

Documentación endpoints de aplicación ApoloRocket

URL: <https://github.com/bioinformaticaInta/apolorocket>

Dependencias

Motor de DB

- PostgreSQL
- Archivo SQL con estructura de DB:
<https://github.com/bioinformaticaInta/apolorocket/blob/main/apoloRocket.sql>
- Modificar archivo “**constants.py**” con datos para conexión con DB:
 - DBNAME
 - DBUSER
 - DBPASSWD
 - DBHOST
 - DBPORT
- Archivo SQL con dump de DB cargada con referencias varias:
https://github.com/bioinformaticaInta/apolorocket/blob/main/exampleData/databaseDumps/apolorocket_dump_withReferences.sql
- Archivo SQL con dump de DB cargada con referencias varias y algunos resultados:
https://github.com/bioinformaticaInta/apolorocket/blob/main/exampleData/databaseDumps/apolorocket_dump_with_references_and_results.sql

Ruta para alojar archivos de referencias:

- Modificar archivo “**constants.py**”:
 - REFERENCEDIR

Los archivos de referencias no están en la DB, sino que deben ser archivos en disco con un formato particular, para los procesos que se deben ejecutar

Software externo

- Bowtie version 1 (<https://bowtie-bio.sourceforge.net/index.shtml>)
- ViennaRNA (<https://github.com/ViennaRNA/ViennaRNA>)

Paquetes Python3

- os
- BCBio
- BioPython
- psycpg2

- datetime
- subprocess
- shutil
- glob
- plotly
- pandas
- math
- Warnings
- json
- random
- string

Lista de endpoints

- createNewProject
- getAllProjectsInformation
- getProjectInformation
- getAllReferencesInformation
- getReferenceInformation
- getAllQueriesInformation
- getQueryInformation
- getProjectComparisons
- getQueryProjects
- getReferenceProjects
- createNewReference
- addNewAnnotationToReference
- getMainTargetsPlot
- getMainTargetsPlotFromDB
- getEfficiencyPlot
- getEfficiencyPlotFromDB
- getOffTargetsPlot
- getOffTargetsPlotFromDB
- getTargetsTable

createNewProject

Entrada: Ruta completa de archivo json

Salida: Tupla con ruta de archivo json de salida

Tiempo de ejecución: 0.1s

Formato de archivo de entrada:

```
{
  name:str NOT NULL -> Project name
  description:str -> Project description
  genus:str NOT NULL -> Genus name
  specie:str NOT NULL -> Specie name
  user_name:str NOT NULL -> Name main user of the project
  outputDir:str NOT NULL -> Output directory complete path
}
```

Ejemplo archivo de entrada:

```
{
  "name":"project1",
  "description":"Project test one",
  "genus":"A",
  "specie":"A",
  "user_name":"sergio",
  "outputDir":"/home/sadosky/backEnd"
}
```

Ejemplo archivo de salida (proyecto creado):

```
{
  "projectID": 5
}
```

Ejemplo archivo de salida (proyecto NO creado: Failed):

```
{
  "projectID": null
}
```

getAllProjectsInformation

Entrada: Ruta completa de archivo json

Salida: Tupla con ruta de archivo json de salida

Tiempo de ejecución: 0.5s

Formato de archivo de entrada:

```
{
  outputDir:str NOT NULL -> Output directory complete path
}
```

Ejemplo archivo de entrada:

```
{
  "outputDir":"/home/sadosky/backEnd"
}
```

Ejemplo archivo de salida:

```
[
  {
    "id": 1,
    "name": "project1",
    "description": "Project test one",
    "genus": "A",
    "specie": "A",
    "user_name": "sergio",
    "initial_date": "2024-05-13",
    "last_modification_date": "2024-05-13"
  },
  {
    "id": 2,
    "name": "project2",
    "description": "Project test two",
    "genus": "A",
    "specie": "A",
    "user_name": "sergio",
    "initial_date": "2024-05-13",
    "last_modification_date": "2024-05-13"
  },
]
```

getProjectInformation

Entrada: Ruta completa de archivo json

Salida: Tupla con ruta de archivo json de salida

Tiempo de ejecución: 0.5s

Formato de archivo de entrada:

```
{  
  projectID:int NOT NULL -> id of project in DB  
  outputDir:str NOT NULL -> Output directory complete path  
}
```

Ejemplo archivo de entrada:

```
{  
  "projectID":1,  
  "outputDir":"/home/sadosky/backEnd"  
}
```

Ejemplo de archivo de salida:

```
{  
  "id": 1,  
  "name": "project1",  
  "description": "Project test one",  
  "genus": "A",  
  "specie": "A",  
  "user_name": "sergio",  
  "initial_date": "2024-05-13",  
  "last_modification_date": "2024-05-13"  
}
```

getAllReferencesInformation

Entrada: Ruta completa de archivo json

Salida: Tupla con ruta de archivo json de salida

Tiempo de ejecución: 0.5s

Formato de archivo de entrada:

```
{
  outputDir:str NOT NULL -> Output directory complete path
}
```

Ejemplo archivo de entrada:

```
{
  "outputDir":"/home/sadosky/backEnd"
}
```

Ejemplo de archivo de salida:

```
[
  {
    "id": 1,
    "genus": "Arabidopsis",
    "specie": "thaliana",
    "version": "1.0",
    "annotation": true,
    "genomic": true
  },
  {
    "id": 2,
    "genus": "Homo",
    "specie": "sapiens",
    "version": "38",
    "annotation": true,
    "genomic": true
  },
]
```

getReferenceInformation

Entrada: Ruta completa de archivo json

Salida: Tupla con ruta de archivo json de salida

Tiempo de ejecución: 0.5s

Formato de archivo de entrada:

```
{  
  referenceID:int NOT NULL -> id of reference in DB  
  outputDir:str NOT NULL -> Output directory complete path  
}
```

Ejemplo archivo de entrada:

```
{  
  "referenceID":5,  
  "outputDir":"/home/sadosky/backEnd"  
}
```

Ejemplo de archivo de salida:

```
{  
  "id": 5,  
  "genus": "Homo",  
  "specie": "sapiens",  
  "version": "38",  
  "annotation": true,  
  "genomic": false  
}
```


getAllQueriesInformation

Entrada: Ruta completa de archivo json

Salida: Tupla con ruta de archivo json de salida

Tiempo de ejecución: 0.5s

Formato de archivo de entrada:

```
{
  outputDir:str NOT NULL -> Output directory complete path
}
```

Ejemplo archivo de entrada:

```
{
  "outputDir":"/home/sadosky/backEnd"
}
```

Ejemplo de archivo de salida:

```
[
  {
    "id": 1,
    "name": "Arabidopsis_thaliana_AT3G03272.1",
    "sequence": "AGAGTAAGCTAGTCCAATTTAAACGATATTTAAGCACT"
  },
  {
    "id": 2,
    "name": "Arabidopsis_thaliana_AT1G45050.2",
    "sequence": "TAAATTAAGAATAGTAAAAAGCTCTTCATTTGCATAAG"
  },
  {
    "id": 3,
    "name": "Arabidopsis_thaliana_AT1G16510.1",
    "sequence": "GTAGTGTAACATAGTATTTGATTTGTATACTGCTGTTGTACC"
  },
  {
    "id": 4,
    "name": "Drosophila_melanogaster_FBtr0343837",
    "sequence": "GAAACACTCATCAAATGTTCAAGGAGTAACATGTCAATTTA"
  }
]
```

getQueryInformation

Entrada: Ruta completa de archivo json

Salida: Tupla con ruta de archivo json de salida

Tiempo de ejecución: 0.5s

Formato de archivo de entrada:

```
{  
  queryID:int NOT NULL -> id of query in DB  
  outputDir:str NOT NULL -> Output directory complete path  
}
```

Ejemplo archivo de entrada:

```
{  
  "queryID":1,  
  "outputDir":"/home/sadosky/backEnd"  
}
```

Ejemplo de archivo de salida:

```
{  
  "id": 2,  
  "name": "Arabidopsis_thaliana_AT1G45050.2",  
  "sequence": "TAAAATTAAGAATAGTAAAAAGCTCTTCATTGCA"  
}
```

getProjectComparisons

Entrada: Ruta completa de archivo json

Salida: Tupla con ruta de archivo json de salida

Tiempo de ejecución: 0.5s

Formato de archivo de entrada:

```
{
  projectID:int NOT NULL -> id of project in DB
  outputDir:str NOT NULL -> Output directory complete path
}
```

Ejemplo archivo de entrada:

```
{
  "projectID":5,
  "outputDir":"/home/sadosky/backEnd"
}
```

Ejemplo de archivo de salida:

```
[
  {
    "maintargetComparisonID": {
      "8": [
        {
          "query_name": "Oryza_sativa_Os06t0225100-01",
          "reference_genus": "Oryza",
          "reference_specie": "sativa",
          "reference_version": "20",
          "reference_genomic": false
        },
        {
          "query_name": "Oryza_sativa_Os06t0225100-01",
          "reference_genus": "Oryza",
          "reference_specie": "sativa",
          "reference_version": "20",
          "reference_genomic": true
        }
      ],
      "9": [
        {
          "query_name": "Oryza_sativa_Os11t0575200-00",
          "reference_genus": "Drosophila",
          "reference_specie": "melanogaster",
```

```

        "reference_version": "3.2",
        "reference_genomic": true
    },
    {
        "query_name": "Oryza_sativa_Os11t0575200-00",
        "reference_genus": "Oryza",
        "reference_specie": "sativa",
        "reference_version": "20",
        "reference_genomic": true
    }
]
}
},
{
    "offtargetComparisonID": {
        "45": [
            {
                "query_name": "Oryza_sativa_Os06t0225100-01_800_1000",
                "query_region_start": 800,
                "query_region_end": 1000,
                "reference_genus": "Arabidopsis",
                "reference_specie": "thaliana",
                "reference_version": "1.0",
                "reference_genomic": true,
                "query_region_selected": false
            }
        ],
        "65": [
            {
                "query_name": "Oryza_sativa_Os11t0575200-00_400_800",
                "query_region_start": 400,
                "query_region_end": 800,
                "reference_genus": "Arabidopsis",
                "reference_specie": "thaliana",
                "reference_version": "1.0",
                "reference_genomic": true,
                "query_region_selected": false
            }
        ],
        "71": [
            {
                "query_name": "Oryza_sativa_Os11t0575200-00_400_800",
                "query_region_start": 400,
                "query_region_end": 800,

```

```
    "reference_genus": "Drosophila",  
    "reference_specie": "melanogaster",  
    "reference_version": "3.2",  
    "reference_genomic": false,  
    "query_region_selected": false  
  }  
]  
}  
]
```

getQueryProjects

Entrada: Ruta completa de archivo json

Salida: Tupla con ruta de archivo json de salida

Tiempo de ejecución: 0.5s

Formato de archivo de entrada:

```
{
  queryID:int NOT NULL -> id of query in DB
  outputDir:str NOT NULL -> Output directory complete path
}
```

Ejemplo archivo de entrada:

```
{
  "queryID":1,
  "outputDir":"/home/sadosky/backEnd"
}
```

Ejemplo de archivo de salida:

```
[
  {
    "id": 1,
    "name": "project1",
    "description": "Project test one",
    "genus": "A",
    "specie": "A",
    "user_name": "sergio",
    "initial_date": "2024-05-13",
    "last_modification_date": "2024-05-14"
  },
  {
    "id": 5,
    "name": "project5",
    "description": "Project test five",
    "genus": "A",
    "specie": "A",
    "user_name": "sergio",
    "initial_date": "2024-05-13",
    "last_modification_date": "2024-05-14"
  }
]
```

getReferenceProjects

Entrada: Ruta completa de archivo json

Salida: Tupla con ruta de archivo json de salida

Tiempo de ejecución: 0.5s

Formato de archivo de entrada:

```
{  
  referenceID:int NOT NULL -> id of reference in DB  
  outputDir:str NOT NULL -> Output directory complete path  
}
```

Ejemplo archivo de entrada:

```
{  
  "referenceID":5,  
  "outputDir":"/home/sadosky/backEnd"  
}
```

Ejemplo de archivo de salida:

```
[  
  {  
    "id": 3,  
    "name": "project3",  
    "description": "Project test three",  
    "genus": "A",  
    "specie": "A",  
    "user_name": "sergio",  
    "initial_date": "2024-05-13",  
    "last_modification_date": "2024-05-14"  
  },  
  {  
    "id": 5,  
    "name": "project5",  
    "description": "Project test five",  
    "genus": "A",  
    "specie": "A",  
    "user_name": "sergio",  
    "initial_date": "2024-05-13",  
    "last_modification_date": "2024-05-14"  
  }  
]
```

createNewReference

Entrada: Ruta completa de archivo json

Salida: Tupla con 2 rutas de archivos generados en el servidor. Si hubo un error lanza una Excepcion.

Tiempo de ejecución:

Genoma Botrytis cinerea (40MB): 30s / 200MB RAM / 1 Thread

Transcriptoma Botrytis cinerea (21MB): 13s / 200MB RAM / 1 Thread

Genoma Arabidopsis thaliana (120MB): 1.4m / 300MB RAM / 1 Thread

Transcriptoma Arabidopsis thaliana (100MB): 2.6m / 300MB RAM / 1 Thread

Genoma Oryza sativa (364MB): 5.3m / 750MB RAM / 1 Thread

Transcriptoma Oryza sativa (73MB): 1.2m / 350MB RAM / 1 Thread

Genoma Homo sapiens (3.2GB): 74 m / 5GB RAM / 1 Thread - 57 m / 7GB RAM / 4 Threads

Transcriptoma Homo sapiens (427MB): 26 m / 1.2GB RAM / 1 Thread

Formato de archivo de entrada:

```
{
  genus:str NOT NULL -> genus name
  specie:str NOT NULL -> Specie name
  version:str NOT NULL -> Reference version
  genomic:bool NOT NULL -> true if the reference is a genome
  referenceFile:str NOT NULL-> Reference file name complete path
  annotationFile:str -> Annotation file name complete path or null if no annotation
}
```

Ejemplo archivo de entrada:

```
{
  "genus":"Arabidopsis",
  "specie":"thaliana",
  "version":"1.0",
  "genomic":true,
  "referenceFile":"/home/sadosky/backEnd/testData/Arabidopsis/genome.fna",
  "annotationFile":"/home/sadosky/backEnd/testData/Arabidopsis/annotation.gff3"
}
```


addNewAnnotationToReference

Entrada: Ruta completa de archivo json

Salida: Tupla con nombre de archivo generado en el servidor. Si falla lanza una Excepcion

Tiempo de ejecución:

Botrytis cinerea (MB): 4s / 150MB RAM / 1 Thread

Arabidopsis thaliana (MB): 20s / 350MB RAM / 1 Thread

Oryza sativa (MB): 14s / 300MB RAM / 1 Thread

Homo sapiens (MB): 1.2m / 500MB RAM / 1 Thread

Formato de archivo de entrada:

```
{  
  referenceID:int -> id value of reference in database  
  annotationFile:str -> Annotation file name complete path  
}
```

Ejemplo archivo de entrada:

```
{  
  "referenceID":9,  
  "annotationFile":"/home/sadosky/backEnd/testData/Drosophila/annotation.gff3"  
}
```

getMaintargetsPlot

Entrada: Ruta completa de archivo json

Salida: Tupla con nombres de archivos salida:

- Posición 0: html con figura
- Posición 1: json para obtener primera página de tabla de datos, con función **getTargertsTable**.

Si no hay alineamientos ambos nombres serán **None**. Si falla el alineamiento lanza una Excepcion

Tiempo de ejecución: (Depende la cantidad de referencias y cantidad de alineamientos, seguimos trabajando en la eficiencia de este paso)

Arabidopsis thaliana:

1 referencia (6 regiones target): 2s / 350MB RAM

2 referencias (2 regiones target): 8s / 350MB RAM

3 referencias (58 regiones target): 14s / 350MB RAM

Oryza sativa:

1 referencia (1250 regiones target): 1.5m / 300 MR RAM

2 referencias (1178 regiones target): 2.1m / 350MB RAM

Formato de archivo de entrada:

```
{
  projectID:int NOT NULL -> ID of the project in database
  sirnaSize:int NOT NULL -> siRNA size
  mismatches:int NOT NULL -> Allowed mismatches (can be 0-3)
  targetsInRegions:bool NOT NULL -> True to take targets separated by regions
  maxGapSize:int -> Maximum number of gap to join 2 regions
  queryGenus:str NOT NULL -> Genus of query sequence
  querySpecie:str NOT NULL -> Specie of query sequence
  querySequenceName:str NOT NULL -> Name of query sequence
  querySequence:str NOT NULL -> Query sequence
  pageSize:int NOT NULL -> Number of rows by page, null for all table in one page
  referenceIDs:list[int] NOT NULL -> List references ID in "reference" DB table
  outputDir:str NOT NULL -> Output directory complete path
}
```

Ejemplo archivo de entrada:

```
{
  "projectID":5,
  "sirnaSize":21,
  "mismatches":0,
  "targetsInRegions":true,
  "maxGapSize":null,
```

```
"queryGenus":"Oryza",
"querySpecie":"sativa",
"querySequenceName":"Os06t0225100-01",
"querySequence":"GTCCAAGAAAAAAAAAAGGAAGAAATACTCGTATGCATTTTGTGCGC",
"outputDir":"/home/sadosky/backEnd/testData/Oryza/query1",
"pageSize":5,
"referenceIDs":[7,8]
}
```

Ejemplo de HTML de salida:

<https://github.com/bioinformaticaInta/apolorocket/blob/main/exampleData/example.mainTargets.figure.html>

Ejemplo de archivo de json salida: Se usa mismo "outputDir" tomado del json de entrada:

```
{
  "totalTargets": 58,
  "pageNumber": 1,
  "pageSize": 5,
  "maintargetComparisonID": 3,
  "outputDir": "/home/sadosky/backEnd/testData/Arabidopsis/query3"
}
```

getMaintargetsPlotFromDB

Entrada: Ruta completa de archivo json

Salida: Tupla con rutas de archivos salida:

- Posición 0: html con figura
- Posición 1: json para obtener primera página de tabla de datos, con función **getTargetsTable**.

Si no hay alineamientos ambos nombres serán **None**

Tiempo de ejecución: 2s

Formato de archivo de entrada:

```
{
  pageSize:int NOT NULL -> Number of rows by page, null for all table in one page
  maintargetComparisonID:int NOT NULL -> Maintarget comparison ID in DB
  outputDir:str NOT NULL -> Output directory complete path
}
```

Ejemplo archivo de entrada:

```
{
  "maintargetComparisonID": 2,
  "pageSize": 5,
  "outputDir": "/home/sadosky/backEnd/testData/Arabidopsis/query2"
}
```

Ejemplo de HTML de salida:

<https://github.com/bioinformaticaInta/apolorocket/blob/main/exampleData/exampleMaintargetFromDB.mainTargets.figure.html>

Ejemplo de archivo de json salida: Se usa mismo "outputDir" tomado del json de entrada:

```
{
  "totalTargets": 6,
  "pageNumber": 1,
  "pageSize": 5,
  "maintargetComparisonID": 2,
  "outputDir": "/home/sadosky/backEnd/testData/Arabidopsis/query2"
}
```

getEfficiencyPlot

Entrada: Ruta completa de archivo json

Salida: Tupla con nombres de archivos salida:

- Posición 0: html con figura
- Posición 1: json para obtener primera página de tabla de datos, con función **getTargertsTable**.

Si no se puede calcular la eficiencia, ambos nombres serán **None**. Si falla el calculo de eficiencia se lanza una Excepcion

Tiempo de ejecución: 2s

Formato de archivo de entrada:

```
{
  sirnaSize:int NOT NULL -> siRNA size
  strandCheck:bool NOT NULL -> Strand selection is enabled or disabled
  endCheck:bool NOT NULL -> End stability selection is enabled or disabled
  accessibilityCheck:bool NOT NULL -> Target site accessibility is enabled or disabled
  accessibilityWindow:int NOT NULL -> Accessibility window
  endStabilityTreshold:float NOT NULL -> End stability treshold
  tsAccessibilityTreshold:float NOT NULL -> Target site accessibility threshold
  terminalCheck:bool NOT NULL
  method:str NOT NULL -> Method to calculate efficiency ('sifi21' or 'IA')
  pageSize:int NOT NULL -> Number of rows by page, null for all table in one page
  maintargetComparisonID:int NOT NULL -> Maintarget_comparison_id value in DB
  maintargetNumbers:list[int] NOT NULL -> List with main targets regions numbers
  outputDir:str NOT NULL -> Output directory path
}
```

Ejemplo archivo de entrada:

```
{
  "sirnaSize":21,
  "strandCheck":true,
  "endCheck":true,
  "accessibilityCheck":true,
  "accessibilityWindow":8,
  "endStabilityTreshold":1.0,
  "tsAccessibilityTreshold":0.1,
  "terminalCheck":true,
  "method":"sifi21",
  "pageSize":10,
  "maintargetComparisonID":6,
  "maintargetNumbers":[1,2],
  "outputDir":"/home/sadosky/backEnd/testData/Drosophila/query1"
```

```
}
```

Ejemplo de HTML de salida:

<https://github.com/bioinformaticaInta/apolorocket/blob/main/exampleData/exampleEfficiencyFigure.html>

Ejemplo de archivo de salida:

```
{  
  "totalTargets": 3,  
  "pageNumber": 1,  
  "pageSize": 10,  
  "maintargetComparisonID": 6,  
  "outputDir": "/home/sadosky/backEnd/testData/Drosophila/query1"  
}
```

getEfficiencyPlotFromDB

Entrada: Ruta completa de archivo json

Salida: Tupla con nombres de archivos salida:

- Posición 0: html con figura
- Posición 1: json para obtener primera página de tabla de datos, con función **getTargertsTable**.

Si no se encuentra calculada la eficiencia, ambos nombres serán **None**.

Tiempo de ejecución: 2s

Formato de archivo de entrada:

```
{
  method:str NOT NULL -> Method to calculate efficiency ('sifi21' or 'IA')
  pageSize:int NOT NULL -> Number of rows by page, null for all table in one page
  maintargetComparisonID:int NOT NULL -> Maintarget_comparison_id value in DB
  outputDir:str NOT NULL -> Output directory path
}
```

Ejemplo archivo de entrada:

```
{
  "method":"sifi21",
  "pageSize":10,
  "maintargetComparisonID":8,
  "outputDir":"/home/sadosky/backEnd/testData/Oryza/query1"
}
```

Ejemplo de HTML de salida:

<https://github.com/bioinformaticalnta/apolorocket/blob/main/exampleData/example.efficiencyFromDB.figure.html>

Ejemplo de archivo de salida:

```
{
  "totalTargets": 1178,
  "pageNumber": 1,
  "pageSize": 10,
  "maintargetComparisonID": 8,
  "outputDir": "/home/sadosky/backEnd/testData/Oryza/query1"
}
```

getOfftargetsPlot

Entrada: Ruta completa de archivo json

Salida: Tupla con nombres de archivos salida:

- Posición 0: html con figura referencia pos 0 de referenceIDs
-
- Posición N: html con figura referencia pos N de referenceIDs

Si no hay alineamientos con una referencia retorna None la posición correspondiente de la tupla

Tiempo de ejecución: 2m (Depende la cantidad de referencias y cantidad de alineamientos, seguimos trabajando en la eficiencia de este paso)

Formato de archivo de entrada:

```
{
  projectID:int NOT NULL -> ID of the project in database
  sirnaSize:int NOT NULL -> siRNA size
  mismatches:int NOT NULL -> Allowed mismatches (can be 0-3)
  targetsInRegions:bool NOT NULL -> True to take targets separated by regions
  maxGapSize:int -> Maximum number of gap to join 2 regions
  queryID:int NOT NULL -> Query id in DB
  queryRegionStart:int NOT NULL -> Start on query (interbase)
  queryRegionEnd:int NOT NULL -> End on query (interbase)
  pageSize:int NOT NULL -> Page number for table pagination
  referenceIDs:list[int] NOT NULL -> List references ID in "reference" DB table
  outputDir:str NOT NULL -> Output directory path
}
```

Ejemplo archivo de entrada:

```
{
  "projectID":5,
  "sirnaSize":21,
  "mismatches":0,
  "targetsInRegions":true,
  "maxGapSize":null,
  "queryID":"8",
  "queryRegionStart":400,
  "queryRegionEnd":800,
  "pageSize":5,
  "referenceIDs":[1,3,4,6,8,9,10],
  "outputDir":"/home/sadosky/backEnd/testData/Oryza/query2"
}
```


Ejemplo de archivo de salida:

<https://github.com/bioinformaticaInta/apolorocket/blob/main/exampleData/example.offTargets.figure.html>

getOfftargetsPlotFromDB

Entrada: Ruta completa de archivo json

Salida: Tupla con ruta de archivo html de salida o None si no hay alineamientos en esa comparación

Tiempo de ejecución:

Formato de archivo de entrada:

```
{
  pageSize:int NOT NULL -> Number of rows by page, null for all table in one page
  offtargetComparisonID:int NOT NULL -> Offtarget comparison ID in DB
  outputDir:str NOT NULL -> Output directory complete path
}
```

Ejemplo archivo de entrada:

```
{
  "pageSize": 5,
  "offtargetComparisonID": 69,
  "outputDir": "/home/sgonzalez/investigador/sadosky/backEnd/testData/Oryza/query2"
}
```

Ejemplo de archivo de salida:

<https://github.com/bioinformaticaInta/apolorocket/blob/main/exampleData/example.offTargetsFromDB.figure.html>

getTargetsTable

Entrada: Ruta completa de archivo json

Salida: Tupla con ruta de archivo json de salida para tabla. None si no hay info para esa página

Tiempo de ejecución: 1s

Formato de archivo de entrada:

```
{
  totalTargets NOT NULL -> Target regions total number
  pageNumber NOT NULL -> Page number for table pagination
  pageSize NOT NULL -> Number of rows by page, null for all table in one page
  maintargetComparisonID NOT NULL -> Maintarget_comparison_id value in DB
  outputDir NOT NULL -> Output directory path
}
```

Ejemplo archivo de entrada:

```
{
  "totalTargets": 58,
  "pageNumber": 10,
  "pageSize": 2,
  "maintargetComparisonID": 3,
  "outputDir": "/home/sadosky/backEnd/testData/Arabidopsis/query3"
}
```

Ejemplo de archivo de salida:

```
[
  {
    "Target number": 19,
    "Reference": {
      "genus": "Arabidopsis",
      "specie": "thaliana",
      "genomic": false,
      "version": "1.0"
    },
    "Target ID": "AT4G38940.1",
    "Target region": "1124-1144",
    "Target region length": 21,
    "Annotation": [
      {
        "gene:AT4G38940": {
          "type": "gene",
          "start": 18152691,
```

```

"end": 18154132,
"qualifiers": {
  "ID": [
    "gene:AT4G38940"
  ],
  "Name": [
    "AT4G38940"
  ],
  "biotype": [
    "protein_coding"
  ],
  "description": [
    "Galactose oxidase/kelch repeat superfamily protein [Source:NCBI gene
(formerly Entrezgene);Acc:830049]"
  ],
  "gene_id": [
    "AT4G38940"
  ],
  "logic_name": [
    "araport11"
  ],
  "source": [
    "araport11"
  ]
},
"subfeatures": {
  "transcript:AT4G38940.1": {
    "type": "mRNA",
    "qualifiers": {
      "ID": [
        "transcript:AT4G38940.1"
      ],
      "Parent": [
        "gene:AT4G38940"
      ],
      "Name": [
        "AT4G38940-201"
      ],
      "biotype": [
        "protein_coding"
      ],
      "tag": [
        "Ensembl_canonical"
      ],
    }
  }
}

```

```

        "transcript_id": [
            "AT4G38940.1"
        ],
        "source": [
            "araport11"
        ]
    }
}
}
}
},
],
"Blocks in target region": 1,
"Blocks data": [
    {
        "Block number": 1,
        "Start block position in target": 1124,
        "End block position in target": 1144,
        "Start block position in query": 698,
        "End block position in query": 718,
        "Missmatches in block": 0,
        "Alignment strand": "-"
    }
]
},
{
    "Target number": 20,
    "Reference": {
        "genus": "Homo",
        "specie": "sapiens",
        "genomic": true,
        "version": "38"
    },
    "Target ID": "1",
    "Target region": "67455843-67455863",
    "Target region length": 21,
    "Annotation": [],
    "Blocks in target region": 1,
    "Blocks data": [
        {
            "Block number": 1,
            "Start block position in target": 67455843,
            "End block position in target": 67455863,
            "Start block position in query": 700,

```

```
    "End block position in query": 720,  
    "Missmatches in block": 0,  
    "Alignment strand": "-"  
  }  
]  
}
```