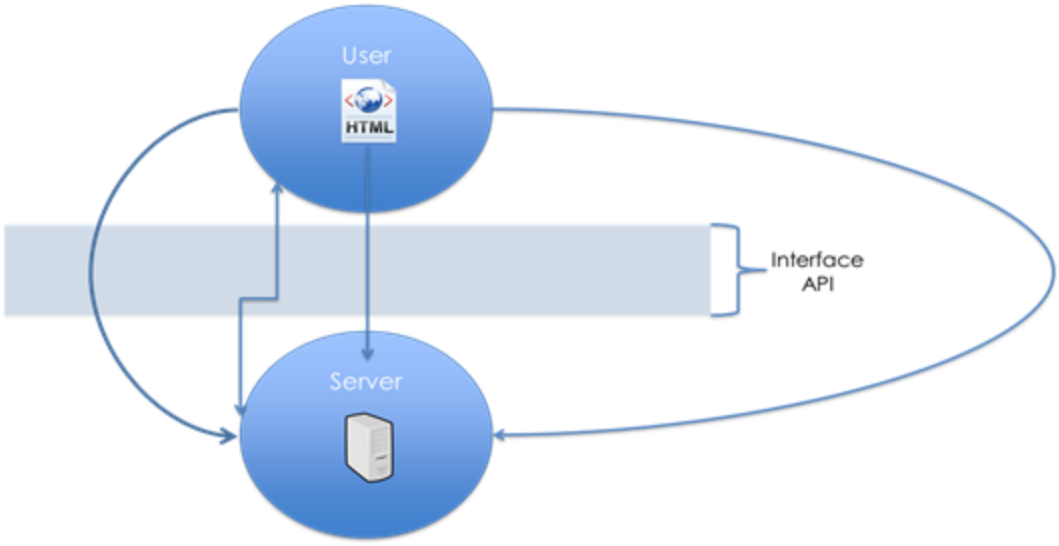
HIVE WebAPI Version 0

High-performance Integrated Virtual Environment Web API.
Version 0 March 13, 2016 Authors: AG, KK

Web App API

When running computations on large data, it is accepted, that the results generated by these calculations will not be immediately available. In order to expedite researchers having access to their results, a mobile web API is being developed to allowing mobile access to HIVE processes. This mobile web app is able to update the user on the progression of their processes by reporting their proximity to completion The interface API provides a language in which different components interact, including users' computational requests, hardware and OS processes. The API accesses html pages, compute hardware, data, and databases, facilitating software development on a cloud-computing environment. The layer, which consists of C/C++ libraries combined with DB stored procedures, implements a variety of internally and externally developed tools, algorithms, and applications. Further, the API layer enables externally developed software to interface with the HIVE environment similar to the manner in which the web user interface interacts with the environment.



Web Interface API Commands

Generic URL: https://domain/hive.cgi?api=0
GWU HIVE: https://hive.biochemistry.gwu.edu/dna.cgi?api=0
HIVE web interface uses CGI standard for requests sent to the server. The first step is establishing a user session through the login command. Upon successful login a set of cookies will be received. These same cookies will be used in each subsequent request to the server and saved upon response, similar to typical internet browser behavior. URL parameter names and values of *cmd* parameter are case sensitive.. Parameter api is always equal 0 (zero).
The following table displays a list of Web API functions for HIVE. Further detail for each command is listed in the sections below the table. Optional URL parameters are in *italic*.

Quick reference

Command	**URL**
Login	?api=0&cmdr=login&login=<email>&pswd=<password>
Logout	?api=0&cmdr=logout
List objects	?api=0&cmdr=objList&mode=json&*type*=<csv type name/regexp list>&*prop*=<csv list of metadata properties names>&*prop_name*=<csv prop names>&*prop_val*=<csv prop filter>&*search*=<text/regexp>&*info*=1&*start*=1&*cnt*=20
Getting data type definition	?api=0&cmdr=propspec&type=<csv type name list>&mode=json
Getting object's metadata by id(s)	?api=0&cmdr=propget&ids=<csv object id list>&*prop*=<csv list of metadata properties names>
Creating and editing object's metadata	?api=0&cmdr=propset&prop.<id>.<propname>[.<enumerator>]=<propertiy value>
Retrieve a File attached to an object	?api=0&cmdr=objFile&ids=<csv ids>&*filename*=<filename>&*propname*=<propname>
Uploading files	/dmUploader.cgi
Submit a job	?api=0&cmdr=-qpProcSubmit&svc=<svc-Name of Service>&prop.svc-Name of Service.svc=<svc-Name of Service>&prop.svc-Name of Service.[…]=[…]
Check status of a job	?api=0&cmdr=-qpRawCheck&reqObjID=<objectID>

Retrieve messages of a job	?api=0&cmdr=-qpReqInfo&reqObjID=<objectID>
URL to a job page	?api=0&cmdr=<svc-name>&id=<objectID>
Retrieve annotations	?api=0&cmdr=ionGenBankAnnotPosMap&ids=<csv object id list>&seqID=NC_xxx&pos_start=..&pos_end=...cnt=..

Commands detail

login

URL: ?api=0&cmdr=login&login=<email>&pswd=<password>
Description: URL to establish new session. The url must be submitted through multipart/form-data POST over SSL 128-bit encrypted https channel
Example: api=0&cmdr=login&login=user@google.com&pswd=password123
Response: Status 200 upon successful session initiation. In case of error additional text in body of response could server as user error message. Cookies received upon success should be saved in between all requests to server.

logout

URL: ?api=0&cmdr=logout
Description: End user session.
Example: api=0&cmdr=logout
Response: Status 200 upon successful session initiation. In case of error additional text in body of response could server as user error message. Cookies received upon success should be saved in between all requests to server.

objList - list objects

URL: ?api=0&cmdr=objList&mode=json&*type*=<csv type name/regexp list>&*prop*=<csv list of metadata properties names>&*prop_name*=<csv prop names>&*prop_val*=<csv prop filter>&*search*=<text search>&*info*=1&*start*=0&*cnt*=20
Example:

List all objects of type u-file (user file) and derived, line genome, nuc-read, etc.:
?api=0&cmdr=objList&mode=json&type=u-file%2B
List all objects with word 'human' in any property:
?api=0&cmdr=objList&mode=json&search=human
List object with '3879' in property 'query' AND 'svc' equal to 'svc-align-hexagon'
?api=0&cmdr=objList&mode=json&prop_name=query,svc&prop_val=3879,svc-align-hexgon

Response: JSON object.
Description:

type – object's type names, comma separated: sysfolder, folder, u-file, genome, nuc-read, process, etc. Adding '+' (%2B) after type name brings objects of that and derived types. Basic regular expressions are supported '^', '$', '*', ranges '[]'
prop – comma separated list of property names to include into output. By default all properties are included. Built-in property '_id' is a string designating unique object id in HIVE, '_type' is name of the object type.
prop_name, prop_val – comma separated **pairs** of AND-ed search criteria rules: property match value, Basic regular expressions are supported.
search – text or regular expression searched through all properties of objects.
Info – if specified provides additional object with total count, and current position
start – 0-based offset in result.
cnt – quantity of objects to return, in result if quantity is less than page size, that page is the last page.

propspec - getting data type definition

URL: ?api=0&cmdr=propspec&type=<csv type name list>&mode=json
Description: Retrieve object type definition. To get list of available types use objList command with type=type

type - csv list of type names.

Response: JSON object(s).

```
{
    "_id": "type.1",
    "name": "type",
    "title": "object type",
    "description": "Object type",
    "_attributes": {
        "created": {
            "type": "datetime",
            "title": "Created",
```

```
            "order": 0,
            "is_readonly": true
        },
        "modified": {
            "type": "datetime",
            "title": "Modified",
            "order": 0,
            "is_readonly": true
        },
        "name": {
            "type": "string",
            "title": "Name",
            "description": "case-insensitive unique name, [a-zA-Z0-9_-] character set, must not be used by any
other type, must not start with _",
            "order": 1,
            "is_key": true,
            "is_readonly": false,
            "is_summary": true,
            "brief": "$_(v) <b>Type</b>"
        },
        "title": {
            "type": "string",
            "title": "Title",
            "description": "human-readable title to show on pages",
            "order": 2,
            "is_readonly": false
        },
        "description": {
            "type": "text",
            "title": "Description",
            "order": 3,
            "is_readonly": false
        },
        "parent": {
            "type": "obj",
            "title": "Parent type(s)",
            "description": "one or more types whose fields will be inherited by this one; only one of them may
be non-abstract",
            "order": 4,
            "is_readonly": false,
            "is_multi": true,
            "constraint": "search",
            "constraint_data": "{\n  \"explorer\": true,\n  \"fetch\": \"id\",\n  \"inline\": \"name\",\n  \"
inline_url\": \"http://?cmd=propget&type=^type$&mode=csv&raw=1&useTypeDomainId=1&useTypeUPObj=1\",\n  \"url\":
\"http://?cmd=objList&type=^type$&mode=csv&raw=1&useTypeDomainId=1&useTypeUPObj=1\"\n}"
        },
        "is_abstract_fg": {
            "type": "bool",
            "default_value": "0",
            "title": "Abstract",
            "description": "if a type is abstract, objects of that type cannot exist in the system - but such a
type can be used for multiple inheritance",
            "order": 5,
            "is_readonly": false
        },
        "fields": {
            "type": "array",
            "title": "Fields",
            "order": 8,
            "is_readonly": false,
            "_children": {
                "field_basics": {
                    "type": "list",
                    "title": "Basics",
                    "order": 1,
                    "is_readonly": false,
                    "is_multi": true,
                    "_children": {
                        "field_name": {
                            "type": "string",
                            "title": "Name",
```

```
                            "description": "WARNING: changing the name in an existing type will break existing
objects of this type or descendant types!!! Case-insensitive unique name within this type, [a-zA-Z0-9_-]
character set, must not be used by any other field within this type, must not start with _",
                            "order": 1,
                            "is_key": true,
                            "is_readonly": false
                        },
                        "field_title": {
                            "type": "string",
                            "title": "Title",
                            "description": "human-readable field title to show on pages",
                            "order": 2,
                            "is_readonly": false
                        },
                        "field_type": {
                            "type": "string",
                            "title": "Value type",
                            "order": 3,
                            "is_readonly": false,
                            "constraint": "choice",
                            "constraint_data":
"string|text|integer|real|bool|array|list|date|time|datetime|url|obj|password|file|type2array///Include an
object type as array|type2list///Include an object type as list"
                        },
                        "field_parent": {
                            "type": "string",
                            "title": "Parent field",
                            "description": "name of parent field",
                            "order": 4,
                            "is_readonly": false
                        },
                        "field_order": {
                            "type": "real",
                            "title": "Order",
                            "description": "numeric (integer or real) order for sorting within parent field",
                            "order": 5,
                            "is_readonly": false
                        },
                        "field_default_value": {
                            "type": "string",
                            "title": "Default value",
                            "order": 6,
                            "is_readonly": false
                        },
                        "field_include_type": {
                            "type": "obj",
                            "title": "Inclusion from type",
                            "description": "type whose fields will be included into this one as sub-fields of
array or list",
                            "order": 7,
                            "is_readonly": false,
                            "constraint": "search",
                            "constraint_data": "{\n  \"explorer\": true,\n  \"fetch\": \"id\",\n  \"inline\": \"
name\",\n  \"inline_url\": \"http://?cmd=propget&type=^type$&mode=csv&raw=1&useTypeDomainId=1&useTypeUPObj=1\",
\n  \"url\": \"http://?cmd=objList&type=^type$&mode=csv&raw=1&useTypeDomainId=1&useTypeUPObj=1\"\n}"
                        }
                    }
                },
                "field_flags": {
                    "type": "list",
                    "title": "Flags",
                    "order": 2,
                    "is_readonly": false,
                    "is_multi": true,
                    "_children": {
                        "field_role": {
                            "type": "string",
                            "title": "Role",
                            "order": 11,
                            "is_readonly": false,
                            "constraint": "choice",
```

```
                            "constraint_data": "in|out"
                        },
                        "field_is_key_fg": {
                            "type": "bool",
                            "default_value": "0",
                            "title": "Key",
                            "description": "field is part of unique identifier of the record or parent list or
array",
                            "order": 12,
                            "is_readonly": false
                        },
                        "field_is_readonly_fg": {
                            "type": "integer",
                            "default_value": "0",
                            "title": "Read-only",
                            "order": 13,
                            "is_readonly": false,
                            "constraint": "choice",
                            "constraint_data": "0///normal field|1///calculated outside and NOT editable on web
pages|-1///write-once|-2///submit-once|2///autofill with default value"
                        },
                        "field_is_optional_fg": {
                            "type": "bool",
                            "default_value": "0",
                            "title": "Optional",
                            "description": "field could be missing completely",
                            "order": 14,
                            "is_readonly": false
                        },
                        "field_is_multi_fg": {
                            "type": "bool",
                            "default_value": "0",
                            "title": "Multi-value",
                            "description": "field can have multiple values",
                            "order": 15,
                            "is_readonly": false
                        },
                        "field_is_virtual_fg": {
                            "type": "bool",
                            "default_value": "0",
                            "title": "Virtual",
                            "order": 15,
                            "is_readonly": false
                        },
                        "field_is_hidden_fg": {
                            "type": "bool",
                            "default_value": "0",
                            "title": "Hidden",
                            "description": "field is not visible on pages",
                            "order": 16,
                            "is_readonly": false
                        },
                        "field_is_summary_fg": {
                            "type": "bool",
                            "default_value": "0",
                            "title": "Summary",
                            "order": 17,
                            "is_readonly": false
                        },
                        "field_is_batch_fg": {
                            "type": "bool",
                            "default_value": "0",
                            "title": "Batchable",
                            "order": 18,
                            "is_readonly": false
                        }
                    }
                },
                "field_constraint_list": {
                    "type": "list",
                    "title": "Value constraints",
```

```
                    "order": 3,
                    "is_readonly": false,
                    "is_multi": true,
                    "_children": {
                        "field_constraint": {
                            "type": "string",
                            "title": "Constraint type",
                            "order": 21,
                            "is_readonly": false,
                            "constraint": "choice",
                            "constraint_data": "choice|choice+|regexp|range|url|search|search+|one-of|type|eval"
                        },
                        "field_constraint_data": {
                            "type": "text",
                            "title": "Constraint data",
                            "order": 22,
                            "is_readonly": false
                        },
                        "field_constraint_description": {
                            "type": "text",
                            "title": "Constraint description",
                            "order": 23,
                            "is_readonly": false
                        }
                    }
                },
                "field_presentation": {
                    "type": "list",
                    "title": "Presentation",
                    "order": 4,
                    "is_readonly": false,
                    "is_multi": true,
                    "_children": {
                        "field_brief": {
                            "type": "string",
                            "title": "Brief format",
                            "description": "Empty or 0 not a brief member; [text]$_(v)[text], where $_(v) is
substituted by prop value",
                            "order": 31,
                            "is_readonly": false
                        },
                        "field_link_url": {
                            "type": "string",
                            "title": "Link URL",
                            "description": "next to the field, adds a link using current field value as $(v)
substitution in url",
                            "order": 32,
                            "is_readonly": false
                        },
                        "field_description": {
                            "type": "text",
                            "title": "Description",
                            "order": 33,
                            "is_readonly": false
                        }
                    }
                }
            }
        }
    }
}
```


propget - getting object's metadata by id(s)

URL: ?api=0&cmdr=propget&mode=json&ids=<csv object id list>*&prop*=<csv list of metadata properties names>
Response: JSON object.
Description:

Ids - csv list of objects ids
prop – comma separated list of property names to include into output. By default all properties are included. . Built-in property '_id' is a string designating unique object id in HIVE, '_type' is name of the object type.


```
[
    {
        "_id": 3101500,
        "_type": "svc-align-hexagon",
        "system": {
            "action": 2,
            "completed": "2016-02-14T21:28:32-05:00",
            "progress": 834,
            "progress100": 100,
            "reqID": 58192008,
            "started": "2016-02-14T21:28:32-05:00",
            "status": 5,
            "svc": "dna-hexagon",
            "svcTitle": "HIVE-hexagon Alignment"
        },
        "created": "2016-02-14T21:28:29-05:00",
        "modified": "2016-02-21T21:33:26-05:00",
        "query": {
            "1.1": 3097913
        },
        "mismatches": {
            "3.0": {
                "considerGoodSubalignments": 1,
                "maxMissQueryPercent": 70
            }
        },
        "complexity": {
            "complexityRef": {
                "4.0.0": {
                    "acceptNNNQuaTrheshold": 0,
                    "complexityRefEntropy": 0,
                    "complexityRefWindow": 0
                }
            },
            "complexityQry": {
                "4.1.0": {
                    "complexityEntropy": 0,
                    "complexityWindow": 0,
                    "maximumPercentLowQualityAllowed": 0
                }
            }
        },
        "basicAlgParam": {
            "searchRepeatsAndTrans": 0
        },
        "alignParam": {
            "isglobal": 0,
            "allowShorterEnds": 0,
            "costMismatchNext": -6,
            "costGapOpen": -12,
            "seed": 11,
            "costMatch": 5,
            "costMismatch": -4,
            "costGapNext": -4
        },
        "advancedParam": {
            "hashCompileStp": "auto",
            "hashStp": "auto",
            "computeDiagonalWidth": "auto",
            "isoptimize": 1,
            "looseExtenderMinimumLengthPercent": 66,
            "looseExtenderMismatchesPercent": 25,
            "alignmentEngine": 1,
            "useRedundSim": 1,
            "selfQueryPosJumpInNonPerfectAlignment": 1,
```

```
            "reverseEngine": 0,
            "isbackward": 1,
            "maxExtensionGaps": 0,
            "maxHashBin": 50,
            "isextendtails": 0
        },
        "name": "Resubmitted 3099194: for test pipeline #1",
        "alignSelector": "svc-align-hexagon",
        "force_reindex": 0,
        "submitter": "dna-hexagon&cmdMode=-",
        "batch_svc": "single",
        "maxHitsPerRead": 200,
        "keepAllMatches": 2,
        "doubleStagePerfect": 1,
        "minMatchLen": "10",
        "doubleHash": 0,
        "maxNumberQuery": "all",
        "slice": 450,
        "split": "query",
        "scissors": "hiveseq",
        "subject": {
            "35": 3082221
        },
        "scoreFilter": "None"
    },
    {
        "_id": 3101501,
        "_type": "folder",
        "created": "2016-02-14T21:43:12-05:00",
        "modified": "2016-02-14T21:43:57-05:00",
        "name": "test_computations_run_2016_02_14_21_42_40",
        "child": {
            "1.1": 3101502,
            "1.1": 3101503
        }
    }
]
```


propset - creating objects, editing object's metadata

URL: ?api=0&cmdr=propset&prop.<id>.<propname>[.<enumerator>]=<propertiy value>
Response: 'prop' formatted text
Description: Each property name/value url parameter has the following fomat: prop.<id>.<name>.<eunmerator>=value. *Prefix prop.* is mandatory. *id* if a number, object with the given id is updated in database, otherwise new object is created and its id is returned in response. All properties with same id get assigned to same new or existing object. *name* of property correspond to filed name in type specifitation (see cmdr=propspec). If a property has typ list or array element of aray should be enmerated with paths like 1.2.12. Path depends on location of propertiy in type specification. Memebers of list or array should have same paths enumerator to constitute a line in list or array.

Request

cmdr=propset&prop.newObj.name.4=some%20file&prop.newObj.description.12=234&prop.newObj._type=u-file&prop.newObj.taxonomy.1.0=123&prop.newObj.taxonomy.1.1=7865867

Response

prop.newObj._id=8765

objFile - retrieve a File attached to an object

URL: ?api=0&cmdr=objFile&ids=<csv ids>&*filename*=<filename>&*propname*=<propname>
Response: File content. If multiple objects given files will be concatenated together (ex: grab a csv table from a bunch of objects as one table). To get individual files separate requests to the server required. Missing files are not reported.
Description:

Objects based on *u-file* type (there are other kinds of object which are not, like processes) has 'primary' upload associated with them. To get that file from object id is sufficient. Received file might be gzipped (.gz.) or zipped (.zip). Zipped file will contain file name as it was originally named upon upload.
Jobs might create more files associated with same object for which you have to know the filename and provide either *filename* or *propname*.
Each file in an upload is treated as a separate object: if user uploads a .tar.gz with 10 .fastq files in it, 10 separate objects will be created.
Ids - csv list of objects ids.
propname – name of the object's property which value to use as filename.
filename – name of the file attached to object.

seqList - retrieve sequences from an object of type Reads or Genome

URL: ?api=0&cmdr=seqList&fformat=fast(a|q|x)&ids=<ids>*&rangeStart*=<0>*&rangeLen*=<0>&start=<0>&cnt=<0>
Response: Sequence data. If multiple objects given files will be concatenated together. To get individual files separate requests to the server required. Invalid ids are not reported.
Description:

Objects based on *u-hiveseq* type has sequence data associated with them. This command allows you to retrieve that data. Request might be done to retrieve sequences with (fastq) or without (fasta) qualities. If quality was not present in original data and fastq format is requested the output will contain an automatically generated value of 30 ('?'). fastx option will detect presence of quality in objects and if one has qualities the output will be in fastq format with generated qualities where missing, fasta otherwise. If given ids has no sequence data associated with them "no available sequences" will be the output. Due to the nature of our compression mechanism, some characters in the output format are not the same as in the original file. IUPAC characters are randomly substituted by one of their valid nucleotide code, for an example 'K' is going to be replaced by either 'G' or 'T'", **'.'** dot character is replaced by N.

Ids - csv list of objects ids.
fformat - *fasta* print in fasta format; *fastq* print in fastq format (if object doesn't have qualities, they will be faked, value of 30 ('?'); *fastx* - print in fasta or fastq format depending on presence of qualities in original data.
rangeStart - print sequence starting at this position (default: 0), if this value exceeds the sequence length, it will not be printed
rangeLen - print number of bases from each row (default: 0 prints the full length)
start - print starting from the 'start' row (default: 0)
cnt - number of rows to print (default: 20), if value is negative, all the rows are printed

upload - uploading files

URL: /dmUploader.cgi
Response: Line "Success ###,###!" second number is the objid id which needs to be tracked for successful completion of upload processing on backend. Always use POST with content-type: multipart/form-data.
Description:

When an upload is analyzed system relies on file extensions to parse recognized formats, like .zip, .fa, .fasta, etc. For a file recognized as sequence data system creates object(s) of 'nuc-read' type.
chkauto_Downloader – "-1" – process file (unpack, index, etc), 0 or missing – store as is.
Idx_Downloader – "1" – index file if applicable (for reads creates internal index), 0 or missing – no indexing.
QC_Downloader – "1" – QC files with applicable algorithms based on file type, 0 or missing – no QC.
screen_Downloader – "1" – screen file with applicable algorithms based on file type, 0 or missing – no screening.
subj_Downloader – CSV list of genome object ids to use as a reference when uploading files of types like, alignments
prop.svc-download._type=svc-download
name="Downloader" – input(s) of type "file".

Header "filename" is mandatory, represented in multipart content as:
------WebKitFormBoundarytRL4e1NkHRsw0qa3
Content-Disposition: form-data; name="Downloader"; filename="local-filename-on-user-disk.txt"
HTML page example:

```
<form action="dmUploader.cgi" method="POST" enctype="multipart/form-data">
<input type="file" name="Downloader" id="Downloader1" multiple />
<input type="file" name="Downloader" id="Downloader2" multiple />
<!—more inputs -->
<label for="chkauto_Downloader"><input checked="" type="checkbox" name="chkauto_Downloader" value="-1"
>automatically process file(s)</label>
<input type="submit" value="Start Upload">
</form>
```

Sample request content:

```
SCRIPT_NAME: /dmUploader.cgi
REQUEST_METHOD: POST
QUERY_STRING:
HTTP_USER_AGENT: Mozilla/5.0 …
HTTP_ACCEPT_LANGUAGE: en-US,en;q=0.8
CONTENT_TYPE: multipart/form-data; boundary=----WebKitFormBoundaryY8elqNEmkMwaonyP
CONTENT_LENGTH: 155
HTTP_COOKIE: …
SERVER_PROTOCOL: HTTP/1.1
REQUEST_URI: /dmUploader.cgi
SERVER_ADDR: xx.xxx.x.xx
HTTP_HOST: xx.xxx.x.xx


------WebKitFormBoundaryY8elqNEmkMwaonyP
Content-Disposition: form-data; name="Downloader"; filename="o3023291-qp-Progress.txt"
Content-Type: text/plain
1,100,done,done
------WebKitFormBoundaryY8elqNEmkMwaonyP
Content-Disposition: form-data; name="Downloader"; filename="align.msf"
Content-Type: application/octet-stream
!!AA_MULTIPLE_ALIGNMENT 1.0
stdout MSF: 131 Type: P 16/01/02 CompCheck: 3003 ..
Name: IXI_234 Len: 131 Check: 6808 Weight: 1.00
------WebKitFormBoundaryY8elqNEmkMwaonyP
Content-Disposition: form-data; name="chkauto_Downloader"
-1
------WebKitFormBoundaryY8elqNEmkMwaonyP-
```

Converting reads to genome

URL: ?api=0&cmdr=scast&ids=3099854,3099852&type=genome
Response:

```
{
"3099854" : { "signal" : "cast", "data" : { "from" : "nuc-read", "to" : "genome" } },
"3099852" : { "signal" : "cast", "data" : { "error" : "cast failed" } }
}
```

Description: Change objects type. For *nuc-read* to change to *genome* initial upload/index must be successfully completed.

-qpProcSubmit - Submit a job

URL: ?api=0&cmdr=-qpProcSubmit&…
Response: "####,####". Second number is the objectID to monitor. In case of failure "error: …."
Description: Submit a backend job

svc – name of service: dna-hexagon
_type – service specific type name: svc-align-hexagon
isPostponed – Boolean (true, false) indicating if process needs to be put on hold vs being executed immediately
name – a free text name of the object
comment – free text filed for additional users notes

Example: Each kind of job has its own specific list of parameters. Below is description of specific types of requests with full parameter description:

Submitting an alignment job

Example:

?api=0**&cmdr=**-qpProcSubmit**&svc=**dna-hexagon**&prop.svc-align-hexagon._type=**svc-align-hexagon**&prop.svc-align-hexagon.acceptNNNQuaTrheshold=**0**&prop.svc-align-hexagon.alignmentEngine=**1**&prop.svc-align-hexagon.alignSelector=**svc-align-hexagon**&prop.svc-align-hexagon.allowShorterEnds=**0**&prop.svc-align-hexagon.batch_svc=**single**&prop.svc-align-hexagon.complexityEntropy=**0**&prop.svc-align-hexagon.complexityRefEntropy=**0**&prop.svc-align-hexagon.complexityRefWindow=**0**&prop.svc-align-hexagon.complexityWindow=**0**&prop.svc-align-hexagon.computeDiagonalWidth=**auto**&prop.svc-align-hexagon.considerGoodSubalignments=**1**&prop.svc-align-hexagon.costGapNext=**-4**&prop.svc-align-hexagon.costGapOpen=**-12**&prop.svc-align-hexagon.costMatch=**5**&prop.svc-align-hexagon.costMismatch=**-4**&prop.svc-align-hexagon.costMismatchNext=**-6**&prop.svc-align-hexagon.doubleHash=**0**&prop.svc-align-hexagon.doubleStagePerfect=**1**&prop.svc-align-hexagon.force_reindex=**0**&prop.svc-align-hexagon.hashCompileStp=**auto**&prop.svc-align-hexagon.hashStp=**auto**&prop.svc-align-hexagon.isbackward=**1**&prop.svc-align-hexagon.isextendtails=**0**&prop.svc-align-hexagon.isglobal=**0**&prop.svc-align-hexagon.isoptimize=**1**&prop.svc-align-hexagon.keepAllMatches=**4**&prop.svc-align-hexagon.looseExtenderMinimumLengthPercent=**66**&prop.svc-align-hexagon.looseExtenderMismatchesPercent=**25**&prop.svc-align-hexagon.maxExtensionGaps=**0**&prop.svc-align-hexagon.maxHashBin=**50**&prop.svc-align-hexagon.maxHitsPerRead=**200**&prop.svc-align-hexagon.maximumPercentLowQualityAllowed=**0**&prop.svc-align-hexagon.maxMissQueryPercent=**15**&prop.svc-align-hexagon.maxNumberQuery=**al**&prop.svc-align-hexagon.minMatchLen=**75**&prop.svc-align-hexagon.name=**o35196-alFasta-24.fa%20versus%20drosophila genome**&prop.svc-align-hexagon.query.1=**3099286**&prop.svc-align-hexagon.query.2=**3099287**&prop.svc-align-hexagon.reverseEngine=**0**&prop.svc-align-hexagon.scissors=**hiveseq**&prop.svc-align-hexagon.scoreFilter=**None**&prop.svc-align-hexagon.searchRepeatsAndTrans=**0**&prop.svc-align-hexagon.seed=**11**&prop.svc-align-hexagon.selfQueryPosJumpInNonPerfectAlignment=**1**&prop.svc-align-hexagon.isPostponed=**false**&prop.svc-align-hexagon.slice=**500000**&prop.svc-align-hexagon.split=**query**&prop.svc-align-hexagon.subject.1=**3098421**&prop.svc-align-hexagon.subject.2=**32455**&prop.svc-align-hexagon.submitter=**dna-hexagon**&cmdMode=-**&prop.svc-align-hexagon.svc=**dna-hexagon**&prop.svc-align-hexagon.svcTitle=**HIVE-hexagon%20Alignment**&prop.svc-align-hexagon.useRedundSim=**1

Multi-value parameters *query* and *subject* might be repeated as necessary increasing the number before equal sign *query.2=123, query.3=456*, etc. Numbers must be different for each new value but not necessarily sequential.

Parameters:

https://hive.biochemistry.gwu.edu/help/tutorials_current/dna_hexagon_tutorial_current.pdf. This document describes parameters based on its user readable title, below is the table to translate titles into names for use with in URL

Name	Title	Value (for empty see PDF)
_type	Process type	svc-align-hexagon
isPostponed	Postpone execution	False
submitter	Submitting Template Page	dna-hexagon
svc	Service ID	dna-hexagon
svcTitle	Service Title	HIVE-hexagon Alignment
query	Sequence Read	Read object id(s)
subject	Reference Genome	Genome object id(s)
scissors	How to split the sequences	hiveseq
slice	Number of Computational Subjects per Single Thread	500000
split	What field contains the object spliting the jobs	Query
batch_svc	System parameter	single
alignSelector	Alignment Algorithm	svc-align-hexagon
maxNumberQuery	Alignment Test-Run Slice	all
comment	Comments	Any text
name	Name	Any text
allowShorterEnds	Shorter Terminus Alignment	
basicAlgParam	Alignment Filters	
computeDiagonalWidth	Width of Intelligent Diagonal	
costGapNext	Gap Continuation Cost	
costGapOpen	Gap Opening Cost	
costMatch	Match Benefit	
costMismatch	Mismatch Penalty	
costMismatchNext	Mismatch Continuation Penalty	
doubleHash	Use Double Length Hash Optimization	

doubleStagePerfect	Use Exact Match Preference Optimization	
hashCompileStp	Intelligent K-mer Jump in Reference	
hashStp	Intelligent K-mer Jump in Read	
isbackward	Alignment Directionality	
isextendtails	Alignment Tail Extension	
isglobal	Alignment Scope	
isoptimize	Intelligent Diagonal	
looseExtenderMinimumLengthPercent	K-mer Extension Minimal Length Percent	
looseExtenderMismatchesPercent	K-mer Extension Mismatch Allowance Percent	
maxExtensionGaps	Allowed number of gaps during extension	
maxHashBin	Over-represented K-mer suppression	
referenceAnnot	Reference Annotation Filter	
referenceAnnotTypes	Reference Annotation Types	
searchRepeatsAndTrans	Repeat and Transposition Discovery	
seed	Seed K-mer	
selfQueryPosJumpInNonPerfectAlignment	Self-overlapping seed matches	
useRedundSim	Use Read Self Similarity	
acceptNNNQuaTrheshold	Filter NN and low quality	
complexity	Low complexity filter	
complexityEntropy	Minimal Shannons Read Entropy	
complexityQry	Short read filtration	
complexityRef	Reference masking	
complexityRefEntropy	Minimal Shannons Reference Entropy	
complexityRefWindow	Window Size Reference	
complexityWindow	Window Size Read	
considerGoodSubalignments	Computed on	
fragmentLength	Length of the fragment	
fragmentLengthMax	Length of fragment Max	
fragmentLengthMin	Length of fragment Min	
fragmentScore	Calculate score on pair	
keepAllMatches	Matches to Keep	
keepMarkovnikovMatches	Markovnikov rule on filtered matches	
keepPairedOnly	Filter unpaired alignments	
keepPairOnOppositeStrand	Filter pairs with wrong stradness	
keepPairOnSameSubject	Filter pairs on different subjects	
maxHitsPerRead	Maximum Number of Hits per Read to Consider	
maximumPercentLowQualityAllowed	Maximum Percent Low Quality Allowed	
maxMissQueryPercent	Percent Mismatches Allowed	
minMatchLen	Minimum Match Length	
force_reindex	Force Reindexing of Subject	

Results: URL to retrieve results in **SAM** format:

?api=0&cmdr=alSam&objs=<process-objectID>&mySubID=-1&found=1&qty=-1&printRefID=1&down=1&useOriginalID=1

Paired end mode

Paired end mode is inferred when any of the following parameters is non empty/zero.

Parameter
fragmentLengthMax
fragmentLengthMin
fragmentScore
keepMarkovnikovMatches
keepPairedOnly
keepPairOnOppositeStrand
keepPairOnSameSubject

Example:

?api=0&cmdr=-qpProcSubmit&svc=dna-hexagon&prop.svc-align-hexagon._type=svc-align-hexagon&prop.svc-align-hexagon.query.1=3099286&prop.svc-align-hexagon.query.2=3099287&prop.svc-align-hexagon.query.3=3099288&prop.svc-align-hexagon.query.4=3099289&prop.svc-align-hexagon.query.5=3099290&prop.svc-align-hexagon.query.6=3099291...

In this example query objects will be paired :

pair 1	pair 2
prop.svc-align-hexagon.query.1=3099286	**prop.svc-align-hexagon.query.4=**3099289
prop.svc-align-hexagon.query.2=3099287	**prop.svc-align-hexagon.query.5=**3099290
prop.svc-align-hexagon.query.3=3099288	**prop.svc-align-hexagon.query.6=**3099291

Submitting a profiling job

Example:

?api=0&cmdr=-qpProcSubmit&svc=dna-heptagon&prop.svc-profiler-heptagon.name=Profile%20based%20on%20alignment%20:%20o35196-alFasta-24.fa%20versus%20drosophila%20genome%20(3099598)&prop.svc-profiler-heptagon.submitter=dna-heptagon&prop.svc-profiler-heptagon.parent_proc_ids.1=3099598&prop.svc-profiler-heptagon.useQuaFilter=20&prop.svc-profiler-heptagon.countAAs=0&prop.svc-profiler-heptagon.maxRptIns=3&prop.svc-profiler-heptagon.entrCutoff=0&prop.svc-profiler-heptagon.disbalanceFR=0&prop.svc-profiler-heptagon.scissors=hiveal&prop.svc-profiler-heptagon.maxMissmatchPercCutoff=0&prop.svc-profiler-heptagon.maxLowQua=no%20cutoff&prop.svc-profiler-heptagon.lenPercCutoff=no%20cutoff&prop.svc-profiler-heptagon.minCover=10&prop.svc-profiler-heptagon.noiseFilterParams=none&prop.svc-profiler-heptagon.noiseProfileMax=0.01&prop.svc-profiler-heptagon.noiseProfileResolution=0.0001&prop.svc-profiler-heptagon.slice=500000&prop.svc-profiler-heptagon._type=svc-profiler-heptagon&prop.svc-profiler-heptagon.SEARCHFORREPEATS=0&prop.svc-profiler-heptagon.snpCompare=1&prop.svc-profiler-heptagon.minFreqIgnoreSNP=0&prop.svc-profiler-heptagon.minFreqPercent=0&prop.svc-profiler-heptagon.minImportantEntropy=0.0&prop.svc-profiler-heptagon.isPostponed=false prop.svc-profiler-heptagon.histograms=0&prop.svc-profiler-heptagon.filterZeros=1&prop.svc-profiler-heptagon.cutEnds=do%20not%20cut&prop.svc-profiler-heptagon.split=parent_proc_ids

Parameter **parent_proc_ids** refers to original alignment process object id

Parameters:

https://hive.biochemistry.gwu.edu/help/tutorials_current/snp_profile_tutorial_april2015.pdf. This document describes parameters based on its user readable title, below is the table to translate titles into names for use with in URL

Name	Title	Value (for empty see PDF)
_type	Process type	svc-profiler-heptagon
scissors	How to split the sequences	Hiveal

slice	Number of Computational Subjects per Single Thread	500000
split	What field contains the object spliting the jobs	parent_proc_ids
isPostponed	Postpone execution	false
submitter	Submitting Template Page	dna-heptagon
svc	Service ID	dna-heptagon
svcTitle	Service Title	HIVE-heptagon Profiler
parent_proc_ids	Alignment(s)	alignment process object id
comment	Comments	Any text
name	Name	Any text
countAAs	Compute AA profile	ADD TO TUTORIAL
cutEnds	Truncate Terminals	
disbalanceFR	Forward/Reverse Disbalance	
entrCutoff	Entropic Cutoff	
filterZeros	Treat filtered positions	
histograms	Tail Histograms	ADD TO TUTORIAL
lenPercCutoff	Minimal Alignment Length	
maxLowQua	Maximum Percentage of Lo-Quality Regions	
maxMissmatchPercCutoff	Maximum Missmatch Percent	ADD TO TUTORIAL
maxRptIns	Count of Inserts Invalidating Repeats	
minCover	Minimal Coverage Allowed	
minFreqIgnoreSNP	Action on SNP Threshold	
minFreqPercent	Minimal Frequency of SNP Threshold	
minImportantEntropy	Safe Entropy Zone	
noise	Noise	
noiseFilterBase	Noise Profiling Process ID	
noiseFilterParams	Noise Filtering Parameters	
noiseProfileMax	Noise Profile Maximum Cutoff	
noiseProfileResolution	Noise Profile Resolution	
referenceAnnot	Annotation File with CDS Field	
SEARCHFORREPEATS	Profile only repeated regions	
snpCompare	Profile type	
thrSNP	SNP Threshold	
useQuaFilter	Base Quality Filter	

Results: URL to retrieve results in **VCF** format:

?api=0&cmdr=profVCF**&objs=**<process-objectID>**&idSub=**-1**&down=**1

-qpRawCheck – Check status of a job

URL: ?api=0&cmdr=-qpRawCheck&showreqs=0&reqObjID=<objectID>
Description: Check a backend job status

reqObjID – objectID of the job
OR
req – requestID of the job, some jobs are objectless, use only one of the ids to check status
showreqs – 0, top level results only, no job status details

Example:

?api=0&cmdr=-qpRawCheck&showreqs=0&reqObjID=3099526

Response: CSV file:

name,parent,cnt,reqID,grpID,svcID,stat,progress,progress100,actTime,takenTime,doneTime,waitTime,execTime,reportType,orderExecute,
runningBefore,act
"Total Progress","root",88,0,58167672,0,4,554,3,1448461439,1448461437,1448461497,139324,60,,,,4
"HIVE-optimized External Alignment","Total Progress",88,0,58167672,382922,4,554,3,1448461439,1448461437,1448461497,139324,60,,,,4
First data line "Total progress" is the only on needed to determine the status of the job. **stat** column equal to **5 is successful completion** indicator and above 5 is an error.

Response columns:

name= may be the request itself, the number of the request ID, or the service
parent= the parent of the total process
cnt=count (ie aligner= the number of hits you found in a certain alignment)
reqID= Request ID
grpID=Group ID
svcID=service ID (ID of the service used for the submission ie aligner service ID)
stat= status (represented by numbers 1-7)

1-waiting
2-processing
3-running
4-suspended
5-done
6-killed
7-program error
8-system error

progress=number processed (ie # reads processed out of total # reads)
progress100= percentage processed
actTime=the first "ping" from the service
takenTime= time request was grabbed (Unix time)
doneTime= time request was complete (Unix time)
waitTime= seconds waited
execTime=seconds executed
reportType=displays 1st alert/error
orderExecute= order in the que that the job was executed
runningBefore= service load in the system
act=last action on the request

-qpReqInfo – Retrieve messages of a job

URL: ?api=0&cmdr=-qpReqInfo&reqObjID=<objectID>
Description: Get information, warnings, errors for a specific job

reqObjID – objectID of the job
OR
req – requestID of the job, some jobs are objectless, use only one of the ids

Example: ?api=0&cmdr=-qpReqInfo&reqObjID=3099526
Response: CSV file:

name,parent,reqID,svcName,date,infoTxt,infoLvl,infoLvlNum
1,57680466,57680466,HIVE-optimized External Alignment,Thu Nov 19 11:27:50 2015,Indexing subject(s),Info,300
2,57680466,57680466,HIVE-optimized External Alignment,Thu Nov 19 11:28:01 2015,Aligner produced empty result,Warning,400
57680466,HIVE-optimized External Alignment,57680466,HIVE-optimized External Alignment,,Aligner produced empty result,Warning,400
"HIVE-optimized External Alignment","Grouped Info",0,,,,Warning,400
"Grouped Info","root",0,,,,Warning,400
Columns are InfoTxt, infoLvl is the main source of information which could be passed on to user.

Response columns:

Name=name of request
Parent= the parent of the total process
reqID= Request ID
svcName= Name of Service
date= Date generated
infoTxt= the actual message
infoLvl= information level: info, warning, error…
infoNumber= the # associated with the info level

-qpProcURL – request URL to process page (under development)

URL: ?api=0&cmdr=-qpProcURL&id=<objectID>

Response: Plain text URL, empty if object is not found or not accessible
Description:

Id – objectID of the object of type process+, not all processes have a page. Download processes do not have a page.

ionGenBankAnnotPosMap – Retrieve annotation information

URL: ?api=0&cmdr=ionGenBankAnnotPosMap&ids=<csv object id list>&seqID=NC_xxx&pos_start=..&pos_end=..cnt=..
Description: URL to retrieve annotation information from objects.

ids - csv list of objects ids
seqID - specifies the sequence identifier of interest (single value)
pos_start - starting position (default: 1). if value is negative, all ranges are retrieved.
pos_end - ending position (default: starting position + 10000). if pos_start is negative pos_end is ignored
start - print starting from the 'start' row (default: 0)
cnt - number of rows to print (default: 20), if value is negative, all the rows are printed
features - list of types of interest, if the field is not specified, all the types are requested

Example: api=0&cmdr=ionGenBankAnnotPosMap&ids=311235&seqID=NC_002929&pos_start=-1&cnt=-1
Response: CSV table:

```
seqID,start,end,idType-id
NC_002929,820,864,FEATURES:misc_feature;strand:+;gene:gidA
NC_002929,1917,2500,FEATURES:CDS|gene;strand:+;gene:gidB
NC_002929,1810,1875,FEATURES:misc_feature;strand:+;gene:gidA
NC_002929,1,1920,FEATURES:CDS|gene;strand:+;gene:gidA
NC_002929,19,1893,FEATURES:misc_feature;strand:+;gene:gidA
```