```
=================================================================================
biocompute object example for "human_gut_metagenomic_detection"
=================================================================================
{
    "_id": "obj.20001",
    "_type":"human_gut_metagenomic_detection",
    "name": "Healthy Human[taxID:9606] Fecal[UBERON:0001988] Metagenomic
           Diversity[PCO:0000019]",
    "version": "1.0",
    "verification:": "manual",
    "creation_date": "Mon Jul 25 13:25:21 EDT 2016",
    "modification_date": "Mon Jul 25 13:25:21 EDT 2016",
    "publication_status": "draft",
    "keywords": [
        "metagenome",
        "metagenomic analysis"
    ],
    "xref": [
        "UBERON:0001988",
        "taxID:9606",
        "PCO:0000019"
        ],
    "usability_domain": [
        "Identify the most common organism present in a human[taxID:9606]
           fecal[UBERON:0001988] sample",
        "Identify the general community composition of organisms in a
           human[taxID:9606] fecal[UBERON:0001988] sample"
    ],
    "authors": [
        {
            "first_name": "Joshua",
            "last_name": "Lederberg",
            "affiliation": "University of Wisconsin"
        },
        {
            "first_name": "Margaret",
            "last_name": "Dayhoff",
            "affiliation": "Columbia University"
        }
    ],
    "pipeline": "obj.20001",
    "DNA_screening": {
        "taxDepth": "leaf",
        "storeAlignments": 1,
        "CensuslimitIterations": 5,
        "Sample": 2500,
        "selfStopping": 0
    },
    "alignment": {
        "minMatchLen": 45,
        "keepAllMatches": 3,
        "keepMarkovnikovMatches": 1,
        "complexityRefEntropy": 1.2,
        "complexityRefWindow": 30,
        "acceptNNNQuaThreshold": 1,
        "complexityEntropy": 1.2,
        "complexityWindow": 30,
        "maximumPercentLowQualityAllowed": 0
    },
    "verification_kit": {
        "reference_sequence": [
            {
                "source": "NCBI",
                "db": "NucCore",
                "id": [
                    "CP000139.1",
                    "FP929042.1",
                    "FP929046.1",
                    "FP929045.1",
                    "FP929043.1",
                    "FP929048.1"
                ]
            }
        ],
        "sequence_reads": [
            {
                "source": "HIVE",
                "db": "Honeycomb",
                "id": [
                    "514683",
                    "514682"
                ]
            }
        ],
        "reference_db": [
            {
                "source": "HIVE",
                "db": "Honeycomb",
                "id": ["513957"]
            }
        ],
        "accession_based_results": {
            "format": [".csv"],
            "filename": "dnaAccessionBasedResult.csv"
        },
        "unalignedReads": {
            "format": [".fasta"],
            "filename": "Unaligned Reads (HIVE_ID).fasta"
        },
        "hit_list": {
            "format": [".csv"],
            "filename": "alCount-Unalignedo541205-alCount--1.csv"
        }
    }
```

```
}

=================================================================================
pipeline definition example for "Human Gut Metagenomic Detection Pipeline"
=================================================================================
{
    "_id": "obj.20001",
    "_type": "pipeline_definition",
    "name": "Human Gut Metagenomic Detection Pipeline",
    "description": [
        "HIVE-Metagenomic Detection Pipeline uses two HIVE tools: CensuScope for
           detection and Hexagon for alignment. Only the options that need to be
           selected or changed are listed.",
        "Iteration 1: Input is raw paired end read file. Files are unthourough
           CensuScope and then the accessions that are detected above our
           threshold (10 hits) are used for the references and then the same read
           files are used as the query files. The Hexagon Hit list and unaligned
           read files are outputs.",
        "Iteration 2: The unaligned read file from the previous iteration is used
           as the input for CensuScope and Hexagon, with the resulting accessions
           from CensuScope serving as references for the alignemnt. A new
           unaligned read file and a new 'HitList' file is created from this
           alignemnt.",
        "Iteration 3: The unaligned read file from the previous iteration is used
           as the input for CensuScope and Hexagon, with the resulting accessions
           from CensuScope serving as references for the alignemnt. All three
           'HitList' file are combined into one sampel result."],
    "workflow": {
        "launcher": "system",
        "script": "o123123"
    },
    "tool_used": [
        {
            "name": "CensuScope",
            "version": "1.1",
            "arguments": [
                {
                    "parameter":  "taxDepth",
                    "value": "species"
                },
                {
                    "parameter": "storeAlignments",
                    "value": 0
                },
                {
                    "parameter": "CensuslimitIterations",
                    "value": 10
                },
                {
                    "parameter": "Sample",
                    "value": 1000
                },
                {
                    "parameter": "selfStopping",
                    "value": 1
                }
            ]
        },
        {
            "name": "HIVE-hexagon",
            "version": "1.3",
            "arguments": [
                {
                    "parameter": "minMatchLen",
                    "value": 45
                },
                {
                    "parameter": "keepAllMatches",
                    "value": 3
                },
                {
                    "parameter": "keepMarkovnikovMatches",
                    "value": 1
                },
                {
                    "parameter": "complexityRefEntropy",
                    "value": 1.2
                },
                {
                    "parameter": "complexityRefWindow",
                    "value": 30
                },
                {
                    "parameter": "acceptNNNQuaTrheshold",
                    "value": 1
                },
                {
                    "parameter": "complexityEntropy",
                    "value": 1.2
                },
                {
                    "parameter": "complexityWindow",
                    "value": 30
                },
                {
                    "parameter": "maximumPercentLowQualityAllowed",
                    "value": 0
                }
            ]
        }
    ]
}
```

```
=================================================================================
typedef for "human_gut_metagenomic_detection"
=================================================================================
{
    "_type": "type",
    "name": "human_gut_metagenomic_detection",
    "description": "This pipeline uses an a screener and then an aligner to
           detect organisms present in a metagenomic sample",
    "_fields": {
        "reference_sequence": {
            "description": "Reference sequences can be either FASTA from known
                sources or identifiers from HIVE database. These must be saved as
                a 'genome' type in HIVE to use in this field",
            "title": "Reference Sequences",
            "_type": "xref",
            "_role": "input",
            "_vital": "true",
            "_plural": "true"
            },
        "sequence_reads": {
            "description": "Sequences can be either FASTA from experimental
                sources or identifiers from HIVE database. These must be saved as
                a 'nuc_read' type in HIVE to use in this field",
            "title": "Nucleotide Read",
            "_type": "xref",
            "_role": "input",
            "_vital": "true",
            "_plural": "true"
            },
        "reference_db": {
            "description": "The db is a collection of sequences that have been
                flagged as a genome and are indexed.",
            "title": "Database",
            "_type": "xref",
            "_role": "input",
            "_vital": "true",
            "_plural": "true"
            },
        "dna-screening":{
            "description": "Arguments for the DNA_screening, HIVE CensuScope",
            "title": "CensuScope",
            "taxDepth": {
                "_type": "string",
                "_default": "species"
            },
            "storeAlignments": {
                "_type": "integer",
                "_default": 0
            },
            "CensuslimitIterations": {
                "_type": "integer",
                "_default": 10
            },
            "Sample": {
                "_type": "integer",
                "_default": 1000
            },
            "selfStopping": {
                "_type": "integer",
                "_default": 1
            }
        },
        "alignment": {
            "description": "Arguments for the alignment, HIVE Hexagon",
            "title": "HIVE-hexagon Alignment",
            "minimum_match_len,": {
                "_type": "integer",
                "_default": 75
            },
            "keepAllMatches": {
                "_type": "integer",
                "_default": 1
            },
            "keepMarkovnikovMatches": {
                "_type": "integer",
                "_default": 0
            },
            "complexityRefEntropy": {
                "_type": "integer",
                "_default": 1.0
            },
            "complexityRefWindow": {
                "_type": "integer",
                "_default": 0
            },
            "acceptNNNQuaTrheshold": {
                "_type": "integer",
                "_default": 0
            },
            "complexityEntropy": {
                "_type": "integer",
                "_default": 1.0
            },
            "complexityWindow": {
                "_type": "integer",
                "_default": 0
            },
            "maximumPercentLowQualityAllowed": {
                "_type": "integer",
                "_default": 30
            }
        },
        "accession_based_results": {
            "description": "A list of the number of reads matched to a refrence
                sequence in a database object",
            "title": "Accession Based Results",
            "_role": "output",
            "_type": "outfile"
        },
        "unalignedReads": {
            "description": "A sequence object of the unaligned reads from an
                alignment",
            "title": "Unaligned Reads",
            "_role": "output",
            "_type": "outfile"
        },
        "hit_list": {
            "description": "A list of all the reads matched to each reference
                sequence",
            "title": "Hit List",
            "_role": "output",
            "_type": "outfile"
        }
    },
    "_inherits": ["base_biocompute_type"]
}

=================================================================================
typedef for “pipeline_definition_type"
=================================================================================
{
    "_id": "obj.10001",
    "_type": "pipeline_definition",
    "name": "HIVE-Viral Mutation Detection Pipeline",
    "description": "HIVE-Viral Mutation Detection uses two HIVE tools: Hexagon
           for alignment and Heptagon for variant calling. Only the options that
           need to be selected or changed are listed.",
    "workflow": {
        "launcher": "system",
        "script": "obj.123123"
    },
    "tool_used": [
        {
            "name": "HIVE-hexagon",
            "version": "1.3",
            "arguments": [
                {
                    "parameter": "seed",
                    "value": 16
                },
                {
                    "parameter": "divergence_threshold_percent",
                    "value":  15
                },
                {
                    "parameter": "minimum_match_len",
                    "value":  100
                }
            ]
        },
        {
            "name": "HIVE-heptagon",
            "version": "1.1",
            "arguments": [
                {
                    "parameter": "min_coverage",
                    "value": 10
                },
                {
                    "parameter": "freq_cutoff",
                    "value": 0.10
                }
            ]
        }
    ]
}

=================================================================================
typedef for "base_biocompute_type"
=================================================================================
{
    "_type": "type",
    "name": "base_biocompute_type",
    "description": "Provides the base fields to populate in a biocompute object",
    "_fields": {
        "validation": {
            "description": "Describes the position of an object in the validation
                process. The unreviewed flag indicates that the object has been
                submitted, but no further curation has occurred. The in_progress
                flag indicates that validation is underway. manual indicates that
                the object has been manually validated. The suspended flag
                indicates an object that was once valid is no longer considered
                valid. The error flag indicates that an error was detected with
                the biocompute object.",
            "_type": "string",
            "_limit": {
                "choice": [
                    "unreviewed",
                    "in_progress",
                    "manual",
                    "suspended",
                    "error"
                ]
            }
        },
        "usability_domain": {
            "description": "Provides a space for the author to define the
                usability
                domane of an object.",
            "_type": "string",
            "_vital": "true",
            "_plural": "true"
        },
        "pipeline" : {
            "description": "defines the fields that have to be populated for a
                tool description in the pipeline of a biocompute object",
            "_type": "pipeline_definition"
        }
    },
    "_inherits": ["base_metadata_type"]
}

=================================================================================
typedef for "base_metadata_type"
=================================================================================
{
    "_type": "datatype",
    "name": "base_metadata_type",
    "description": "Definition of the base metadata type providing fields for
        the inclusion of outside links, submissions, and information pertinent
        to the experiment but not critical for the computation.",
    "_fields": {
        "name": {
            "_type": "string"
        },
        "creation_date": {
            "description": "The time and date that the object is
                created/computation was run",
            "_type": " datetime"
        },
        "modification_date": {
            "description": "This field will list all the dates and time an
                object ismodified.",
            "_type":" datetime",
            "_plural": "true"
        },
        "publication_status": {
            "description": "This is a choice field with four options. The draft
                flag indicates that an object is in draft form and is still
                being added to or changed. The open access flag indicates that
                an object has been published and is freely available to anyone.
                The objects with the private flag have restrictions on who can
                view and access them. This is a way for researchers using
                restricted data or metadata to insure the confidentially is
                maintained.",
            "_type": "string",
            "_limit": {
                "choice": ["draft", "open access",  "private", "recalled"]
            }
        },
        "authors": {
            "description": "This is the list of authors submitting this object.
                First and last name are required, middle name and affiliation are
                not. The affiliation field is for listing the academic,
                governmental, or private entity that supported the project
                responsible for generating the biocompute object.",
            "_plural": "true",
            "first_name": {
                "_type":"string",
                "_vital": "true"
            },
            "middle_name": {
                "_type": "string"
            },
            "last_name": {
                "_type": "string",
                "_vital": "true"
            },
            "affiliation": {
                "_type": "string"
            }
        },
        "keywords": {
            "description": "Keywords to aid in search-ability and description of
                the object. These are a controlled vocabulary.",
            "_type": "string",
            "_plural": "true"
        }
    }
}
```