Jumpstart your genomics pipelines with genomepy

This manuscript (permalink) was automatically generated from vanheeringen-lab/genomepy_manuscript@e0e1b23 on March 24, 2021.

Authors

- **Siebren Frolich**
  0000-0001-6925-8446 · siebrenf
  Department of Molecular Developmental Biology, Radboud University

- **Maarten van der Sande**
  0000-0001-7803-1526 · Maarten-vd-Sande · MaartenvdSande
  Department of Molecular Developmental Biology, Radboud University

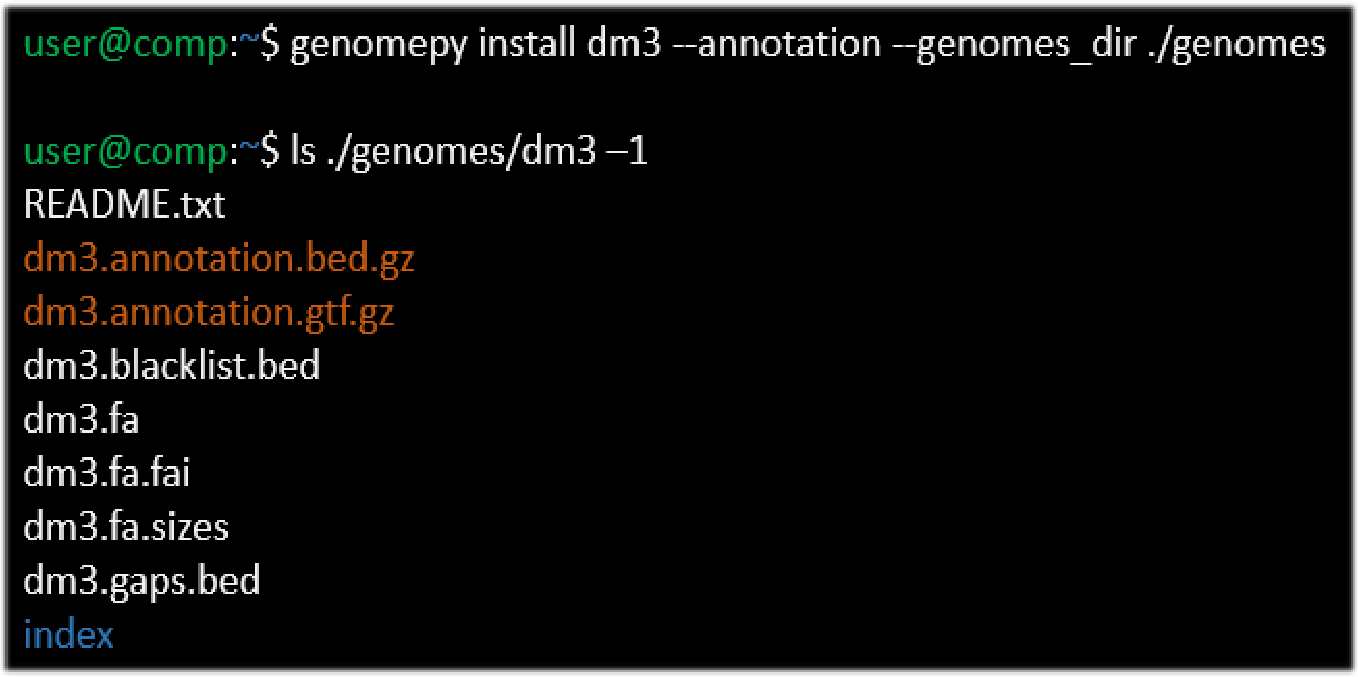
- **Simon van Heeringen**
  0000-0002-0411-3219 · simonvh · svheeringen
  Department of Molecular Developmental Biology, Radboud University

Abstract

Analyzing Omics data, including ATAC, ChIP and RNA-sequencing, requires genomic data such as a genome assembly and gene annotations. These resources can generally be retrieved from multiple organizations, at multiple versions, and generated with varying methods. Which data to use depends on the research context, data reuse and quality. Meanwhile, many bioinformatic workflows and pipelines require the user to supply this genomic data manually, which can be a tedious and error-prone process.

Here we present genomepy, a quality-of-life enhancement tool, that can navigate the assembly databases of Ensembl, UCSC and NCBI. Genomepy can `search` and `install` genome assemblies and gene annotation data in a consistent, reproducible and documented manner. The search function retrieves genomes related to the search term, and can do so for one or all providers to allow the user to make an informed decision. The install function can download a specified genome and gene annotation, from any database, with sensible yet controllable defaults. Additional supporting data can be automatically generated, such as aligner indexes, genome metadata and blacklists.

Genomepy provides these functionalities and more via command line interface and Python application programming interface, aimed at ease of use and integration in automated pipelines.



```
user@comp:~$ genomepy install dm3 --annotation --genomes_dir ./genomes

user@comp:~$ ls ./genomes/dm3 –1
README.txt
dm3.annotation.bed.gz
dm3.annotation.gtf.gz
dm3.blacklist.bed
dm3.fa
dm3.fa.fai
dm3.fa.sizes
dm3.gaps.bed
index
```

Figure 1: executive overview.

Introduction

In order to cope with the explosive increase in Omics data, robust and scalable bioinformatics tools are required. Many tools have been developed to perform the functions of preprocessing 1, analysis 2 and workflow management [3,4] However, many tools cannot obtain all input data automatically, notably the genomic data, which includes the genome assembly, gene annotation and derived files. These data can be obtained from a variety of different providers, including three major providers, Ensembl 5, UCSC 6 and NCBI 7, and many niche providers, such as flybase 8, wormbase 9 or xenbase 10. Each provider has a different method of generating genome assemblies and gene annotations, which can affect available data formats, naming schema, information density, as well as availability, accessibility and relevance. The differences between these data significantly impact the compatibility of the reference data with research tools [11], other reference databases, and other research.

In order to assist in searching through genome providers for, and standardize the processing of genomic data, we developed genomepy. The genomepy search function returns all genomes on the three major providers containing the search term in their name, description or accession identifier, as well as genomes matching a taxonomy identifier. The genomepy install function retrieves a specified genome assembly and related data in a format ready for downstream use.

Related Work

Ensembl, UCSC and NCBI all support downloading from their individual databases via accessible FTP archives, web portals, and REST APIs. To access these databases programmatically, there exists several external tools, such as the ncbi-genome-download tool [12] and ucsc-genomes-downloader [13]. However, to our knowledge no tool exists that can consistently search or download from all three major genome providers.

There are several existing tools for reproducibly sharing reference data between projects. Data management tools accept reference data and derived assets such as aligner indexes, and include iGenomes [14], refGenie [15] and Go Get Data [16]. These tools excel in their ability to reproducibly share data, a feature which is not present in genomepy, and can be used to obtain and manage previously generated data with ease. However, these tools require the user to supply the reference data to any new assembly, newer version, or certain assets.

We conclude that there was a need for a tool that can programmatically obtain and preprocess genomic data, which is how genomepy came to be.

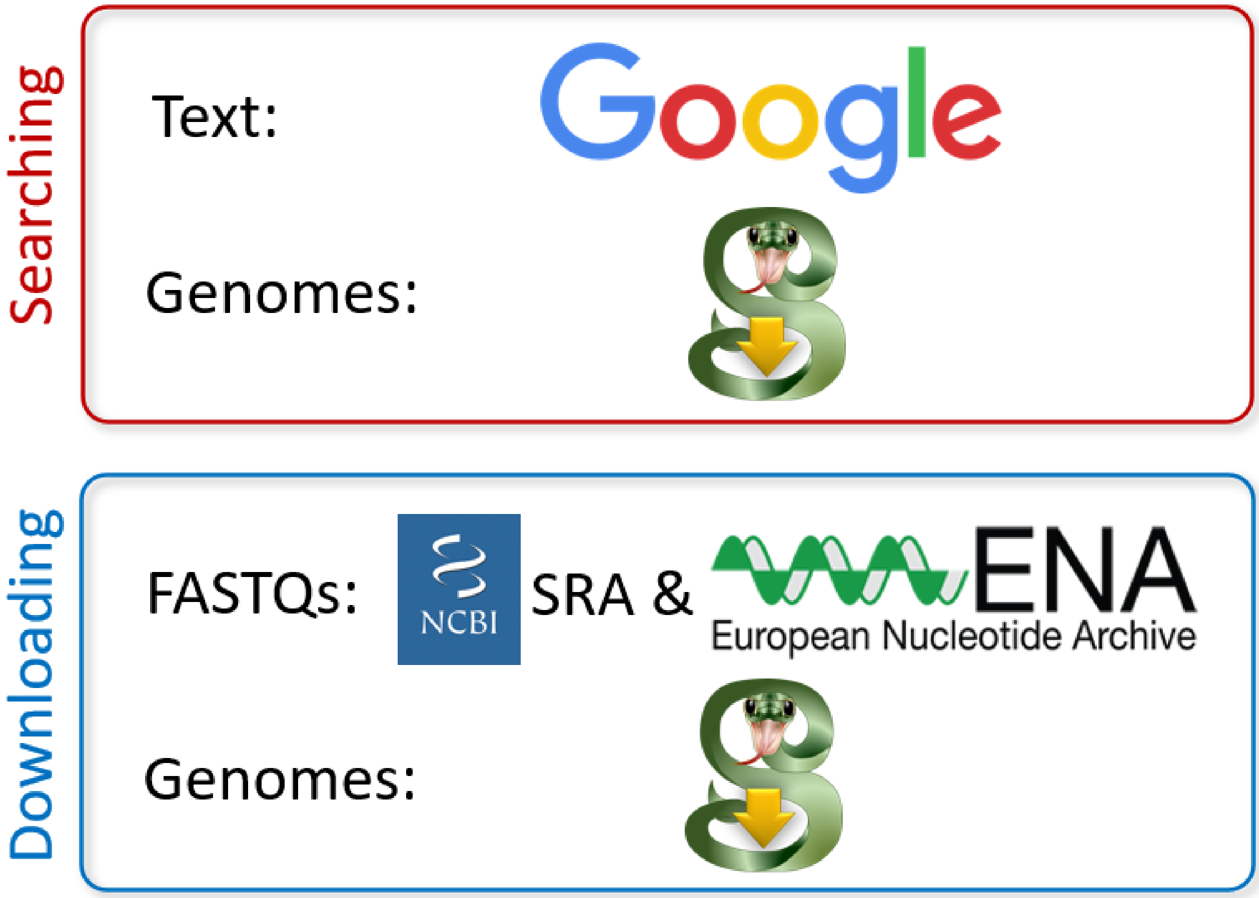


Figure 2: niche for genomepy in the bioinformatics ecosystem.

genomepy

The core functionalities of genomepy are to search, download and prepare genomes and gene annotations. These functions are split over the `search` and `install` functions.

Search will query the three major providers for a given search term. Genomepy can search for text terms in genome names or descriptions, taxonomy identifiers and accession numbers, and will automatically detect which. The search results are returned with available metadata for review.

An assembly name from a major provider can be passed to the install function, along with the name of the provider if the data is available on multiple. Alternatively, if the assembly originates from another source, the url to the genome can be passed. Next, the genome assembly is downloaded with the desired sequence masking level [17,18]. By default soft masked genomes are downloaded, but unmasked or hard masked can be downloaded (or generated if required) as well. Reference assemblies often contain alternate sequences to reflect biological diversity. For the purpose of sequence alignment however, the best results are obtained if there is one reference per nucleotide. Therefore genomepy filters out alternative regions, unless specified otherwise. Additionally, regex filters may be passed to either include or exclude contigs (chromosomes, scaffolds, etc.) by name. Once filtering is performed, genomepy generates commonly used support files. The genome is indexed using pyfaidx 19, and contig sizes and contig gap sizes are collected in separate files.

If specified and available, genomepy will download the gene annotation. Gene annotations are output in the commonly used GTF and BED formats. Contig names of the genome and gene annotation are checked for compatibility. Should these mismatch, genomepy will attempt to match the names in the annotations to the genome.

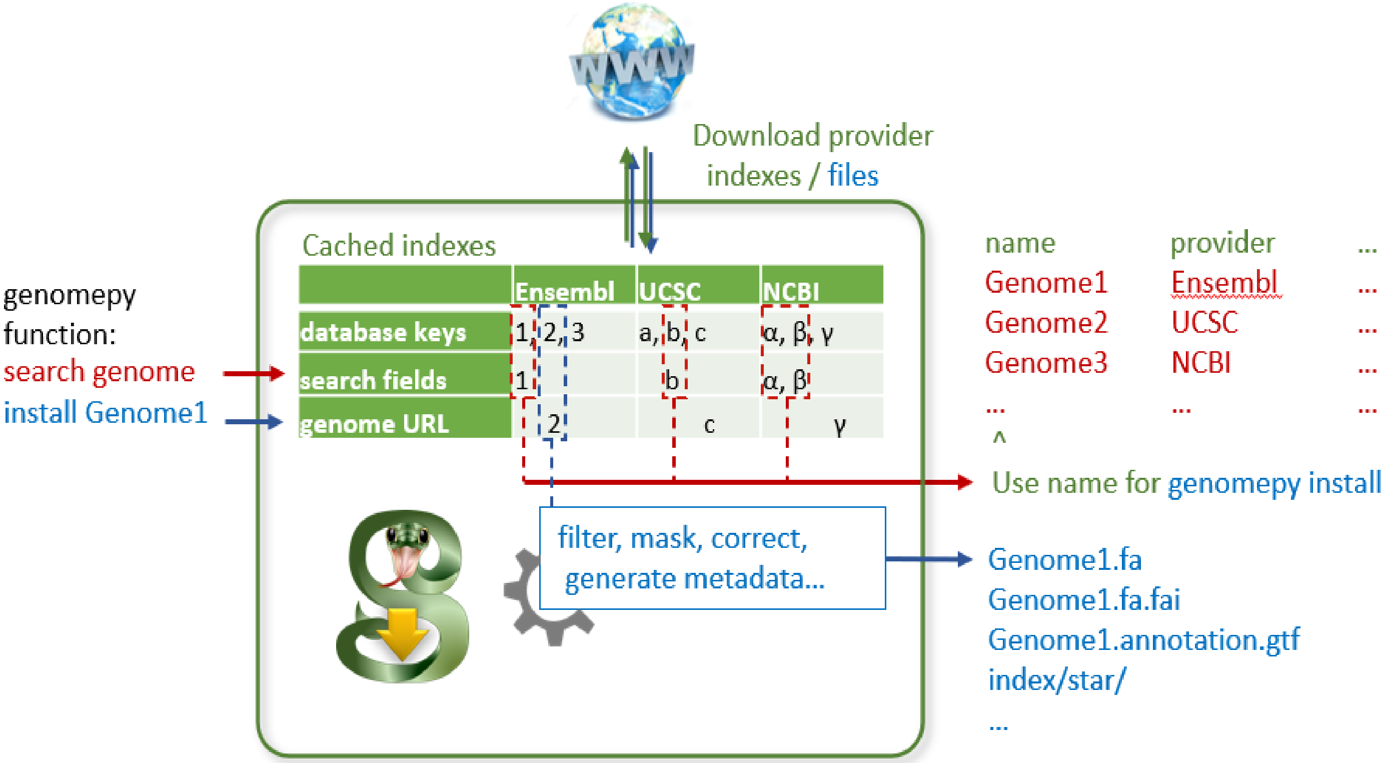


Figure 3: workflow for `genomepy search` and `genomepy install`.

Genomepy facilitates optional processing steps via plugins. These options can be inspected and toggled with the `genomepy plugin` command line function. The blacklist plugin downloads blacklists by the Kundaje lab [20] for the supported genomes. Other plugins support the generation of aligner indexes, including DNA aligner indexes for Bowtie2 [21], BWA [22], GMAP [23] or Minimap2 [24], and splice-aware aligners such as STAR [25] and HISAT2 [26].

For data provenance and reproducibility, a README file is kept with the timestamp, URLs to the source files, the steps performed, and filtered contigs.

Conclusion

Obtaining suitable genomic data is a principal step in any genomics project. Here we demonstrated how to generate an overview of genomes on the three major providers, and how reproducibly download and process genomic data using genomepy. Genomepy provides full control via its command line and Python application programming interfaces. This allows genomepy to automate a step in Omics research that was previously required to be performed by hand.

Code availability

Genomepy can be installed using Bioconda and Pip. Code and documentation are available at https://github.com/vanheeringen-lab/genomepy.

References

1. **seq2science**
Maarten Van Der Sande, Siebren Frölich, Jos Smits, Tilman Schäfers, Rebecca Snabel, Simon Van Heeringen
Zenodo (2021-03-03) https://doi.org/ghktg7
DOI: 10.5281/zenodo.3921913

2. **vanheeringen-lab/gimmemotifs: Version 0.15.2**
Simon Van Heeringen, Siebren Frölich, Maarten-Vd-Sande, Aaron Statham
Zenodo (2020-11-26) https://doi.org/gjjkww
DOI: 10.5281/zenodo.824117

3. **Sustainable data analysis with Snakemake**
Felix Mölder, Kim Philipp Jablonski, Brice Letcher, Michael B. Hall, Christopher H. Tomkins-Tinch, Vanessa Sochat, Jan Forster, Soohyun Lee, Sven O. Twardziok, Alexander Kanitz, … Johannes Köster
F1000Research (2021-01-18) https://doi.org/gjjkwv
DOI: 10.12688/f1000research.29032.1

4. **GENOMEPY — Snakemake Wrappers tags/0.73.0 documentation** https://snakemake-wrappers.readthedocs.io/en/stable/wrappers/genomepy.html

5. **Validate User** https://academic.oup.com/crawlprevention/governor?content=%2fnar%2fadvance-article%2fdoi%2f10.1093%2fnar%2fgkz966%2f5613682

6. **The Human Genome Browser at UCSC**
W. J. Kent, C. W. Sugnet, T. S. Furey, K. M. Roskin, T. H. Pringle, A. M. Zahler, a. D. Haussler
Genome Research (2002-05-16) https://doi.org/fpf5rm
DOI: 10.1101/gr.229102 · PMID: 12045153 · PMCID: PMC186604

7. **The UCSC Table Browser data retrieval tool**
Donna Karolchik, Angela S Hinrichs, Terrence S Furey, Krishna M Roskin, Charles W Sugnet, David Haussler, W James Kent
Nucleic acids research (2004-01-01) https://www.ncbi.nlm.nih.gov/pmc/articles/PMC308837/
DOI: 10.1093/nar/gkh103 · PMID: 14681465 · PMCID: PMC308837

8. **Validate User** https://academic.oup.com/crawlprevention/governor?content=%2fnar%2farticle%2f47%2fD1%2fD759%2f5144957

9. **Validate User** https://academic.oup.com/crawlprevention/governor?content=%2fnar%2fadvance-article%2fdoi%2f10.1093%2fnar%2fgkz920%2f5603222

10. **Xenbase: a genomic, epigenomic and transcriptomic model organism database**
Kamran Karimi, Joshua D Fortriede, Vaneet S Lotay, Kevin A Burns, Dong Zhou Wang, Malcom E Fisher, Troy J Pells, Christina James-Zorn, Ying Wang, V G Ponferrada, … Peter D Vize
Nucleic Acids Research (2018-01-04) https://doi.org/ghk99d
DOI: 10.1093/nar/gkx936 · PMID: 29059324 · PMCID: PMC5753396

11. **A comprehensive evaluation of ensembl, RefSeq, and UCSC annotations in the context of RNA-seq read mapping and gene quantification**
Shanrong Zhao, Baohong Zhang

BMC Genomics (2015-02-18) https://doi.org/10.1186/s12864-015-1308-8
DOI: 10.1186/s12864-015-1308-8

12. **kblin/ncbi-genome-download**
Kai Blin
(2021-03-24) https://github.com/kblin/ncbi-genome-download

13. **ucsc-genomes-downloader: Python package to quickly download genomes from the UCSC.**
Luca Cappelletti
https://github.com/LucaCappelletti94/ucsc_genomes_downloader

14. **iGenomes** https://support.illumina.com/sequencing/sequencing_software/igenome.html

15. **Validate User** https://academic.oup.com/crawlprevention/governor?content=%2fgigascience%2farticle%2fdoi%2f10.1093%2fgigascience%2fgiz149%2f5717403

16. **Go Get Data (GGD): simple, reproducible access to scientific data**
Michael J. Cormier, Jonathan R. Belyeu, Brent S. Pedersen, Joseph Brown, Johannes Koster, Aaron R. Quinlan
bioRxiv (2020-09-11) https://www.biorxiv.org/content/10.1101/2020.09.10.291377v2
DOI: 10.1101/2020.09.10.291377

17. **RepeatMasker Home Page** http://repeatmasker.org/

18. **A Fast and Symmetric DUST Implementation to Mask Low-Complexity DNA Sequences**
Aleksandr Morgulis, E. Michael Gertz, Alejandro A. Schäffer, Richa Agarwala
Journal of Computational Biology (2006-06) https://doi.org/fqs85g
DOI: 10.1089/cmb.2006.13.1028 · PMID: 16796549

19. **Efficient "pythonic" access to FASTA files using pyfaidx**
Matthew D. Shirley, Zhaorong Ma, Brent S. Pedersen, Sarah J. Wheelan
PeerJ PrePrints (2015-04-08) https://peerj.com/preprints/970

20. **The ENCODE Blacklist: Identification of Problematic Regions of the Genome**
Haley M. Amemiya, Anshul Kundaje, Alan P. Boyle
Scientific Reports (2019-06-27) https://www.nature.com/articles/s41598-019-45839-z
DOI: 10.1038/s41598-019-45839-z

21. **Fast gapped-read alignment with Bowtie 2**
Ben Langmead, Steven L. Salzberg
Nature Methods (2012-04) https://www.nature.com/articles/nmeth.1923
DOI: 10.1038/nmeth.1923

22. **Validate User** https://academic.oup.com/crawlprevention/governor?content=%2fbioinformatics%2farticle-lookup%2fdoi%2f10.1093%2fbioinformatics%2fbtp324

23. **Validate User** https://academic.oup.com/crawlprevention/governor?content=%2fbioinformatics%2farticle-lookup%2fdoi%2f10.1093%2fbioinformatics%2fbti310

24. **Validate User** https://academic.oup.com/crawlprevention/governor?content=%2fbioinformatics%2farticle%2f34%2f18%2f3094%2f4994778

25. **Validate User** https://academic.oup.com/crawlprevention/governor?content=%2fbioinformatics%2farticle-lookup%2fdoi%2f10.1093%2fbioinformatics%2fbts635

26. **HISAT: a fast spliced aligner with low memory requirements**
Daehwan Kim, Ben Langmead, Steven L. Salzberg
Nature Methods (2015-04) https://www.nature.com/articles/nmeth.3317
DOI: 10.1038/nmeth.3317