worms Documentation

Release 0.1.26

Will Sheffler

Jul 10, 2018

Contents

1 worms **3**
- 1.1 Features . . . 4
- 1.2 Credits . . . 4

2 Installation **5**
- 2.1 Stable release . . . 5
- 2.2 From sources . . . 5

3 Usage **7**

4 worms **9**
- 4.1 worms package . . . 9

5 Contributing **35**
- 5.1 Types of Contributions . . . 35
- 5.2 Get Started! . . . 36
- 5.3 Pull Request Guidelines . . . 36
- 5.4 Tips . . . 37

6 History **39**
- 6.1 0.1.24 (2018-04-25) . . . 39
- 6.2 0.1.23 (2018-04-24) . . . 39
- 6.3 0.1.22 (2018-04-20) . . . 39
- 6.4 0.1.21 (2018-04-19) . . . 39
- 6.5 0.1.20 (2018-04-16) . . . 39
- 6.6 0.1.19 (2018-04-6) . . . 40
- 6.7 0.1.17 (2018-04-4) . . . 40
- 6.8 0.1.16 (2018-03-15) . . . 40
- 6.9 0.1.15 (2018-03.05) . . . 40
- 6.10 0.1.14 (2018-02-28) . . . 40
- 6.11 0.1.13 (2018-02-16) . . . 40
- 6.12 0.1.12 (2018-02-16) . . . 40
- 6.13 0.1.11 (2018-02-15) . . . 40
- 6.14 0.1.10 (2018-02-15) . . . 41
- 6.15 0.1.9 (2018-02-08) . . . 41
- 6.16 0.1.8 (2018-02-07) . . . 41
- 6.17 0.1.6 (2018-02-01) . . . 41

6.18 0.1.6 (2018-02-01) . . . 41
6.19 0.1.5 (2018-02-01) . . . 41
6.20 0.1.4 (2018-02-01) . . . 41
6.21 0.1.3 (2018-02-01) . . . 41
6.22 0.1.2 (2018-01-23) . . . 42
6.23 0.1.1 (2018-01-23) . . . 42

7 Indices and tables **43**

Python Module Index **45**

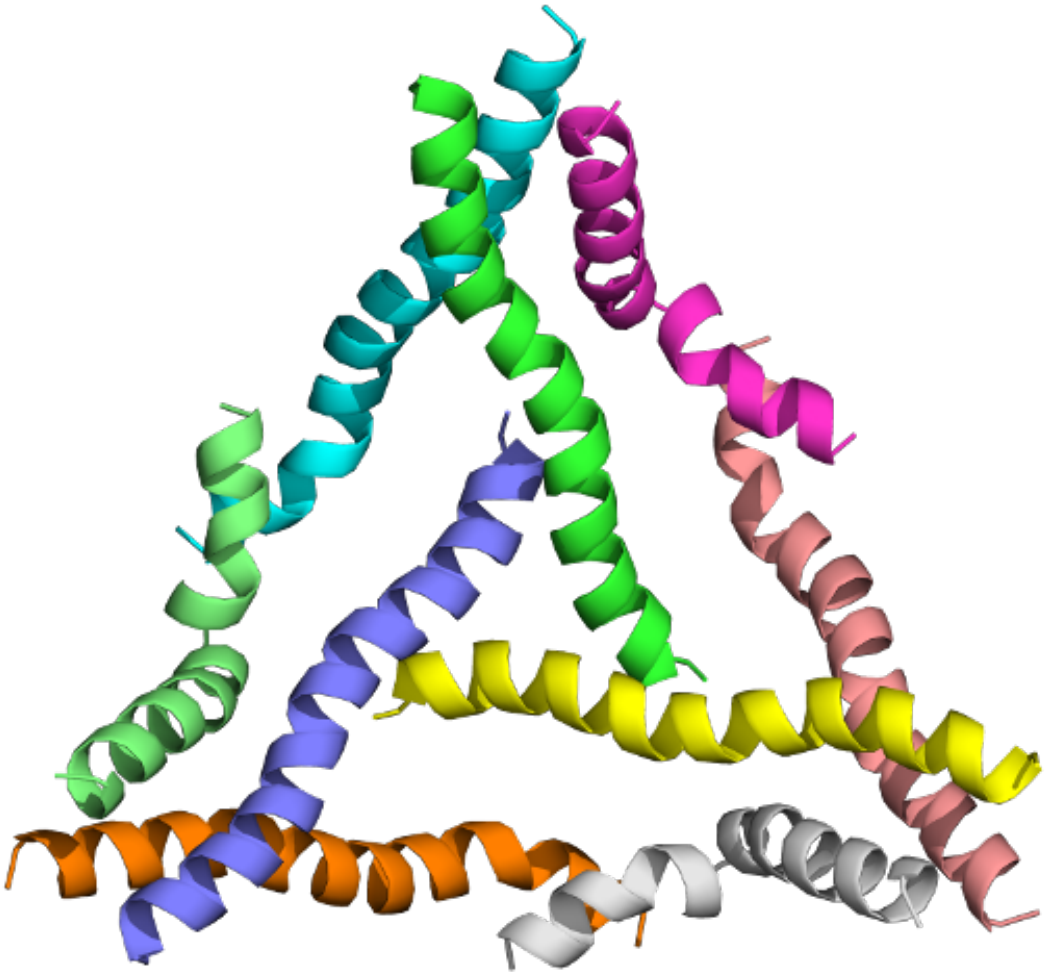
Contents:

CHAPTER 1

worms

Protein assembly design without protein interface design.

- Free software: Apache Software License 2.0
- Documentation: https://worms.readthedocs.io.



docs/img/worms_triangle_600.png

1.1 Features

please send angry emails to willsheffler@gmail.com until there are docs here.

1.2 Credits

This package was created with Cookiecutter and the audreyr/cookiecutter-pypackage project template.

CHAPTER 2

Installation

2.1 Stable release

To install worms, run this command in your terminal:

```
$ pip install worms
```

This is the preferred method to install worms, as it will always install the most recent stable release.

If you don't have pip installed, this Python installation guide can guide you through the process.

2.2 From sources

The sources for worms can be downloaded from the Github repo.

You can either clone the public repository:

```
$ git clone git://github.com/willsheffler/worms
```

Or download the tarball:

```
$ curl  -OL https://github.com/willsheffler/worms/tarball/master
```

Once you have a copy of the source, you can install it with:

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

CHAPTER 3

Usage

To use worms in a project:

```
import worms
```

CHAPTER 4

worms

4.1 worms package

4.1.1 Subpackages

worms.criteria package

Submodules

worms.criteria.base module

TODO: Summary

worms.criteria.base.**Ux**
TYPE – Description

worms.criteria.base.**Uy**
TYPE – Description

worms.criteria.base.**Uz**
TYPE – Description

class worms.criteria.base.**CriteriaList**(*children*)
Bases: *worms.criteria.base.WormCriteria*

TODO: Summary

children
TYPE – Description

score(***kw*)
TODO: Summary

Parameters **kw** – passthru args

Returns Description

Return type TYPE

class worms.criteria.base.**NullCriteria**(*from_seg=0*, *to_seg=-1*, *origin_seg=None*)

Bases: *worms.criteria.base.WormCriteria*

TODO: Summary

from_seg
TYPE – Description

to_seg
TYPE – Description

alignment(*segpos*, ***kw*)
TODO: Summary

Parameters

- **segpos** (*TYPE*) – Description
- **kw** – passthru args

Returns Description

Return type TYPE

jit_lossfunc()

score(*segpos*, ***kw*)
TODO: Summary

Parameters

- **segpos** (*TYPE*) – Description
- **kw** – passthru args

Returns Description

Return type TYPE

class worms.criteria.base.**WormCriteria**

Bases: abc.ABC

TODO: Summary

allowed_attributes
TYPE – Description

allowed_attributes = ('last_body_same_as', 'symname', 'is_cyclic', 'alignment', 'from_

score(***kw*)
TODO: Summary

Parameters **kw** – passthru args

worms.criteria.bounded module

class worms.criteria.bounded.**AxesIntersect**(*symname*, *tgtaxis1*, *tgtaxis2*, *from_seg*, ***, *tol=1.0*, *lever=50*, *to_seg=-1*, *distinct_axes=False*)

Bases: *worms.criteria.base.WormCriteria*

TODO: Summary

angle
TYPE – Description

distinct_axes
TYPE – Description

from_seg
TYPE – Description

lever
TYPE – Description

rot_tol
TYPE – Description

sym_axes
TYPE – Description

symname
TYPE – Description

tgtaxis1
TYPE – Description

tgtaxis2
TYPE – Description

to_seg
TYPE – Description

tol
TYPE – Description

alignment(*segpos*, *debug=0*, ***kw*)
TODO: Summary

Parameters

- **segpos** (*TYPE*) – Description
- **debug** (*int, optional*) – Description
- **kw** – passthru args

Returns Description

Return type TYPE

Raises `AssertionError` – Description

score(*segpos*, *verbosity=False*, ***kw*)
TODO: Summary

Parameters

- **segpos** (*TYPE*) – Description
- **verbosity** (*bool, optional*) – Description
- **kw** – passthru args

Returns Description

Return type TYPE

worms.criteria.bounded.**D2**(*c2=0*, *c2b=-1*, ***kw*)
TODO: Summary

Parameters

- **c2** (*int, optional*) – Description
- **c2b** (*TYPE, optional*) – Description
- **kw** – passthru args

Returns Description

Return type TYPE

worms.criteria.bounded.**D3**(*c3=0*, *c2=-1*, ***kw*)

TODO: Summary

Parameters

- **c3** (*int, optional*) – Description
- **c2** (*TYPE, optional*) – Description
- **kw** – passthru args

Returns Description

Return type TYPE

worms.criteria.bounded.**D4**(*c4=0*, *c2=-1*, ***kw*)

TODO: Summary

Parameters

- **c4** (*int, optional*) – Description
- **c2** (*TYPE, optional*) – Description
- **kw** – passthru args

Returns Description

Return type TYPE

worms.criteria.bounded.**D5**(*c5=0*, *c2=-1*, ***kw*)

TODO: Summary

Parameters

- **c5** (*int, optional*) – Description
- **c2** (*TYPE, optional*) – Description
- **kw** – passthru args

Returns Description

Return type TYPE

worms.criteria.bounded.**D6**(*c6=0*, *c2=-1*, ***kw*)

TODO: Summary

Parameters

- **c6** (*int, optional*) – Description
- **c2** (*TYPE, optional*) – Description
- **kw** – passthru args

Returns Description

Return type TYPE

worms.criteria.bounded.**Icosahedral**(*c5=None*, *c3=None*, *c2=None*, ***kw*)
TODO: Summary

Parameters

- **c5** (*None, optional*) – Description
- **c3** (*None, optional*) – Description
- **c2** (*None, optional*) – Description
- **kw** – passthru args

Returns Description

Return type TYPE

Raises ValueError – Description

worms.criteria.bounded.**Octahedral**(*c4=None*, *c3=None*, *c2=None*, ***kw*)
TODO: Summary

Parameters

- **c4** (*None, optional*) – Description
- **c3** (*None, optional*) – Description
- **c2** (*None, optional*) – Description
- **kw** – passthru args

Returns Description

Return type TYPE

Raises ValueError – Description

worms.criteria.bounded.**Tetrahedral**(*c3=None*, *c2=None*, *c3b=None*, ***kw*)
TODO: Summary

Parameters

- **c3** (*None, optional*) – Description
- **c2** (*None, optional*) – Description
- **c3b** (*None, optional*) – Description
- **kw** – passthru args

Returns Description

Return type TYPE

Raises ValueError – Description

worms.criteria.cyclic module

class worms.criteria.cyclic.**Cyclic**(*symmetry=1*, *from_seg=0*, ***, *tol=1.0*, *origin_seg=None*, *lever=50.0*, *to_seg=-1*, *min_radius=0*)
Bases: *worms.criteria.base.WormCriteria*

alignment(*segpos*, ***kw*)

jit_lossfunc()

score (*segpos*, *, *verbosity=False*, ***kw*)
TODO: Summary

Parameters **kw** – passthru args

worms.criteria.unbounded module

class worms.criteria.unbounded.**AxesAngle** (*symname*, *tgtaxis1*, *tgtaxis2*, *from_seg*, *, *tol=1.0*, *lever=50*, *to_seg=-1*, *space_group_str=None*)
Bases: *worms.criteria.base.WormCriteria*

alignment (*segpos*, *out_cell_spacing=False*, ***kw*)
Alignment to move stuff to be in line with symdef file

Parameters

- **segpos** (*lst*) – List of segment positions / coordinates.
- **I'll accept any "non-positional" argument as name = value, and store in a dictionary** (***kw*) –

crystinfo (*segpos*)

score (*segpos*, ***kw*)
Score

Parameters

- **segpos** (*lst*) – List of segment positions / coordinates.
- **I'll accept any "non-positional" argument as name = value, and store in a dictionary** (***kw*) –

symfile_modifiers (*segpos*)

worms.criteria.unbounded.**Crystal_F432_C3_C4** (*c3a=None*, *c4b=None*, ***kw*)

worms.criteria.unbounded.**Crystal_I213_C2_C3** (*c2a=None*, *c3b=None*, ***kw*)

worms.criteria.unbounded.**Crystal_I432_C2_C4** (*c2a=None*, *c4b=None*, ***kw*)

worms.criteria.unbounded.**Crystal_P213_C3_C3** (*c3a=None*, *c3b=None*, ***kw*)

worms.criteria.unbounded.**Crystal_P4132_C2_C3** (*c2a=None*, *c3b=None*, ***kw*)

worms.criteria.unbounded.**Crystal_P432_C4_C4** (*c4a=None*, *c4b=None*, ***kw*)

worms.criteria.unbounded.**Sheet_P321** (*c3=None*, *c2=None*, ***kw*)

worms.criteria.unbounded.**Sheet_P4212** (*c4=None*, *c2=None*, ***kw*)

worms.criteria.unbounded.**Sheet_P6** (*c6=None*, *c2=None*, ***kw*)

Module contents

worms.data package

Module contents

class worms.data.**PoseLib**
Bases: object

get

worms.khash package

Submodules

worms.khash.khash_cffi module

Module contents

4.1.2 Submodules

4.1.3 worms.bblock module

worms.bblock.**BBlock**(*entry*, *pdbfile*, *filehash*, *pose*, *ss*)

worms.bblock.**bblock_components**(*bblock*)
TODO: Summary

Parameters **bblock** (*TYPE*) – Description

Returns Description

Return type TYPE

worms.bblock.**bblock_dump_pdb**(*out, bblock, dirn, splice, join=True, pos=array([[1., 0., 0., 0.], [0., 1., 0., 0.], [0., 0., 1., 0.], [0., 0., 0., 1.]]), chain=0, anum=1, rnum=1*)

worms.bblock.**bblock_str**(*bblock*)
TODO: Summary

Parameters **bblock** (*TYPE*) – Description

Returns Description

Return type TYPE

worms.bblock.**chain_of_ires**
Summary

Parameters

- **bb** (*TYPE*) – Description
- **ires** (*TYPE*) – Description

Returns Description

Return type TYPE

4.1.4 worms.database module

TODO: Summary

class worms.database.**BBlockDB**(*cachedir=None*, *bakerdb_files=[]*, *load_poses=False*, *nprocs=1*, *lazy=True*, *read_new_pdbs=False*, *verbosity=0*)
Bases: object

stores Poses and BBlocks in a disk cache

bblock(*pdbkey*)

bblockfile(*pdbkey*)

build_pdb_data(*entry*)
return Nnew, Nmissing

check_lock_cachedir()

islocked_cachedir()

load_cached_bblock_into_memory(*pdbkey*)

load_cached_pose_into_memory(*pdbfile*)

load_from_pdbs()

load_from_pdbs_inner(*exe*)

lock_cachedir()

pose(*pdbfile*)
load pose from _bblock_cache, read from file if not in memory

posefile(*pdbfile*)

query(*query*, *, *useclass=True*, *max_bblocks=150*, *shuffle=True*)
match name, _type, _class if one match, use it if _type and _class match, check useclass option Het:NNCx/y require exact number or require extra

Parameters

- **query** (*TYPE*) – Description
- **useclass** (*bool, optional*) – Description

Returns Description

Return type TYPE

query_names(*query*, *, *useclass=True*)
query for names only

unlock_cachedir()

class worms.database.**SpliceDB**(*cachedir=None*)
Bases: object

Stores valid NC splices for bblock pairs

add(*params*, *pdbkey0*, *pdbkey1*, *val*)

cachepath(*params*, *pdbkey*)

partial(*params*, *pdbkey*)

sync_to_disk()

worms.database.**flatten_path**(*pdbfile*)

4.1.5 worms.edge module

worms.edge.**Edge**(*u*, *ublks*, *v*, *vblks*, *verbosity=0*, ***kw*)

worms.edge.**get_allowed_splices**(*u*, *ublks*, *v*, *vblks*, *splicedb=None*, *max_splice_rms=0.7*, *ncontact_cut=10*, *clashd2=9.0*, *contactd2=100.0*, *rms_range=5*, *clash_contact_range=9*, *skip_on_fail=True*, *parallel=False*, *verbosity=1*, *sync_to_disk_every=0.001*)

worms.edge.**splice_metrics_pair**(*blk0*, *blk1*, *max_splice_rms=0.7*, *clashd2=9.0*, *contactd2=100.0*, *rms_range=9*, *clash_contact_range=9*, *skip_on_fail=True*)

4.1.6 worms.jitsearch module

4.1.7 worms.pose_contortions module

TODO: Summary

class worms.pose_contortions.**AnnoPose**(*pose*, *iseg*, *srcpose*, *src_lb*, *src_ub*, *cyclic_entry*)

Bases: object

TODO: Summary

cyclic_entry
TYPE – Description

iseg
TYPE – Description

pose
TYPE – Description

src_lb
TYPE – Description

src_ub
TYPE – Description

srcpose
TYPE – Description

seq()
TODO: Summary

Returns Description

Return type TYPE

srcseq()
TODO: Summary

Returns Description

Return type TYPE

class worms.pose_contortions.**CyclicTrim**(*sym_seg_from*, *sym_seg_to*)

Bases: tuple

sym_seg_from
Alias for field number 0

sym_seg_to
Alias for field number 1

worms.pose_contortions.**contort_pose_chains**(*pose*, *chains*, *nseg*, *ir_en*, *ir_ex*, *pl_en*, *pl_ex*, *chain_start*, *chain_end*, *position=None*, *pad=(0, 0)*, *iseg=None*, *cyclictrim=None*, *last_seg_entrypol=None*, *first_seg_exitpol=None*, *sym_ir=None*, *sym_pol=None*)

make pose chains from 'segment' info

what a monster this has become. returns (segchains, rest) segchains elems are [enterexitchain] or, [enterchain, ..., exitchain] rest holds other chains IFF enter and exit in same chain each element is a pair [pose, source] where source is (origin_pose, start_res, stop_res) cyclictrim specifies segments which are spliced across the symmetric interface. segments only needed if cyclictrim==True if cyclictrim, last segment will only be a single entry residue

Args: No

worms.pose_contortions.**make_contorted_pose**(*, *entryexits*, *entrypol*, *exitpol*, *indices*, *from_seg*, *to_seg*, *origin_seg*, *seg_pos*, *position*, *is_cyclic*, *align*, *cryst_info*, *end*, *iend*, *only_connected*, *join*, *cyclic_permute*, *cyclictrim*, *provenance*, *make_chain_list*)

there be dragons here

worms.pose_contortions.**reorder_spliced_as_N_to_C**(*body_chains*, *polarities*)

remap chains of each body such that concatenated chains are N->C

Parameters

- **body_chains** (*TYPE*) – Description
- **polarities** (*TYPE*) – Description

Returns Description

Return type TYPE

Raises `ValueError` – Description

4.1.8 worms.search module

worms.search.**random**() → x in the interval [0, 1).

4.1.9 worms.segments module

TODO: Summary

class worms.segments.**Segment**(*spliceables*, *entry=None*, *exit=None*, *expert=False*)

Bases: `object`

TODO: Summary

bodyid
TYPE – Description

entrypol
TYPE – Description

entryresid
TYPE – Description

entrysiteid
TYPE – Description

exitpol
TYPE – Description

exitresid
TYPE – Description

exitsiteid
TYPE – Description

expert
TYPE – Description

max_sites
TYPE – Description

min_sites
TYPE – Description

nchains
TYPE – Description

spliceables
TYPE – Description

x2exit
TYPE – Description

x2orgn
TYPE – Description

init_segment_data()
TODO: Summary

Raises `ValueError` – Description

make_head()
TODO: Summary

Returns Description

Return type TYPE

make_pose_chains(*indices*, *position=None*, *pad=(0, 0)*, *iseg=None*, *segments=None*, *cyclictrim=None*)
what a monster this has become. returns (segchains, rest) segchains elems are [enterexitchain] or, [enterchain, . . . , exitchain] rest holds other chains IFF enter and exit in same chain each element is a pair [pose, source] where source is (origin_pose, start_res, stop_res) cyclictrim specifies segments which are spliced across the symmetric interface. segments only needed if cyclictrim==True if cyclictrim, last segment will only be a single entry residue

Parameters

- **indices** (*TYPE*) – Description
- **position** (*None, optional*) – Description
- **pad** (*tuple, optional*) – Description
- **iseg** (*None, optional*) – Description
- **segments** (*None, optional*) – Description
- **cyclictrim** (*None, optional*) – Description

Returns Description

Return type TYPE

make_tail()
TODO: Summary

Returns Description

Return type TYPE

merge_idx(*head*, *head_idx*, *tail*, *tail_idx*)
TODO: Summary

Parameters

- **head** (*TYPE*) – Description
- **head_idx** (*TYPE*) – Description
- **tail** (*TYPE*) – Description
- **tail_idx** (*TYPE*) – Description

Returns Description

Return type TYPE

merge_idx_slow(*head*, *head_idx*, *tail*, *tail_idx*)
TODO: Summary

Parameters

- **head** (*TYPE*) – Description
- **head_idx** (*TYPE*) – Description
- **tail** (*TYPE*) – Description
- **tail_idx** (*TYPE*) – Description

Returns return joint index, -1 if head/tail pairing is invalid

Return type TYPE

same_bodies_as(*other*)
TODO: Summary

Parameters **other** (*TYPE*) – Description

Returns Description

Return type TYPE

split_idx(*idx*, *head*, *tail*)
return indices for separate head and tail segments

Parameters

- **idx** (*TYPE*) – Description
- **head** (*TYPE*) – Description
- **tail** (*TYPE*) – Description

Returns Description

Return type TYPE

class worms.segments.**Segments**(*segments*)

Bases: object

light wrapper around list of Segments

segments
TYPE – Description

index(*val*)
TODO: Summary

Parameters **val** (*TYPE*) – Description

Returns Description

Return type TYPE

split_at(*idx*)
TODO: Summary

Parameters **idx** (*TYPE*) – Description

Returns Description

Return type TYPE

class worms.segments.**SpliceSite**(*sele*, *polarity*, *chain=None*)

Bases: object

TODO: Summary

chain
TYPE – Description

polarity
TYPE – Description

selections
TYPE – Description

resid(*id*, *pose*)
TODO: Summary

Parameters

- **id** (*TYPE*) – Description
- **pose** (*TYPE*) – Description

Returns Description

Return type TYPE

Raises ValueError – Description

class worms.segments.**Spliceable**(*body*, *sites*, ***, *bodyid=None*, *min_seg_len=1*, *allowed_pairs=None*)

Bases: object

TODO: Summary

allowed_pairs
TYPE – Description

body
TYPE – Description

bodyid
TYPE – Description

chains
TYPE – Description

end_of_chain
TYPE – Description

min_seg_len
TYPE – Description

nsite
TYPE – Description

sites
TYPE – Description

start_of_chain
TYPE – Description

is_compatible(*isite*, *ires*, *jsite*, *jres*)
TODO: Summary

Parameters

- **isite** (*TYPE*) – Description
- **ires** (*TYPE*) – Description
- **jsite** (*TYPE*) – Description
- **jres** (*TYPE*) – Description

Returns Description

Return type TYPE

resids(*isite*)
TODO: Summary

Parameters **isite** (*TYPE*) – Description

Returns Description

Return type TYPE

sitepair_allowed(*isite*, *jsite*)
TODO: Summary

Parameters

- **isite** (*TYPE*) – Description
- **jsite** (*TYPE*) – Description

Returns Description

Return type TYPE

spliceable_positions()
selection of resids, and map 'global' index to selected index

Returns Description

Return type TYPE

class worms.segments.**Worms**(*segments*, *scores*, *indices*, *positions*, *criteria*, *detail*)

Bases: object

TODO: Summary

criteria
TYPE – Description

detail
TYPE – Description

indices
TYPE – Description

positions
TYPE – Description

score0
TYPE – Description

score0sym
TYPE – Description

scores
TYPE – Description

segments
TYPE – Description

splicepoint_cache
dict – Description

clear_caches()
TODO: Summary

pose(*which*, ***, *align=True*, *end=None*, *only_connected='auto'*, *join=True*, *cyclic_permute=None*, *cyclictrim=None*, *provenance=False*, *make_chain_list=False*, ***kw*)
makes a pose for the ith worm

splicepoints(*which*)
TODO: Summary

Parameters **which** (*TYPE*) – Description

Returns Description

Return type TYPE

splices(*which*)
TODO: Summary

Parameters **which** (*TYPE*) – Description

Returns Description

Return type TYPE

sympose(*which*, *score=False*, *provenance=False*, *fullatom=False*, *asym_score_thresh=50*, *min_cell_spacing=130*, ***, *parallel=False*)
TODO: Summary

Parameters

- **which** (*TYPE*) – Description
- **score** (*bool, optional*) – Description

- **provenance** (*bool, optional*) – Description
- **fullatom** (*bool, optional*) – Description
- **parallel** (*bool, optional*) – Description
- **asym_score_thresh** (*int, optional*) – Description

Returns Description

Return type TYPE

Raises IndexError – Description

worms.segments.**lineno**()
Returns the current line number in our program.

Returns Description

Return type TYPE

4.1.10 worms.util module

TODO: Summary

class worms.util.**InProcessExecutor**(**args*, ***kw*)
Bases: object

TODO: Summary

map(*func*, **iterables*)
TODO: Summary

Parameters

- **func** (*TYPE*) – Description
- **iterables** – Description

Returns Description

Return type TYPE

submit(*fn*, **args*, ***kw*)
TODO: Summary

Parameters

- **fn** (*TYPE*) – Description
- **args** – Description
- **kw** – passthru args

Returns Description

Return type TYPE

class worms.util.**MultiRange**(*nside*)
Bases: object

TODO: Summary

len
TYPE – Description

nside
TYPE – Description

psum
TYPE – Description

class worms.util.**NonFuture**(*result*)
Bases: object

TODO: Summary

result()
TODO: Summary

Returns Description

Return type TYPE

worms.util.**bigprod**(*iterable*)
TODO: Summary

Parameters **iterable** (*TYPE*) – Description

Returns Description

Return type TYPE

worms.util.**contig_idx_breaks**

worms.util.**cpu_count**()
TODO: Summary

Returns Description

Return type TYPE

worms.util.**dicts_to_items**(*inp*)
TODO: Summary

Parameters **inp** (*TYPE*) – Description

Returns Description

Return type TYPE

worms.util.**expand_array_if_needed**

worms.util.**first_duplicate**(*segs*)
TODO: Summary

Parameters **segs** (*TYPE*) – Description

Returns Description

Return type TYPE

worms.util.**get_bb_coords**(*pose*, *which_resi=None*)
extract rif style stubs from rosetta pose

Parameters

- **pose** (*TYPE*) – Description
- **which_resi** (*None, optional*) – Description

Returns Description

Return type TYPE

Raises `ValueError` – Description

worms.util.**get_bb_stubs**(*pose*, *which_resi=None*)
extract rif style stubs from rosetta pose

Parameters

- **pose** (*TYPE*) – Description
- **which_resi** (*None, optional*) – Description

Returns Description

Return type TYPE

Raises `ValueError` – Description

worms.util.**get_chain_bounds**(*pose*)
TODO: Summary

Parameters **pose** (*TYPE*) – Description

Returns Description

Return type TYPE

worms.util.**get_symdata**
TODO – Summary

Parameters **name** (*TYPE*) – Description

Returns Description

Return type TYPE

worms.util.**get_symdata_modified**(*name*, *string_substitutions=None*, *scale_positions=None*)
TODO: Summary

Parameters

- **name** (*TYPE*) – Description
- **string_substitutions** (*None, optional*) – Description
- **scale_positions** (*None, optional*) – Description

Returns Description

Return type TYPE

worms.util.**get_symfile_contents**
TODO – Summary

Parameters **name** (*TYPE*) – Description

Returns Description

Return type TYPE

worms.util.**hash_str_to_int**(*s*)

worms.util.**infer_cyclic_symmetry**(*pose*)
TODO: Summary

Parameters **pose** (*TYPE*) – Description

Raises `NotImplementedError` – Description

worms.util.**items_to_dicts**(*inp*)
TODO: Summary

Parameters **inp** (*TYPE*) – Description

Returns Description

Return type TYPE

worms.util.**no_overlapping_adjacent_residues**(*p*)
TODO: Summary

Parameters **p** (*TYPE*) – Description

Returns Description

Return type TYPE

worms.util.**no_overlapping_residues**(*p*)
TODO: Summary

Parameters **p** (*TYPE*) – Description

Returns Description

Return type TYPE

worms.util.**numpy_stub_from_rosetta_stub**(*rosstub*)
TODO: Summary

Parameters **rosstub** (*TYPE*) – Description

Returns Description

Return type TYPE

worms.util.**parallel_batch_map**(*pool*, *function*, *accumulator*, *batch_size*, *map_func_args*, ***kw*)
TODO: Summary

Parameters

- **pool** (*TYPE*) – Description
- **function** (*TYPE*) – Description
- **accumulator** (*TYPE*) – Description
- **batch_size** (*TYPE*) – Description
- **map_func_args** (*TYPE*) – Description
- **kw** – passthru args

Yields *TYPE* – Description

worms.util.**parallel_nobatch_map**(*pool*, *function*, *accumulator*, *batch_size*, *map_func_args*, ***kw*)
TODO: Summary

Parameters

- **pool** (*TYPE*) – Description
- **function** (*TYPE*) – Description
- **accumulator** (*TYPE*) – Description
- **batch_size** (*TYPE*) – Description
- **map_func_args** (*TYPE*) – Description

- **kw** – passthru args

Yields *TYPE* – Description

worms.util.**pose_bounds**(*pose*, *lb*, *ub*)
TODO: Summary

Parameters

- **pose** (*TYPE*) – Description
- **lb** (*TYPE*) – Description
- **ub** (*TYPE*) – Description

Returns Description

Return type TYPE

Raises ValueError – Description

worms.util.**residue_coords**(*p*, *ir*, *n=3*)

worms.util.**residue_sym_err**(*p, ang, ir, jr, n=1, axis=[0, 0, 1], verbose=0*)

worms.util.**rosetta_stub_from_numpy_stub**(*npstub*)
TODO: Summary

Parameters **npstub** (*TYPE*) – Description

Returns Description

Return type TYPE

worms.util.**subpose**(*pose*, *lb*, *ub=-1*)
TODO: Summary

Parameters

- **pose** (*TYPE*) – Description
- **lb** (*TYPE*) – Description
- **ub** (*TYPE, optional*) – Description

Returns Description

Return type TYPE

worms.util.**symfile_path**(*name*)
TODO: Summary

Parameters **name** (*TYPE*) – Description

Returns Description

Return type TYPE

worms.util.**tqdm_parallel_map**(*pool*, *function*, *accumulator*, *map_func_args*, *batch_size*, ***kw*)
TODO: Summary

Parameters

- **pool** (*TYPE*) – Description
- **function** (*TYPE*) – Description
- **accumulator** (*TYPE*) – Description
- **map_func_args** (*TYPE*) – Description

- **batch_size** (*TYPE*) – Description
- **kw** – passthru args

worms.util.**trim_pose**(*pose*, *resid*, *direction*, *pad=0*)
trim end of pose from direction, leaving <=pad residues beyond resid

Parameters

- **pose** (*TYPE*) – Description
- **resid** (*TYPE*) – Description
- **direction** (*TYPE*) – Description
- **pad** (*int, optional*) – Description

Returns Description

Return type TYPE

Raises ValueError – Description

worms.util.**unique_key**(*a*, *b=None*)

worms.util.**unique_key_int32s**(*a*, *b*)

worms.util.**worst_CN_connect**(*p*)
TODO: Summary

Parameters **p** (*TYPE*) – Description

Returns Description

Return type TYPE

worms.util.**xform_pose**(*xform*, *pose*, *lb=1*, *ub=-1*)
TODO: Summary

Parameters

- **xform** (*TYPE*) – Description
- **pose** (*TYPE*) – Description
- **lb** (*int, optional*) – Description
- **ub** (*TYPE, optional*) – Description

4.1.11 worms.vertex module

TODO: Summary

worms.vertex.**Vertex**(*bbs*, *dirn*, *bbids=None*, *min_seg_len=1*, *verbosity=0*)
Summary

Parameters

- **bbs** (*TYPE*) – Description
- **bbids** (*TYPE*) – Description
- **dirn** (*TYPE*) – Description
- **min_seg_len** (*TYPE*) – Description

Returns Description

Return type TYPE

worms.vertex.**vertex_single**(*bbstate*, *bbid*, *din*, *dout*, *min_seg_len*, *verbosity=0*)
build on bblock's worth of vertex

4.1.12 worms.vis module

TODO: Summary

worms.vis.**numcom**
int – Description

worms.vis.**numline**
int – Description

worms.vis.**numray**
int – Description

worms.vis.**numseg**
int – Description

worms.vis.**numvec**
int – Description

worms.vis.**showme_state**
TYPE – Description

worms.vis.**cgo_cyl**(*c1*, *c2*, *r*, *col=(1, 1, 1)*, *col2=None*)
TODO: Summary

Parameters

- **c1** (*TYPE*) – Description
- **c2** (*TYPE*) – Description
- **r** (*TYPE*) – Description
- **col** (*tuple, optional*) – Description
- **col2** (*None, optional*) – Description

Returns Description

Return type TYPE

worms.vis.**cgo_lineabs**(*a*, *c*, *col=(1, 1, 1)*)
TODO: Summary

Parameters

- **a** (*TYPE*) – Description
- **c** (*TYPE*) – Description
- **col** (*tuple, optional*) – Description

Returns Description

Return type TYPE

worms.vis.**cgo_segment**(*c1*, *c2*, *col=(1, 1, 1)*)
TODO: Summary

Parameters

- **c1** (*TYPE*) – Description
- **c2** (*TYPE*) – Description
- **col** (*tuple, optional*) – Description

Returns Description

Return type TYPE

worms.vis.**cgo_sphere**(*c*, *r=1*, *col=(1*, *1*, *1)*)
TODO: Summary

Parameters

- **c** (*TYPE*) – Description
- **r** (*int, optional*) – Description
- **col** (*tuple, optional*) – Description

Returns Description

Return type TYPE

worms.vis.**format_atom**(*atomi=0*, *atomn='ATOM'*, *idx=' '*, *resn='RES'*, *chain='A'*, *resi=0*, *insert=' '*, *x=0*, *y=0*, *z=0*, *occ=0*, *b=0*)

worms.vis.**is_rosetta_pose**(*to_show*)

worms.vis.**pymol_load**(*to_show*, *state=None*, *name=None*, ***kw*)

worms.vis.**pymol_load_pose**(*pose*, *name*)

worms.vis.**pymol_xform**(*name*, *xform*)

worms.vis.**show_with_axis**(*worms*, *idx=0*)
TODO: Summary

Parameters

- **worms** (*TYPE*) – Description
- **idx** (*int, optional*) – Description

worms.vis.**show_with_z_axes**(*worms*, *idx=0*, *only_connected=0*, ***kw*)
TODO: Summary

Parameters

- **worms** (*TYPE*) – Description
- **idx** (*int, optional*) – Description
- **only_connected** (*int, optional*) – Description
- **kw** – passthru args

worms.vis.**showcom**(*sel='all'*)
TODO: Summary

Parameters **sel** (*str, optional*) – Description

worms.vis.**showcyl**(*c1*, *c2*, *r*, *col=(1*, *1*, *1)*, *col2=None*, *lbl=''*)
TODO: Summary

Parameters

- **c1** (*TYPE*) – Description

- **c2** (*TYPE*) – Description
- **r** (*TYPE*) – Description
- **col** (*tuple, optional*) – Description
- **col2** (*None, optional*) – Description
- **lbl** (*str, optional*) – Description

worms.vis.**showline**(*a*, *c*, *col=(1*, *1*, *1)*, *lbl=''*)
TODO: Summary

Parameters

- **a** (*TYPE*) – Description
- **c** (*TYPE*) – Description
- **col** (*tuple, optional*) – Description
- **lbl** (*str, optional*) – Description

worms.vis.**showlineabs**(*a*, *c*, *col=(1*, *1*, *1)*, *lbl=''*)
TODO: Summary

Parameters

- **a** (*TYPE*) – Description
- **c** (*TYPE*) – Description
- **col** (*tuple, optional*) – Description
- **lbl** (*str, optional*) – Description

worms.vis.**showme**(**args*, *how='pymol'*, ***kw*)
TODO: Summary

Parameters

- **args** – passthru args
- **how** (*str, optional*) – Description
- **kw** – passthru args

Returns Description

Return type TYPE

Raises NotImplementedError – Description

worms.vis.**showme_pymol**(*what*, *headless=False*, *block=False*, ***kw*)
TODO: Summary

Parameters

- **what** (*TYPE*) – Description
- **headless** (*bool, optional*) – Description
- **block** (*bool, optional*) – Description
- **kw** – passthru args

Returns Description

Return type TYPE

worms.vis.**showsegment**(*c1*, *c2*, *col=(1, 1, 1)*, *lbl=''*)
TODO: Summary

Parameters

- **c1** (*TYPE*) – Description
- **c2** (*TYPE*) – Description
- **col** (*tuple, optional*) – Description
- **lbl** (*str, optional*) – Description

worms.vis.**showsphere**(*c*, *r=1*, *col=(1, 1, 1)*, *lbl=''*)
TODO: Summary

Parameters

- **c** (*TYPE*) – Description
- **r** (*int, optional*) – Description
- **col** (*tuple, optional*) – Description
- **lbl** (*str, optional*) – Description

worms.vis.**showvecfrompoint**(*a*, *c*, *col=(1, 1, 1)*, *lbl=''*)
TODO: Summary

Parameters

- **a** (*TYPE*) – Description
- **c** (*TYPE*) – Description
- **col** (*tuple, optional*) – Description
- **lbl** (*str, optional*) – Description

4.1.13 Module contents

Top-level package for worms.

CHAPTER 5

Contributing

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

You can contribute in many ways:

5.1 Types of Contributions

5.1.1 Report Bugs

Report bugs at https://github.com/willsheffler/worms/issues.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

5.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with "bug" and "help wanted" is open to whoever wants to implement it.

5.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with "enhancement" and "help wanted" is open to whoever wants to implement it.

5.1.4 Write Documentation

worms could always use more documentation, whether as part of the official worms docs, in docstrings, or even on the web in blog posts, articles, and such.

5.1.5 Submit Feedback

The best way to send feedback is to file an issue at https://github.com/willsheffler/worms/issues.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

5.2 Get Started!

Ready to contribute? Here's how to set up *worms* for local development.

1. Fork the *worms* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/worms.git
```

3. Install your local copy into a conda environment

   $ conda env update

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

   Now you can make your changes locally.

5. When you're done making changes, check that your changes pass the tests:

```
$ pytest
```

   To get flake8 and tox, just pip install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

5.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.

2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 2.6, 2.7, 3.3, 3.4 and 3.5, and for PyPy. Check https://travis-ci.org/willsheffler/worms/pull_requests and make sure that the tests pass for all supported Python versions.

5.4 Tips

Hook up your editor to run this (see your editor's docs. . .):

$ python util/runtests.py CURRENT_FILE

Credits —-~~~~~ - Will Sheffler - Ivan Vulovic - Una Natterman - David Baker - Yang Hisa - Rubal Mout

CHAPTER 6

History

6.1 0.1.24 (2018-04-25)

- change two-stage xindex thershold multiplier for 1st stage from 1.25 to 2.0

6.2 0.1.23 (2018-04-24)

- add initial draft of Una's criteria for P213 xtal

6.3 0.1.22 (2018-04-20)

- add initial draft of Una's criteria for P3, P4 and P6 layers

6.4 0.1.21 (2018-04-19)

- peace sign heterotrimer site compatibility bug fix
- xindex search now stores lists instead of one-per-bin

6.5 0.1.20 (2018-04-16)

- add more dihedral sym files

6.6 0.1.19 (2018-04-6)

- minor performance impovements in hash-index based search

6.7 0.1.17 (2018-04-4)

- hash-index based search for Cyclic(. . . , origin_seg=. . .)

6.8 0.1.16 (2018-03-15)

- I52 symmetry bug fix

6.9 0.1.15 (2018-03.05)

- add NullCriteria that always returns 0 err

6.10 0.1.14 (2018-02-28)

- fix provenance bug in 'cyclic entry' cases
- try to make serialization of Segments more efficient

6.11 0.1.13 (2018-02-16)

- raise exception if system too large

6.12 0.1.12 (2018-02-16)

- partial bignum fix

6.13 0.1.11 (2018-02-15)

- fix memory bug
- make distribution work better
- maybe fix pose bug, still some logic err, but maybe ok

6.14 0.1.10 (2018-02-15)

- add max_results option to grow
- fix C2 sym bug
- fix xform axis cen bug
- fix memory "bug" with batch parallel processing

6.15 0.1.9 (2018-02-08)

- add max_samples option to grow

6.16 0.1.8 (2018-02-07)

- origin_seg bug fix

6.17 0.1.6 (2018-02-01)

- middle-to-end cyclic fusions working
- add pretty logo of mid-to-end C3 fusion

6.18 0.1.6 (2018-02-01)

- bug fix in fullatom option

6.19 0.1.5 (2018-02-01)

- add fullatom option to Worms.sympose
- cyclic premutation working for simple beginning-to-end case

6.20 0.1.4 (2018-02-01)

- pypi deplolment derp

6.21 0.1.3 (2018-02-01)

- pypi deplolment derp

6.22 0.1.2 (2018-01-23)

- Add __main__ for module to run tests
- move worms.pdb to worms.data because pdb is kinda reserved
- move utility stuff to util.py
- add some interactive visualization utils for debugging

6.23 0.1.1 (2018-01-23)

- First release on PyPI.

CHAPTER 7

Indices and tables

- genindex
- modindex
- search

Python Module Index

W

worms, 33
worms.bblock, 15
worms.criteria, 14
worms.criteria.base, 9
worms.criteria.bounded, 10
worms.criteria.cyclic, 13
worms.criteria.unbounded, 14
worms.data, 14
worms.database, 15
worms.edge, 16
worms.khash, 15
worms.khash.khash_cffi, 15
worms.pose_contortions, 17
worms.search, 18
worms.segments, 18
worms.util, 24
worms.vertex, 29
worms.vis, 30

Index

A

add() (worms.database.SpliceDB method), 16
alignment() (worms.criteria.base.NullCriteria method), 10
alignment() (worms.criteria.bounded.AxesIntersect method), 11
alignment() (worms.criteria.cyclic.Cyclic method), 13
alignment() (worms.criteria.unbounded.AxesAngle method), 14
allowed_attributes (worms.criteria.base.WormCriteria attribute), 10
allowed_pairs (worms.segments.Spliceable attribute), 21
angle (worms.criteria.bounded.AxesIntersect attribute), 10
AnnoPose (class in worms.pose_contortions), 17
AxesAngle (class in worms.criteria.unbounded), 14
AxesIntersect (class in worms.criteria.bounded), 10

B

BBlock() (in module worms.bblock), 15
bblock() (worms.database.BBlockDB method), 15
bblock_components() (in module worms.bblock), 15
bblock_dump_pdb() (in module worms.bblock), 15
bblock_str() (in module worms.bblock), 15
BBlockDB (class in worms.database), 15
bblockfile() (worms.database.BBlockDB method), 16
bigprod() (in module worms.util), 25
body (worms.segments.Spliceable attribute), 21
bodyid (worms.segments.Segment attribute), 18
bodyid (worms.segments.Spliceable attribute), 21
build_pdb_data() (worms.database.BBlockDB method), 16

C

cachepath() (worms.database.SpliceDB method), 16
cgo_cyl() (in module worms.vis), 30
cgo_lineabs() (in module worms.vis), 30
cgo_segment() (in module worms.vis), 30
cgo_sphere() (in module worms.vis), 31
chain (worms.segments.SpliceSite attribute), 21
chain_of_ires (in module worms.bblock), 15
chains (worms.segments.Spliceable attribute), 22
check_lock_cachedir() (worms.database.BBlockDB method), 16
children (worms.criteria.base.CriteriaList attribute), 9
clear_caches() (worms.segments.Worms method), 23
contig_idx_breaks (in module worms.util), 25
contort_pose_chains() (in module worms.pose_contortions), 17
cpu_count() (in module worms.util), 25
criteria (worms.segments.Worms attribute), 23
CriteriaList (class in worms.criteria.base), 9
Crystal_F432_C3_C4() (in module worms.criteria.unbounded), 14
Crystal_I213_C2_C3() (in module worms.criteria.unbounded), 14
Crystal_I432_C2_C4() (in module worms.criteria.unbounded), 14
Crystal_P213_C3_C3() (in module worms.criteria.unbounded), 14
Crystal_P4132_C2_C3() (in module worms.criteria.unbounded), 14
Crystal_P432_C4_C4() (in module worms.criteria.unbounded), 14
crystinfo() (worms.criteria.unbounded.AxesAngle method), 14
Cyclic (class in worms.criteria.cyclic), 13
cyclic_entry (worms.pose_contortions.AnnoPose attribute), 17
CyclicTrim (class in worms.pose_contortions), 17

D

D2() (in module worms.criteria.bounded), 11
D3() (in module worms.criteria.bounded), 12
D4() (in module worms.criteria.bounded), 12
D5() (in module worms.criteria.bounded), 12
D6() (in module worms.criteria.bounded), 12
detail (worms.segments.Worms attribute), 23
dicts_to_items() (in module worms.util), 25

distinct_axes (worms.criteria.bounded.AxesIntersect attribute), 11

E

Edge() (in module worms.edge), 16
end_of_chain (worms.segments.Spliceable attribute), 22
entrypol (worms.segments.Segment attribute), 18
entryresid (worms.segments.Segment attribute), 18
entrysiteid (worms.segments.Segment attribute), 18
exitpol (worms.segments.Segment attribute), 19
exitresid (worms.segments.Segment attribute), 19
exitsiteid (worms.segments.Segment attribute), 19
expand_array_if_needed (in module worms.util), 25
expert (worms.segments.Segment attribute), 19

F

first_duplicate() (in module worms.util), 25
flatten_path() (in module worms.database), 16
format_atom() (in module worms.vis), 31
from_seg (worms.criteria.base.NullCriteria attribute), 10
from_seg (worms.criteria.bounded.AxesIntersect attribute), 11

G

get (worms.data.PoseLib attribute), 14
get_allowed_splices() (in module worms.edge), 16
get_bb_coords() (in module worms.util), 25
get_bb_stubs() (in module worms.util), 26
get_chain_bounds() (in module worms.util), 26
get_symdata (in module worms.util), 26
get_symdata_modified() (in module worms.util), 26
get_symfile_contents (in module worms.util), 26

H

hash_str_to_int() (in module worms.util), 26

I

Icosahedral() (in module worms.criteria.bounded), 13
index() (worms.segments.Segments method), 21
indices (worms.segments.Worms attribute), 23
infer_cyclic_symmetry() (in module worms.util), 26
init_segment_data() (worms.segments.Segment method), 19
InProcessExecutor (class in worms.util), 24
is_compatible() (worms.segments.Spliceable method), 22
is_rosetta_pose() (in module worms.vis), 31
iseg (worms.pose_contortions.AnnoPose attribute), 17
islocked_cachedir() (worms.database.BBlockDB method), 16
items_to_dicts() (in module worms.util), 26

J

jit_lossfunc() (worms.criteria.base.NullCriteria method), 10
jit_lossfunc() (worms.criteria.cyclic.Cyclic method), 13

L

len (worms.util.MultiRange attribute), 24
lever (worms.criteria.bounded.AxesIntersect attribute), 11
lineno() (in module worms.segments), 24
load_cached_bblock_into_memory() (worms.database.BBlockDB method), 16
load_cached_pose_into_memory() (worms.database.BBlockDB method), 16
load_from_pdbs() (worms.database.BBlockDB method), 16
load_from_pdbs_inner() (worms.database.BBlockDB method), 16
lock_cachedir() (worms.database.BBlockDB method), 16

M

make_contorted_pose() (in module worms.pose_contortions), 18
make_head() (worms.segments.Segment method), 19
make_pose_chains() (worms.segments.Segment method), 19
make_tail() (worms.segments.Segment method), 20
map() (worms.util.InProcessExecutor method), 24
max_sites (worms.segments.Segment attribute), 19
merge_idx() (worms.segments.Segment method), 20
merge_idx_slow() (worms.segments.Segment method), 20
min_seg_len (worms.segments.Spliceable attribute), 22
min_sites (worms.segments.Segment attribute), 19
MultiRange (class in worms.util), 24

N

nchains (worms.segments.Segment attribute), 19
no_overlapping_adjacent_residues() (in module worms.util), 27
no_overlapping_residues() (in module worms.util), 27
NonFuture (class in worms.util), 25
nside (worms.util.MultiRange attribute), 24
nsite (worms.segments.Spliceable attribute), 22
NullCriteria (class in worms.criteria.base), 10
numcom (in module worms.vis), 30
numline (in module worms.vis), 30
numpy_stub_from_rosetta_stub() (in module worms.util), 27
numray (in module worms.vis), 30
numseg (in module worms.vis), 30
numvec (in module worms.vis), 30

O

Octahedral() (in module worms.criteria.bounded), 13

P

parallel_batch_map() (in module worms.util), 27
parallel_nobatch_map() (in module worms.util), 27
partial() (worms.database.SpliceDB method), 16
polarity (worms.segments.SpliceSite attribute), 21
pose (worms.pose_contortions.AnnoPose attribute), 17
pose() (worms.database.BBlockDB method), 16
pose() (worms.segments.Worms method), 23
pose_bounds() (in module worms.util), 28
posefile() (worms.database.BBlockDB method), 16
PoseLib (class in worms.data), 14
positions (worms.segments.Worms attribute), 23
psum (worms.util.MultiRange attribute), 25
pymol_load() (in module worms.vis), 31
pymol_load_pose() (in module worms.vis), 31
pymol_xform() (in module worms.vis), 31

Q

query() (worms.database.BBlockDB method), 16
query_names() (worms.database.BBlockDB method), 16

R

random() (in module worms.search), 18
reorder_spliced_as_N_to_C() (in module worms.pose_contortions), 18
resid() (worms.segments.SpliceSite method), 21
resids() (worms.segments.Spliceable method), 22
residue_coords() (in module worms.util), 28
residue_sym_err() (in module worms.util), 28
result() (worms.util.NonFuture method), 25
rosetta_stub_from_numpy_stub() (in module worms.util), 28
rot_tol (worms.criteria.bounded.AxesIntersect attribute), 11

S

same_bodies_as() (worms.segments.Segment method), 20
score() (worms.criteria.base.CriteriaList method), 9
score() (worms.criteria.base.NullCriteria method), 10
score() (worms.criteria.base.WormCriteria method), 10
score() (worms.criteria.bounded.AxesIntersect method), 11
score() (worms.criteria.cyclic.Cyclic method), 14
score() (worms.criteria.unbounded.AxesAngle method), 14
score0 (worms.segments.Worms attribute), 23
score0sym (worms.segments.Worms attribute), 23
scores (worms.segments.Worms attribute), 23
Segment (class in worms.segments), 18
Segments (class in worms.segments), 20
segments (worms.segments.Segments attribute), 21
segments (worms.segments.Worms attribute), 23
selections (worms.segments.SpliceSite attribute), 21
seq() (worms.pose_contortions.AnnoPose method), 17
Sheet_P321() (in module worms.criteria.unbounded), 14
Sheet_P4212() (in module worms.criteria.unbounded), 14
Sheet_P6() (in module worms.criteria.unbounded), 14
show_with_axis() (in module worms.vis), 31
show_with_z_axes() (in module worms.vis), 31
showcom() (in module worms.vis), 31
showcyl() (in module worms.vis), 31
showline() (in module worms.vis), 32
showlineabs() (in module worms.vis), 32
showme() (in module worms.vis), 32
showme_pymol() (in module worms.vis), 32
showme_state (in module worms.vis), 30
showsegment() (in module worms.vis), 32
showsphere() (in module worms.vis), 33
showvecfrompoint() (in module worms.vis), 33
sitepair_allowed() (worms.segments.Spliceable method), 22
sites (worms.segments.Spliceable attribute), 22
splice_metrics_pair() (in module worms.edge), 17
Spliceable (class in worms.segments), 21
spliceable_positions() (worms.segments.Spliceable method), 22
spliceables (worms.segments.Segment attribute), 19
SpliceDB (class in worms.database), 16
splicepoint_cache (worms.segments.Worms attribute), 23
splicepoints() (worms.segments.Worms method), 23
splices() (worms.segments.Worms method), 23
SpliceSite (class in worms.segments), 21
split_at() (worms.segments.Segments method), 21
split_idx() (worms.segments.Segment method), 20
src_lb (worms.pose_contortions.AnnoPose attribute), 17
src_ub (worms.pose_contortions.AnnoPose attribute), 17
srcpose (worms.pose_contortions.AnnoPose attribute), 17
srcseq() (worms.pose_contortions.AnnoPose method), 17
start_of_chain (worms.segments.Spliceable attribute), 22
submit() (worms.util.InProcessExecutor method), 24
subpose() (in module worms.util), 28
sym_axes (worms.criteria.bounded.AxesIntersect attribute), 11
sym_seg_from (worms.pose_contortions.CyclicTrim attribute), 17
sym_seg_to (worms.pose_contortions.CyclicTrim attribute), 17
symfile_modifiers() (worms.criteria.unbounded.AxesAngle method), 14
symfile_path() (in module worms.util), 28
symname (worms.criteria.bounded.AxesIntersect attribute), 11
sympose() (worms.segments.Worms method), 23
sync_to_disk() (worms.database.SpliceDB method), 16

T

Tetrahedral() (in module worms.criteria.bounded), 13
tgtaxis1 (worms.criteria.bounded.AxesIntersect attribute), 11
tgtaxis2 (worms.criteria.bounded.AxesIntersect attribute), 11
to_seg (worms.criteria.base.NullCriteria attribute), 10
to_seg (worms.criteria.bounded.AxesIntersect attribute), 11
tol (worms.criteria.bounded.AxesIntersect attribute), 11
tqdm_parallel_map() (in module worms.util), 28
trim_pose() (in module worms.util), 29

U

unique_key() (in module worms.util), 29
unique_key_int32s() (in module worms.util), 29
unlock_cachedir() (worms.database.BBlockDB method), 16
Ux (in module worms.criteria.base), 9
Uy (in module worms.criteria.base), 9
Uz (in module worms.criteria.base), 9

V

Vertex() (in module worms.vertex), 29
vertex_single() (in module worms.vertex), 30

W

WormCriteria (class in worms.criteria.base), 10
Worms (class in worms.segments), 22
worms (module), 33
worms.bblock (module), 15
worms.criteria (module), 14
worms.criteria.base (module), 9
worms.criteria.bounded (module), 10
worms.criteria.cyclic (module), 13
worms.criteria.unbounded (module), 14
worms.data (module), 14
worms.database (module), 15
worms.edge (module), 16
worms.khash (module), 15
worms.khash.khash_cffi (module), 15
worms.pose_contortions (module), 17
worms.search (module), 18
worms.segments (module), 18
worms.util (module), 24
worms.vertex (module), 29
worms.vis (module), 30
worst_CN_connect() (in module worms.util), 29

X

x2exit (worms.segments.Segment attribute), 19
x2orgn (worms.segments.Segment attribute), 19
xform_pose() (in module worms.util), 29