
ZARP-cli

Release 0.1.0

Zavolan Lab

May 05, 2022

MODULES

1	zarp	1
1.1	zarp package	1
2	Indices and tables	9
	Python Module Index	11
	Index	13

1.1 zarp package

ZARP project root

1.1.1 Subpackages

zarp.config package

Submodules

zarp.config.cli module

zarp.config.init module

zarp.config.models module

Models and enumerators.

```
class zarp.config.models.Config(*, samples: typing.List[zarp.config.models.Sample] = [], run:
    zarp.config.models.Run = Run(identifier=None, description=None,
    cores=1, htsinfer_config=None, execution_mode=<ExecModes.RUN:
    'run'>, tool_packaging=<ToolPackaging.CONDA: '--use-conda'>,
    execution_profile=None, snakemake_config=None,
    keep_files=[<OutputFiles.CONFIGS: 'configs'>, <OutputFiles.LOGS:
    'logs'>, <OutputFiles.RESULTS: 'results'>]), user:
    zarp.config.models.User = User(surname=None, first_name=None,
    email=None, affiliations=None, urls=None, logo_location=None))
```

Bases: pydantic.main.BaseModel

ZARP-cli main configuration.

Parameters

- **sample** – Sample-specific parameters.
- **run** – Run-specific parameters.
- **user** – User-specific parameters.

sample

Sample-specific parameters.

run

Run-specific parameters.

Type *zarp.config.models.Run*

user

User-specific parameters.

Type *zarp.config.models.User*

run: *zarp.config.models.Run*

samples: `List[zarp.config.models.Sample]`

user: *zarp.config.models.User*

class `zarp.config.models.ExecModes`(*value*)

Bases: `enum.Enum`

Execution modes.

Parameters

- **DRY_RUN** – Do not download any files, infer parameters or start the analysis workflow.
- **PREPARE_RUN** – Download files and infer parameters, but do not start the analysis workflow.
- **RUN** – Download files, infer parameters and start the analysis workflow.

DRY_RUN

Do not download any files, infer parameters or start the analysis workflow.

PREPARE_RUN

Download files and infer parameters, but do not start the analysis workflow.

RUN

Download files, infer parameters and start the analysis workflow.

DRY_RUN = 'dry_run'

PREPARE_RUN = 'prepare_run'

RUN = 'run'

class `zarp.config.models.FragLenDist`(*, *mean: float = 300, sd: float = 100*)

Bases: `pydantic.main.BaseModel`

Fragment length distribution parameters of the sequencing library.

Required for single-end libraries. If not provided, default values will be used. For paired-end libraries any provided values will be ignored, as values will be automatically inferred.

Parameters

- **mean** – Mean of the fragment length distribution.
- **sd** – Standard deviation of the fragment length distribution.

mean

Mean of the fragment length distribution.

Type float

sd

Standard deviation of the fragment length distribution.

Type float

mean: float

sd: float

class zarp.config.models.**GenomeResources**(**reference_sequences: str = None, annotations: str = None*)

Bases: pydantic.main.BaseModel

Genome resources to be used for mapping reads and annotating alignments.

Parameters

- **reference_sequences** – Path to FASTA file containing reference sequences to align reads against, typically chromosome sequences.
- **annotations** – Path to GTF file containing gene annotations for the *reference_sequences*.

reference_sequences

Path to FASTA file containing reference sequences to align reads against, typically chromosome sequences.

Type Optional[str]

annotations

Path to GTF file containing gene annotations for the *reference_sequences*.

Type Optional[str]

annotations: Optional[str]

reference_sequences: Optional[str]

class zarp.config.models.**OutputFiles**(*value*)

Bases: enum.Enum

Output file types.

Parameters

- **CONFIGS** – Configuration files.
- **LOGS** – Log files.
- **RESULTS** – Result files.
- **TEMPORARY** – Temporary files.

CONFIGS

Configuration files.

LOGS

Log files.

RESULTS

Result files.

TEMPORARY

Temporary files.

CONFIGS = 'configs'

LOGS = 'logs'

RESULTS = 'results'

TEMPORARY = 'temporary'

class zarp.config.models.**ReadLayout**(* , *three: str = None*)

Bases: pydantic.main.BaseModel

Adapters and linkers that may be present in a library's reads.

Parameters **three** – 3'-end adapter, truncated during preprocessing.

three

3'-end adapter, truncated during preprocessing.

Type Optional[str]

three: Optional[str]

class zarp.config.models.**Run**(* , *identifier: str = None, description: str = None, cores: int = 1, htsinfer_config: str = None, execution_mode: zarp.config.models.ExecModes = ExecModes.RUN, tool_packaging: zarp.config.models.ToolPackaging = ToolPackaging.CONDA, execution_profile: str = None, snakemake_config: str = None, keep_files: typing.List[zarp.config.models.OutputFiles] = [<OutputFiles.CONFIGS: 'configs'>, <OutputFiles.LOGS: 'logs'>, <OutputFiles.RESULTS: 'results'>])*

Bases: pydantic.main.BaseModel

Run-specific parameters.

Parameters

- **identifier** – Unique identifier for a run.
- **description** – Run description.
- **cores** – Cores to use when running the analysis workflow.
- **configuration** – Configuration file for parameter inference, general Snakemake parameters and workflow-specific parameters.
- **execution_mode** – Execution mode to use.
- **tool_packaging** – Tool packaging option to use.
- **execution_profile** – Configuration options for execution environment.
- **keep_files** – Types of output files to keep.

cores: int

description: Optional[str]

execution_mode: *zarp.config.models.ExecModes*

execution_profile: Optional[str]


```

htsinfer_config: Optional[str]
identifier: Optional[str]
keep_files: List[zarp.config.models.OutputFiles]
snakemake_config: Optional[str]
tool_packaging: zarp.config.models.ToolPackaging

```

```

class zarp.config.models.Sample(*, file_paths: Tuple[str, None] = None, ids: zarp.config.models.SampleIds
                               = SampleIds(name=None, sra=None), read_layout:
                               zarp.config.models.ReadLayout = ReadLayout(three=None),
                               fragment_length_distribution: zarp.config.models.FragLenDist =
                               FragLenDist(mean=300, sd=100))

```

Bases: pydantic.main.BaseModel

Sample-specific parameters.

Parameters

- **file_paths** – Paths to FASTQ files of a sequencing library. Either a tuple of two paths (paired-end library) or a tuple of one path and *None* (for single-end libraries).
- **ids** – Sample identifiers and aliases.
- **read_layout** – Adapters and linkers that may be present in a library’s reads.
- **fragment_length_distribution** – Fragment length distribution parameters of the sequencing library.

file_paths

Paths to FASTQ files of a sequencing library.

Type Optional[Tuple[str, None]]

ids

Sample identifiers and aliases.

Type *zarp.config.models.SampleIds*

read_layout

Adapters and linkers that may be present in a library’s reads.

Type *zarp.config.models.ReadLayout*

fragment_length_distribution

Fragment length distribution parameters of the sequencing library.

Type *zarp.config.models.FragLenDist*

file_paths: Optional[Tuple[str, None]]

fragment_length_distribution: *zarp.config.models.FragLenDist*

ids: *zarp.config.models.SampleIds*

read_layout: *zarp.config.models.ReadLayout*

```
class zarp.config.models.SampleIds(*name: str = None, sra: str = None)
```

Bases: pydantic.main.BaseModel

Sample identifiers and aliases.

Parameters

- **name** – Human-friendly sample name.
- **sra** – SRA sample or run identifier.

name

Human-friendly sample name.

Type Optional[str]

sra

SRA sample or run identifier.

Type Optional[str]

name: Optional[str]

sra: Optional[str]

```
class zarp.config.models.ToolPackaging(value)
```

Bases: enum.Enum

Supported tool packaging options.

Parameters

- **CONDA** – Use binaries from Conda.
- **SINGULARITY** – Use Singularity containers.

CONDA

Use binaries from Conda.

SINGULARITY

Use Singularity containers.

CONDA = '--use-conda'

SINGULARITY = '--use-singularity'

```
class zarp.config.models.User(*surname: str = None, first_name: str = None, email: str = None,  
affiliations: List[str] = None, urls: List[str] = None, logo_location: str =  
None)
```

Bases: pydantic.main.BaseModel

User-specific parameters.

Parameters

- **surname** – Surname of the person running the analysis.
- **other_names** – First/other names of the person running the analysis.
- **email** – Email address of the person running the analysis.
- **affiliations** – Affiliations of the person running the analysis.
- **url** – One or more URLs with additional information about the author or their affiliation.

- **logo_location** – Path or URL pointing to image file to be used as a logo in the run report.

affiliations: Optional[List[str]]

email: Optional[str]

first_name: Optional[str]

logo_location: Optional[str]

surname: Optional[str]

urls: Optional[List[str]]

zarp.config.samples module

zarp.extensions package

Submodules

zarp.extensions.ensembl module

zarp.extensions.htsinfer module

zarp.extensions.sra module

zarp.run package

Submodules

zarp.run.cleanup module

zarp.run.snakemake module

1.1.2 Submodules

1.1.3 zarp.zarp module

CLI runner for the ZARP RNA-Seq analysis pipeline.

class zarp.zarp.LogLevels(*value*)

Bases: enum.Enum

Log level enumerator.

CRITICAL = 50

DEBUG = 10

ERROR = 40

INFO = 20

WARN = 30

WARNING = 30

`zarp.zarp.main()` → None

Main function.

Parameters `args` – Command-line arguments and their values.

`zarp.zarp.parse_args(args: Optional[Sequence[str]] = None)` → `argparse.Namespace`

Parse CLI arguments.

`zarp.zarp.setup_logging(verbosity: str = 'INFO')` → None

Configure logging.

INDICES AND TABLES

- genindex
- modindex

PYTHON MODULE INDEX

Z

- zarp, 1
- zarp.config, 1
- zarp.config.cli, 1
- zarp.config.init, 1
- zarp.config.models, 1
- zarp.config.samples, 7
- zarp.extensions, 7
- zarp.extensions.ensembl, 7
- zarp.extensions.htsinfer, 7
- zarp.extensions.sra, 7
- zarp.run, 7
- zarp.run.cleanup, 7
- zarp.run.snakemake, 7
- zarp.zarp, 7

A

affiliations (*zarp.config.models.User* attribute), 7
 annotations (*zarp.config.models.GenomeResources* attribute), 3

C

CONDA (*zarp.config.models.ToolPackaging* attribute), 6
 Config (*class in zarp.config.models*), 1
 CONFIGS (*zarp.config.models.OutputFiles* attribute), 3, 4
 cores (*zarp.config.models.Run* attribute), 4
 CRITICAL (*zarp.zarp.LogLevels* attribute), 7

D

DEBUG (*zarp.zarp.LogLevels* attribute), 7
 description (*zarp.config.models.Run* attribute), 4
 DRY_RUN (*zarp.config.models.ExecModes* attribute), 2

E

email (*zarp.config.models.User* attribute), 7
 ERROR (*zarp.zarp.LogLevels* attribute), 7
 ExecModes (*class in zarp.config.models*), 2
 execution_mode (*zarp.config.models.Run* attribute), 4
 execution_profile (*zarp.config.models.Run* attribute), 4

F

file_paths (*zarp.config.models.Sample* attribute), 5
 first_name (*zarp.config.models.User* attribute), 7
 FragLenDist (*class in zarp.config.models*), 2
 fragment_length_distribution
 (*zarp.config.models.Sample* attribute), 5

G

GenomeResources (*class in zarp.config.models*), 3

H

htsinfer_config (*zarp.config.models.Run* attribute), 4

I

identifier (*zarp.config.models.Run* attribute), 5
 ids (*zarp.config.models.Sample* attribute), 5

INFO (*zarp.zarp.LogLevels* attribute), 7

K

keep_files (*zarp.config.models.Run* attribute), 5

L

LogLevels (*class in zarp.zarp*), 7
 logo_location (*zarp.config.models.User* attribute), 7
 LOGS (*zarp.config.models.OutputFiles* attribute), 3, 4

M

main() (*in module zarp.zarp*), 8
 mean (*zarp.config.models.FragLenDist* attribute), 2, 3
 module
 zarp, 1
 zarp.config, 1
 zarp.config.cli, 1
 zarp.config.init, 1
 zarp.config.models, 1
 zarp.config.samples, 7
 zarp.extensions, 7
 zarp.extensions.ensembl, 7
 zarp.extensions.htsinfer, 7
 zarp.extensions.sra, 7
 zarp.run, 7
 zarp.run.cleanup, 7
 zarp.run.snakemake, 7
 zarp.zarp, 7

N

name (*zarp.config.models.SampleIds* attribute), 6

O

OutputFiles (*class in zarp.config.models*), 3

P

parse_args() (*in module zarp.zarp*), 8
 PREPARE_RUN (*zarp.config.models.ExecModes* attribute),
 2

R

read_layout (*zarp.config.models.Sample* attribute), 5

ReadLayout (*class in zarp.config.models*), 4
reference_sequences
 (*zarp.config.models.GenomeResources attribute*), 3
RESULTS (*zarp.config.models.OutputFiles attribute*), 3, 4
Run (*class in zarp.config.models*), 4
run (*zarp.config.models.Config attribute*), 2
RUN (*zarp.config.models.ExecModes attribute*), 2

S

Sample (*class in zarp.config.models*), 5
sample (*zarp.config.models.Config attribute*), 1
SampleIds (*class in zarp.config.models*), 5
samples (*zarp.config.models.Config attribute*), 2
sd (*zarp.config.models.FragLenDist attribute*), 3
setup_logging() (*in module zarp.zarp*), 8
SINGULARITY (*zarp.config.models.ToolPackaging attribute*), 6
snakemake_config (*zarp.config.models.Run attribute*), 5
sra (*zarp.config.models.SampleIds attribute*), 6
surname (*zarp.config.models.User attribute*), 7

T

TEMPORARY (*zarp.config.models.OutputFiles attribute*), 3, 4
three (*zarp.config.models.ReadLayout attribute*), 4
tool_packaging (*zarp.config.models.Run attribute*), 5
ToolPackaging (*class in zarp.config.models*), 6

U

urls (*zarp.config.models.User attribute*), 7
User (*class in zarp.config.models*), 6
user (*zarp.config.models.Config attribute*), 2

W

WARN (*zarp.zarp.LogLevels attribute*), 7
WARNING (*zarp.zarp.LogLevels attribute*), 8

Z

zarp
 module, 1
zarp.config
 module, 1
zarp.config.cli
 module, 1
zarp.config.init
 module, 1
zarp.config.models
 module, 1
zarp.config.samples
 module, 7
zarp.extensions
 module, 7
zarp.extensions.ensembl
 module, 7
zarp.extensions.htsinfer
 module, 7
zarp.extensions.sra
 module, 7
zarp.run
 module, 7
zarp.run.cleanup
 module, 7
zarp.run.snakemake
 module, 7
zarp.zarp
 module, 7