definition_parser

DefinitionParser
+ field: type
+ parse: Model

return

gene

Exchangeable
+ gene_ref: ModelGene
+ method(type): type

Extends

GeneStatus
+ MANDATORY
+ ACCESSORY
+ NEUTRAL
+ FORBIDEN

ModelGene
+ model: Model
+ gene: CoreGene
+ method(type): type

CoreGene
1
+ method(type): type
+ profile: Profile

GeneBank
+ field: type
+ method(type): type

databases

RepliconDB
+ field: type
+ method(type): type

Indexes
+ field: type
+ method(type): type

model

ModelBank
+ field: type
+ method(type): type

Model
mandatory: List[ModelGene]
accessory: List[ModelGene]
neutral: List[ModelGene]
accessory: List[ModelGene]
+ method(type): type

registries

ModelRegistry
- _registry: Dict[name:str, ModelLocatic
+ method(type): type

ModelLocation
- _profiles: Dict[name str: path:str]
- _definitions: Dict[name str: DefLocatior
+ method(type): type

DefinitionLocation
+ name: str
+ fqn:str
+ path: str
+ subdefinition: Dict[name:str: DefinitionLocat
+ method(type): type

cluster

Cluster
+ hits: Hit/ValidHit
+ model: Model
+ method(type): type

build_clusters

scaffold_to_cluster

get_loners

filter_loners

profile

ProfileFactory
+ field: type
+ method(type): type

Profile
+ field: type
+ method(type): type

model_package

AbstractModelIndex
+ field: type
+ method(type): type

RemoteModelIndex
+ field: type
+ method(type): type

LocalModelIndex
+ field: type
+ method(type): type

ModelPackage
+ field: type
+ method(type): type

hit

Hit
1
+ gene: CoreGene
+ id: str
+ seq_length: int
+ replicon_name: str
+ position: int
...
+ method(type): type

HitWeight
+ itself: float
+ exchangeable: float
+ mandatory: float
+ accessory: float
+ neutral: float
+ out_of_cluster: float

ModelHit
+ hit: Hit
+ status: GeneStatus
+ gene_ref: ModelGene

Extends

system

**<<Abstract>>
AbstractSetOfHits**
+ model: Model
+ hits

MetaSetOfHits
+ method(type): type

RejectedClusters
+ clusters: List[Cluster]
+ model: Model
+ method(type): type

System
+ clusters: List[Cluster]
+ model: Model
+ method(type): type

**<<Abstract>>
AbstractSetOfHits**
+ model: Model
+ mandatory_hits: List[ValidH
+ accessory_hits: List[ValidH
+ neutral_hits: List[ValidHit]
+ forbidden_hits: List[ValidHi

**<<Abstract>>
AbstractMatchMaker**
+ sort_hits_by_status
+ present_genes
+ match

OrderedMatchMaker
+ match(clusters)

UnOrderedMatchMaker
+ match(clusters)

UnlikelySytem
+ reasons: List of str
+ __str__

LikelySystem
+ __str__

solution

Solution
+ systems: [System, ...]

config

MacsyDefaults
+ field: type
+ method(type): typ

Config
+ field: type
+ method(type): typ

model_conf_parser

ModelConfParser
+ parse: type

Use

metadata

Metadata
+ field: type
+ method(type): type

Use

Maintainer
+ field: type
+ method(type): type

<<Abstract>>
CounterpartHit
+ hit: Hit
+ status: GeneStatus
+ gene_ref: ModelGene
+ loner: False
+ multi_system: True
+ counterpart: [ModelHit,]

implements   implements

Loner
+ loner: True

MultiSystem
+ multi_system: True

Extends   Extends

LonerMultiSystem

io

likely_systems_to_tsv   loner_warning

loners_to_tsv   multi_systems_to_tsv

rejected_candidates_to_tsv   rejected_candidates_to_txt

solutions_to_tsv   summary_best_solution

systems_to_tsv   systems_to_txt

+ score: float
+ average_wholeness: typ
+ hits_positions: [int, ...]
+ hits_number: int

serialization

<<Abstract>>
SystemSerilizer
+ system: System
+ serialize(): type

TxtSystemSerializer
+ serialize(system):String

TxtLikelySystemSerializer
+ serialize(LikelySystem):Strin

TxtUnlikelySystemSerializer
+ serialize(UnlikelySystem):String

TsvSystemSerializer
+ serialize(system):String

TsvLikelySystemSerializer
+ serialize(LikelySystem):Strin

TsvSpecialHitSerializer
+ serialize(CounterpartHit):Stri

Use

TsvSolutionSerializer
+ system: System
+ serialize():String

Use