G-OnRamp Glossary

# Virtualization

* Virtual machine (VM)

A virtual machine emulates a computer system within another computer system running on either the same or different hardware architecture. G-OnRamp provides a Linux virtual machine that can be deployed on a local computer running macOS or MS Window, or on the Cloud (*e.g.*, Amazon EC2). The “Genome Browser in a Box” product developed by the UCSC Bioinformatics group is another example of a virtual machine.

* Virtual appliance

A pre-configured virtual machine that can be easily installed on multiple virtualization platforms. For example, G-OnRamp provides a virtual appliance in the Open Virtualization Format (OVF) that can be imported into VirtualBox.

* Hypervisor

A piece of hardware or software on a computer (host) that runs and manages the guest virtual machines. A Type-1 hypervisor (*e.g.*, Hyper-V, Xen) runs directly on the host’s hardware. A Type-2 hypervisor (*e.g.*, VirtualBox, VMWare) is a software program that runs on the host OS. Type-1 hypervisors generally have better performance than Type-2 hypervisors. The G-OnRamp virtual appliance is designed to run on a Type-2 hypervisor.

* Host / Guest Operating System (OS)

The computer system that runs and manages virtual machines is called the “host”. Each virtual machine running on the host is called the “guest”. In a Type-2 hypervisor, the virtual machines are managed by the host OS, and the guest OS runs inside a virtual machine. For example, if the G-OnRamp virtual machine is running on macOS, then the Ubuntu OS running inside the G-OnRamp virtual machine is the guest OS, and macOS is the host OS. Depending on the amount of compute resources available, a host OS can run multiple guest OS’s simultaneously.

* VirtualBox

An open-source Type-2 hypervisor that runs on MS Windows, macOS, and Linux. VirtualBox is available for download at <https://www.virtualbox.org/>.

# Deployment

* Ansible

An open-source program for automating the configuration and deployment of multiple applications. G-OnRamp uses Ansible playbooks to automate the installation and configuration of Galaxy, Galaxy Tools, and the bioinformatics tool dependencies. The Ansible role for installing and managing Galaxy servers is available at <https://github.com/galaxyproject/ansible-galaxy>.

* Amazon Web Services (AWS)

A cloud computing platform developed by Amazon, AWS has two primary service models: Infrastructure as a Service (IaaS), whereby users access the hardware resources (*e.g.*, storage, compute, networking) from the cloud provider, and Platform as a Service (PaaS), whereby users develop applications based on the development tools and platforms (*e.g.*, AWS Lambda) managed by the cloud provider. G-OnRamp uses AWS IaaS to provide a cloud deployment option for G-OnRamp.

* Amazon EC2

The Amazon Elastic Compute Cloud (EC2) enables users to run virtual machines (instances) on AWS. EC2 provides instance types that have different compute, memory, storage, and network capacities (see <https://aws.amazon.com/ec2/instance-types/>). Persistent data for Amazon EC2 instances are stored in the Amazon Elastic Block Store (EBS) volumes. The cloud version of the G-OnRamp virtual machine is designed to run on Amazon EC2.

* Amazon Machine Image (AMI)

An AMI is a virtual appliance that is designed specifically for Amazon EC2. G-OnRamp provides an AMI image with Galaxy and all the G-OnRamp tools already installed. Users can use this AMI image to launch one or more instances of G-OnRamp on Amazon EC2.

# General Computer / Networking Terms

* Extensible Markup Language (XML)

A standardized machine and human readable document format defined by the W3C’s XML 1.0 Specification (see <https://www.w3.org/TR/xml/>). XML files are used to configure Galaxy, and to develop Galaxy wrappers for bioinformatics tools (see <https://docs.galaxyproject.org/en/master/dev/schema.html>).

* Application Programming Interface (API)

An API provides a specification (contract) that facilitates the communication between the provider and the consumer of a software component. Consumers interact with the software component through the APIs, which enable the providers to change the internal implementations of the software component without affecting the consumer. Galaxy has a web-based API (<https://docs.galaxyproject.org/en/master/api_doc.html>), and the BioBlend Python library can be used to interact with the Galaxy API (see <http://bioblend.readthedocs.io/en/latest/>).

* Secure Shell (SSH)

SSH enables the secure communication between two computers via an insecure network. SSH is often used to securely access a remote machine. For example, one can use an SSH client to access a G-OnRamp instance deployed on Amazon EC2 (see <http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AccessingInstancesLinux.html>).

# Galaxy

* History

Galaxy History records each step of the analysis performed using Galaxy. To ensure reproducibility, the History keeps track of the metadata associated with each dataset, and the tool parameters used in each step of the analysis. See the “Histories” tutorial on the Galaxy web site for details: <https://galaxyproject.org/tutorials/histories/>.

* Tools

The bioinformatics programs available on a Galaxy instance can be accessed through the Tools panel on the left side of the Galaxy interface. Galaxy tools provide a standardized web interface for defining the inputs, program parameters, and outputs of bioinformatics programs. The standardized input and output formats enable the composition of multiple Galaxy tools into an analysis workflow. G-OnRamp is primarily a collection of Galaxy tool wrappers for bioinformatics programs that have been designed for genome annotation and for constructing genome browsers.

* Workflow

Most bioinformatics analyses require the use of multiple tools, and each tool has its own set of parameters. Galaxy can encapsulate these tools and tool parameters into a Workflow in order to ensure reproducibility and to facilitate reuse. Workflows can either be derived from a Galaxy History, or be created *de novo* using the Workflow Canvas (see <https://galaxyproject.org/learn/advanced-workflow/>). G-OnRamp provides Workflows that could be used to construct Assembly Hubs for the UCSC Genome Browser and for JBrowse.

# Datatypes

Detailed descriptions of common Galaxy datatypes are available on the Galaxy web site at <https://galaxyproject.org/learn/datatypes/>. Detailed descriptions of common bioinformatics data formats are available on the “Data File Formats” page on the UCSC Genome Browser web site at <https://genome.ucsc.edu/FAQ/FAQformat.html>.

* FASTA

A FASTA file contains one or more nucleotide or protein sequences. Each sequence in the FASTA file begins with a definition line (defline). The defline is demarcated by the “>” symbol at the beginning of the line, followed by the sequence identifier and an optional description. The lines after the defline correspond to the nucleotides or amino acids for that sequence record (see the section A of the BLAST help page for details: <https://blast.ncbi.nlm.nih.gov/Blast.cgi?CMD=Web&PAGE_TYPE=BlastDocs&DOC_TYPE=BlastHelp>). In the context of the G-OnRamp workflows, the genome assembly and the protein sequences from the informant genome must be in FASTA format.

* FASTQ

The FASTQ format is most commonly used to store second and third generation sequencing data. Each sequence record in the FASTQ format consists of four lines. The first line begins with an “@” symbol followed by the sequence ID. The second line contains either the nucleotide or amino acid sequences for that sequence record. The third line begins with a “+” symbol, followed optionally by the sequence ID. The fourth line contains the quality scores for the sequence on the second line. For the G-OnRamp workflow, the RNA-Seq input dataset must be in FASTQ format.

* FASTQSANGER

For Illumina sequencing data, the scheme used to encode the quality scores of the sequence in a FASTQ record depends on the version of the Illumina CASAVA pipeline used to produce the sequencing data. (See the “Encoding” section of the “FASTQ format” Wikipedia page at <https://en.wikipedia.org/wiki/FASTQ_format> for details.) Since 2011 (CASAVA 1.8+), Illumina uses the FASTQSANGER format to encode the quality scores.

The FASTQSANGER format uses the ASCII character codes 33 to 126 to encode the quality of each base, which corresponds to the Phred quality scores of 0 to 93 (i.e., ASCII character code = 33 + Phred quality score). The Phred quality score corresponds to the estimated error rate of the base call in log scale, where a base with a Phred score of 10 has an estimated error rate of 1/10, and a Phred score of 20 has an estimated error rate of 1/100 (i.e., 1/102). Tools such as FastQC can estimate the quality encoding of a FASTQ file (see <https://www.bioinformatics.babraham.ac.uk/projects/fastqc/> for details).

* SAM

The Sequence Alignment/Map format is a plain text format that is most often used to store the alignment between sequencing reads from second or third generation sequencers against a reference genome. The headers section of a SAM file begins with the “@” character. The alignment section consists of 11 required fields, followed by any number of optional fields. The SAM format specification is available online at <https://samtools.github.io/hts-specs/SAMv1.pdf>.

* BAM

BAM is a (BGZF) compressed version of the SAM file. Most Galaxy Next Generation Sequencing (NGS) alignment tools (*e.g.*, Bowtie2, HISAT2) will automatically convert the alignment results to BAM format. Programs such as SAMtools (<http://www.htslib.org/>) and Picard Tools (<https://broadinstitute.github.io/picard/>) can be used to convert and manipulate BAM files.

* BED

The Browser Extensible Data (BED) format is a flexible tab-delimited text file for describing genomic features. The BED format consists of three required fields, followed by nine optional fields. A BED file uses a zero-start, half-open coordinate system. (For a sequence with length n, the first base of the sequence is at position 0 and the end of the sequence is at n-1; see <http://genome.ucsc.edu/blog/the-ucsc-genome-browser-coordinate-counting-systems/> for details.)

The most common types of BED file formats are:

* + BED4: chrom, chromStart, chromEnd, name
	+ BED6: BED4 + score, strand
	+ BED9: BED6 + thickStart, thickEnd, itemRgb
	+ BED12: BED9 + blockCounts, blockSizes, blockStarts

The thickStart and thickEnd fields in the BED9 format could be used to demarcate coding regions within a transcript, while the blocks in the BED12 format could be used to denote exons. A BED file may also include extra fields that follow the required and optional fields. For example, the TRF BED format used by G-OnRamp corresponds to the BED4+12 format, where the first four BED fields are used to specify the location of each tandem repeat, followed by 12 additional fields that describe the characteristics of each tandem repeat (*e.g.*, period, alignment score, entropy).

* bigBed

The bigBed format is a binary, indexed version of the BED format. This file format is designed for the UCSC Track Hubs and Assembly Hubs. The bigBed files are hosted on a remote server, and only the portion of the file required to display the evidence track for the genomic region of interests is transferred to the UCSC Genome Browser. Most of the files produced by the Hub Archive Creator are in bigBed format. Similar to the BED format, a bigBed entry may contain extra fields in addition to the three required and nine optional fields. An AutoSql schema (a file with the .as extension) is used to define the extra fields in a bigBed file (see <http://genomewiki.ucsc.edu/index.php/AutoSql>). File formats such as bigGenePred, bigPsl, bigMaf, and bigChain are bigBed files with extra fields defined by their respective AutoSql schemas. (See the “bigBed Track Format” page for details: <https://genome.ucsc.edu/goldenPath/help/bigBed.html>.)

* bigGenePred

The bigGenePred format is used to represent gene predictions on the UCSC Genome Browser. The bigGenePred format is a variant of the bigBed file format that consists of 20 fields (BED12+8), and it is derived from the Gene Predictions (Extended) format (genePredExt; <https://genome.ucsc.edu/FAQ/FAQformat.html#format9>). The Hub Archive Creator constructs bigGenePred files to depict the results for the SNAP, GlimmerHMM, and Augustus gene predictions. (See the “bigGenePred Track Format” page for details: <https://genome.ucsc.edu/goldenPath/help/bigGenePred.html>.)

* PSL

The Pattern Space Layout (PSL) format consists of 21 columns, and it is used to represent sequence alignments. PSL files are usually produced by BLAT (see <https://genome.ucsc.edu/goldenpath/help/blatSpec.html>). The UCSC Bioinformatics group also provides a tools for converting NCBI BLAST results to PSL format (*e.g.*, blastXmlToPsl) See the “Examples” section of the documentation for the PSL format on the UCSC Genome Browser web site for details on the special rules used to handle alignments on the minus strand and for protein queries (<https://genome.ucsc.edu/FAQ/FAQformat.html#format2>).

* bigPsl

The bigPsl format is used to represent sequence alignments on the UCSC Genome Browser. It is a variant of the bigBed file format that consists of either 24 (BED12+12) or 25 fields (BED12+13), and it is derived from the PSL format. (See the “bigPsl Track Format” page for details: <https://genome.ucsc.edu/goldenPath/help/bigPsl.html>.) The Hub Archive Creator constructs bigPsl files to represent the BLAT alignments of transcripts from an informant species against a target genome assembly, and the tblastn alignments of protein sequences against a target genome.

* GenBank (gb)

The GenBank flatfile format is developed by NCBI to describe the annotations and the sequence data for nucleotide and protein sequences. G-OnRamp uses the mRNA sequence records of the informant species in GenBank flatfile format as the input dataset for the BLAT subworkflow because it contains the location of the coding regions within each transcript. The specifications for the GenBank flatfile format is available on the NCBI FTP site at ftp://ftp.ncbi.nlm.nih.gov/genbank/gbrel.txt. The feature table descriptions within a GenBank record is defined by the International Nucleotide Sequence Database Collaboration (INSDC), and it is available at <http://www.insdc.org/files/feature_table.html>.

* bigWig

The bigWig format is a binary, compressed version of the Wiggle format used to display high density, continuous datasets (*e.g.*, read coverage from an RNA-Seq experiment) on the UCSC Genome Browser and JBrowse. Similar to the bigBed format, the bigWig format is optimized for UCSC Track Hubs and Assembly Hubs because only the portion of the file required to display the evidence track for the genomic region of interests is transferred to the UCSC Genome Browser. Note that unlike BED files, locations in the bigWig format are described by a 1-based, fully closed coordinate system. For sequence of length n, the first position of the sequence is at position 1, and the last position of the sequence is at position n. (See the “BigWig Track Format” page on the UCSC web site for details: <https://genome.ucsc.edu/goldenpath/help/bigWig.html>.)

* GFF3

General Feature Format Version 3 (GFF3) is a 9-column, tab-delimited text file format used to describe genomic features and alignments. This is the primary input format for JBrowse, for many gene predictors, and for most model organism databases (*e.g.*, FlyBase). The last column of the GFF3 line consists of attributes in the ‘key=value’ format, and multiple attributes are separated by semicolons. The Parent attribute is used to describe the parent-child relationship between different items in the GFF3 file. For example, the Parent of an exon feature is a mRNA feature, and the parent of a mRNA feature is a gene feature. The GFF3 format can also use the attributes column to describe a sequence alignment. The GFF3 match lines contain the “Gap” attribute that specifies the alignment operations (i.e. matches, insertions, and deletions). G-OnRamp uses the GFF3 format to store the results from different gene predictors, and to display the tblastn alignment results on JBrowse. The GFF3 specification is available at <https://github.com/The-Sequence-Ontology/Specifications/blob/master/gff3.md>.

* GTF

The Gene Transfer Format (GTF) is similar to the GFF3 format, but it is designed specifically to represent gene models. The last column of a GTF item contains a list of attributes. Each attribute has a key, followed by a space, and a value enclosed by double quotes (i.e. ‘key “value”’). The list of attributes must begin with the two required attributes “gene\_id” and “transcript\_id”. Multiple attributes are separated by semicolons. RNA-Seq tools such as Cufflinks and StringTie produces the transcripts assembled from RNA-Seq data in GTF format. The reference annotations used to guide the Cufflinks and StringTie transcript assemblers are also often in GTF format. GTF annotation files for many organisms are available for download through the Ensembl web site (<http://www.ensembl.org/info/data/ftp/index.html>). The UCSC Genome Browser Wiki provides instructions on how to create GTF files using the tools provided by UCSC (<http://genomewiki.ucsc.edu/index.php/Genes_in_gtf_or_gff_format>). The GTF specification is available at <http://mblab.wustl.edu/GTF22.html>.

# Bioinformatics tools

This section provides a brief overview of the bioinformatics tools used by the G-OnRamp workflows. See the references within the description for additional information on the implementation and the algorithms used by each tool.

## Repeat finders

* WindowMasker

WindowMasker uses a window-based approach to identify low complexity sequences and transposon remnants within a DNA sequence. The WindowMasker mk\_counts module counts the number of times that each (short) sequence of length k (k-mer) that appears in the genome. This information is used by the ustat module to mask repetitive sequences in a genome. The repeat masking information can be incorporated into BLAST database, thereby reducing the number of spurious matches in the BLAST results, and improve the performance of BLAST (Morgulis A *et al.*, 2006; PMID: 16287941).

* TrfBig

TrfBig is a wrapper developed by the UCSC Bioinformatics group to facilitate the identification of tandem repeats within genomic sequences using Tandem Repeats Finder (TRF). Tandem repeats are approximate copies of the same DNA sequence that are located next to each other. See Benson G, 1999 (PMID: 9862982) and the Tandem Repeats Finder web site for details (<https://tandem.bu.edu/trf/trfdesc.html>).

## Sequence similarity searches

* NCBI BLAST+

See Camacho C *et al.*, 2009 (PMID: 20003500) for an overview of NCBI BLAST+. See the “BLAST Sequence Analysis Tool” section of the NCBI Handbook for an overview of NCBI BLAST: <https://www.ncbi.nlm.nih.gov/books/NBK153387/>.

* + makeblastdb

Creates a BLAST database from a collection of nucleotide or protein sequences in FASTA format. The BLAST database can include repeat masking data from tools such as WindowMasker and Tandem Repeats Finder (TRF).

* + tblastn

The tblastn program searches a translated nucleotide database using one or more protein query sequences. For G-OnRamp, tblastn is used to detect sequence similarity between protein sequences from an informant species and the genome assembly of the target species.

* + blastXmlToPsl

This UCSC tool converts the XML alignment output file from a tblastn search into the PSL format used by the UCSC Genome Browser.

* + pslToBigPsl

This UCSC tool converts the PSL alignment files into the bigPsl format to facilitate the display of sequence alignments in UCSC Assembly Hubs.

* BLAT mRNA alignments
	+ BLAT

The BLAST-like Alignment Tool (BLAT) is designed to detect sequence similarity between nucleotide and protein sequences. It is originally designed to map transcripts and protein sequences against the genome assembly of the same species. BLAT is much faster but is less sensitive than NCBI BLAST+ (Kent WJ, 2002; PMID: 11932250). BLAT is used in the G-OnRamp workflow to detect sequence similarity between transcript sequences from an informant species and the genome assembly of the target species.

* + gbToFasta

This UCSC tool converts the transcript sequence records in the GenBank flatfile format into the FASTA format. It also creates an additional file that contains the metadata associated with each transcript, such as the location of the coding region within the transcript.

* + pslCDnaFilter

This UCSC tool is used to filter the cDNA alignments against a genome assembly produced by BLAT. G-OnRamp uses this tool to filter the transcript alignments produced by BLAT in order to identify the best placement of each transcript from the informant species on the genome assembly of the target species. Transcript alignments with scores that are close to the best score are also kept in order to facilitate the identification of partial genes and paralogs.

* + pslCheck

This UCSC tool validates the PSL alignment output produced by BLAT. This tool verifies that the PSL file conforms to the PSL specification, and that the sequences and the sequence lengths described in the PSL file are consistent with the lengths of the query and target sequences in the original FASTA files.

* + pslPosTarget

This UCSC tool converts the PSL alignments so that they are with respect to the orientation of the target sequence.

## RNA-Seq

* HISAT

An alignment program for second and third generation sequencing reads, HISAT uses multiple indices to enable the efficient mapping of unspliced and spliced reads against a target genome (Kim D *et al.*, 2015; PMID: 25751142). See the HISAT2 Manual for the list of available alignment options (<https://ccb.jhu.edu/software/hisat2/manual.shtml>). The G-OnRamp workflow uses HISAT to map RNA-Seq reads against the target genome.

* regtools junction extract

The junction extract command of regtools (<http://regtools.readthedocs.io/en/latest/>) identifies splice junctions based on the spliced alignments in an RNA-Seq BAM file. The documentation for the regtools junction extract tool is available at <https://regtools.readthedocs.io/en/latest/commands/junctions-extract/>.

* Convert BAM to bigWig

This tool calculates the number of reads that align to each genomic position, and converts the results into the bigWig format for display on the UCSC Genome Browser and on JBrowse. G-OnRamp uses this tool to show the RNA-Seq alignment coverage.

* StringTie

A reference-based transcriptome assembler, StringTie uses a network flow algorithm to assemble transcripts based on the spliced and unspliced RNA-Seq reads that have been mapped to the target genome (Pertea M *et al.*, 2015; PMID: 25690850). StringTie also provides an estimate of the expression levels of each assembled transcript that could be used to identify differentially expressed genes within a sample.

## Gene predictions

* Augustus

The Augustus gene predictor is based on a Generalized Hidden Markov Model (GHMM), and it can predict multiple isoforms for each gene (Stanke M *et al.*, 2006; PMID: 16845043). Augustus shows high sensitivity and specificity for the gene predictions in many different genomes (see <http://augustus.gobics.de/accuracy>). Augustus also provides a web server for generating species-specific gene prediction parameters (<http://bioinf.uni-greifswald.de/webaugustus/trainingtutorial.gsp>).

* GlimmerHMM

GlimmerHMM is one of the first *ab initio* gene predictor for eukaryotes and it generally shows higher sensitivity and specificity than Genscan (Majoros WH *et al.*, 2004; PMID: 15145805). The GlimmerHMM user manual is available online at <https://ccb.jhu.edu/software/glimmerhmm/man.shtml>.

* SNAP

SNAP is an *ab initio* gene predictor for prokaryotic and eukaryotic genomes that shows better sensitivity and specificity than Genscan (Korf I 2004; PMID: 15144565). SNAP can estimate the gene prediction parameters for a novel genome that has limited amount of experimental data or manually curated gene models.

## UCSC Assembly Hub

* Hub Archive Creator

The Hub Archive Creator converts the genome assembly and the output from repeat finders, sequence similarity searches, RNA-Seq tools, and gene predictors into the UCSC Assembly Hub format (see: <http://genomewiki.ucsc.edu/index.php/Assembly_Hubs>). The specification for UCSC Track Hubs and Assembly Hubs are available on the UCSC web site at <http://genome.ucsc.edu/goldenPath/help/trackDb/trackDbHub.html>. The Hub Archive Creator is available at <https://github.com/goeckslab/hub-archive-creator>.

## JBrowse Assembly Hub

* JBrowse Archive Creator

The JBrowse Archive Creator converts the genome assembly and the output from repeat finders, sequence similarity searches, RNA-Seq tools, and gene predictors into a format that is compatible with JBrowse. JBrowse (<https://jbrowse.org/>) is a web-based genome browser where the genome browser image is rendered by the web browser (using HTML5 and JavaScript), which results in improved performance and scalability (Buels R *et al.*, 2016; PMID: 27072794). The JBrowse Archive Creator is available at <https://github.com/Yating-L/jbrowse-archive-creator>.

## Galaxy Tools for interface with Apollo

* Create or Update Organism

The tool takes a JBrowse instance generated by the JBrowse Archive Creator, and create an organism (an Apollo instance) on Apollo.

* Apollo User Manager

This tool is used to manage Apollo users. The currently supported operation including:

* + Create a new user
	+ Delete a user
	+ Create a user group
	+ Delete a user group
	+ Add a user to a user group
	+ Remove a user from a user group
* Delete an Apollo record

The tool is used to delete an organism from Apollo and export all annotations on the organism.