Argp-Type': 'custom_bool_from_str',
                          'Def-Val': False,
                          'Desc.': 'Allow use of flags not defined in the '
                                  'hybkit-specification order when reading and '
                                  'writing hyb records. As the preferred '
                                  'alternative to using this setting, the '
                                  '--custom_flags argument can be used to '
                                  'supply custom allowed flags.'},
'allow_unknown_seg_types': {'Argp-Flag': None,
                            'Argp-Opts': {'const': True, 'nargs': '?'},
                            'Argp-Type': 'custom_bool_from_str',
                            'Def-Val': False,
                            'Desc.': 'Allow unknown segment types when assigning '
                                    'segment types.'},
'custom_flags': {'Argp-Flag': None,
                 'Argp-Opts': {'nargs': '+'},
                 'Argp-Type': 'str',
                 'Def-Val': [],
                 'Desc.': 'Custom flags to allow in addition to those specified in '
                         'the hybkit specification.'},
'hyb_placeholder': {'Argp-Flag': None,
                   'Argp-Opts': {},
                   'Argp-Type': 'str',
                   'Def-Val': '.',
                   'Desc.': 'placeholder character/string for missing data in hyb '
                           'files.'},
'mirna_types': {'Argp-Flag': None,
                'Argp-Opts': {'nargs': '+'},
                'Argp-Type': 'str',
                'Def-Val': ['miRNA', 'microRNA'],
                'Desc.': '"seg_type" fields identifying a miRNA'},
'reorder_flags': {'Argp-Flag': None,
                  'Argp-Opts': {},
                  'Argp-Type': 'custom_bool_from_str',
                  'Def-Val': True,
                  'Desc.': 'Re-order flags to the hybkit-specification order when '
                          'writing hyb records.'}
}
```

hybkit.settings.HybFile\_settings\_info

Information for settings of *HybFile* (page 21) class. Copied into *HybFile\_settings* (page 55) for use at

runtime.

```
hybkit.settings.HybFile_settings_info = {
'hybformat_id': {'Argp-Flag': None,
  'Argp-Opts': {'const': True, 'nargs': '?'},
  'Argp-Type': 'custom_bool_from_str',
  'Def-Val': False,
  'Desc.': 'The Hyb Software Package places further information in '
    'the "id" field of the hybrid record that can be used to '
    'infer the number of contained read counts. When set to '
    'True, the identifiers will be parsed as: '
    '"<read_id>_<read_count>"'},
'hybformat_ref': {'Argp-Flag': None,
  'Argp-Opts': {'const': True, 'nargs': '?'},
  'Argp-Type': 'custom_bool_from_str',
  'Def-Val': False,
  'Desc.': 'The Hyb Software Package uses a reference database '
    'with identifiers that contain sequence type and other '
    'sequence information. When set to True, all hyb file '
    'identifiers will be parsed as: '
    '"<gene_id>_<transcript_id>_<gene_name>_<seg_type>"'}
}
```

#### hybkit.settings.FoldRecord\_settings\_info

Information for settings of *FoldRecord* (page 23) class. Copied into *FoldRecord\_settings* (page 55) for use at runtime.

```
hybkit.settings.FoldRecord_settings_info = {
'allowed_mismatches': {'Argp-Flag': None,
  'Argp-Opts': {},
  'Argp-Type': 'int',
  'Def-Val': 0,
  'Desc.': 'For DynamicFoldRecords, allowed number of '
    'mismatches with a HybRecord.'},
'error_mode': {'Argp-Flag': None,
  'Argp-Opts': {'choices': ['raise', 'warn_return', 'return']},
  'Argp-Type': 'str',
  'Def-Val': 'raise',
  'Desc.': 'Mode for handling errors during reading of HybFiles '
    '"(overridden by HybFoldIter.settings[\'iter_error_mode\'] "
    'when using HybFoldIter). Options: "raise": Raise an error '
    'when encountered and exit program ; "warn_return": Print '
    'a warning and return the error_value ; "return": Return '
    'the error value with no program output. record is '
    'encountered.'},
'fold_placeholder': {'Argp-Flag': None,
  'Argp-Opts': {},
  'Argp-Type': 'str',
  'Def-Val': '.',
  'Desc.': 'Placeholder character/string for missing data for '
    'reading/writing fold records.'},
'seq_type': {'Argp-Flag': '-y',
  'Argp-Opts': {'choices': ['static', 'dynamic']},
```

(continues on next page)



(continued from previous page)

```

    'Argp-Type': 'str',
    'Def-Val': 'static',
    'Desc.': 'Type of fold record object to use. Options: "static": '
              'FoldRecord, requires an exact sequence match to be paired '
              'with a HybRecord; "dynamic": DynamicFoldRecord, requires a '
              'sequence match to the "dynamic" annotated regions of a '
              'HybRecord, and may be shorter/longer than the original '
              'sequence.'}
}

```

**hybkit.settings.FoldFile\_settings\_info**

Information for settings of FoldFile class. Copied into *FoldFile\_settings* (page 55) for use at runtime.

```

hybkit.settings.FoldFile_settings_info = {
}

```

**hybkit.settings.HybFoldIter\_settings\_info**

Information for settings of *HybFoldIter* (page 33) class. Copied into *HybFoldIter\_settings* (page 56) for use at runtime.

```

hybkit.settings.HybFoldIter_settings_info = {
  'error_checks': {'Argp-Flag': None,
                   'Argp-Opts': {'choices': ['hybrecord_indel',
                                              'foldrecord_nofold',
                                              'max_mismatch',
                                              'energy_mismatch']}},
  'Argp-Type': 'str',
  'Def-Val': ['hybrecord_indel',
              'foldrecord_nofold',
              'max_mismatch',
              'energy_mismatch'],
  'Desc.': 'Error checks for simultaneous HybFile and FoldFile '
            'parsing. Options: "hybrecord_indel": Error for '
            'HybRecord objects where one/both sequences have '
            'insertions/deletions in alignment, which prevents '
            'matching of sequences; "foldrecord_nofold": Error when '
            'failure in reading a fold_record object; '
            '"max_mismatch": Error when mismatch between hybrecord '
            'and foldrecord sequences is greater than FoldRecord '
            '"allowed_mismatches" setting; "energy_mismatch": Error '
            'when a mismatch exists between HybRecord and FoldRecord '
            'energy values.',
  'iter_error_mode': {'Argp-Flag': None,
                      'Argp-Opts': {'choices': ['raise',
                                                'warn_return',
                                                'warn_skip',
                                                'skip',
                                                'return']}},
  'Argp-Type': 'str',
  'Def-Val': 'warn_skip',
  'Desc.': 'Mode for handling errors found during error checks. '
}

```

(continues on next page)



(continued from previous page)

```

        'Overrides HybRecord "error_mode" setting when using '
        'HybFoldIter. Options: "raise": Raise an error when '
        'encountered; "warn_return": Print a warning and '
        'return the value; "warn_skip": Print a warning and '
        'continue to the next iteration; "skip": Continue to '
        'the next iteration without any output; "return": '
        'return the value without any error output;'},
'max_sequential_skips': {'Argp-Flag': None,
                        'Argp-Opts': {},
                        'Argp-Type': 'int',
                        'Def-Val': 100,
                        'Desc.': 'Maximum number of record(-pairs) to skip in a '
                                'row. Limited as several sequential skips '
                                'usually indicates an issue with record '
                                'formatting or a desynchronization between '
                                'files.'}
}

```

**hybkit.settings.Analysis\_settings\_info**

Information for settings of Analysis class. Copied into *Analysis\_settings* (page 56) for use at runtime.

```

hybkit.settings.Analysis_settings_info = {
'out_delim': {'Argp-Flag': None,
              'Argp-Opts': {},
              'Argp-Type': 'str',
              'Def-Val': ',',
              'Desc.': 'Delimiter-string to place between fields in analysis '
                      'output.'},
'quant_mode': {'Argp-Flag': None,
               'Argp-Opts': {'choices': ['single', 'reads', 'records']},
               'Argp-Type': 'str',
               'Def-Val': 'single',
               'Desc.': 'Method for counting records. Options: "single": Count '
                       'each record as a single entry; "reads": Use the number of '
                       'reads per hyb record as the count (may contain PCR '
                       'duplicates); "records": Count the number of records '
                       'represented by each hyb record entry (1 for "unmerged" '
                       'records, >= 1 for "merged" records)'}
}

```

```

hybkit.settings.HybRecord_settings = {'allow_undefined_flags': False,
'allow_unknown_seg_types': False, 'custom_flags': [], 'hyb_placeholder': '.',
'mirna_types': ['miRNA', 'microRNA'], 'reorder_flags': True}

```

Settings for *HybRecord* (page 9), created from *HybRecord\_settings\_info* (page 52)

```

hybkit.settings.HybFile_settings = {'hybformat_id': False, 'hybformat_ref': False}

```

Settings for *HybFile* (page 21), created from *HybFile\_settings\_info* (page 52)

```

hybkit.settings.FoldRecord_settings = {'allowed_mismatches': 0, 'error_mode': 'raise',
'fold_placeholder': '.', 'seq_type': 'static'}

```

Settings for *FoldRecord* (page 23), created from *FoldRecord\_settings\_info* (page 53)

```
hybkit.settings.FoldFile_settings = {}
```

Settings for FoldFile, created from [FoldFile\\_settings\\_info](#) (page 54)

```
hybkit.settings.HybFoldIter_settings = {'error_checks': ['hybrecord_indel',
'foldrecord_nofold', 'max_mismatch', 'energy_mismatch'], 'iter_error_mode': 'warn_skip',
'max_sequential_skips': 100}
```

Settings for [HybFoldIter](#) (page 33), created from [HybFoldIter\\_settings\\_info](#) (page 54)

```
hybkit.settings.Analysis_settings = {'out_delim': ',', 'quant_mode': 'single'}
```

Settings for BaseAnalysis, created from [Analysis\\_settings\\_info](#) (page 55)

## 2.6 hybkit.util

This module contains helper functions for hybkit's command line scripts.

```
hybkit.util.get_argparse_doc(docstring: str738) → str739
```

Get the argparse description from a docstring.

### Parameters

**docstring** (str<sup>740</sup>) -- A docstring.

### Returns

A string containing the argparse description.

```
hybkit.util.dir_exists(dir_name: str741) → str742
```

Check if a directory exists at the provided path (else raise), and return a normalized path.

### Parameters

**dir\_name** (str<sup>743</sup>) -- Name of directory to check for existence.

### Returns

A normalized version of the path passed to dir\_name.

```
hybkit.util.file_exists(file_name: str744, required_suffixes: Optional745[List746[str747]] = None) → str748
```

Check if a file exists at the provided path, and return a normalized path.

### Parameters

- **file\_name** (str<sup>749</sup>) -- Name of file to check for existence.
- **required\_suffixes** (list<sup>750</sup>, optional) -- List of strings containing file-name suffixes. If provided, a file passed to file-exists must end with one of the strings provided. Otherwise an error will be raised.

### Returns

A normalized version of the path passed to file\_name.

<sup>738</sup> <https://docs.python.org/3/library/stdtypes.html#str>

<sup>739</sup> <https://docs.python.org/3/library/stdtypes.html#str>

<sup>740</sup> <https://docs.python.org/3/library/stdtypes.html#str>

<sup>741</sup> <https://docs.python.org/3/library/stdtypes.html#str>

<sup>742</sup> <https://docs.python.org/3/library/stdtypes.html#str>

<sup>743</sup> <https://docs.python.org/3/library/stdtypes.html#str>

<sup>744</sup> <https://docs.python.org/3/library/stdtypes.html#str>

<sup>745</sup> <https://docs.python.org/3/library/typing.html#typing.Optional>

<sup>746</sup> <https://docs.python.org/3/library/typing.html#typing.List>

<sup>747</sup> <https://docs.python.org/3/library/stdtypes.html#str>

<sup>748</sup> <https://docs.python.org/3/library/stdtypes.html#str>

<sup>749</sup> <https://docs.python.org/3/library/stdtypes.html#str>

<sup>750</sup> <https://docs.python.org/3/library/stdtypes.html#list>

`hybkit.util.hyb_exists(file_name: str751) → str752`

Check if a .hyb file exists at the provided path, and return a normalized path.

Wrapper for `file_exists()` (page 56) that includes the required suffixes in `hybkit.settings.HYB_SUFFIXES` (page 51).

**Parameters**

`file_name (str753)` -- Name of file to check for existence.

**Returns**

A normalized version of the path passed to `file_name`.

`hybkit.util.vienna_exists(file_name: str754) → str755`

Check if a .vienna file exists at the provided path, and return a normalized path.

Wrapper for `file_exists()` (page 56) that includes the required suffixes in `hybkit.settings.VIENNA_SUFFIXES` (page 51).

**Parameters**

`file_name (str756)` -- Name of file to check for existence.

**Returns**

A normalized version of the path passed to `file_name`.

`hybkit.util.ct_exists(file_name: str757) → str758`

Check if a .ct file exists at the provided path, and return a normalized path.

Wrapper for `file_exists()` (page 56) that includes the required suffixes in `hybkit.settings.CT_SUFFIXES` (page 51).

**Parameters**

`file_name (str759)` -- Name of file to check for existence.

**Returns**

A normalized version of the path passed to `file_name`.

`hybkit.util.fold_exists(file_name: str760) → str761`

Check if a fold-representing file exists at the provided path, and return a normalized path.

Wrapper for `file_exists()` (page 56) that includes the required suffixes in `hybkit.settings.FOLD_SUFFIXES` (page 51).

**Parameters**

`file_name (str762)` -- Name of file to check for existence.

**Returns**

A normalized version of the path passed to `file_name`.

`hybkit.util.out_path_exists(file_name: str763) → str764`

Check if the directory of the specified output path exists, and return a normalized path.

**Parameters**

`file_name (str765)` -- Name of path to an output file to check.

<sup>751</sup> <https://docs.python.org/3/library/stdtypes.html#str>

<sup>752</sup> <https://docs.python.org/3/library/stdtypes.html#str>

<sup>753</sup> <https://docs.python.org/3/library/stdtypes.html#str>

<sup>754</sup> <https://docs.python.org/3/library/stdtypes.html#str>

<sup>755</sup> <https://docs.python.org/3/library/stdtypes.html#str>

<sup>756</sup> <https://docs.python.org/3/library/stdtypes.html#str>

<sup>757</sup> <https://docs.python.org/3/library/stdtypes.html#str>

<sup>758</sup> <https://docs.python.org/3/library/stdtypes.html#str>

<sup>759</sup> <https://docs.python.org/3/library/stdtypes.html#str>

<sup>760</sup> <https://docs.python.org/3/library/stdtypes.html#str>

<sup>761</sup> <https://docs.python.org/3/library/stdtypes.html#str>

<sup>762</sup> <https://docs.python.org/3/library/stdtypes.html#str>

**Returns**

A normalized version of the path passed to `file_name`.

`hybkit.util.make_out_file_name(in_file_name: str766, name_suffix: str767 = 'out', in_suffix: str768 = '', out_suffix: str769 = '', out_dir: str770 = '', seg_sep: str771 = '_' ) → str772`

Given an input file name, generate an output file name.

**Parameters**

- **in\_file\_name** (str<sup>773</sup>) -- Name of input file as template.
- **name\_suffix** (str<sup>774</sup>) -- Suffix to add to name before file type.
- **in\_suffix** (str<sup>775</sup>) -- File type suffix on in\_file\_name (to remove).
- **out\_suffix** (str<sup>776</sup>) -- File type suffix to add to final output file.
- **out\_dir** (str<sup>777</sup>) -- Directory path in which to place output file.
- **seg\_sep** (str<sup>778</sup>) -- Separator string between file name segments.

**Returns**

An output file path based on the input file template.

`hybkit.util.validate_args(args: Namespace779, parser: Optional780[ArgumentParser781] = None) → bool782`

Check supplied arguments to make sure there are no hidden contradictions.

**Current checks:**

- If explicit output file names supplied, be sure that they match the number of input files provided.
- If fold files provided, make sure that they match the number of input hyb files provided.

**Parameters**

- **args** (argparse.Namespace<sup>783</sup>) -- The arguments produced by argparse.
- **parser** (argparse.ArgumentParser<sup>784</sup>, optional) -- Argparse parser object to use for verbose outputting of help message.

`hybkit.util.validate_args_exit(args: Namespace785, parser: Optional786[ArgumentParser787] = None) → None788`

Check supplied arguments using `validate_args()` (page 58), and exit if a conflict exists.

**Parameters**

<sup>763</sup> <https://docs.python.org/3/library/stdtypes.html#str>  
<sup>764</sup> <https://docs.python.org/3/library/stdtypes.html#str>  
<sup>765</sup> <https://docs.python.org/3/library/stdtypes.html#str>  
<sup>766</sup> <https://docs.python.org/3/library/stdtypes.html#str>  
<sup>767</sup> <https://docs.python.org/3/library/stdtypes.html#str>  
<sup>768</sup> <https://docs.python.org/3/library/stdtypes.html#str>  
<sup>769</sup> <https://docs.python.org/3/library/stdtypes.html#str>  
<sup>770</sup> <https://docs.python.org/3/library/stdtypes.html#str>  
<sup>771</sup> <https://docs.python.org/3/library/stdtypes.html#str>  
<sup>772</sup> <https://docs.python.org/3/library/stdtypes.html#str>  
<sup>773</sup> <https://docs.python.org/3/library/stdtypes.html#str>  
<sup>774</sup> <https://docs.python.org/3/library/stdtypes.html#str>  
<sup>775</sup> <https://docs.python.org/3/library/stdtypes.html#str>  
<sup>776</sup> <https://docs.python.org/3/library/stdtypes.html#str>  
<sup>777</sup> <https://docs.python.org/3/library/stdtypes.html#str>  
<sup>778</sup> <https://docs.python.org/3/library/stdtypes.html#str>  
<sup>779</sup> <https://docs.python.org/3/library/argparse.html#argparse.Namespace>  
<sup>780</sup> <https://docs.python.org/3/library/typing.html#typing.Optional>  
<sup>781</sup> <https://docs.python.org/3/library/argparse.html#argparse.ArgumentParser>  
<sup>782</sup> <https://docs.python.org/3/library/functions.html#bool>  
<sup>783</sup> <https://docs.python.org/3/library/argparse.html#argparse.Namespace>  
<sup>784</sup> <https://docs.python.org/3/library/argparse.html#argparse.ArgumentParser>

- **args** (*argparse.Namespace*<sup>789</sup>) -- The arguments produced by argparse.
- **parser** (*argparse.ArgumentParser*<sup>790</sup>, optional) -- Argparse parser object to use for verbose outputting of help message.

`hybkit.util.set_setting(setting: str791, set_value: Any792, verbose: bool793 = False) → str794`

Take a namespace object as from an argparse parser and update settings.

Each setting in the following settings dictionaries are checked and set where applicable:

<a href="#">HybRecord</a> (page 9) Settings	<a href="#">hybkit.settings.HybRecord_settings</a> (page 55)
<a href="#">HybFile</a> (page 21) Settings	<a href="#">hybkit.settings.HybFile_settings</a> (page 55)
<a href="#">FoldRecord</a> (page 23) Settings	<a href="#">hybkit.settings.FoldRecord_settings</a> (page 55)
FoldFile Settings	<a href="#">hybkit.settings.FoldFile_settings</a> (page 55)
<a href="#">HybFoldIter</a> (page 33) Settings	<a href="#">hybkit.settings.HybFoldIter_settings</a> (page 56)
Analysis Settings	<a href="#">hybkit.settings.Analysis_settings</a> (page 56)

#### Parameters

- **setting** (str<sup>795</sup>) -- Name of setting to change
- **set\_value** (str<sup>796</sup>) -- New value for setting
- **verbose** (bool<sup>797</sup>, optional) -- If True, print when changing setting.

`hybkit.util.set_settings_from_namespace(nspace: Namespace798, verbose: bool799 = False) → None800`

Take a namespace object as from an argparse parser and update settings.

See [set\\_setting\(\)](#) (page 59) for details

#### Parameters

- **nspace** (*argparse.Namespace*<sup>801</sup>) -- Namespace containing settings
- **verbose** (bool<sup>802</sup>, optional) -- If True, print when changing setting.

<sup>785</sup> <https://docs.python.org/3/library/argparse.html#argparse.Namespace>

<sup>786</sup> <https://docs.python.org/3/library/typing.html#typing.Optional>

<sup>787</sup> <https://docs.python.org/3/library/argparse.html#argparse.ArgumentParser>

<sup>788</sup> <https://docs.python.org/3/library/constants.html#None>

<sup>789</sup> <https://docs.python.org/3/library/argparse.html#argparse.Namespace>

<sup>790</sup> <https://docs.python.org/3/library/argparse.html#argparse.ArgumentParser>

<sup>791</sup> <https://docs.python.org/3/library/stdtypes.html#str>

<sup>792</sup> <https://docs.python.org/3/library/typing.html#typing.Any>

<sup>793</sup> <https://docs.python.org/3/library/functions.html#bool>

<sup>794</sup> <https://docs.python.org/3/library/stdtypes.html#str>

<sup>795</sup> <https://docs.python.org/3/library/stdtypes.html#str>

<sup>796</sup> <https://docs.python.org/3/library/stdtypes.html#str>

<sup>797</sup> <https://docs.python.org/3/library/functions.html#bool>

<sup>798</sup> <https://docs.python.org/3/library/argparse.html#argparse.Namespace>

<sup>799</sup> <https://docs.python.org/3/library/functions.html#bool>

<sup>800</sup> <https://docs.python.org/3/library/constants.html#None>

<sup>801</sup> <https://docs.python.org/3/library/argparse.html#argparse.Namespace>

<sup>802</sup> <https://docs.python.org/3/library/functions.html#bool>

## 2.7 hybkit.errors

Module storing hybkit error classes.

**exception** `hybkit.errors.HybkitError`

Base class for Hybkit errors.

**Variables**

**message** (`str`<sup>803</sup>) -- Human-readable string describing the error.

**exception** `hybkit.errors.HybkitArgError`

Error raised when an invalid argument is provided to a Hybkit function.

Subclass of `HybkitError` (page 60).

**Variables**

**message** (`str`<sup>804</sup>) -- Human-readable string describing the error.

**exception** `hybkit.errors.HybkitConstructorError`

Error raised when a read error occurs.

Subclass of `HybkitError` (page 60).

**Variables**

**message** (`str`<sup>805</sup>) -- Human-readable string describing the error.

**exception** `hybkit.errors.HybkitIterError`

Error raised when an error is encountered during Hybkit iteration.

Subclass of `HybkitError` (page 60).

**Variables**

**message** (`str`<sup>806</sup>) -- Human-readable string describing the error.

**exception** `hybkit.errors.HybkitMiscError`

Error raised when an error is encountered during Hybkit usage.

Subclass of `HybkitError` (page 60).

**Variables**

**message** (`str`<sup>807</sup>) -- Human-readable string describing the error.

<sup>803</sup> <https://docs.python.org/3/library/stdtypes.html#str>

<sup>804</sup> <https://docs.python.org/3/library/stdtypes.html#str>

<sup>805</sup> <https://docs.python.org/3/library/stdtypes.html#str>

<sup>806</sup> <https://docs.python.org/3/library/stdtypes.html#str>

<sup>807</sup> <https://docs.python.org/3/library/stdtypes.html#str>

## HYBKIT TOOLKIT

The hybkit toolkit contains command-line scripts for analysis and manipulation of hyb and fold files. Scripts are implemented in Python3, and the hybkit module must be on the user's PYTHONPATH for script execution.

The command-line options and flags are generated with the Python3 argparse module. Relevant settings pertaining to specific hybkit classes are accessible via command-line flags, as demonstrated in the "shell\_analysis" implementations in the *Example Analyses* (page 94).

This version of hybkit includes the following executables:

Utility	Description
<i>hyb_check</i> (page 61)	Parse a hyb (/fold) file and check for errors
<i>hyb_eval</i> (page 77)	Evaluate hyb (/fold) records to identify segment types and miRNAs
<i>hyb_filter</i> (page 67)	Filter a hyb (/fold) file to a specific subset of sequences
<i>hyb_analyze</i> (page 86)	Perform a type, miRNA, summary, or target analysis on a hyb (/fold) file

Detailed descriptions and usage information are available at each respective script page.

### 3.1 hyb\_check

Read one or more hyb (and optional fold) format files and check for errors.

This utility reads in one or more files in hyb-format (see the *hybkit Hyb File Specification* (page 5)) and uses hybkit's internal file error checking to check for errors. This can be useful as an initial preparation step for further analyses.

#### Example system calls:

```
hyb_check -i my_file_1.hyb -f my_file_1.vienna
hyb_check -i my_file_1.hyb my_file_2.hyb -f my_file_1.vienna \\\
my_file_2.vienna -v --custom_flags myflag
```

```
usage: hyb_check [-h] -i PATH_TO/MY_FILE.HYB [PATH_TO/MY_FILE.HYB ...]
               [-f [PATH_TO/MY_FILE.VIENNA [PATH_TO/MY_FILE.VIENNA ...]]]
               [--version] [-v | -s]
               [--mirna_types MIRNA_TYPES [MIRNA_TYPES ...]]
               [--custom_flags CUSTOM_FLAGS [CUSTOM_FLAGS ...]]
               [--hyb_placeholder HYB_PLACEHOLDER]
               [--reorder_flags {True,False}]
```

(continues on next page)

(continued from previous page)

```

[--allow_undefined_flags [{True,False}]]
[--allow_unknown_seg_types [{True,False}]]
[--hybformat_id [{True,False}]]
[--hybformat_ref [{True,False}]]
[--allowed_mismatches ALLOWED_MISMATCHES]
[--fold_placeholder FOLD_PLACEHOLDER] [-y {static,dynamic}]
[--error_mode {raise,warn_return,return}]
[--error_checks {hybrecord_indel,foldrecord_nofold,max_mismatch,energy_
↪ mismatch}]
[--iter_error_mode {raise,warn_return,warn_skip,skip,return}]
[--max_sequential_skips MAX_SEQUENTIAL_SKIPS]

```

### 3.1.1 Named Arguments

<b>-i, --in_hyb</b>	REQUIRED path to one or more hybformat files with a ".hyb" suffix for use in the evaluation.
<b>-f, --in_fold</b>	REQUIRED path to one or more RNA secondary-structure files with a ".vienna" or ".ct" suffix for use in the evaluation.
<b>--version</b>	Print version and exit.
<b>-v, --verbose</b>	Print verbose output during run.
	Default: False
<b>-s, --silent</b>	Print no output during run.
	Default: False

### 3.1.2 Hyb Record Settings

<b>--mirna_types</b>	"seg_type" fields identifying a miRNA Default: ['miRNA', 'microRNA']
<b>--custom_flags</b>	Custom flags to allow in addition to those specified in the hybkit specification. Default: []
<b>--hyb_placeholder</b>	placeholder character/string for missing data in hyb files.



Default: "."

**--reorder\_flags** Possible choices: True, False

Re-order flags to the hybkit-specification order when writing hyb records.

Default: True

**--allow\_undefined\_flags** Possible choices: True, False

Allow use of flags not defined in the hybkit-specification order when reading and writing hyb records. As the preferred alternative to using this setting, the `--custom_flags` argument can be used to supply custom allowed flags.

Default: False

**--allow\_unknown\_seg\_types** Possible choices: True, False

Allow unknown segment types when assigning segment types.

Default: False

### 3.1.3 Hyb File Settings

**--hybformat\_id** Possible choices: True, False

The Hyb Software Package places further information in the "id" field of the hybrid record that can be used to infer the number of contained read counts. When set to True, the identifiers will be parsed as: "`<read_id>_<read_count>`"

Default: False

**--hybformat\_ref** Possible choices: True, False

The Hyb Software Package uses a reference database with identifiers that contain sequence type and other sequence information. When set to True, all hyb file identifiers will be parsed as: "<gene\_id>\_<transcript\_id>\_<gene\_name>\_<seg\_type>"

Default: False

3.1.4 Fold Record Settings

**--allowed\_mismatches**

For DynamicFoldRecords, allowed number of mismatches with a HybRecord.

Default: 0

**--fold\_placeholder**

Placeholder character/string for missing data for reading/writing fold records.

Default: "."

**-y, --seq\_type**

Possible choices: static, dynamic

Type of fold record object to use. Options: "static": FoldRecord, requires an exact sequence match to be paired with a HybRecord; "dynamic": DynamicFoldRecord, requires a sequence match to the "dynamic" annotated regions of a HybRecord, and may be shorter/longer than the original sequence.

Default: "static"

**--error\_mode**

Possible choices: raise, warn\_return, return

Mode for handling errors during read-

ing of HybFiles  
(overridden by Hyb-  
FoldIter.settings['iter\_error\_mode']  
when using Hyb-  
FoldIter). Options:  
"raise": Raise an  
error when en-  
countered and  
exit program ;  
"warn\_return": Print  
a warning and return  
the error\_value ;  
"return": Return  
the error value  
with no program  
output. record is  
encountered.

Default: "raise"

### 3.1.5 Hyb-Fold Iterator Settings

**--error\_checks** Possible choices: hybrecord\_indel, foldrecord\_nofold, max\_mismatch, en-  
ergy\_mismatch

Error checks for  
simultaneous Hyb-  
File and FoldFile  
parsing. Options:  
"hybrecord\_indel":  
Error for Hy-  
bRecord objects  
where one/both  
sequences have  
insertions/deletions  
in alignment, which  
prevents matching  
of sequences; "fol-  
drecord\_nofold":  
Error when fail-  
ure in reading a  
fold\_record object;  
"max\_mismatch":  
Error when mis-  
match between  
hybrecord and fol-  
drecord sequences  
is greater than  
FoldRecord "al-  
lowed\_mismatches"  
setting; "en-  
ergy\_mismatch":  
Error when a mis-  
match exists between

	HybRecord and FoldRecord energy values.
	Default: ['hybrecord_indel', 'foldrecord_nofold', 'max_mismatch', 'energy_mismatch']
<b>--iter_error_mode</b>	<p>Possible choices: raise, warn_return, warn_skip, skip, return</p> <p>Mode for handling errors found during error checks. Overrides HybRecord "error_mode" setting when using HybFoldIter. Options: "raise": Raise an error when encountered; "warn_return": Print a warning and return the value; "warn_skip": Print a warning and continue to the next iteration; "skip": Continue to the next iteration without any output; "return": return the value without any error output;</p> <p>Default: "warn_skip"</p>
<b>--max_sequential_skips</b>	<p>Maximum number of record(-pairs) to skip in a row. Limited as several sequential skips usually indicates an issue with record formatting or a desynchronization between files.</p> <p>Default: 100</p>

### Output File Naming:

Output files can be named in two fashions: via automatic name generation, or by providing specific out file names.

#### Automatic Name Generation:

For output name generation, the default respective naming scheme is used:

```
hyb_script -i PATH_TO/MY_FILE_1.HYB [...]
--> OUT_DIR/MY_FILE_1_ADDSUFFIX.HYB
```

This output file path can be modified with the arguments {--out\_dir, --out\_suffix} described below.

The output directory defaults to the current working directory (\$PWD), and can be modified with the --out\_dir <dir> argument. Note: The provided directory must exist, or an error will be raised. For Example:

```
hyb_script -i PATH_TO/MY_FILE_1.HYB [...] --out_dir MY_OUT_DIR
--> MY_OUT_DIR/MY_FILE_1_ADDSUFFIX.HYB
```

The suffix used for output files is based on the primary actions of the script. It can be specified using --out\_suffix <suffix>. This can optionally include the ".hyb" final suffix. for Example:

```
hyb_script -i PATH_TO/MY_FILE_1.HYB [...] --out_suffix MY_SUFFIX
--> OUT_DIR/MY_FILE_1_MY_SUFFIX.HYB
#OR
hyb_script -i PATH_TO/MY_FILE_1.HYB [...] --out_suffix MY_SUFFIX.HYB
--> OUT_DIR/MY_FILE_1_MY_SUFFIX.HYB
```

### Specific Output Names:

Alternatively, specific file names can be provided via the -o/--out\_hyb argument, ensuring that the same number of input and output files are provided. This argument takes precedence over all automatic output file naming options (--out\_dir, --out\_suffix), which are ignored if -o/--out\_hyb is provided. For Example:

```
hyb_script [...] --out_hyb MY_OUT_DIR/OUT_FILE_1.HYB MY_OUT_DIR/OUT_FILE_2.HYB
--> MY_OUT_DIR/OUT_FILE_1.hyb
--> MY_OUT_DIR/OUT_FILE_2.hyb
```

Note: The directory provided with output file paths (MY\_OUT\_DIR above) must exist, otherwise an error will be raised.

## 3.2 hyb\_filter

Filter hyb (and corresponding fold) files to meet (or exclude) specific criteria.

This script takes one or more filter and/or exclusion criteria and outputs only those records matching (/excluding) those criteria.

The filter criteria and options are based on the options provided by the `hybkit.HybRecord.prop()` (page 16) method of the Hybkit API. For more information see the full documentation for the `HybRecord` (page 9) class.

### Example System Calls:

```
hyb_filter -i my_file_1.hyb --filter has_seg_types
# Outputs records that have completed a segtype analysis

hyb_filter -i my_file_1.hyb -f my_file_1.vienna \
--include seg_type mRNA
# Outputs hyb and fold records where hyb record has either segtype of mRNA

hyb_filter -i my_file_1.hyb --exclude seg_type mRNA
# Outputs records without either segtype of mRNA

hyb_filter -i my_file_1.hyb --include seg1_type mRNA
# Outputs records with only the first / 5p segtype of mRNA
```

(continues on next page)

(continued from previous page)

```
hyb_filter -i my_file_1.hyb my_file_2.hyb -f my_file_1.vienna my_file_2.vienna \\  
  --include seg_type_contains RNA  
# Outputs all records with a segtype that includes  
# the string "RNA" (case-sensitive)  
  
hyb_filter -i my_file_1.hyb --filter seg_contains kshv  
# Outputs records where either segment identifier contains the  
# the string: "kshv" (case-sensitive)
```

Multiple filtering options can be used together. The `-m / --filter_mode` argument determines whether "any" (DEFAULT) or "all" filters are required to be true for inclusion. Note: Matching any exclusion criteria results in exclusion of the record.

#### Example System Calls (match ALL criteria):

```
hyb_filter -i my_file_1.hyb -f my_file_1.vienna \\  
  --filter seg_contains kshv \\  
  --filter_2 seg_type miRNA  
# Outputs records with either reference sequence identifier containing "kshv"  
# and with either segment having an assigned segtype of miRNA
```

#### Example System Calls (match ANY criteria):

```
hyb_filter -i my_file_1.hyb --filter_mode any \\  
  --filter seg_type miRNA \\  
  --filter_2 seg_type lncRNA  
# Outputs records containing either segment type matching  
# either "miRNA" or "lncRNA" (case-sensitive)
```

```
usage: hyb_filter [-h] -i PATH_TO/MY_FILE.HYB [PATH_TO/MY_FILE.HYB ...]  
  [-f [PATH_TO/MY_FILE.VIENNA [PATH_TO/MY_FILE.VIENNA ...]]]  
  [-o PATH_TO/OUT_FILE.HYB [PATH_TO/OUT_FILE.HYB ...]]  
  [-l PATH_TO/OUT_FILE.VIENNA [PATH_TO/OUT_FILE.VIENNA ...]]  
  [-d OUT_DIR] [-u OUT_SUFFIX] [-m {all,any}]  
  [--skip_dup_id_before] [--skip_dup_id_after]  
  [--filter FILTER [FILTER ...]]  
  [--filter_2 FILTER_2 [FILTER_2 ...]]  
  [--filter_3 FILTER_3 [FILTER_3 ...]]  
  [--exclude EXCLUDE [EXCLUDE ...]]  
  [--exclude_2 EXCLUDE_2 [EXCLUDE_2 ...]]  
  [--exclude_3 EXCLUDE_3 [EXCLUDE_3 ...]] [--set_dataset]  
  [--version] [-v | -s]  
  [--mirna_types MIRNA_TYPES [MIRNA_TYPES ...]]  
  [--custom_flags CUSTOM_FLAGS [CUSTOM_FLAGS ...]]  
  [--hyb_placeholder HYB_PLACEHOLDER]  
  [--reorder_flags {True,False}]  
  [--allow_undefined_flags [{True,False}]]  
  [--allow_unknown_seg_types [{True,False}]]  
  [--hybformat_id [{True,False}]]  
  [--hybformat_ref [{True,False}]]  
  [--allowed_mismatches ALLOWED_MISMATCHES]  
  [--fold_placeholder FOLD_PLACEHOLDER] [-y {static,dynamic}]
```

(continues on next page)

(continued from previous page)

```

--error_mode {raise,warn_return,return}}
--error_checks {hybrecord_indel,foldrecord_nofold,max_mismatch,energy_
↪mismatch}}
--iter_error_mode {raise,warn_return,warn_skip,skip,return}}
--max_sequential_skips MAX_SEQUENTIAL_SKIPS]

```

### 3.2.1 Named Arguments

**-i, --in\_hyb**

REQUIRED path to one or more hyb-format files with a ".hyb" suffix for use in the evaluation.

**-f, --in\_fold**

REQUIRED path to one or more RNA secondary-structure files with a ".vienna" or ".ct" suffix for use in the evaluation.

**-o, --out\_hyb**

Optional path to one or more hyb-format file for output (should include a ".hyb" suffix). If not provided, the output for input file "PATH\_TO/MY\_FILE.HYB" will be used as a template for the output "OUT\_DIR/MY\_FILE\_OUT.HYB".

**-l, --out\_fold**

Optional path to one or more ".vienna" or ".ct"-format files for output (should include appropriate ".vienna"/".ct" suffix). If not provided, the output for input file "PATH\_TO/MY\_FILE.VIENNA" will be used as a template for the output "OUT\_DIR/MY\_FILE\_OUT.VIENNA".

**-d, --out\_dir**

Path to directory for output of files. Defaults to the current working directory.

Default: \$PWD

<b>-u, --out_suffix</b>		Suffix to add to the name of output files, before any file- or analysis-specific suffixes. The file-type appropriate suffix will be added automatically.
	Default: "_filtered"	
<b>-m, --filter_mode</b>	Possible choices: all, any	Modes for evaluating multiple filters. The "all" mode requires all provided filters to be true for inclusion. The "any" mode requires only one provided filter to be true for inclusion. (Note: matching any exclusion filter is grounds for exclusion of record.)
	Default: "all"	
<b>--skip_dup_id_before</b>		Skip sequential duplicate read IDs before filtering.
	Default: False	
<b>--skip_dup_id_after</b>		Skip sequential duplicate read IDs after filtering.
	Default: False	
<b>--filter</b>		Filter criteria #1. Records matching the criteria will be included in output. Includes a filter type, Ex: "seg_name_contains", and an argument, Ex: "ENST00000340384". (Note: not all filter types require a second argument, for Example: "has_mirna_seg")
<b>--filter_2</b>		Filter criteria #2. Records matching the criteria will



	<p>be included in output. Includes a filter type, Ex: "seg_name_contains", and an argument, Ex: "ENST00000340384". (Note: not all filter types require a second argument, for Example: "has_mirna_seg")</p>
<b>--filter_3</b>	<p>Filter criteria #3. Records matching the criteria will be included in output. Includes a filter type, Ex: "seg_name_contains", and an argument, Ex: "ENST00000340384". (Note: not all filter types require a second argument, for Example: "has_mirna_seg")</p>
<b>--exclude</b>	<p>Exclusion filter criteria #1. Records matching the criteria will be excluded from output. Includes a filter type, Ex: "seg_name_contains", and an argument, Ex: "ENST00000340384". (Note: not all filter types require a second argument, for Example: "has_mirna_seg")</p>
<b>--exclude_2</b>	<p>Exclusion filter criteria #2. Records matching the criteria will be excluded from output. Includes a filter type, Ex: "seg_name_contains", and an argument, Ex: "ENST00000340384". (Note: not all filter types require a second argument, for Example:</p>

	"has_mirna_seg")
<b>--exclude_3</b>	Exclusion filter criteria #3. Records matching the criteria will be excluded from output. Includes a filter type, Ex: "seg_name_contains", and an argument, Ex: "ENST00000340384". (Note: not all filter types require a second argument, for Example: "has_mirna_seg")
<b>--set_dataset</b>	Set "dataset" flag to value of the input file name.
	Default: False
<b>--version</b>	Print version and exit.
<b>-v, --verbose</b>	Print verbose output during run.
	Default: False
<b>-s, --silent</b>	Print no output during run.
	Default: False

### 3.2.2 Hyb Record Settings

<b>--mirna_types</b>	"seg_type" fields identifying a miRNA Default: ['miRNA', 'microRNA']
<b>--custom_flags</b>	Custom flags to allow in addition to those specified in the hybkit specification. Default: []
<b>--hyb_placeholder</b>	placeholder character/string for missing data in hyb files. Default: "."
<b>--reorder_flags</b>	Possible choices: True, False Re-order flags to the hybkit-specification order when writing hyb records.

Default: True

**--allow\_undefined\_flags** Possible choices: True, False

Allow use of flags not defined in the hybkit-specification order when reading and writing hyb records. As the preferred alternative to using this setting, the `--custom_flags` argument can be used to supply custom allowed flags.

Default: False

**--allow\_unknown\_seg\_types** Possible choices: True, False

Allow unknown segment types when assigning segment types.

Default: False

### 3.2.3 Hyb File Settings

**--hybformat\_id** Possible choices: True, False

The Hyb Software Package places further information in the "id" field of the hybrid record that can be used to infer the number of contained read counts. When set to True, the identifiers will be parsed as: "`<read_id>_<read_count>`"

Default: False

**--hybformat\_ref** Possible choices: True, False

The Hyb Software Package uses a reference database with identifiers that contain sequence type and other sequence information. When set to True, all

hyb file identifiers  
will be parsed as:  
"<gene\_id>\_<transcript\_id>\_<gene\_name>\_<seg\_type>"

Default: False

3.2.4 Fold Record Settings

**--allowed\_mismatches** For DynamicFoldRecords, allowed number of mismatches with a HybRecord.

Default: 0

**--fold\_placeholder** Placeholder character/string for missing data for reading/writing fold records.

Default: "."

**-y, --seq\_type** Possible choices: static, dynamic  
Type of fold record object to use. Options: "static": FoldRecord, requires an exact sequence match to be paired with a HybRecord; "dynamic": DynamicFoldRecord, requires a sequence match to the "dynamic" annotated regions of a HybRecord, and may be shorter/longer than the original sequence.

Default: "static"

**--error\_mode** Possible choices: raise, warn\_return, return  
Mode for handling errors during reading of HybFiles (overridden by HybFoldIter.settings['iter\_error\_mode'] when using HybFoldIter). Options: "raise": Raise an error when encountered and

```

exit    program    ;
"warn_return": Print
a warning and return
the    error_value ;
"return":   Return
the    error    value
with no program
output.    record is
encountered.

```

Default: "raise"

### 3.2.5 Hyb-Fold Iterator Settings

**--error\_checks** Possible choices: hybrecord\_indel, foldrecord\_nofold, max\_mismatch, energy\_mismatch

Error checks for simultaneous Hyb-File and FoldFile parsing. Options: "hybrecord\_indel": Error for HybRecord objects where one/both sequences have insertions/deletions in alignment, which prevents matching of sequences; "foldrecord\_nofold": Error when failure in reading a fold\_record object; "max\_mismatch": Error when mismatch between hybrecord and foldrecord sequences is greater than FoldRecord "allowed\_mismatches" setting; "energy\_mismatch": Error when a mismatch exists between HybRecord and FoldRecord energy values.

Default: ['hybrecord\_indel', 'foldrecord\_nofold', 'max\_mismatch', 'energy\_mismatch']

**--iter\_error\_mode** Possible choices: raise, warn\_return, warn\_skip, skip, return

Mode for handling errors found during error checks. Overrides HybRecord "error\_mode" setting when using HybFoldIter. Options: "raise": Raise an error when encountered; "warn\_return": Print a warning and return the value; "warn\_skip": Print a warning and continue to the next iteration; "skip": Continue to the next iteration without any output; "return": return the value without any error output;

Default: "warn\_skip"

#### **--max\_sequential\_skips**

Maximum number of record(-pairs) to skip in a row. Limited as several sequential skips usually indicates an issue with record formatting or a desynchronization between files.

Default: 100

### **Output File Naming:**

Output files can be named in two fashions: via automatic name generation, or by providing specific out file names.

#### **Automatic Name Generation:**

For output name generation, the default respective naming scheme is used:

```
hyb_script -i PATH_TO/MY_FILE_1.HYB [...]
--> OUT_DIR/MY_FILE_1_ADDSUFFIX.HYB
```

This output file path can be modified with the arguments {--out\_dir, --out\_suffix} described below.

The output directory defaults to the current working directory (\$PWD), and can be modified with the --out\_dir <dir> argument. Note: The provided directory must exist, or an error will be raised. For Example:

```
hyb_script -i PATH_TO/MY_FILE_1.HYB [...] --out_dir MY_OUT_DIR
--> MY_OUT_DIR/MY_FILE_1_ADDSUFFIX.HYB
```

The suffix used for output files is based on the primary actions of the script. It can be specified using `--out_suffix <suffix>`. This can optionally include the ".hyb" final suffix. for Example:

```
hyb_script -i PATH_TO/MY_FILE_1.HYB [...] --out_suffix MY_SUFFIX
--> OUT_DIR/MY_FILE_1_MY_SUFFIX.HYB
#OR
hyb_script -i PATH_TO/MY_FILE_1.HYB [...] --out_suffix MY_SUFFIX.HYB
--> OUT_DIR/MY_FILE_1_MY_SUFFIX.HYB
```

#### Specific Output Names:

Alternatively, specific file names can be provided via the `-o/--out_hyb` argument, ensuring that the same number of input and output files are provided. This argument takes precedence over all automatic output file naming options (`--out_dir`, `--out_suffix`), which are ignored if `-o/--out_hyb` is provided. For Example:

```
hyb_script [...] --out_hyb MY_OUT_DIR/OUT_FILE_1.HYB MY_OUT_DIR/OUT_FILE_2.HYB
--> MY_OUT_DIR/OUT_FILE_1.hyb
--> MY_OUT_DIR/OUT_FILE_2.hyb
```

Note: The directory provided with output file paths (MY\_OUT\_DIR above) must exist, otherwise an error will be raised.

## 3.3 hyb\_eval

Read hyb files (and optional matched fold files) and evaluate the contained hybrids.

This utility reads in one or more files in hyb-format (see the [hybkit Hyb File Specification](#) (page 5)) and corresponding fold files (.vienna or .ct) and evaluates hybrid record properties.

Evaluation Types:

type	Assigns types to each segment within hyb records
mirna	Assigns which segments are a miRNA based on segment types.

#### type Evaluation:

The 'type' evaluation utilizes the `hybkit.HybRecord.eval_types()` (page 14) method to assign the record flags: `seg1_type` (page 6) and `seg2_type` (page 6)

#### Example system calls:

```
$ hyb_eval -t type -i my_file_1.hyb

$ hyb_eval -t type -i my_file_1.hyb -f my_file_1.vienna

$ hyb_eval -t type \\\
-i my_file_1.hyb my_file_2.hyb \\\
-f my_file_1.vienna my_file_2.vienna \\\
--type_method string_match \\\
--type_parameters my_parameters_file.csv \\\
--allow_unknown_seg_types
```

**mirna Evaluation:**

The 'mirna' evaluation uses the `hybkit.HybRecord.eval_mirna()` (page 15) method to identify properties relating to mirna within the hybrids, including mirna presence and positions. This evaluation requires the `seg_type` flags to be filled, either by a type evaluation, or by parsing the read using the `--hybformat_ref` `True` option with a hyb-format reference. The `mirna_seg` (page 6) flag is then set for each record, indicating the presence and position of any miRNA within the hybrid.

**Example system calls:**

```
$ hyb_eval -t mirna -i my_file_1.hyb

$ hyb_eval -t mirna -i my_file_1.hyb -f my_file_1.vienna

$ hyb_eval -t mirna -i my_file_1.hyb my_vile_2.hyb \\\
  -f my_file_1.vienna my_file_2.vienna \\\
  --mirna_types miRNA kshv-miRNA
```

This can also be combined with the type evaluation, as such:

```
$ hyb_eval -t type mirna -i my_file_1.hyb \\\
  --type_method string_match \\\
  --type_parameters my_parameters_file.csv \\\
  --allow_unknown_seg_types \\\
  --mirna_types miRNA kshv-miRNA
```

```
usage: hyb_analysis [-h] -i PATH_TO/MY_FILE.HYB [PATH_TO/MY_FILE.HYB ...]
                  [-f [PATH_TO/MY_FILE.VIENNA [PATH_TO/MY_FILE.VIENNA ...]]]
                  [-o PATH_TO/OUT_FILE.HYB [PATH_TO/OUT_FILE.HYB ...]]
                  [-l PATH_TO/OUT_FILE.VIENNA [PATH_TO/OUT_FILE.VIENNA ...]]
                  [-d OUT_DIR] [-u OUT_SUFFIX]
                  [-t {type,mirna} [{type,mirna} ...]]
                  [--type_method {hybformat,string_match,id_map}]
                  [--type_params_file PATH_TO/PARAMETERS_FILE]
                  [--set_dataset] [--version] [-v | -s]
                  [--mirna_types MIRNA_TYPES [MIRNA_TYPES ...]]
                  [--custom_flags CUSTOM_FLAGS [CUSTOM_FLAGS ...]]
                  [--hyb_placeholder HYB_PLACEHOLDER]
                  [--reorder_flags {True,False}]
                  [--allow_undefined_flags [{True,False}]]
                  [--allow_unknown_seg_types [{True,False}]]
                  [--hybformat_id [{True,False}]]
                  [--hybformat_ref [{True,False}]]
                  [--allowed_mismatches ALLOWED_MISMATCHES]
                  [--fold_placeholder FOLD_PLACEHOLDER]
                  [-y {static,dynamic}]
                  [--error_mode {raise,warn_return,return}]
                  [--error_checks {hybrecord_indel,foldrecord_nofold,max_mismatch,
↪energy_mismatch}]
                  [--iter_error_mode {raise,warn_return,warn_skip,skip,return}]
                  [--max_sequential_skips MAX_SEQUENTIAL_SKIPS]
```



### 3.3.1 Named Arguments

<b>-i, --in_hyb</b>	REQUIRED path to one or more hyb-format files with a ".hyb" suffix for use in the evaluation.
<b>-f, --in_fold</b>	REQUIRED path to one or more RNA secondary-structure files with a ".vienna" or ".ct" suffix for use in the evaluation.
<b>-o, --out_hyb</b>	Optional path to one or more hyb-format file for output (should include a ".hyb" suffix). If not provided, the output for input file "PATH_TO/MY_FILE.HYB" will be used as a template for the output "OUT_DIR/MY_FILE_OUT.HYB".
<b>-l, --out_fold</b>	Optional path to one or more ".vienna" or ".ct"-format files for output (should include appropriate ".vienna"/".ct" suffix). If not provided, the output for input file "PATH_TO/MY_FILE.VIENNA" will be used as a template for the output "OUT_DIR/MY_FILE_OUT.VIENNA".
<b>-d, --out_dir</b>	Path to directory for output of files. Defaults to the current working directory.
	Default: \$PWD
<b>-u, --out_suffix</b>	Suffix to add to the name of output files, before any file- or analysis-specific suffixes. The file-type appropriate suffix will be added automatically.

	Default: "_evaluated"	
<b>-t, --eval_types</b>	Possible choices: type, mirna	Types of evaluations to perform on input hyb file. (Note: evaluations can be combined, such as "--eval_types type mirna")
	Default: ['type']	
<b>--set_dataset</b>		Set "dataset" flag to value of the input file name.
	Default: False	
<b>--version</b>		Print version and exit.
<b>-v, --verbose</b>		Print verbose output during run.
	Default: False	
<b>-s, --silent</b>		Print no output during run.
	Default: False	

### 3.3.2 type Analysis Options

<b>--type_method</b>	Possible choices: hybformat, string_match, id_map	Segment-type finding method to use for type evaluation. For a description of the different methods, see the HybRecord documentation for the eval_types method.
	Default: "hybformat"	
<b>--type_params_file</b>		Segment-type finding parameters file to use for type evaluation with some type finding methods: {string_match, id_map}. For a description of the different methods, see the HybRecord documentation for

the `find_seg_types` method.

### 3.3.3 Hyb Record Settings

<b>--mirna_types</b>	"seg_type" fields identifying a miRNA Default: ['miRNA', 'microRNA']
<b>--custom_flags</b>	Custom flags to allow in addition to those specified in the hybkit specification. Default: []
<b>--hyb_placeholder</b>	placeholder character/string for missing data in hyb files. Default: "."
<b>--reorder_flags</b>	Possible choices: True, False Re-order flags to the hybkit-specification order when writing hyb records. Default: True
<b>--allow_undefined_flags</b>	Possible choices: True, False Allow use of flags not defined in the hybkit-specification order when reading and writing hyb records. As the preferred alternative to using this setting, the <code>--custom_flags</code> argument can be used to supply custom allowed flags. Default: False
<b>--allow_unknown_seg_types</b>	Possible choices: True, False Allow unknown segment types when assigning segment types. Default: False

### 3.3.4 Hyb File Settings

<b>--hybformat_id</b>	Possible choices: True, False	The Hyb Software Package places further information in the "id" field of the hybrid record that can be used to infer the number of contained read counts. When set to True, the identifiers will be parsed as: " <code>&lt;read_id&gt;_&lt;read_count&gt;</code> "
	Default: False	
<b>--hybformat_ref</b>	Possible choices: True, False	The Hyb Software Package uses a reference database with identifiers that contain sequence type and other sequence information. When set to True, all hyb file identifiers will be parsed as: " <code>&lt;gene_id&gt;_&lt;transcript_id&gt;_&lt;gene_name&gt;_&lt;seg_type&gt;</code> "
	Default: False	

### 3.3.5 Fold Record Settings

<b>--allowed_mismatches</b>	For DynamicFoldRecords, allowed number of mismatches with a HybRecord.
	Default: 0
<b>--fold_placeholder</b>	Placeholder character/string for missing data for reading/writing fold records.
	Default: "."
<b>-y, --seq_type</b>	Possible choices: static, dynamic
	Type of fold record object to use. Options: "static":

		<p>FoldRecord, requires an exact sequence match to be paired with a HybRecord;</p> <p>"dynamic": DynamicFoldRecord, requires a sequence match to the "dynamic" annotated regions of a HybRecord, and may be shorter/longer than the original sequence.</p>
	Default: "static"	
<b>--error_mode</b>	Possible choices: raise, warn_return, return	<p>Mode for handling errors during reading of HybFiles (overridden by HybFoldIter.settings['iter_error_mode'] when using HybFoldIter). Options:</p> <p>"raise": Raise an error when encountered and exit program ;</p> <p>"warn_return": Print a warning and return the error_value ;</p> <p>"return": Return the error value with no program output. record is encountered.</p>
	Default: "raise"	

### 3.3.6 Hyb-Fold Iterator Settings

<b>--error_checks</b>	Possible choices: hybrecord_indel, foldrecord_nofold, max_mismatch, energy_mismatch	<p>Error checks for simultaneous HybFile and FoldFile parsing. Options:</p> <p>"hybrecord_indel": Error for HybRecord objects where one/both sequences have insertions/deletions</p>
-----------------------	---	--

in alignment, which prevents matching of sequences; "foldrecord\_nofold": Error when failure in reading a fold\_record object; "max\_mismatch": Error when mismatch between hybrecord and foldrecord sequences is greater than FoldRecord "allowed\_mismatches" setting; "energy\_mismatch": Error when a mismatch exists between HybRecord and FoldRecord energy values.

Default: ['hybrecord\_indel', 'foldrecord\_nofold', 'max\_mismatch', 'energy\_mismatch']

**--iter\_error\_mode** Possible choices: raise, warn\_return, warn\_skip, skip, return

Mode for handling errors found during error checks. Overrides HybRecord "error\_mode" setting when using HybFoldIter. Options: "raise": Raise an error when encountered; "warn\_return": Print a warning and return the value; "warn\_skip": Print a warning and continue to the next iteration; "skip": Continue to the next iteration without any output; "return": return the value without any error output;

Default: "warn\_skip"

**--max\_sequential\_skips**

Maximum number of record(-pairs) to skip in a row. Lim-

ited as several sequential skips usually indicates an issue with record formatting or a desynchronization between files.

Default: 100

### Output File Naming:

Output files can be named in two fashions: via automatic name generation, or by providing specific out file names.

#### Automatic Name Generation:

For output name generation, the default respective naming scheme is used:

```
hyb_script -i PATH_TO/MY_FILE_1.HYB [...]
--> OUT_DIR/MY_FILE_1_ADDSUFFIX.HYB
```

This output file path can be modified with the arguments {--out\_dir, --out\_suffix} described below.

The output directory defaults to the current working directory (\$PWD), and can be modified with the --out\_dir <dir> argument. Note: The provided directory must exist, or an error will be raised. For Example:

```
hyb_script -i PATH_TO/MY_FILE_1.HYB [...] --out_dir MY_OUT_DIR
--> MY_OUT_DIR/MY_FILE_1_ADDSUFFIX.HYB
```

The suffix used for output files is based on the primary actions of the script. It can be specified using --out\_suffix <suffix>. This can optionally include the ".hyb" final suffix. for Example:

```
hyb_script -i PATH_TO/MY_FILE_1.HYB [...] --out_suffix MY_SUFFIX
--> OUT_DIR/MY_FILE_1_MY_SUFFIX.HYB
#OR
hyb_script -i PATH_TO/MY_FILE_1.HYB [...] --out_suffix MY_SUFFIX.HYB
--> OUT_DIR/MY_FILE_1_MY_SUFFIX.HYB
```

#### Specific Output Names:

Alternatively, specific file names can be provided via the -o/--out\_hyb argument, ensuring that the same number of input and output files are provided. This argument takes precedence over all automatic output file naming options (--out\_dir, --out\_suffix), which are ignored if -o/--out\_hyb is provided. For Example:

```
hyb_script [...] --out_hyb MY_OUT_DIR/OUT_FILE_1.HYB MY_OUT_DIR/OUT_FILE_2.HYB
--> MY_OUT_DIR/OUT_FILE_1.hyb
--> MY_OUT_DIR/OUT_FILE_2.hyb
```

Note: The directory provided with output file paths (MY\_OUT\_DIR above) must exist, otherwise an error will be raised.

## 3.4 hyb\_analyze

Read hyb / vienna files and analyze the fold information in the contained hybrid sequences.

Analysis Types:

<a href="#">Energy</a> (page 40)	Analysis of values of predicted intra-hybrid folding energy
<a href="#">Type</a> (page 40)	Analysis of segment types
<a href="#">miRNA</a> (page 41)	Analysis of miRNA segments distributions
<a href="#">Target</a> (page 41)	Analysis of mirna target segment names and types
<a href="#">Fold</a> (page 42)	Analysis of folding data included in the analyzed hyb_records

This utility reads in one or more files in hyb-format (see the [hybkit Hyb File Specification](#) (page 5)) along with a corresponding predicted RNA secondary structure file in the "Vienna" ([Vienna Format](#) (page 23)) or "CT" ([CT\\_Format](#) (page 23)) formats, and analyzes hybrid secondary structure properties.

For full information on the different analysis types, see the [Analyses](#) (page 40) section of the hybkit documentation.

**Example system calls:**

```
$ hyb_analyze -a fold -i my_file_1.hyb -f my_file_1.vienna

$ hyb_analyze -a fold -i my_file_2.hyb -f my_file_2.ct \\  
--make_plots False
```

```
usage: hyb_analysis [-h] -i PATH_TO/MY_FILE.HYB [PATH_TO/MY_FILE.HYB ...]
                  [-f [PATH_TO/MY_FILE.VIENNA [PATH_TO/MY_FILE.VIENNA ...]]]
                  [-o PATH_TO/OUT_BASENAME [PATH_TO/OUT_BASENAME ...]]
                  [-d OUT_DIR] [-u OUT_SUFFIX]
                  [-a {energy,type,mirna,target,fold} [{energy,type,mirna,target,fold}
↪ ...]]
                  [-n ANALYSIS_NAME] [-p {True,False}] [--version] [-v | -s]
                  [--mirna_types MIRNA_TYPES [MIRNA_TYPES ...]]
                  [--custom_flags CUSTOM_FLAGS [CUSTOM_FLAGS ...]]
                  [--hyb_placeholder HYB_PLACEHOLDER]
                  [--reorder_flags {True,False}]
                  [--allow_undefined_flags [{True,False}]]
                  [--allow_unknown_seg_types [{True,False}]]
                  [--hybformat_id [{True,False}]]
                  [--hybformat_ref [{True,False}]]
                  [--allowed_mismatches ALLOWED_MISMATCHES]
                  [--fold_placeholder FOLD_PLACEHOLDER]
                  [-y {static,dynamic}]
                  [--error_mode {raise,warn_return,return}]
                  [--error_checks {hybrecord_indel,foldrecord_nofold,max_mismatch,
↪ energy_mismatch}]
                  [--iter_error_mode {raise,warn_return,warn_skip,skip,return}]
                  [--max_sequential_skips MAX_SEQUENTIAL_SKIPS]
                  [--quant_mode {single,reads,records}]
                  [--out_delim OUT_DELIM]
```



### 3.4.1 Named Arguments

<b>-i, --in_hyb</b>	REQUIRED path to one or more hyb-format files with a ".hyb" suffix for use in the evaluation.
<b>-f, --in_fold</b>	REQUIRED path to one or more RNA secondary-structure files with a ".vienna" or ".ct" suffix for use in the evaluation.
<b>-o, --out_basename</b>	Optional path to one or more basename prefixes to use for output. The appropriate suffix will be added based on the specific name. If not provided, the output for input file "PATH_TO/MY_FILE.HYB" will be used as a template for the basename "OUT_DIR/MY_FILE".
<b>-d, --out_dir</b>	Path to directory for output of files. Defaults to the current working directory.
	Default: \$PWD
<b>-u, --out_suffix</b>	Suffix to add to the name of output files, before any file- or analysis-specific suffixes. The file-type appropriate suffix will be added automatically.
<b>-a, --analysis_types</b>	Possible choices: energy, type, mirna, target, fold  Analysis to perform on input hyb and fold files.
	Default: "fold"
<b>-n, --analysis_name</b>	Name / title of analysis data.
<b>-p, --make_plots</b>	Possible choices: True, False

		Create plots of analysis output.
	Default: True	
<b>--version</b>		Print version and exit.
<b>-v, --verbose</b>		Print verbose output during run.
	Default: False	
<b>-s, --silent</b>		Print no output during run.
	Default: False	

### 3.4.2 Hyb Record Settings

<b>--mirna_types</b>	"seg_type" fields identifying a miRNA Default: ['miRNA', 'microRNA']	
<b>--custom_flags</b>		Custom flags to allow in addition to those specified in the hybkit specification.
	Default: []	
<b>--hyb_placeholder</b>		placeholder character/string for missing data in hyb files.
	Default: "."	
<b>--reorder_flags</b>	Possible choices: True, False	Re-order flags to the hybkit-specification order when writing hyb records.
	Default: True	
<b>--allow_undefined_flags</b>	Possible choices: True, False	Allow use of flags not defined in the hybkit-specification order when reading and writing hyb records. As the preferred alternative to using this setting, the <code>--custom_flags</code> argument can be used to supply custom allowed flags.

Default: False

**--allow\_unknown\_seg\_types** Possible choices: True, False

Allow unknown segment types when assigning segment types.

Default: False

### 3.4.3 Hyb File Settings

**--hybformat\_id** Possible choices: True, False

The Hyb Software Package places further information in the "id" field of the hybrid record that can be used to infer the number of contained read counts. When set to True, the identifiers will be parsed as: "<read\_id>\_<read\_count>"

Default: False

**--hybformat\_ref** Possible choices: True, False

The Hyb Software Package uses a reference database with identifiers that contain sequence type and other sequence information. When set to True, all hyb file identifiers will be parsed as: "<gene\_id>\_<transcript\_id>\_<gene\_name>\_<seg\_type>"

Default: False

### 3.4.4 Fold Record Settings

**--allowed\_mismatches**

For DynamicFoldRecords, allowed number of mismatches with a HybRecord.

Default: 0

<b>--fold_placeholder</b>	Placeholder character/string for missing data for reading/writing fold records.
	Default: "."
<b>-y, --seq_type</b>	Possible choices: static, dynamic
	Type of fold record object to use. Options: "static": FoldRecord, requires an exact sequence match to be paired with a HybRecord; "dynamic": DynamicFoldRecord, requires a sequence match to the "dynamic" annotated regions of a HybRecord, and may be shorter/longer than the original sequence.
	Default: "static"
<b>--error_mode</b>	Possible choices: raise, warn_return, return
	Mode for handling errors during reading of HybFiles (overridden by HybFoldIter.settings['iter_error_mode'] when using HybFoldIter). Options: "raise": Raise an error when encountered and exit program ; "warn_return": Print a warning and return the error_value ; "return": Return the error value with no program output. record is encountered.
	Default: "raise"

### 3.4.5 Hyb-Fold Iterator Settings

**--error\_checks** Possible choices: hybrecord\_indel, foldrecord\_nofold, max\_mismatch, energy\_mismatch

Error checks for simultaneous HybFile and FoldFile parsing. Options: "hybrecord\_indel": Error for HybRecord objects where one/both sequences have insertions/deletions in alignment, which prevents matching of sequences; "foldrecord\_nofold": Error when failure in reading a fold\_record object; "max\_mismatch": Error when mismatch between hybrecord and foldrecord sequences is greater than FoldRecord "allowed\_mismatches" setting; "energy\_mismatch": Error when a mismatch exists between HybRecord and FoldRecord energy values.

Default: ['hybrecord\_indel', 'foldrecord\_nofold', 'max\_mismatch', 'energy\_mismatch']

**--iter\_error\_mode** Possible choices: raise, warn\_return, warn\_skip, skip, return

Mode for handling errors found during error checks. Overrides HybRecord "error\_mode" setting when using HybFoldIter. Options: "raise": Raise an error when encountered; "warn\_return": Print a warning and return the value; "warn\_skip": Print

a warning and continue to the next iteration; "skip": Continue to the next iteration without any output; "return": return the value without any error output;

Default: "warn\_skip"

**--max\_sequential\_skips**

Maximum number of record(-pairs) to skip in a row. Limited as several sequential skips usually indicates an issue with record formatting or a desynchronization between files.

Default: 100

3.4.6 Analysis Settings

**--quant\_mode**

Possible choices: single, reads, records

Method for counting records. Options: "single": Count each record as a single entry; "reads": Use the number of reads per hyb record as the count (may contain PCR duplicates); "records": Count the number of records represented by each hyb record entry (1 for "unmerged" records, >= 1 for "merged" records)

Default: "single"

**--out\_delim**

Delimiter-string to place between fields in analysis output.

Default: ","

**Output File Naming:**

Output files can be named in two fashions: via automatic name generation, or by providing specific out file names.

**Automatic Name Generation:**

For output name generation, the default respective naming scheme is used:

```
hyb_script -i PATH_TO/MY_FILE_1.HYB [...]
--> OUT_DIR/MY_FILE_1_ADDSUFFIX.HYB
```

This output file path can be modified with the arguments {--out\_dir, --out\_suffix} described below.

The output directory defaults to the current working directory (\$PWD), and can be modified with the --out\_dir <dir> argument. Note: The provided directory must exist, or an error will be raised. For Example:

```
hyb_script -i PATH_TO/MY_FILE_1.HYB [...] --out_dir MY_OUT_DIR
--> MY_OUT_DIR/MY_FILE_1_ADDSUFFIX.HYB
```

The suffix used for output files is based on the primary actions of the script. It can be specified using --out\_suffix <suffix>. This can optionally include the ".hyb" final suffix. for Example:

```
hyb_script -i PATH_TO/MY_FILE_1.HYB [...] --out_suffix MY_SUFFIX
--> OUT_DIR/MY_FILE_1_MY_SUFFIX.HYB
#OR
hyb_script -i PATH_TO/MY_FILE_1.HYB [...] --out_suffix MY_SUFFIX.HYB
--> OUT_DIR/MY_FILE_1_MY_SUFFIX.HYB
```

**Specific Output Names:**

Alternatively, specific file names can be provided via the -o/--out\_hyb argument, ensuring that the same number of input and output files are provided. This argument takes precedence over all automatic output file naming options (--out\_dir, --out\_suffix), which are ignored if -o/--out\_hyb is provided. For Example:

```
hyb_script [...] --out_hyb MY_OUT_DIR/OUT_FILE_1.HYB MY_OUT_DIR/OUT_FILE_2.HYB
--> MY_OUT_DIR/OUT_FILE_1.hyb
--> MY_OUT_DIR/OUT_FILE_2.hyb
```

Note: The directory provided with output file paths (MY\_OUT\_DIR above) must exist, otherwise an error will be raised.

## EXAMPLE ANALYSES

This section includes multiple example stepwise analyses of data from a qCLASH experiment described in [Gay2018], with data acquired from the NCBI Gene Expression Omnibus (GEO) accession GSE101978<sup>808</sup>.

Each analysis is implemented both using the Python3 API, and as a sequence of shell executable commands in a bash script. The Python API implementations are generally significantly more efficient as more steps can be performed on a single iteration over the input data.

Each analysis performs quality control steps on the data by checking data integrity (*hyb\_check* (page 61)) and removing artifactual ribosomal- and mitochondrial-RNA hybrids (*hyb\_filter* (page 67)). Further filtration may be performed, and then each described analysis is carried out.

Pipeline	Description
<i>Example Type-miRNA Analysis</i> (page 94)	Quantify the sequence and miRNA types in a hyb file
<i>Example Target Analysis</i> (page 97)	Analyze targets of a set of miRNAs from a single experiment
<i>Example Grouped Target Analysis</i> (page 98)	Analyze and plot targets of a set of miRNAs from pooled experimental replicates
<i>Example Fold Analysis</i> (page 99)	Analyze and plot predicted miRNA folding patterns in miRNA-containing hybrids

Further details on each respective example analysis can be found in each section.

### 4.1 Example Type-miRNA Analysis

This directory contains an example analysis of Hyb-format data, published in the quick Crosslinking and Sequencing of Hybrids (qCLASH) experiment described in: Gay, Lauren A., et al. "Modified cross-linking, ligation, and sequencing of hybrids (qCLASH) identifies Kaposi's Sarcoma-associated herpesvirus microRNA targets in endothelial cells." *Journal of virology* 92.8 (2018): e02138-17.

**The analysis is carried out in multiple example implementations which produce identical output:**

- via the *Command-Line*<sup>809</sup>
- via the *Python3 API*<sup>810</sup>

---

<sup>808</sup> <https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE101978>

<sup>809</sup> [https://github.com/RenneLab/hybkit/blob/master/example\\_01\\_type\\_mirna\\_analysis/analysis\\_shell.sh/](https://github.com/RenneLab/hybkit/blob/master/example_01_type_mirna_analysis/analysis_shell.sh/)

<sup>810</sup> [https://github.com/RenneLab/hybkit/blob/master/example\\_01\\_type\\_mirna\\_analysis/analysis\\_python.py/](https://github.com/RenneLab/hybkit/blob/master/example_01_type_mirna_analysis/analysis_python.py/)



This analysis first performs quality control on the data. It then summarizes and analyzes the hybrid, segment, and miRNA characteristics of each input file. Analyses for each individual file, and a combined summary analysis are all produced.

The sequencing information is available at NCBI Gene Expression Omnibus (GEO) GSE101978, at:

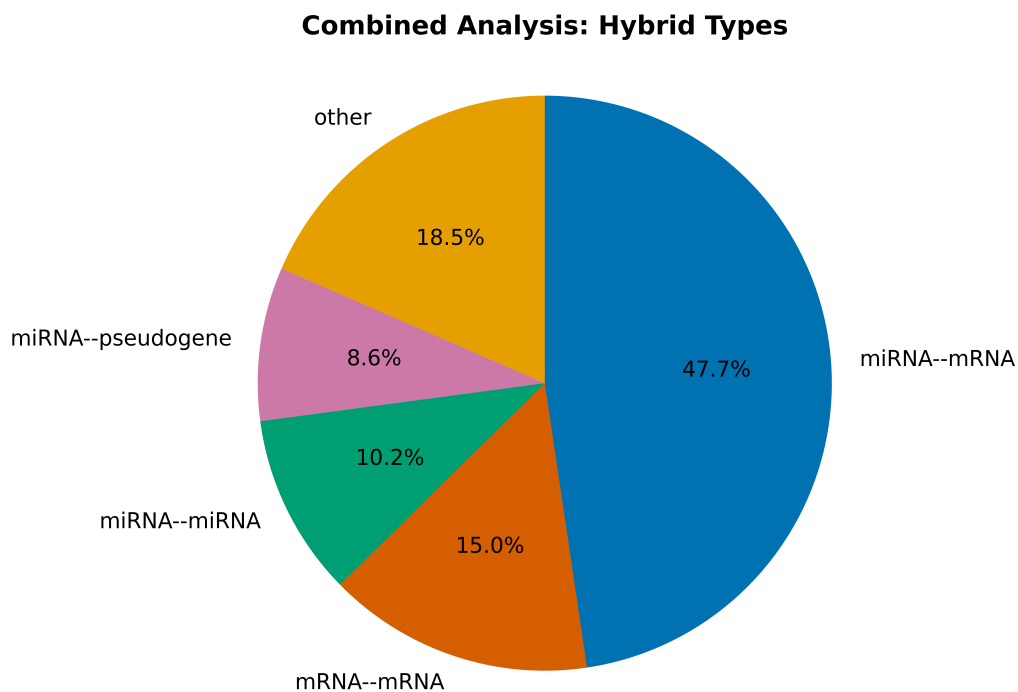
<https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE101978>

The data files can be downloaded and uncompressed by using the command:

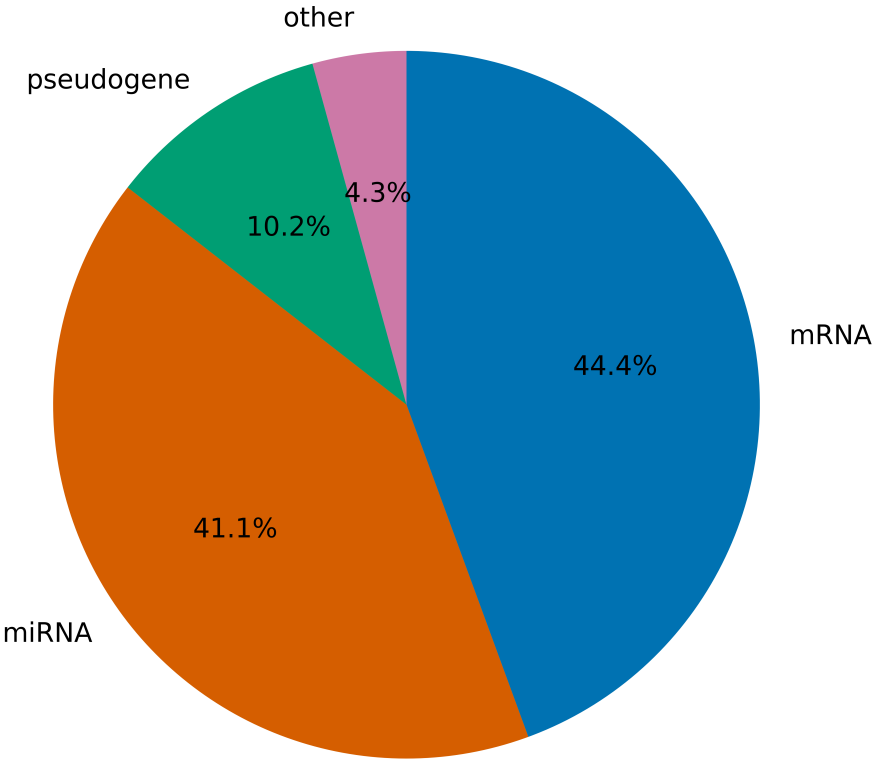
```
$ sh ./download_data.sh
```

The unpacked hyb data-files require ~2 GB of space. The completed output of the analysis requires ~1.5 GB of space.

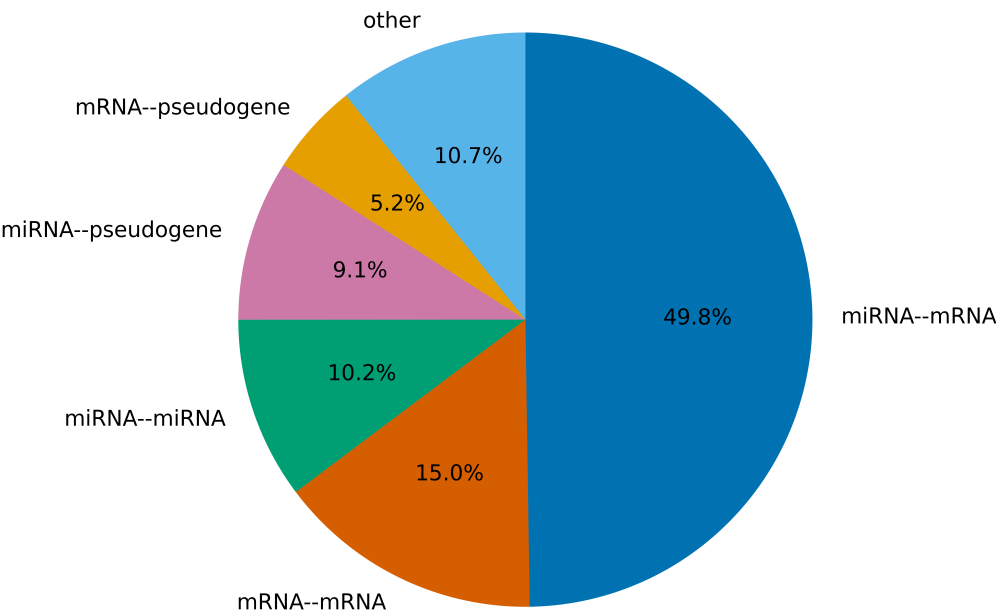
#### 4.1.1 Type-miRNA Analysis Example Output



Combined Analysis: All Segment Types



Combined Analysis: Reordered miRNA' Hybrid Types



## 4.2 Example Target Analysis

This directory contains an example analysis of Hyb-format data, published in the quick Crosslinking and Sequencing of Hybrids (qCLASH) experiment described in: Gay, Lauren A., et al. "Modified cross-linking, ligation, and sequencing of hybrids (qCLASH) identifies Kaposi's Sarcoma-associated herpesvirus microRNA targets in endothelial cells." *Journal of virology* 92.8 (2018): e02138-17.

The analysis is carried out in multiple example implementations which produce identical output:

- via the [Command-Line](#)<sup>811</sup>
- via the [Python3 API](#)<sup>812</sup>

This analysis specifically filters and analyzes the kshv-miR-K12-5 miRNA arising from Kaposi's Sarcoma-Associated Herpesvirus (KSHV), which has the assigned type "KSHV-miRNA". Both individual and summary output files are produced.

Hybrid sequences generated by the Hyb program are available at NCBI Gene Expression Omnibus (GEO) GSE101978, at:

<https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE101978>

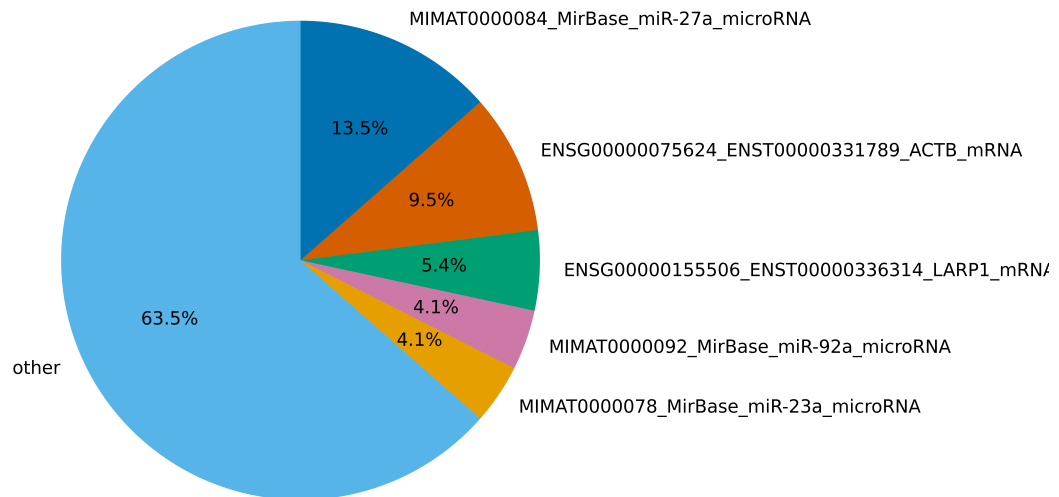
The data files can be downloaded and uncompressed by using the command:

```
$ sh ./download_data.sh
```

The unpacked hyb data-file require ~130 MB of space. The completed output of the analysis requires ~20 MB of space.

### 4.2.1 Target Analysis Example Output

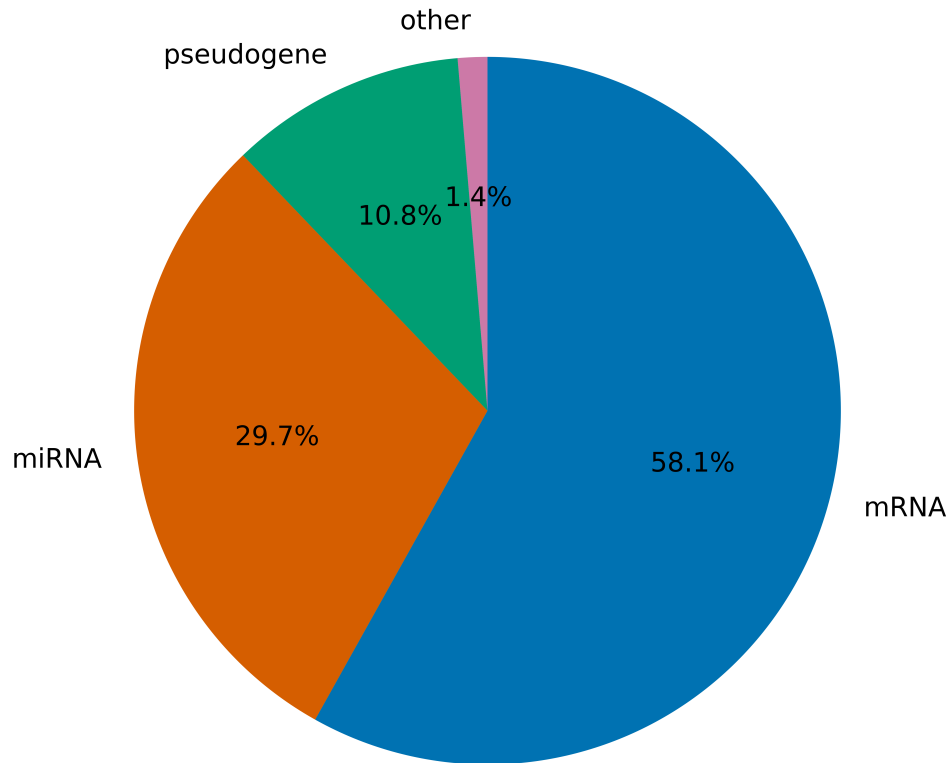
**GSM2720020\_WT\_BR1: miRNA Target Names**



<sup>811</sup> [https://github.com/RenneLab/hybkit/blob/master/example\\_02\\_target\\_analysis/analysis\\_shell.sh/](https://github.com/RenneLab/hybkit/blob/master/example_02_target_analysis/analysis_shell.sh/)

<sup>812</sup> [https://github.com/RenneLab/hybkit/blob/master/example\\_02\\_target\\_analysis/analysis\\_python.py/](https://github.com/RenneLab/hybkit/blob/master/example_02_target_analysis/analysis_python.py/)

### GSM2720020\_WT\_BR1: miRNA Target Types



## 4.3 Example Grouped Target Analysis

This directory contains an example analysis of Hyb-format data, published in the quick Crosslinking and Sequencing of Hybrids (qCLASH) experiment described in: Gay, Lauren A., et al. "Modified cross-linking, ligation, and sequencing of hybrids (qCLASH) identifies Kaposi's Sarcoma-associated herpesvirus microRNA targets in endothelial cells." *Journal of virology* 92.8 (2018): e02138-17.

**The analysis is carried out in multiple example implementations which produce identical output:**

- via the [Command-Line](#)<sup>813</sup>
- via the [Python3 API](#)<sup>814</sup>

This analysis specifically investigates and characterizes miRNA arising from six experimental replicates from two conditions with cells infected with Kaposi's Sarcoma Herpesvirus, which are given the type name "KSHV\_miRNA". The hybrid reads from KSHV miRNA are grouped and analyzed together. Both individual and summary output files are produced.

Hybrid sequence information created by the Hyb program information is available at NCBI Gene Expression Omnibus (GEO) GSE101978, at:

<https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE101978>

<sup>813</sup> [https://github.com/RenneLab/hybkit/blob/master/example\\_03\\_grouped\\_target\\_analysis/analysis\\_shell.sh/](https://github.com/RenneLab/hybkit/blob/master/example_03_grouped_target_analysis/analysis_shell.sh/)

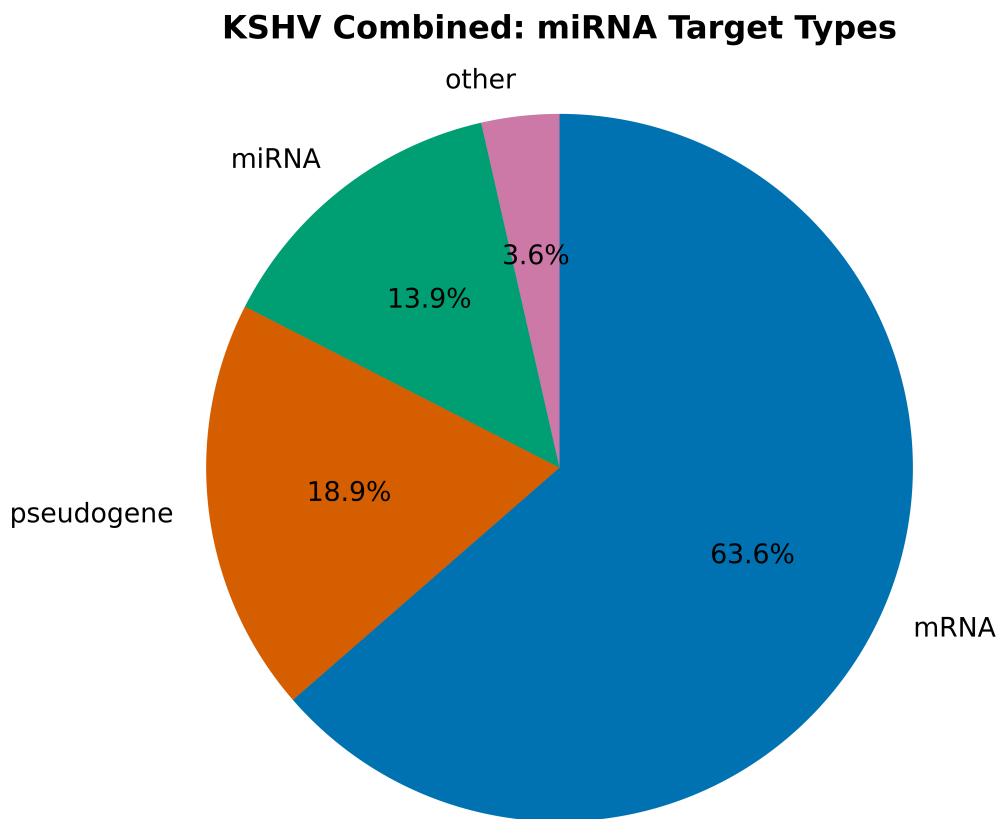
<sup>814</sup> [https://github.com/RenneLab/hybkit/blob/master/example\\_03\\_grouped\\_target\\_analysis/analysis\\_python.py/](https://github.com/RenneLab/hybkit/blob/master/example_03_grouped_target_analysis/analysis_python.py/)

The data files can be downloaded and uncompressed by using the command:

```
$ sh ./download_data.sh"
```

The unpacked hyb data-file require ~1.3 GB of space. The completed output of the analysis requires ~40 MB of space.

### 4.3.1 Grouped Target Analysis Example Output



## 4.4 Example Fold Analysis

This directory contains a example analysis of Hyb-format and Vienna-format data, published in the quick Crosslinking and Sequencing of Hybrids (qCLASH) experiment described in: Gay, Lauren A., et al. "Modified cross-linking, ligation, and sequencing of hybrids (qCLASH) identifies Kaposi's Sarcoma-associated herpesvirus microRNA targets in endothelial cells." Journal of virology 92.8 (2018): e02138-17.

**The analysis is carried out in multiple example implementations which produce identical output:**

- via the [Command-Line](#)<sup>815</sup>
- via the [Python3 API](#)<sup>816</sup>

<sup>815</sup> [https://github.com/RenneLab/hybkit/blob/master/example\\_04\\_fold\\_analysis/analysis\\_shell.sh/](https://github.com/RenneLab/hybkit/blob/master/example_04_fold_analysis/analysis_shell.sh/)

<sup>816</sup> [https://github.com/RenneLab/hybkit/blob/master/example\\_04\\_fold\\_analysis/analysis\\_python.py/](https://github.com/RenneLab/hybkit/blob/master/example_04_fold_analysis/analysis_python.py/)

This analysis investigates the predicted folding of miRNA from an experimental replicate infected with Kaposi's Sarcoma-Associated Herpesvirus (KSHV), which are given the type name "KSHV-miRNA". Data from the predicted folding fold for each hybrid record produced by the "Hyb" program are analyzed, and the folds of each KSHV miRNA with a non-miRNA target are characterized to determine the per-base folding folds.

Hybrid sequence information created by the Hyb program and the fold output are provided with the hybkit package in the databases directory, created by downstream analysis of files available at NCBI Gene Expression Omnibus (GEO) GSE101978, at:

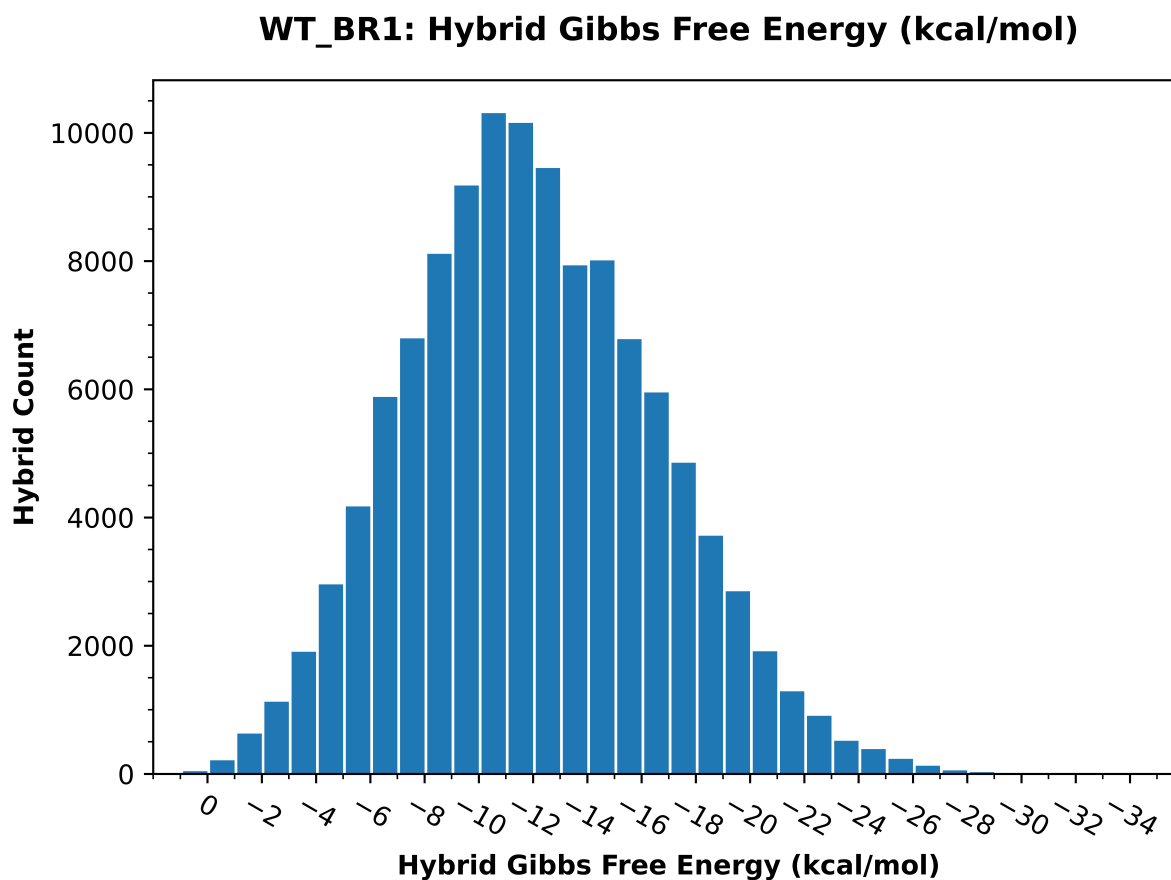
<https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE101978>

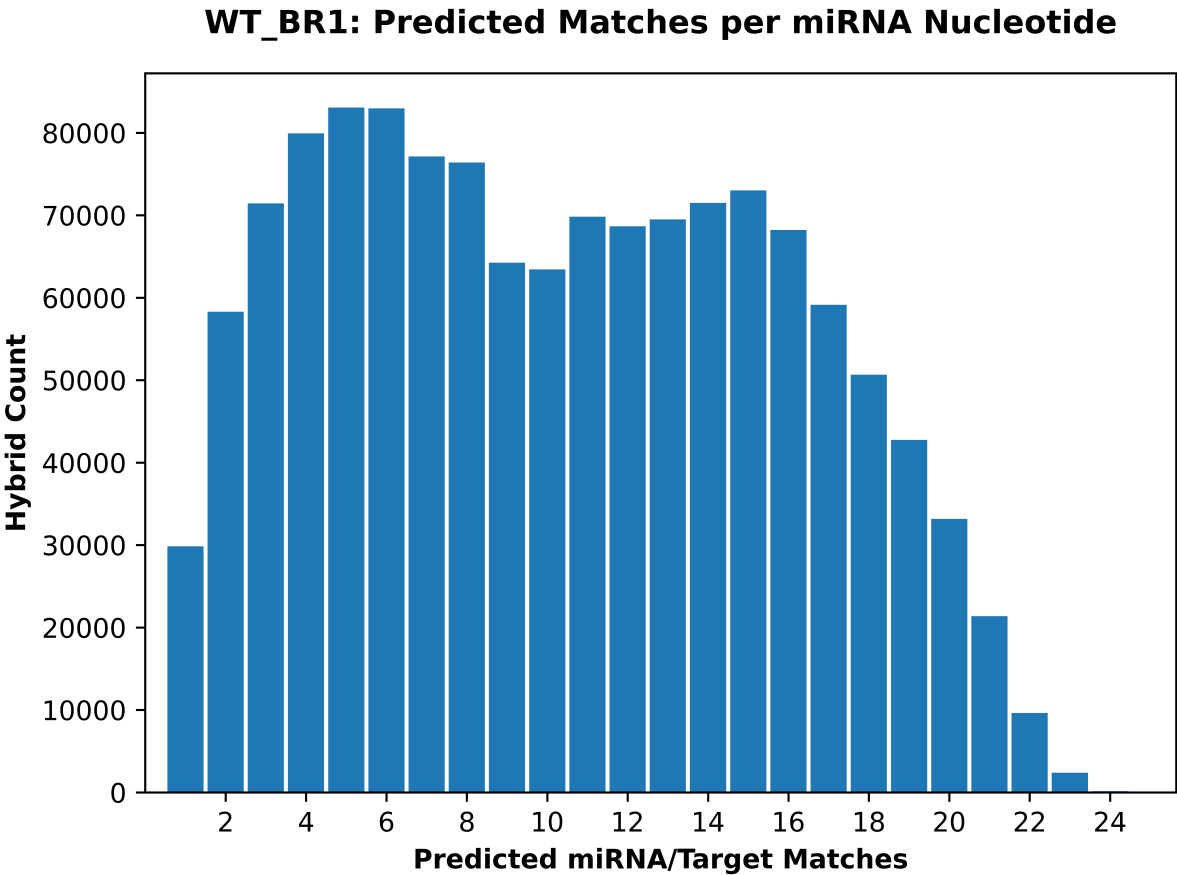
The data files can be copied and uncompressed by using the command:

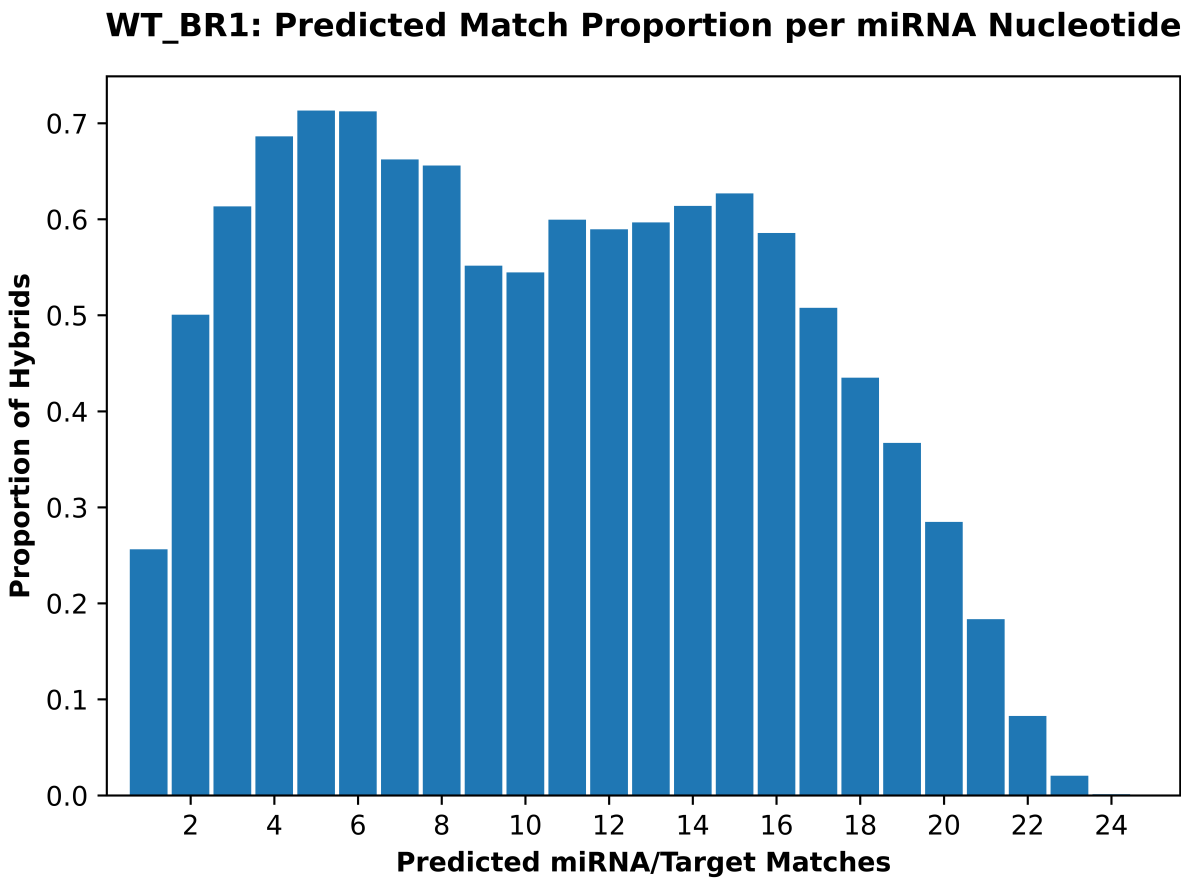
```
$ sh ./prepare_data.sh
```

The unpacked data-files require ~150 MB of space. The completed output of the analysis requires ~30 MB of space.

#### 4.4.1 Fold Analysis Example Output









## 5.1 Renne Lab

Principal Investigator: Rolf Renne  
Henry E. Innes Professor of Cancer Research  
University of Florida  
UF Health Cancer Center  
UF Department of Molecular Genetics and Microbiology  
UF Genetics Institute  
<http://www.rennelab.com>

## 5.2 Lead Developer

- Daniel Stribling <[ds@ufl.edu](mailto:ds@ufl.edu)>  
<https://www.danthescienceman.com>  
<https://orcid.org/0000-0002-0649-9506>  
University of Florida, Renne Lab

## 5.3 Changelog

- 0.3.4 (2023-11) Changes include:
  - Misc Bugfixes and Refinements
  - Switch code linting to Ruff
  - Add hybkit.errors module and HybkitError classes
  - Moved printing of warnings to python logging module
  - Add option for direct passage of file-like object for construction of HybFile and ViennaFile
  - Add HybRecord.to\_csv\_header(), HybRecord.to\_fields(), and HybRecord.to\_fields\_header() methods
  - Refine description of HybFile.open() constructor method
  - Add typing\_extensions dependency
  - Add Python3.8-compatible type hints to API
  - Documentation Updates

- 0.3.3 (2023-09) Changes include:
  - Misc Bugfixes and Refinements
  - Update integer bar-plot functions
- 0.3.2 (2023-08) Changes include:
  - Misc Bugfixes and Refinements
  - Add duplicate hybrid filtration (by HybRecord.id) options to hyb\_filter
  - Add duplicate hybrid filtration to example analyses
- 0.3.1 (2023-08) Changes include:
  - Misc Bugfixes and Refinements
  - Add --version flag to scripts
  - Change move scripts output file description to argparse epilog
  - Refine plot functions
  - Change default plot colors to the Bang Wong scheme [Wong2011] for colorblind accessibility
  - Documentation corrections
  - Spellcheck
- 0.3.0 (2023-04) Major Codebase And API Overhaul. Changes include:
  - Simplifying HybRecord API
  - Simplifying FoldRecord API
  - Unifying settings information for argparse and modules
  - Removing Support for ViennaD format
  - Moving identifier-parsing code to module type\_finder
  - Moving target region analysis code to module region\_finder
  - Moving code for settings into a "settings" module
  - Renamed HybRecord type\_analysis and mirna\_analysis to eval\_types and eval\_mirna, respectively to differentiate from analysis module functions
  - Reimplemented analyses methods within a single Analysis class
  - Added error checking / catching to HybFoldIter
  - Removed Target-Region Analysis and associated files due to lack of archival database information, pending future development
  - Added "dynamic" seq\_type to FoldRecord for non-identical fold/hybrid sequence handling
  - Added shell implementation to all example analyses
  - Remove support for Python3.6, Python3.7
  - Migrate to CircleCI for CI/CD
  - Added Pytest unit testing integrated with CircleCI
  - Other Misc. Improvements / Bugfixes
- 0.2.0 (2020-03) Added Command-line Toolkit. Code Refinements.
- 0.1.9 (2020-03) Fix for Module Path Finding for Python > 3.6

- 0.1.8 (2020-03) Streamlining, PyPI / PIP Initial Release
- 0.1.0 (2020-01) Initial Implementation

## REFERENCES

- [ViennaFormat]
- [CTFormat]
- [Zuker2003]
- [Hunter2007]
- [Cock2009]
- [Lorenz2011]
- [Wong2011]
- [Helwak2013]
- [Travis2014]
- [Gay2018]

## BIBLIOGRAPHY

- [ViennaFormat] ViennaRNA Vienna File Format Description<sup>817</sup>  
 UNAFold Vienna File Format Description<sup>818</sup>
- [CTFormat] UNAFold CT Format Description<sup>819</sup>  
 RNAstructure CT Format Description<sup>820</sup>
- [Zuker2003] Zuker M. Mfold web server for nucleic acid folding and hybridization prediction. *Nucleic Acids Res.* 2003 Jul 1;31(13):3406-15. doi: [10.1093/nar/gkg595](https://doi.org/10.1093/nar/gkg595)<sup>821</sup>. PMID: 12824337; PMCID: PMC169194.
- [Hunter2007] J. Hunter, "Matplotlib: A 2D Graphics Environment" in *Computing in Science & Engineering*, vol. 9, no. 03, pp. 90-95, 2007. doi: [10.1109/MCSE.2007.55](https://doi.org/10.1109/MCSE.2007.55)<sup>822</sup>
- [Cock2009] Cock PJ, Antao T, Chang JT, Chapman BA, Cox CJ, Dalke A, Friedberg I, Hamelryck T, Kauff F, Wilczynski B, de Hoon MJ. Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics*. 2009 Jun 1;25(11):1422-3. doi: [10.1093/bioinformatics/btp163](https://doi.org/10.1093/bioinformatics/btp163)<sup>823</sup>. Epub 2009 Mar 20. PMID: 19304878; PMCID: PMC2682512.
- [Lorenz2011] Lorenz, R., Bernhart, S.H., Höner zu Siederdissen, C. et al. ViennaRNA Package 2.0. *Algorithms Mol Biol* 6, 26 (2011). doi: [10.1186/1748-7188-6-26](https://doi.org/10.1186/1748-7188-6-26)<sup>824</sup>
- [Wong2011] Wong, B. Points of view: Color blindness. *Nat Methods* 8, 441 (2011). doi: [10.1038/nmeth.1618](https://doi.org/10.1038/nmeth.1618)<sup>825</sup>
- [Helwak2013] Helwak A, Kudla G, Dudnakova T, Tollervey D. Mapping the human miRNA interactome by CLASH reveals frequent noncanonical binding. *Cell*. 2013 Apr 25;153(3):654-65. doi: [10.1016/j.cell.2013.03.043](https://doi.org/10.1016/j.cell.2013.03.043)<sup>826</sup>. PMID: 23622248; PMCID: PMC3650559.
- [Travis2014] Travis AJ, et al. Hyb: a bioinformatics pipeline for the analysis of CLASH (crosslinking, ligation and sequencing of hybrids) data. *Methods*. 2014 Feb;65(3):263-73. doi: [10.1016/j.ymeth.2013.10.015](https://doi.org/10.1016/j.ymeth.2013.10.015)<sup>827</sup>.
- [Gay2018] Gay LA, Sethuraman S, Thomas M, Turner PC, Renne R. Modified Cross-Linking, Ligation, and Sequencing of Hybrids (qCLASH) Identifies Kaposi's Sarcoma-Associated Herpesvirus MicroRNA Targets in Endothelial Cells. *J Virol*. 2018 Mar 28;92(8):e02138-17. doi: [10.1128/JVI.02138-17](https://doi.org/10.1128/JVI.02138-17)<sup>828</sup>. PMID: 29386283; PMCID: PMC5874430.

<sup>817</sup> [https://www.tbi.univie.ac.at/RNA/tutorial/#sec2\\_7](https://www.tbi.univie.ac.at/RNA/tutorial/#sec2_7)

<sup>818</sup> <http://www.unafold.org/doc/formats.php#VIENNA>

<sup>819</sup> <http://www.unafold.org/doc/formats.php#CT>

<sup>820</sup> [https://rna.urmc.rochester.edu/Text/File\\_Formats.html#CT](https://rna.urmc.rochester.edu/Text/File_Formats.html#CT)

<sup>821</sup> <https://doi.org/10.1093/nar/gkg595>

<sup>822</sup> <https://doi.org/10.1109/MCSE.2007.55>

<sup>823</sup> <https://doi.org/10.1093/bioinformatics/btp163>

<sup>824</sup> <https://doi.org/10.1186/1748-7188-6-26>

<sup>825</sup> <https://doi.org/10.1038/nmeth.1618>

<sup>826</sup> <https://doi.org/10.1016/j.cell.2013.03.043>

<sup>827</sup> <https://doi.org/10.1016/j.ymeth.2013.10.015>

<sup>828</sup> <https://doi.org/10.1128/JVI.02138-17>

## PYTHON MODULE INDEX

### h

- `hyb_analyze`, 86
- `hyb_check`, 61
- `hyb_eval`, 77
- `hyb_filter`, 67
- `hybkit`, 8
- `hybkit.analysis`, 40
- `hybkit.errors`, 60
- `hybkit.plot`, 47
- `hybkit.settings`, 51
- `hybkit.type_finder`, 34
- `hybkit.util`, 56

## A

add\_hyb\_record() (*hybkit.analysis.Analysis* method), 42  
 add\_hyb\_records() (*hybkit.analysis.Analysis* method), 42  
 ALL\_FLAGS (*hybkit.HybRecord* attribute), 11  
 Analysis (*class* in *hybkit.analysis*), 40  
 analysis\_options (*hybkit.analysis.Analysis* attribute), 42  
 Analysis\_settings (*in module hybkit.settings*), 56  
 Analysis\_settings\_info (*in module hybkit.settings*), 55

## B

BAR\_DEFAULTS (*in module hybkit.plot*), 47  
 BAR\_INT\_DEFAULTS (*in module hybkit.plot*), 47

## C

check\_set\_method() (*hybkit.type\_finder.TypeFinder* class method), 35  
 close() (*hybkit.CtFile* method), 32  
 close() (*hybkit.HybFile* method), 21  
 close() (*hybkit.ViennaFile* method), 29  
 COLOR\_DICT (*in module hybkit.plot*), 47  
 COLOR\_LIST (*in module hybkit.plot*), 47  
 count\_hyb\_record\_mismatches() (*hybkit.FoldRecord* method), 25  
 ct\_exists() (*in module hybkit.util*), 57  
 CT\_SUFFIXES (*in module hybkit.settings*), 51  
 CtFile (*class* in *hybkit*), 31

## D

default\_method (*hybkit.type\_finder.TypeFinder* attribute), 34  
 dir\_exists() (*in module hybkit.util*), 56  
 dynamic\_count\_hyb\_record\_mismatches() (*hybkit.FoldRecord* method), 25

## E

ENERGY\_DEFAULTS (*in module hybkit.plot*), 48  
 ENERGY\_HIST\_RC\_PARAMS (*in module hybkit.plot*), 47

energy\_histogram() (*in module hybkit.plot*), 48  
 ensure\_matches\_hyb\_record() (*hybkit.FoldRecord* method), 26  
 eval\_mirna() (*hybkit.HybRecord* method), 15  
 eval\_types() (*hybkit.HybRecord* method), 14

## F

file\_exists() (*in module hybkit.util*), 56  
 find() (*hybkit.type\_finder.TypeFinder* class method), 35  
 find\_with\_params (*hybkit.type\_finder.TypeFinder* attribute), 34  
 fold\_exists() (*in module hybkit.util*), 57  
 fold\_match\_counts\_histogram() (*in module hybkit.plot*), 50  
 FOLD\_MATCH\_HIST\_RC\_PARAMS (*in module hybkit.plot*), 47  
 fold\_mirna\_nt\_counts\_histogram() (*in module hybkit.plot*), 51  
 FOLD\_NT\_COUNTS\_HIST\_RC\_PARAMS (*in module hybkit.plot*), 47  
 FOLD\_SUFFIXES (*in module hybkit.settings*), 51  
 FoldFile\_settings (*in module hybkit.settings*), 55  
 FoldFile\_settings\_info (*in module hybkit.settings*), 54  
 FoldRecord (*class* in *hybkit*), 23  
 FoldRecord\_settings (*in module hybkit.settings*), 55  
 FoldRecord\_settings\_info (*in module hybkit.settings*), 53  
 from\_ct\_lines() (*hybkit.FoldRecord* class method), 26  
 from\_ct\_string() (*hybkit.FoldRecord* class method), 27  
 from\_fasta\_records() (*hybkit.HybRecord* class method), 18  
 from\_line() (*hybkit.HybRecord* class method), 17  
 from\_vienna\_lines() (*hybkit.FoldRecord* class method), 26  
 from\_vienna\_string() (*hybkit.FoldRecord* class method), 26

## G

GEN\_PROPS (*hybkit.HybRecord* attribute), 11

- get\_all\_results() (*hybkit.analysis.Analysis method*), 43  
 get\_analysis\_delim\_str() (*hybkit.analysis.Analysis method*), 43  
 get\_analysis\_results() (*hybkit.analysis.Analysis method*), 43  
 get\_argparse\_doc() (*in module hybkit.util*), 56  
 get\_mirna\_props() (*hybkit.HybRecord method*), 14  
 get\_read\_count() (*hybkit.HybRecord method*), 14  
 get\_record\_count() (*hybkit.HybRecord method*), 14  
 get\_seg1\_type() (*hybkit.HybRecord method*), 13  
 get\_seg2\_type() (*hybkit.HybRecord method*), 13  
 get\_seg\_types() (*hybkit.HybRecord method*), 13  
 get\_specific\_result() (*hybkit.analysis.Analysis method*), 43  
 get\_target\_props() (*hybkit.HybRecord method*), 14
- ## H
- has\_prop() (*hybkit.HybRecord method*), 16  
 HAS\_PROPS (*hybkit.HybRecord attribute*), 13  
 hyb\_analyze  
   module, 86  
 hyb\_check  
   module, 61  
 hyb\_eval  
   module, 77  
 hyb\_exists() (*in module hybkit.util*), 56  
 hyb\_filter  
   module, 67  
 HYB\_SUFFIXES (*in module hybkit.settings*), 51  
 HybFile (*class in hybkit*), 21  
 HybFile\_settings (*in module hybkit.settings*), 55  
 HybFile\_settings\_info (*in module hybkit.settings*), 52  
 HybFoldIter (*class in hybkit*), 33  
 HybFoldIter\_settings (*in module hybkit.settings*), 56  
 HybFoldIter\_settings\_info (*in module hybkit.settings*), 54  
 hybkit  
   module, 8  
 hybkit.analysis  
   module, 40  
 hybkit.errors  
   module, 60  
 hybkit.plot  
   module, 47  
 hybkit.settings  
   module, 51  
 hybkit.type\_finder  
   module, 34  
 hybkit.util  
   module, 56  
 HybkitArgError, 60  
 HybkitConstructorError, 60
- HybkitError, 60  
 HybkitIterError, 60  
 HybkitMiscError, 60  
 HybRecord (*class in hybkit*), 9  
 HybRecord\_settings (*in module hybkit.settings*), 55  
 HybRecord\_settings\_info (*in module hybkit.settings*), 52  
 HYBRID\_COLUMNS (*hybkit.HybRecord attribute*), 11
- ## I
- is\_set() (*hybkit.HybRecord method*), 16
- ## K
- key (*hybkit.analysis.Analysis attribute*), 44
- ## M
- make\_id\_map\_params() (*hybkit.type\_finder.TypeFinder static method*), 38  
 make\_out\_file\_name() (*in module hybkit.util*), 58  
 make\_string\_match\_params() (*hybkit.type\_finder.TypeFinder static method*), 37  
 matches\_hyb\_record() (*hybkit.FoldRecord method*), 25  
 method\_hybformat() (*hybkit.type\_finder.TypeFinder static method*), 36  
 method\_id\_map() (*hybkit.type\_finder.TypeFinder static method*), 37  
 method\_is\_set() (*hybkit.type\_finder.TypeFinder class method*), 35  
 method\_string\_match() (*hybkit.type\_finder.TypeFinder static method*), 36  
 methods (*hybkit.type\_finder.TypeFinder attribute*), 34  
 mirna\_detail() (*hybkit.HybRecord method*), 16  
 mirna\_details() (*hybkit.HybRecord method*), 15  
 MIRNA\_PROPS (*hybkit.HybRecord attribute*), 12  
 MIRNA\_STR\_PROPS (*hybkit.HybRecord attribute*), 12  
 MIRNA\_TYPES (*in module hybkit.settings*), 52  
 module  
   hyb\_analyze, 86  
   hyb\_check, 61  
   hyb\_eval, 77  
   hyb\_filter, 67  
   hybkit, 8  
   hybkit.analysis, 40  
   hybkit.errors, 60  
   hybkit.plot, 47  
   hybkit.settings, 51  
   hybkit.type\_finder, 34  
   hybkit.util, 56
- ## N
- not\_set() (*hybkit.HybRecord method*), 16



## O

`open()` (*hybkit.CtFile* class method), 32  
`open()` (*hybkit.HybFile* class method), 22  
`open()` (*hybkit.ViennaFile* class method), 29  
`out_path_exists()` (in module *hybkit.util*), 57

## P

`param_methods` (*hybkit.type\_finder.TypeFinder* attribute), 35  
`param_methods_needs_file` (*hybkit.type\_finder.TypeFinder* attribute), 35  
`params` (*hybkit.type\_finder.TypeFinder* attribute), 34  
`PIE_DEFAULTS` (in module *hybkit.plot*), 47  
`plot_analysis_results()` (*hybkit.analysis.Analysis* method), 44  
`print_report()` (*hybkit.HybFoldIter* method), 34  
`prop()` (*hybkit.HybRecord* method), 16

## R

`read_record()` (*hybkit.CtFile* method), 32  
`read_record()` (*hybkit.HybFile* method), 21  
`read_record()` (*hybkit.ViennaFile* method), 29  
`read_records()` (*hybkit.CtFile* method), 32  
`read_records()` (*hybkit.HybFile* method), 21  
`read_records()` (*hybkit.ViennaFile* method), 30  
`report()` (*hybkit.HybFoldIter* method), 34

## S

`SEGMENT_COLUMNS` (*hybkit.HybRecord* attribute), 11  
`set_custom_method()` (*hybkit.type\_finder.TypeFinder* class method), 36  
`set_flag()` (*hybkit.HybRecord* method), 13  
`set_fold_record()` (*hybkit.HybRecord* method), 15  
`set_method()` (*hybkit.type\_finder.TypeFinder* class method), 35  
`SET_PROPS` (*hybkit.HybRecord* attribute), 11  
`set_setting()` (in module *hybkit.util*), 59  
`set_settings_from_namespace()` (in module *hybkit.util*), 59  
`settings` (*hybkit.analysis.Analysis* attribute), 42  
`settings` (*hybkit.CtFile* attribute), 33  
`settings` (*hybkit.FoldRecord* attribute), 25  
`settings` (*hybkit.HybFile* attribute), 21  
`settings` (*hybkit.HybFoldIter* attribute), 34  
`settings` (*hybkit.HybRecord* attribute), 11  
`settings` (*hybkit.ViennaFile* attribute), 30  
`static_count_hyb_record_mismatches()` (*hybkit.FoldRecord* method), 25  
`STR_PROPS` (*hybkit.HybRecord* attribute), 12

## T

`target_count()` (in module *hybkit.plot*), 49  
`TARGET_PIE_RC_PARAMS` (in module *hybkit.plot*), 47

`to_csv()` (*hybkit.HybRecord* method), 16  
`to_csv_header()` (*hybkit.HybRecord* class method), 19  
`to_fasta_record()` (*hybkit.HybRecord* method), 17  
`to_fasta_str()` (*hybkit.HybRecord* method), 17  
`to_fields()` (*hybkit.HybRecord* method), 17  
`to_fields_header()` (*hybkit.HybRecord* class method), 18  
`to_line()` (*hybkit.HybRecord* method), 16  
`to_vienna_lines()` (*hybkit.FoldRecord* method), 25  
`to_vienna_string()` (*hybkit.FoldRecord* method), 25  
`type_count()` (in module *hybkit.plot*), 48  
`type_count_dual()` (in module *hybkit.plot*), 49  
`TYPE_PIE_DUAL_RC_PARAMS` (in module *hybkit.plot*), 47  
`TYPE_PIE_SINGLE_RC_PARAMS` (in module *hybkit.plot*), 47  
`TypeFinder` (class in *hybkit.type\_finder*), 34  
`TypeFinder` (*hybkit.HybRecord* attribute), 11

## V

`validate_args()` (in module *hybkit.util*), 58  
`validate_args_exit()` (in module *hybkit.util*), 58  
`vienna_exists()` (in module *hybkit.util*), 57  
`VIENNA_SUFFIXES` (in module *hybkit.settings*), 51  
`ViennaFile` (class in *hybkit*), 29

## W

`write()` (*hybkit.HybFile* method), 22  
`write_analysis_delim_str()` (*hybkit.analysis.Analysis* method), 44  
`write_analysis_results_special()` (*hybkit.analysis.Analysis* method), 44  
`write_fh()` (*hybkit.CtFile* method), 33  
`write_fh()` (*hybkit.HybFile* method), 22  
`write_fh()` (*hybkit.ViennaFile* method), 30  
`write_record` (*hybkit.CtFile* attribute), 32  
`write_record()` (*hybkit.HybFile* method), 21  
`write_record()` (*hybkit.ViennaFile* method), 30  
`write_records` (*hybkit.CtFile* attribute), 32  
`write_records()` (*hybkit.HybFile* method), 21  
`write_records()` (*hybkit.ViennaFile* method), 30