BioCompute Object for Regulatory Review

BCO Title: RNA-seq Alignment - TopHat

BCO Generation Date: October 22, 2019

BCO Specification Version: v1.3.0

BCO Generator: Seven Bridges

Contents

1 BioCompute Object Domain Entries

1.1 Top Level Fields

["https://w3id.org/biocompute/1.3.0/",
"http://biocompute.sbgenomics.com/bco/5f28ac19-ea58-424b-a6e9-639fc3171302",
"935e8347674716fe3cdfb3f8b40e3c6b6a559debdeaf77cdad9b780f735d85ea"]

1.2 Provenance Domain

{
"name": "RNA-seq Alignment - TopHat",
"version": "1.0.0",
"review": [],
"derived_from":
"https://cgc-api.sbgenomics.com/v2/apps/Dennis/el-bco-zip/rna-seq-alignment-tophat/0/raw/",
"obsolete_after": "2019-10-21T00:00:00+0000",
"embargo": ["2019-10-21T00:00:00+0000",
"2019-10-21T00:00:00+0000"],
"created": "2019-10-21T00:00:00+0000",
"modified": "2019-10-21T00:00:00+0000",
"contributors": [
{
"name": "Elizabeth Christine Lee",
"affiliation": "George Washington University School of
Medicine & Health Sciences, Washington, D.C. 20037",
"email": "eclee314@gwu.edu",
"contribution": "createdBy",
"orcid": "https://orcid.org/0000-0003-0384-595X"
},
{
"name": "Ana Damljanovic",
"affiliation": "Seven Bridges Genomics",

"email": "ana.damljanovic@sbgenomics.com",
"contribution": "authoredBy",
"orcid": ""
},
{
"name": "Ana Damljanovic",
"affiliation": "Seven Bridges Genomics",
"email": "ana.damljanovic@sbgenomics.com",
"contribution": "derivedFrom",
"orcid": ""
}
],
"license": "https://spdx.org/licenses/CC-BY-4.0.html"
}

1.3 Usability Domain

"RNA-Seq technology represents a powerful method to interrogate gene expression. In addition to determining total gene expression levels, RNA-Seq allows quantitation of isoforms, identification of novel transcripts, and interrogation of RNA editing events. The first step in profiling the transcriptome is the alignment of RNA-Seq reads against the reference genome. This step reveals the location in the genome from which the reads originated.\n\nThis pipeline uses the popular split-read aligner, TopHat, to map reads to a reference genome, and it is set up to accommodate the most common experimental conditions (e.g. RNA-Seq experiments of samples from well annotated transcriptomes such as Human and Mouse). It utilizes a transcript annotation file (GTF) to speed read mapping across known splice junctions. This pipeline will

```
generate alignment files that can then be compared for
differential expression, analyzed to discover novel
transcripts, or viewed directly in a genome browser. TopHat
is highly versatile and by building pipelines, you are able
to exploit its many functions including the use of
experimentally identified junctions, insertions and
deletions.\n\nThis pipeline can be used in combination with
the \"RNA-Seq Differential Expression\" (available in
Public Apps) to take you all the way from raw sequencing
reads to a list of differentially expressed
genes.\n\nAlignment of RNA-Seq reads to a reference genome
is performed using the split read aligner TopHat. TopHat
incorporates the ultrafast short read aligner **Bowtie 2**.
While Bowtie 2 is able to align tens of millions of reads
per CPU hour, it does not allow alignments between the read
and genome to contain large gaps. This limitation precludes
the use of Bowtie 2 to align reads that span introns.
TopHat was built to overcome this restriction - any reads
that cannot be initially aligned to the genome are broken
up by TopHat into smaller pieces which, when processed
independently can be aligned by Bowtie 2. When read
segments are found to align to the genome far apart from
each other, TopHat infers that the read spans a splice
junction and estimates the location of the splice sites.
While TopHat can build up an index of splice sites in the
transcriptome without a priori gene or splice site
annotations, alignment speed and accuracy is increased by
providing this information during the mapping process.
\n\n###Inputs###\n\n**Reads**: This pipeline accepts both
single stranded or paired-end RNA-Seq data in FASTQ format.
If paired-end reads are used, the read pair metadata fields
must be set as 1 and 2. The metadata field Sample ID should
```

```
be unique for each biological sample.\n\n**Reference or
index files**: For proper TopHat performance (which relies
on Bowtie 2 for alignment) Bowtie 2 requires that the
reference genome is indexed before read alignment can be
performed. We have added **Bowtie 2 Indexer** to this
pipeline for reference file indexing. This indexing can be
time intensive, and in order to optimize for execution time
of this pipeline, you can index reference file separately
using Bowtie 2 Indexer. You may create a short pipeline
with this tool and reuse index archive if you intend to
perform several alignments with the same reference. By
default, you will be provided with the tar bundle
containing index files
(`human_g1k_v37_decoy.phiX174_bowtie2-2.2.6.tar` obtained
from `human_g1k_v37_decoy.phiX174.fasta`) as a suggested
file.\n\n**FASTA reference**: FASTA file containing
reference genome. Pipeline is provided with
`human_g1k_v37_decoy.phiX174.fasta` as a suggested
reference file.\n\n**GTF annotations**: Gene Transfer
Format file containing known gene annotations. Using a GTF
file will increase mapping speed and accuracy but it is not
required. It is critical that the chromosome numbering
schema used in the GTF file matches that used in the
Reference file (UCSC convention is to number chromosomes as
Chr _number_, whereas ensembl simply numbers chromosomes
just with _number_). Suggested file for this input is
`Homo_sapiens.GRCh37.75.gtf` annotation
file.\n\n###Q&A###\n\n***Q: What should I do if I already
have Bowtie2 index files, not archived as tar
bundle?***\n\n***A***: You can provide your *.bt2 files to
**SBG Compressor** app from our public apps and set \"TAR\"
as your output format. After the task is finished, **you
```

should assign common prefix of the index files to the `Reference genome` metadata field** and your TAR is ready for use.\n\n***Example:***\nIndexed files: chr20.1.bt2, chr20.2.bt2, chr20.3.bt2, chr20.4.bt2, chr20.rev.1.bt2, chr20.rev.2.bt2\n\nMetadata - `Reference genome`: **chr20**\n\n###Common issues###\nOne of the most common issues when running TopHat is incompatibility between reference genome and annotations. Please, make sure that you are using compatible FASTA (from which you have created tar bundle with index files) and GTF files in order to run tasks successfully. If you suspect your task has failed due to this incompatibility, you can check the last line in `job.err.log` file which would look as following if your assumptions are correct: `Error: Couldn't build bowtie index with err = 1`.\n\n**Important note: In case of paired-end alignment it is crucial to set metadata `Paired-end` field to \"1\" or \"2\". Sequences specified as \"1\" must correspond file-for-file and read-for-read with those specified as \"2\". Reads may be a mix of different lengths. In case of unpaired reads, the same metadata field should be set to '-'. Only one type of alignment can be performed at once, so all specified reads should be either paired or unpaired.**"

1.4 Extension Domain

{
"fhir_extension": {
"fhir_endpoint": "",
"fhir_version": "",
"fhir_resources": {}
},

```
"scm_extension": {
"scm_repository":
"https://github.com/ECL314/CGC-BCO-Generator-App-Extra-Credit-Assignment",
"scm_type": "git",
"scm_commit": "",
"scm_path": "",
"scm_preview": ""
}
}
```


1.5 Description Domain


```
{
"keywords": [],
"xref": [],
"platform": "Seven Bridges Platform",
"pipeline_steps": [
{
"step_number": "1",
"name": "#FastQC",
"description": "FastQC reads a set of sequence files and
produces a quality control (QC) report from each one. These
reports consist of a number of different modules, each of
which will help identify a different type of potential
problem in your data.\n\nFastQC is a tool which takes a
FastQ file and runs a series of tests on it to generate a
comprehensive QC report.  This report will tell you if
there is anything unusual about your sequence.  Each test
is flagged as a pass, warning, or fail depending on how far
it departs from what you would expect from a normal large
dataset with no significant biases.  It is important to
stress that warnings or even failures do not necessarily
```

```
mean that there is a problem with your data, only that it
is unusual.  It is possible that the biological nature of
your sample means that you would expect this particular
bias in your results.",
"version": "0.11.4",
"prerequisite": [],
"input_list": [
{
"uri": "http://example.com/RNASeqdata.fastq",
"access_time": "2019-10-22"
}
],
"output_list": [
{
"uri": "http://example.com/QC_report.zip",
"access_time": "2019-10-22"
}
]
},
{
"step_number": "2",
"name": "#BamTools_Index",
"description": "BamTools Index creates an index file (BAI
or BTI) for a BAM file. If BAI file is present on the input
the tool will skip indexing step and output BAM with
provided BAI file.\n\n**Common issues:** Providing a BAI
file on input will result in a pass-through without
execution, even if a different index format is requested on
the output (BTI instead of BAI).",
"version": "2.4.0",
"prerequisite": [],
"input_list": [
```

```
{
"uri": "http://example.com/input_file.bam",
"access_time": null
}
],
"output_list": [
{
"uri": "http://example.com/output_bam_file.bam",
"access_time": "2019-10-22"
},
{
"uri": "http://example.com/generated_index.bai",
"access_time": "2019-10-22"
}
]
},
{
"step_number": "3",
"name": "#SBG_FASTQ_Quality_Detector",
"description": "FASTQ Quality Scale Detector detects which
quality encoding scheme was used in your reads and
automatically enters the proper value in the \"Quality
Scale\" metadata field.",
"version": null,
"prerequisite": [
{
"name": "reference_genome_used_for_alignment.fasta",
"uri": {
"uri":
"http://example.com/human_g1k_v37_decoy.phiX174.fasta",
"access_time": "2019-10-22"
}
```

```
}
],
"input_list": [
{
"uri": "http://example.com/RNA-Seq_data.fastq",
"access_time": "2019-10-22"
}
],
"output_list": [
{
"uri": "http://example.com/RNASeq_data_updated
metadata.fastq",
"access_time": "2019-10-22"
},
{
"uri": "http://example.com/quality_scale.strings",
"access_time": "2019-10-22"
}
]
},
{
"step_number": "4",
"name": "#Picard_CollectAlignmentSummaryMetrics",
"description": "Picard CollectAlignmentSummaryMetrics
assesses the quality of alignment by analyzing a SAM or BAM
file. It compares it with the reference file (FASTA) and
provides alignment statistics, such as the number of input
reads and the percent of reads that are mapped. It produces
a file which contains summary alignment metrics from a SAM
or BAM file.\n\nNote: This tool requires the exact same
FASTA file as the one to which raw reads were aligned.",
"version": "1.140",
```

```
"prerequisite": [],
"input_list": [
{
"uri": "http://example.com/input_file.bam",
"access_time": "2019-10-22"
},
{
"uri":
"http://example.com/human_g1k_v37_decoy.phiX174.fasta",
"access_time": "2019-10-22"
}
],
"output_list": [
{
"uri": "http://example.com/summary_metrics.txt",
"access_time": "2019-10-22"
}
]
},
{
"step_number": "5",
"name": "#TopHat2",
"description": "## TopHat2 (version
2.1.1)\n\n[TopHat2](https://ccb.jhu.edu/software/tophat/manual.shtml)
is a program that aligns RNA-Seq reads to a genome in order
to identify exon-exon splice junctions. It is built on the
ultrafast short read mapping program
[Bowtie2](http://bowtie-bio.sourceforge.net/bowtie2/manual.shtml).\n\nTopHat2
can also align reads directly to a transcriptome. If
provided with annotation of known transcripts, TopHat2
constructs a virtual transcriptome and uses Bowtie2 to
align reads to this reference first. Reads that do not
```

```
align to the transcriptome are then mapped on the reference
genome. The reads that did align on the transcriptome will
be converted to genomic mappings (spliced as needed) and
merged with the novel mappings and junctions in the final
output.\n\nThis version of TopHat only accepts reads in
FASTQ format. It is optimized for reads that are at least
75bp long.\n\n**Common issues:**\nOne of the most common
issues when running TopHat is incompatibility between
reference genome and annotations. Please, make sure that
you are using compatible FASTA (from which you have created
tar bundle with index files) and GTF files in order to run
tasks successfully. If you suspect your task has failed due
to this incompatibility, you can check the last line in
`job.err.log` file which would look as following if your
assumptions are correct: `Error: Couldn't build bowtie
index with err = 1`.\n\n**Q&A:**\n\n***Q: What should I do
if I already have Bowtie2 index files, not archived as tar
bundle?***\n\n***A***: You can provide your *.bt2 files to
[SBG
Compressor](https://igor.sbgenomics.com/public/apps#admin/sbg-public-data/sbg-compressor-1-0/)
app from our public apps and set \"TAR\" as your output
format. After the task is finished, **you should assign
common prefix of the index files to the `Reference genome`
metadata field** and your TAR is ready for
use.\n\n***Example:***\nIndexed files: chr20.1.bt2,
chr20.2.bt2, chr20.3.bt2, chr20.4.bt2, chr20.rev.1.bt2,
chr20.rev.2.bt2\n\nMetadata - `Reference genome`:
**chr20**\n\n__Important note: In case of paired-end
alignment it is crucial to set metadata 'paired-end' field
to 1/2. Sequences specified as mate 1s must correspond
file-for-file and read-for-read with those specified for
mate 2s. Reads may be a mix of different lengths. In case
```

```
of unpaired reads, the same metadata field should be set to
'-'. Only one type of alignment can be performed at once,
so all specified reads should be either paired or
unpaired.__",
"version": "2.1.0",
"prerequisite": [
{
"name": "Bowtie2",
"uri": {
"uri":
"https://cgc.sbgenomics.com/public/apps#admin/sbg-public-data/bowtie2-aligner/",
"access_time": "2019-10-22"
}
}
],
"input_list": [
{
"uri":
"http://example.com/human_g1k_v37_decoy.phiX174_bowtie2-2.2.6.tar",
"access_time": "2019-10-22"
},
{
"uri": "http://example.com/RNASeq_reads.fastq",
"access_time": "2019-10-22"
},
{
"uri": "http://example.com/Homo_sapiens.GRCh37.75.gtf",
"access_time": "2019-10-22"
}
],
"output_list": [
{
```

```
"uri": "http://example.com/alignment_summary.txt",
"access_time": "2019-10-22"
},
{
"uri": "http://example.com/read_alignments.bam",
"access_time": "2019-10-22"
},
{
"uri": "http://example.com/tophat_deletions.bed",
"access_time": "2019-10-22"
},
{
"uri": "http://example.com/tophat_insertions.bed",
"access_time": "2019-10-22"
},
{
"uri": "http://example.com/tophat_junctions.bed",
"access_time": "2019-10-22"
},
{
"uri": "http://example.com/unmapped_reads.bam",
"access_time": "2019-10-22"
}
]
},
{
"step_number": "6",
"name": "#Bowtie2_Indexer",
"description": "Bowtie2 Indexer is a tool for indexing
reference genomes of any size used in an alignment. It was
built from `bowtie2-build` script and used for reference
genome indexing aimed at assisting Bowtie2 in fast and
```

```
memory-efficient alignment. It outputs an archive which
consists of 6 files with suffixes .1.bt2, .2.bt2, .3.bt2,
.4.bt2, .rev.1.bt2, and .rev.2.bt2. This archive
constitutes the index and should be provided when aligning
the reads (either with [Bowtie2
Aligner](https://igor.sbgenomics.com/public/apps#tool/admin/sbg-public-data/bowtie2-aligner)
or
[TopHat2](https://igor.sbgenomics.com/public/apps#tool/admin/sbg-public-data/tophat2)).
\n\n###Common issues###\nNo issues have been reported.",
"version": "2.2.6",
"prerequisite": [],
"input_list": [
{
"uri":
"http://example.com/human_g1k_v37_decoy.phiX174_bowtie2-2.2.6.tar",
"access_time": "2019-10-22"
}
],
"output_list": [
{
"uri": "http://example.com/bowtie_index_archive.tar",
"access_time": "2019-10-22"
}
]
}
]
}
```


1.6 Execution Domain


```
{
"keywords": [],
```

```
"xref": [],
"platform": "Seven Bridges Platform",
"pipeline_steps": [
{
"step_number": "1",
"name": "#FastQC",
"description": "FastQC reads a set of sequence files and
produces a quality control (QC) report from each one. These
reports consist of a number of different modules, each of
which will help identify a different type of potential
problem in your data.\n\nFastQC is a tool which takes a
FastQ file and runs a series of tests on it to generate a
comprehensive QC report.  This report will tell you if
there is anything unusual about your sequence.  Each test
is flagged as a pass, warning, or fail depending on how far
it departs from what you would expect from a normal large
dataset with no significant biases.  It is important to
stress that warnings or even failures do not necessarily
mean that there is a problem with your data, only that it
is unusual.  It is possible that the biological nature of
your sample means that you would expect this particular
bias in your results.",
"version": "0.11.4",
"prerequisite": [],
"input_list": [
{
"uri": "http://example.com/RNASeqdata.fastq",
"access_time": "2019-10-22"
}
],
"output_list": [
{
```

```
"uri": "http://example.com/QC_report.zip",
"access_time": "2019-10-22"
}
]
},
{
"step_number": "2",
"name": "#BamTools_Index",
"description": "BamTools Index creates an index file (BAI
or BTI) for a BAM file. If BAI file is present on the input
the tool will skip indexing step and output BAM with
provided BAI file.\n\n**Common issues:** Providing a BAI
file on input will result in a pass-through without
execution, even if a different index format is requested on
the output (BTI instead of BAI).",
"version": "2.4.0",
"prerequisite": [],
"input_list": [
{
"uri": "http://example.com/input_file.bam",
"access_time": null
}
],
"output_list": [
{
"uri": "http://example.com/output_bam_file.bam",
"access_time": "2019-10-22"
},
{
"uri": "http://example.com/generated_index.bai",
"access_time": "2019-10-22"
}
```

```
]
},
{
"step_number": "3",
"name": "#SBG_FASTQ_Quality_Detector",
"description": "FASTQ Quality Scale Detector detects which
quality encoding scheme was used in your reads and
automatically enters the proper value in the \"Quality
Scale\" metadata field.",
"version": null,
"prerequisite": [
{
"name": "reference_genome_used_for_alignment.fasta",
"uri": {
"uri":
"http://example.com/human_g1k_v37_decoy.phiX174.fasta",
"access_time": "2019-10-22"
}
}
],
"input_list": [
{
"uri": "http://example.com/RNA-Seq_data.fastq",
"access_time": "2019-10-22"
}
],
"output_list": [
{
"uri": "http://example.com/RNASeq_data_updated
metadata.fastq",
"access_time": "2019-10-22"
},
```

```
{
"uri": "http://example.com/quality_scale.strings",
"access_time": "2019-10-22"
}
]
},
{
"step_number": "4",
"name": "#Picard_CollectAlignmentSummaryMetrics",
"description": "Picard CollectAlignmentSummaryMetrics
assesses the quality of alignment by analyzing a SAM or BAM
file. It compares it with the reference file (FASTA) and
provides alignment statistics, such as the number of input
reads and the percent of reads that are mapped. It produces
a file which contains summary alignment metrics from a SAM
or BAM file.\n\nNote: This tool requires the exact same
FASTA file as the one to which raw reads were aligned.",
"version": "1.140",
"prerequisite": [],
"input_list": [
{
"uri": "http://example.com/input_file.bam",
"access_time": "2019-10-22"
},
{
"uri":
"http://example.com/human_g1k_v37_decoy.phiX174.fasta",
"access_time": "2019-10-22"
}
],
"output_list": [
{
```

```
"uri": "http://example.com/summary_metrics.txt",
"access_time": "2019-10-22"
}
]
},
{
"step_number": "5",
"name": "#TopHat2",
"description": "## TopHat2 (version
2.1.1)\n\n[TopHat2](https://ccb.jhu.edu/software/tophat/manual.shtml)
is a program that aligns RNA-Seq reads to a genome in order
to identify exon-exon splice junctions. It is built on the
ultrafast short read mapping program
[Bowtie2](http://bowtie-bio.sourceforge.net/bowtie2/manual.shtml).\n\nTopHat2
can also align reads directly to a transcriptome. If
provided with annotation of known transcripts, TopHat2
constructs a virtual transcriptome and uses Bowtie2 to
align reads to this reference first. Reads that do not
align to the transcriptome are then mapped on the reference
genome. The reads that did align on the transcriptome will
be converted to genomic mappings (spliced as needed) and
merged with the novel mappings and junctions in the final
output.\n\nThis version of TopHat only accepts reads in
FASTQ format. It is optimized for reads that are at least
75bp long.\n\n**Common issues:**\nOne of the most common
issues when running TopHat is incompatibility between
reference genome and annotations. Please, make sure that
you are using compatible FASTA (from which you have created
tar bundle with index files) and GTF files in order to run
tasks successfully. If you suspect your task has failed due
to this incompatibility, you can check the last line in
`job.err.log` file which would look as following if your
```

```
assumptions are correct: `Error: Couldn't build bowtie
index with err = 1`.\n\n**Q&A:**\n\n***Q: What should I do
if I already have Bowtie2 index files, not archived as tar
bundle?***\n\n***A***: You can provide your *.bt2 files to
[SBG
Compressor](https://igor.sbgenomics.com/public/apps#admin/sbg-public-data/sbg-compressor-1-0/)
app from our public apps and set \"TAR\" as your output
format. After the task is finished, **you should assign
common prefix of the index files to the `Reference genome`
metadata field** and your TAR is ready for
use.\n\n***Example:***\nIndexed files: chr20.1.bt2,
chr20.2.bt2, chr20.3.bt2, chr20.4.bt2, chr20.rev.1.bt2,
chr20.rev.2.bt2\n\nMetadata - `Reference genome`:
**chr20**\n\n__Important note: In case of paired-end
alignment it is crucial to set metadata 'paired-end' field
to 1/2. Sequences specified as mate 1s must correspond
file-for-file and read-for-read with those specified for
mate 2s. Reads may be a mix of different lengths. In case
of unpaired reads, the same metadata field should be set to
'-'. Only one type of alignment can be performed at once,
so all specified reads should be either paired or
unpaired.__",
"version": "2.1.0",
"prerequisite": [
{
"name": "Bowtie2",
"uri": {
"uri":
"https://cgc.sbgenomics.com/public/apps#admin/sbg-public-data/bowtie2-aligner/",
"access_time": "2019-10-22"
}
}
```

```
],
"input_list": [
{
"uri":
"http://example.com/human_g1k_v37_decoy.phiX174_bowtie2-2.2.6.tar",
"access_time": "2019-10-22"
},
{
"uri": "http://example.com/RNASeq_reads.fastq",
"access_time": "2019-10-22"
},
{
"uri": "http://example.com/Homo_sapiens.GRCh37.75.gtf",
"access_time": "2019-10-22"
}
],
"output_list": [
{
"uri": "http://example.com/alignment_summary.txt",
"access_time": "2019-10-22"
},
{
"uri": "http://example.com/read_alignments.bam",
"access_time": "2019-10-22"
},
{
"uri": "http://example.com/tophat_deletions.bed",
"access_time": "2019-10-22"
},
{
"uri": "http://example.com/tophat_insertions.bed",
"access_time": "2019-10-22"
```

```
},
{
"uri": "http://example.com/tophat_junctions.bed",
"access_time": "2019-10-22"
},
{
"uri": "http://example.com/unmapped_reads.bam",
"access_time": "2019-10-22"
}
]
},
{
"step_number": "6",
"name": "#Bowtie2_Indexer",
"description": "Bowtie2 Indexer is a tool for indexing
reference genomes of any size used in an alignment. It was
built from `bowtie2-build` script and used for reference
genome indexing aimed at assisting Bowtie2 in fast and
memory-efficient alignment. It outputs an archive which
consists of 6 files with suffixes .1.bt2, .2.bt2, .3.bt2,
.4.bt2, .rev.1.bt2, and .rev.2.bt2. This archive
constitutes the index and should be provided when aligning
the reads (either with [Bowtie2
Aligner](https://igor.sbgenomics.com/public/apps#tool/admin/sbg-public-data/bowtie2-aligner)
or
[TopHat2](https://igor.sbgenomics.com/public/apps#tool/admin/sbg-public-data/tophat2)).
\n\n###Common issues###\nNo issues have been reported.",
"version": "2.2.6",
"prerequisite": [],
"input_list": [
{
"uri":
```

```
"http://example.com/human_g1k_v37_decoy.phiX174_bowtie2-2.2.6.tar",
"access_time": "2019-10-22"
}
],
"output_list": [
{
"uri": "http://example.com/bowtie_index_archive.tar",
"access_time": "2019-10-22"
}
]
}
]
}
```


1.7 Parametric Domain


```
{
"keywords": [],
"xref": [],
"platform": "Seven Bridges Platform",
"pipeline_steps": [
{
"step_number": "1",
"name": "#FastQC",
"description": "FastQC reads a set of sequence files and
produces a quality control (QC) report from each one. These
reports consist of a number of different modules, each of
which will help identify a different type of potential
problem in your data.\n\nFastQC is a tool which takes a
FastQ file and runs a series of tests on it to generate a
comprehensive QC report.  This report will tell you if
there is anything unusual about your sequence.  Each test
```

```
is flagged as a pass, warning, or fail depending on how far
it departs from what you would expect from a normal large
dataset with no significant biases.  It is important to
stress that warnings or even failures do not necessarily
mean that there is a problem with your data, only that it
is unusual.  It is possible that the biological nature of
your sample means that you would expect this particular
bias in your results.",
"version": "0.11.4",
"prerequisite": [],
"input_list": [
{
"uri": "http://example.com/RNASeqdata.fastq",
"access_time": "2019-10-22"
}
],
"output_list": [
{
"uri": "http://example.com/QC_report.zip",
"access_time": "2019-10-22"
}
]
},
{
"step_number": "2",
"name": "#BamTools_Index",
"description": "BamTools Index creates an index file (BAI
or BTI) for a BAM file. If BAI file is present on the input
the tool will skip indexing step and output BAM with
provided BAI file.\n\n**Common issues:** Providing a BAI
file on input will result in a pass-through without
execution, even if a different index format is requested on
```

```
the output (BTI instead of BAI).",
"version": "2.4.0",
"prerequisite": [],
"input_list": [
{
"uri": "http://example.com/input_file.bam",
"access_time": null
}
],
"output_list": [
{
"uri": "http://example.com/output_bam_file.bam",
"access_time": "2019-10-22"
},
{
"uri": "http://example.com/generated_index.bai",
"access_time": "2019-10-22"
}
]
},
{
"step_number": "3",
"name": "#SBG_FASTQ_Quality_Detector",
"description": "FASTQ Quality Scale Detector detects which
quality encoding scheme was used in your reads and
automatically enters the proper value in the \"Quality
Scale\" metadata field.",
"version": null,
"prerequisite": [
{
"name": "reference_genome_used_for_alignment.fasta",
"uri": {
```

```
"uri":
"http://example.com/human_g1k_v37_decoy.phiX174.fasta",
"access_time": "2019-10-22"
}
}
],
"input_list": [
{
"uri": "http://example.com/RNA-Seq_data.fastq",
"access_time": "2019-10-22"
}
],
"output_list": [
{
"uri": "http://example.com/RNASeq_data_updated
metadata.fastq",
"access_time": "2019-10-22"
},
{
"uri": "http://example.com/quality_scale.strings",
"access_time": "2019-10-22"
}
]
},
{
"step_number": "4",
"name": "#Picard_CollectAlignmentSummaryMetrics",
"description": "Picard CollectAlignmentSummaryMetrics
assesses the quality of alignment by analyzing a SAM or BAM
file. It compares it with the reference file (FASTA) and
provides alignment statistics, such as the number of input
reads and the percent of reads that are mapped. It produces
```

```
a file which contains summary alignment metrics from a SAM
or BAM file.\n\nNote: This tool requires the exact same
FASTA file as the one to which raw reads were aligned.",
"version": "1.140",
"prerequisite": [],
"input_list": [
{
"uri": "http://example.com/input_file.bam",
"access_time": "2019-10-22"
},
{
"uri":
"http://example.com/human_g1k_v37_decoy.phiX174.fasta",
"access_time": "2019-10-22"
}
],
"output_list": [
{
"uri": "http://example.com/summary_metrics.txt",
"access_time": "2019-10-22"
}
]
},
{
"step_number": "5",
"name": "#TopHat2",
"description": "## TopHat2 (version
2.1.1)\n\n[TopHat2](https://ccb.jhu.edu/software/tophat/manual.shtml)
is a program that aligns RNA-Seq reads to a genome in order
to identify exon-exon splice junctions. It is built on the
ultrafast short read mapping program
[Bowtie2](http://bowtie-bio.sourceforge.net/bowtie2/manual.shtml).\n\nTopHat2
```

```
can also align reads directly to a transcriptome. If
provided with annotation of known transcripts, TopHat2
constructs a virtual transcriptome and uses Bowtie2 to
align reads to this reference first. Reads that do not
align to the transcriptome are then mapped on the reference
genome. The reads that did align on the transcriptome will
be converted to genomic mappings (spliced as needed) and
merged with the novel mappings and junctions in the final
output.\n\nThis version of TopHat only accepts reads in
FASTQ format. It is optimized for reads that are at least
75bp long.\n\n**Common issues:**\nOne of the most common
issues when running TopHat is incompatibility between
reference genome and annotations. Please, make sure that
you are using compatible FASTA (from which you have created
tar bundle with index files) and GTF files in order to run
tasks successfully. If you suspect your task has failed due
to this incompatibility, you can check the last line in
`job.err.log` file which would look as following if your
assumptions are correct: `Error: Couldn't build bowtie
index with err = 1`.\n\n**Q&A:**\n\n***Q: What should I do
if I already have Bowtie2 index files, not archived as tar
bundle?***\n\n***A***: You can provide your *.bt2 files to
[SBG
Compressor](https://igor.sbgenomics.com/public/apps#admin/sbg-public-data/sbg-compressor-1-0/)
app from our public apps and set \"TAR\" as your output
format. After the task is finished, **you should assign
common prefix of the index files to the `Reference genome`
metadata field** and your TAR is ready for
use.\n\n***Example:***\nIndexed files: chr20.1.bt2,
chr20.2.bt2, chr20.3.bt2, chr20.4.bt2, chr20.rev.1.bt2,
chr20.rev.2.bt2\n\nMetadata - `Reference genome`:
**chr20**\n\n__Important note: In case of paired-end
```

```
alignment it is crucial to set metadata 'paired-end' field
to 1/2. Sequences specified as mate 1s must correspond
file-for-file and read-for-read with those specified for
mate 2s. Reads may be a mix of different lengths. In case
of unpaired reads, the same metadata field should be set to
'-'. Only one type of alignment can be performed at once,
so all specified reads should be either paired or
unpaired.__",
"version": "2.1.0",
"prerequisite": [
{
"name": "Bowtie2",
"uri": {
"uri":
"https://cgc.sbgenomics.com/public/apps#admin/sbg-public-data/bowtie2-aligner/",
"access_time": "2019-10-22"
}
}
],
"input_list": [
{
"uri":
"http://example.com/human_g1k_v37_decoy.phiX174_bowtie2-2.2.6.tar",
"access_time": "2019-10-22"
},
{
"uri": "http://example.com/RNASeq_reads.fastq",
"access_time": "2019-10-22"
},
{
"uri": "http://example.com/Homo_sapiens.GRCh37.75.gtf",
"access_time": "2019-10-22"
```

```
}
],
"output_list": [
{
"uri": "http://example.com/alignment_summary.txt",
"access_time": "2019-10-22"
},
{
"uri": "http://example.com/read_alignments.bam",
"access_time": "2019-10-22"
},
{
"uri": "http://example.com/tophat_deletions.bed",
"access_time": "2019-10-22"
},
{
"uri": "http://example.com/tophat_insertions.bed",
"access_time": "2019-10-22"
},
{
"uri": "http://example.com/tophat_junctions.bed",
"access_time": "2019-10-22"
},
{
"uri": "http://example.com/unmapped_reads.bam",
"access_time": "2019-10-22"
}
]
},
{
"step_number": "6",
"name": "#Bowtie2_Indexer",
```

```
"description": "Bowtie2 Indexer is a tool for indexing
reference genomes of any size used in an alignment. It was
built from `bowtie2-build` script and used for reference
genome indexing aimed at assisting Bowtie2 in fast and
memory-efficient alignment. It outputs an archive which
consists of 6 files with suffixes .1.bt2, .2.bt2, .3.bt2,
.4.bt2, .rev.1.bt2, and .rev.2.bt2. This archive
constitutes the index and should be provided when aligning
the reads (either with [Bowtie2
Aligner](https://igor.sbgenomics.com/public/apps#tool/admin/sbg-public-data/bowtie2-aligner)
or
[TopHat2](https://igor.sbgenomics.com/public/apps#tool/admin/sbg-public-data/tophat2)).
\n\n###Common issues###\nNo issues have been reported.",
"version": "2.2.6",
"prerequisite": [],
"input_list": [
{
"uri":
"http://example.com/human_g1k_v37_decoy.phiX174_bowtie2-2.2.6.tar",
"access_time": "2019-10-22"
}
],
"output_list": [
{
"uri": "http://example.com/bowtie_index_archive.tar",
"access_time": "2019-10-22"
}
]
}
]
}
```

1.8 Input/Output Domain


```
{
  "input_subdomain": [
    {
      "uri": [
        {
          "filename": "",
          "uri": "",
          "access_time": ""
        }
      ]
    }
  ],
  "output_subdomain": [
    {
      "mediatype": "",
      "uri": [
        {
          "uri": "",
          "access_time": ""
        }
      ]
    }
  ]
}
```


1.9 Error Domain


```
{
  "empirical_error": [],
  "algorithmic_error": []
}
```

2 Funding

The Seven Bridges Cancer Genomics Cloud has been funded in whole or in part with Federal funds from the National Cancer Institute, National Institutes of Health, Contract No. HHSN261201400008C and ID/IQ Agreement No. 17X146 under Contract No. HHSN261201500003I.

3 References

Lau et al (2017) The Cancer Genomics Cloud: Collaborative, Reproducible, and Democratized—A New Paradigm in Large-Scale Computational Research. Cancer Res. 77(21):e3-e6. doi: 10.1158/0008-5472.CAN-17-0387.

4 Appendix 1: BioCompute Object Specification v1.3.0

Name	ID	Description
Top Level Fields		
BioCompute Object Indetifier	BCO_id	Unique identifier that should be apllied to each BCO instance. Assigned by a BCO database engine, like URL. It never be reused.
Type	type	As any object of the type, it has its own fields.
Digital signature	digital_signature	A string-type, read-only generated and stored by a BCO database, protecting the object from internal or external alterations without proper validation. It can be used for validation, downloading, and transferring BCOs.
BCO version	bco_spec_version	The version of the BCO specification used to define this document.
Provenance Domain		
Name	name	Name of the BCO.
Structured name	structured_name	Computable text field designed to represent a BCO instance name in visible interfaces
Version	version	Records the versioning of this BCO instance object. A change in the BCO affecting the outcome of the computation should be deposited as a new BCO, not as a new version.
Review	review	Describes the status of an object in the review process. Status flags: unreviewed, in-review, approved, suspended, rejected.
Inheritance/derivation	derived_from	If the object is derived from another, this field will specify the parent object, in the form of the objectid. It is null, if inherits only from the base BioCompute Object or a type definition.
Obsolescence	obsolete	If the object has an expiration date this field will specify that using the datetime type.
Embargo	embargo	If the object has a period of time that it is not public, that range can be specified using these fields. Using the datetime type a start and end time are specified for the embargo.
Created	created	Using the datetime type the time of initial creation of the BCO is recorded.
Modification	modified	Using the datetime type the time of most recent modification of the BCO is recorded.
Contributors	contributors	List to hold contributor identifiers and a description of their type of contribution, including a field for ORCIDs to record author information, as they allow for the author to curate their information after submission.

(continued)

Name	ID	Description
License	license	A space for Creative commons licence or other licence information. The default or recommended licence can be Attribution 4.0 International.
Usability Domain		
Usability Domain	usability_domain	Provides a space for the author to define the usability domain of the BCO. It is an array of free text values. This field is to aid in search-ability and provide a specific description of the object. It helps determine when and how the BCO can be used.
Extension Domain		
Extension Domain	extension_domain	For a user to add more structured information that is defined in the type definition. This section is not evaluated by checks for BCO validity or computational correctness.
Extension to External References: SMART on FHIR Genomics	Extension to External References: SMART on FHIR Genomics	SMART on FHIR Genomics provides a framework for HER-based apps to built on FHIR that integrate clinical and genomics information.
Extension to External References: GitHub	Extension to External References: GitHub	Include an extension to GitHub repositories where HTS computational analysis pipelins, workflows, protocols, and tool or software source code can be stored, deposited, downloaded.
Description Domain		
Description Domain	description_domain	Structured field for description of external references, the pipeline steps, and the relationship of IO objects. Information in this domain is not used for computation. Capture information that is currently being provided in FDA submission in journal format.
Keywords	keywords	List of key map fields to hold a list of keywords to aid in search-ability and description of the object.
External References	xref	It contains a list of the databases and/or ontology IDs that are cross-referenced in the BCO. It provides more specificity in the information related to BCO entries.
Pipeline tools	pipeline_steps	For recording the specifics of a pipeline. Each individual tool is represented as step, at the discretion of the author. Step Number (step_number), Name (name), Tool Description (description), Tool Version (version), Tool Prerequisites (prerequisite), Input List (input_list), Output List (output_list).
Execution Domain		

(continued)

Name	ID	Description
Execution Domain	execution_domain	The filelds required for execution of the BCO have been encapsulated together in order to clearly separate information needed for deployment, software configuration, and running applications in a dependent enviroment.
Script Access Type	script_access_type	This field indicates whether the code of the script to execute the BioCompute Object is access as an external file via HTTP or in-line text in the script field.
Script	script	Points to an internal or external reference to a script object that was used to perform computations for this BCO instance. This may be reference to Galaxy Project or Seven Bridges Genomics pipeline, a Common Workflow Language (CWL) object in GitHub, HIVE computational service or any other type of script.
Pipeline Version	pipeline_version	This field records the version of the pipeline implementation.
Platform/Environment	platform	The multi-value reference to a particular deployment of an existing platform where this BCO can be reproduced (Galaxy or HIVE or CASAVA).
Script Driver	script_driver	The reference to an executable that can be launched in order to perform a sequence of commands described in the script. For example if the pipeline is driven by a HIVE script, the script driver is the hive execution engine. For CWL based scripts specify cwl-runner. Another very general commonly used in Linux based operating systems is shell.
Algorithmic tools and Software Prerequisites	software_prerequisites	Field listing the minimal necessary prerequisites, library, tool versions needed to successfully run the script to produce BCO.
Domain Prerequisites	domain_prerequisites	Listing the minimal necessary domain specific external data source access in order to successfully run the script to produce BCO.
Enviromental parameters	env_parameters	Multi-value additional key value pairs useful to configure the execution environment on the target platform, like compute cores, available memory use of the script.
Parametric Domain		
Parametric Domain	parametric_domain	List of parameters customizing the computational flow which can affect the output of the calculations. These fields are custom to each type of analysis and are tied to a particular pipeline implementation.
Input and Output Domain		

(continued)

Name	ID	Description
Input and output Domain	io_domain	This represents the list of global input and output files created by the computational workflow, excluding the intermediate files.
Input Subdomain	input_subdomain	This field records the references and input files for the entire pipeline. Each type of input file is listed under a key for that type.
Output Subdomain	output_subdomain	This field records the outputs for the entire pipeline .
Error Domain, acceptable range of variability	error_domain	Consists of two subdomains: empirical and algorithmic._The empirical subdomain contains the limits of_detectability_ fps, fns, statistical confidence of outcomes, etc. The algorithmic subdomain is descriptive of errors that originated by fuzziness of the algorithms, driven by stochastic processes, in dynamically parallelized multi-threaded executions, or in machine learning methodologies where the state of the machine can affect the outcomeConsists of two subdomains: empirical and algorithmic. The empirical subdomain contains the limits of detectability FPs, FNs, statistical confidence of outcomes, etc. The algorithmic subdomain is descriptive of errors that originated by fuzziness of the algorithms, driven by stochastic processes, in dynamically parallelized multi-threaded executions, or in machine learning methodologies where the state of the machine can affect the outcome.

5 Appendix 2: The Complete BioCompute Object


```
{
  "bco_spec_version": "https://w3id.org/biocompute/1.3.0/",
  "bco_id": "http://biocompute.sbgenomics.com/bco/5f28ac19-ea58-424b-a6e9-639fc3171302",
  "checksum": "935e8347674716fe3cdfb3f8b40e3c6b6a559debdeaf77cdad9b780f735d85ea",
  "provenance_domain": {
    "name": "RNA-seq Alignment - TopHat",
    "version": "1.0.0",
    "review": [],
    "derived_from": "https://cgc-api.sbgenomics.com/v2/apps/Dennis/el-bco-zip/rna-seq-alignment
    "obsolete_after": "2019-10-21T00:00:00+0000",
    "embargo": ["2019-10-21T00:00:00+0000", "2019-10-21T00:00:00+0000"],
    "created": "2019-10-21T00:00:00+0000",
    "modified": "2019-10-21T00:00:00+0000",
    "contributors": [
      {
        "name": "Elizabeth Christine Lee",
        "affiliation": "George Washington University School of Medicine & Health Sciences, Wash
        "email": "eclee314@gwu.edu",
        "contribution": "createdBy",
        "orcid": "https://orcid.org/0000-0003-0384-595X"
      },
      {
        "name": "Ana Damljanovic",
        "affiliation": "Seven Bridges Genomics",
        "email": "ana.damljanovic@sbgenomics.com",
        "contribution": "authoredBy",
        "orcid": ""
      },
      {
        "name": "Ana Damljanovic",
```

```
        "affiliation": "Seven Bridges Genomics",
        "email": "ana.damljanovic@sbgenomics.com",
        "contribution": "derivedFrom",
        "orcid": ""
      }
    ],
    "license": "https://spdx.org/licenses/CC-BY-4.0.html"
  },
  "usability_domain": "RNA-Seq technology represents a powerful method to interrogate gene expr
  "extension_domain": {
    "fhir_extension": {
      "fhir_endpoint": "",
      "fhir_version": "",
      "fhir_resources": {}
    },
    "scm_extension": {
      "scm_repository": "https://github.com/ECL314/CGC-BCO-Generator-App-Extra-Credit-Assignmen
      "scm_type": "git",
      "scm_commit": "",
      "scm_path": "",
      "scm_preview": ""
    }
  },
  "description_domain": {
    "keywords": [],
    "xref": [],
    "platform": "Seven Bridges Platform",
    "pipeline_steps": [
      {
        "step_number": "1",
        "name": "#FastQC",
        "description": "FastQC reads a set of sequence files and produces a quality control (QC
```

```
    "version": "0.11.4",
    "prerequisite": [],
    "input_list": [
      {
        "uri": "http://example.com/RNASeqdata.fastq",
        "access_time": "2019-10-22"
      }
    ],
    "output_list": [
      {
        "uri": "http://example.com/QC_report.zip",
        "access_time": "2019-10-22"
      }
    ]
  },
  {
    "step_number": "2",
    "name": "#BamTools_Index",
    "description": "BamTools Index creates an index file (BAI or BTI) for a BAM file. If BA
    "version": "2.4.0",
    "prerequisite": [],
    "input_list": [
      {
        "uri": "http://example.com/input_file.bam",
        "access_time": null
      }
    ],
    "output_list": [
      {
        "uri": "http://example.com/output_bam_file.bam",
        "access_time": "2019-10-22"
      },
```

```
      {
        "uri": "http://example.com/generated_index.bai",
        "access_time": "2019-10-22"
      }
    ]
  },
  {
    "step_number": "3",
    "name": "#SBG_FASTQ_Quality_Detector",
    "description": "FASTQ Quality Scale Detector detects which quality encoding scheme was
    "version": null,
    "prerequisite": [
      {
        "name": "reference_genome_used_for_alignment.fasta",
        "uri": {
          "uri": "http://example.com/human_g1k_v37_decoy.phiX174.fasta",
          "access_time": "2019-10-22"
        }
      }
    ],
    "input_list": [
      {
        "uri": "http://example.com/RNA-Seq_data.fastq",
        "access_time": "2019-10-22"
      }
    ],
    "output_list": [
      {
        "uri": "http://example.com/RNASeq_data_updated metadata.fastq",
        "access_time": "2019-10-22"
      },
      {
```

```
        "uri": "http://example.com/quality_scale.strings",
        "access_time": "2019-10-22"
      }
    ]
  },
  {
    "step_number": "4",
    "name": "#Picard_CollectAlignmentSummaryMetrics",
    "description": "Picard CollectAlignmentSummaryMetrics assesses the quality of alignment
    "version": "1.140",
    "prerequisite": [],
    "input_list": [
      {
        "uri": "http://example.com/input_file.bam",
        "access_time": "2019-10-22"
      },
      {
        "uri": "http://example.com/human_g1k_v37_decoy.phiX174.fasta",
        "access_time": "2019-10-22"
      }
    ],
    "output_list": [
      {
        "uri": "http://example.com/summary_metrics.txt",
        "access_time": "2019-10-22"
      }
    ]
  },
  {
    "step_number": "5",
    "name": "#TopHat2",
    "description": "## TopHat2 (version 2.1.1)\n\n[TopHat2](https://ccb.jhu.edu/software/to
```

```
"version": "2.1.0",
"prerequisite": [
  {
    "name": "Bowtie2",
    "uri": {
      "uri": "https://cgc.sbgenomics.com/public/apps#admin/sbg-public-data/bowtie2-alig
      "access_time": "2019-10-22"
    }
  }
],
"input_list": [
  {
    "uri": "http://example.com/human_g1k_v37_decoy.phiX174_bowtie2-2.2.6.tar",
    "access_time": "2019-10-22"
  },
  {
    "uri": "http://example.com/RNASeq_reads.fastq",
    "access_time": "2019-10-22"
  },
  {
    "uri": "http://example.com/Homo_sapiens.GRCh37.75.gtf",
    "access_time": "2019-10-22"
  }
],
"output_list": [
  {
    "uri": "http://example.com/alignment_summary.txt",
    "access_time": "2019-10-22"
  },
  {
    "uri": "http://example.com/read_alignments.bam",
    "access_time": "2019-10-22"
```

```
      },
      {
        "uri": "http://example.com/tophat_deletions.bed",
        "access_time": "2019-10-22"
      },
      {
        "uri": "http://example.com/tophat_insertions.bed",
        "access_time": "2019-10-22"
      },
      {
        "uri": "http://example.com/tophat_junctions.bed",
        "access_time": "2019-10-22"
      },
      {
        "uri": "http://example.com/unmapped_reads.bam",
        "access_time": "2019-10-22"
      }
    ]
  },
  {
    "step_number": "6",
    "name": "#Bowtie2_Indexer",
    "description": "Bowtie2 Indexer is a tool for indexing reference genomes of any size us
    "version": "2.2.6",
    "prerequisite": [],
    "input_list": [
      {
        "uri": "http://example.com/human_g1k_v37_decoy.phiX174_bowtie2-2.2.6.tar",
        "access_time": "2019-10-22"
      }
    ],
    "output_list": [
```

```
          {
            "uri": "http://example.com/bowtie_index_archive.tar",
            "access_time": "2019-10-22"
          }
        ]
      }
    ]
  },
  "execution_domain": {
    "script": "https://cgc-api.sbgenomics.com/v2/apps/Dennis/el-bco-zip/rna-seq-alignment-topha
    "script_driver": "Seven Bridges Common Workflow Language Executor",
    "software_prerequisites": [
      {
        "name": "Cancer Genomics Cloud",
        "version": "",
        "uri": [
          {
            "uri": "https://cgc.sbgenomics.com/",
            "access_time": "2019-10-22",
            "sha1_chksum": ""
          }
        ]
      },
      {
        "name": "Seven Bridges Platform",
        "version": "2019-10-22",
        "uri": [
          {
            "uri": "https://igor.sbgenomics.com/",
            "access_time": "2019-10-22",
            "sha1_chksum": ""
          }
```

```
        ]
      }
    ],
    "external_data_endpoints": [],
    "environment_variables": []
  },
  "parametric_domain": [
    {
      "param": "optimized for read length",
      "value": "at least 75bp",
      "step": "5"
    },
    {
      "param": "metadata for Paired-end field",
      "value": "1 or 2",
      "step": "5"
    }
  ],
  "io_domain": {
    "input_subdomain": [
      {
        "uri": [
          {
            "filename": "",
            "uri": "",
            "access_time": ""
          }
        ]
      }
    ],
    "output_subdomain": [
      {
```

```
        "mediatype": "",
        "uri": [
          {
            "uri": "",
            "access_time": ""
          }
        ]
      }
    ]
  },
  "error_domain": {
    "empirical_error": [],
    "algorithmic_error": []
  }
}
```