



# genomepy: genes and genomes made easy

This manuscript ([permalink](#)) was automatically generated from [vanheeringen-lab/genomepy\\_manuscript@b31db1c](#) on December 23, 2021.

## Authors

---

- **Siebre 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

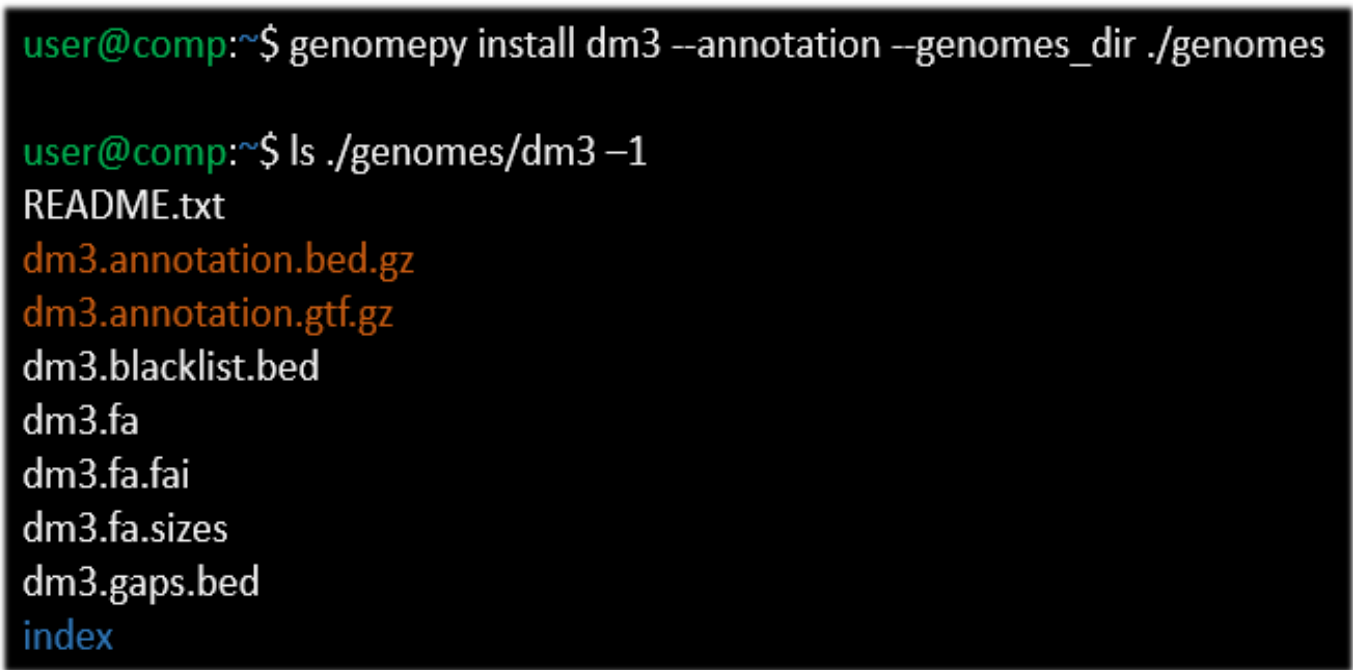
# Abstract

---

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.

A terminal window with a dark background and light green text. The first command is 'genomepy install dm3 --annotation --genomes\_dir ./genomes'. The second command is 'ls ./genomes/dm3 -1'. The output lists several files: README.txt, dm3.annotation.bed.gz, dm3.annotation.gtf.gz, dm3.blacklist.bed, dm3.fa, dm3.fa.fai, dm3.fa.sizes, dm3.gaps.bed, and index.

```
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

---

Data analysis is increasingly important in biological research. Whether you are analysing gene expression in two samples or protein binding in genome atlases, you will need external information such as a reference genome or a gene annotation. For these types of data, there are three major providers: Ensembl [1], UCSC [2] and NCBI [3], and many model-system specific providers, including GENCODE [4], ZFIN [5], FlyBase [6], WormBase [7], Xenbase [8] and more. Providers have different approaches to compiling genome assemblies and gene annotations, which effect formats, format compliance, naming, data quality, available versions and release cycle. These differences significantly impact compatibility with research [9], tools and (data based on) other genomic data.

You could try to find genomic data yourself, but there are many options with no clear metric for the “best”. Ensembl, UCSC and NCBI each have FTP archives, web portals, and REST APIs to search their individual databases. Alternatively, there are several tools to access some of these databases programmatically, such as [ncbi-genome-download \[10\]](#) and [ucsc-genomes-downloader \[11\]](#). However, none of these can search, compare or download from all major genome providers data. Furthermore, downloading and processing genomic data manually can be tedious, error-prone, and poorly reproducible. Although the latter could be remedied by a data management tool, such as [iGenomes \[12\]](#), [refGenie \[13\]](#) or [Go Get Data \[14\]](#), data managers still require the user to supply new data manually.

We have developed `genomepy` to 1) find genomic data on major providers, 2) compare gene annotations, 3) select the genomic data best suited for your analysis and 4) provide a suite of functions to peruse and manipulate the data. Selected data can be downloaded from anywhere, and is processed automatically. Sources and processing steps are documents to ensure reproducibility, and could be combined with data managers to greater effect. Genomic data can be loaded into `genomepy`, which can utilize and extend upon packages including [pyfaidx \[15\]](#), [pandas \[16\]](#) and [MyGene.info \[17\]](#) to rapidly work with gene and genome sequences and metadata. Similarly, `genomepy` has been incorporated into other packages, such as [pybedtools \[18\]](#) and [CellOracle \[19\]](#). `Genomepy` can be used on command line and via its fully documented Python API, for a one-time analysis or integration in pipelines and workflow managers such as [Nextflow \[20\]](#), [Galaxy \[21\]](#) or [Snakemake \[22\]](#).

## genomepy

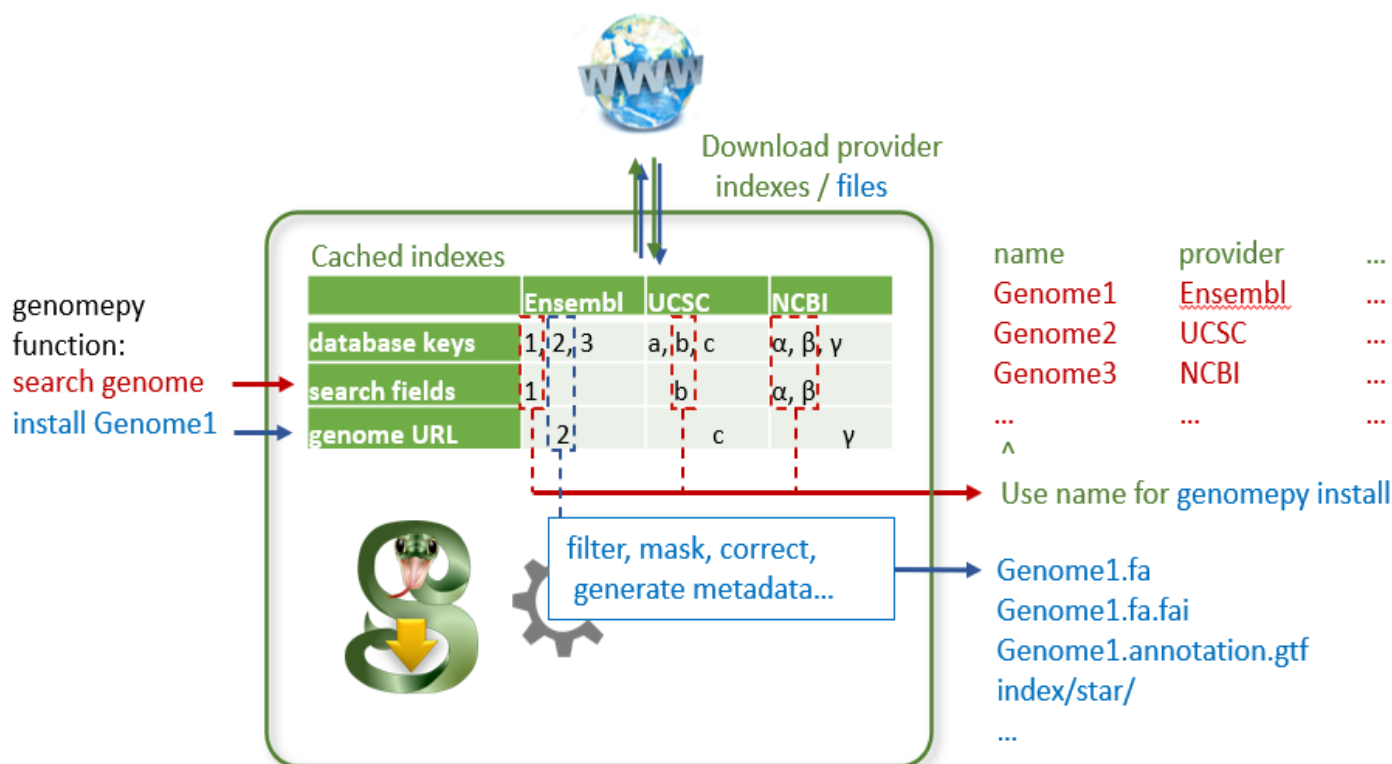
---

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 [\[23,24\]](#). 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 \[15\]](#), 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 2:** 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 [25] for the supported genomes. Other plugins support the generation of aligner indexes, including DNA aligner indexes for Bowtie2 [26], BWA [27], GMAP [28] or Minimap2 [29], and splice-aware aligners such as STAR [30] and HISAT2 [31].

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. <https://academic.oup.com/nar/advance-article/doi/10.1093/nar/gkz966/5613682>
2. **The Human Genome Browser at UCSC**  
WJames Kent, Charles W Sugnet, Terrence S Furey, Krishna M Roskin, Tom H Pringle, Alan M Zahler, and David Haussler  
*Genome Research* (2002-06-01) <https://genome.cshlp.org/content/12/6/996>  
DOI: [10.1101/gr.229102](https://doi.org/10.1101/gr.229102)
3. **The UCSC Table Browser data retrieval tool**  
Donna Karolchik, Angela S Hinrichs, Terrence S Furey, Krishna M Roskin, Charles W Sugnet, David Haussler, WJames Kent  
*Nucleic acids research* (2004-01-01) <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC308837/>  
DOI: [10.1093/nar/gkh103](https://doi.org/10.1093/nar/gkh103) · PMID: [14681465](https://pubmed.ncbi.nlm.nih.gov/14681465/) · PMCID: [PMC308837](https://pubmed.ncbi.nlm.nih.gov/PMC308837/)
4. <https://academic.oup.com/nar/article/49/D1/D916/6018430>
5. <https://academic.oup.com/nar/article/47/D1/D867/5165261>
6. <https://academic.oup.com/nar/article/47/D1/D759/5144957>
7. <https://academic.oup.com/nar/advance-article/doi/10.1093/nar/gkz920/5603222>
8. <https://academic.oup.com/nar/article/46/D1/D861/4559118>
9. **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](https://doi.org/10.1186/s12864-015-1308-8)
10. **GitHub - kblin/ncbi-genome-download: Scripts to download genomes from the NCBI FTP servers**  
GitHub  
<https://github.com/kblin/ncbi-genome-download>
11. **ucsc-genomes-downloader: Python package to quickly download genomes from the UCSC.**  
Luca Cappelletti  
[https://github.com/LucaCappelletti94/ucsc\\_genomes\\_downloader](https://github.com/LucaCappelletti94/ucsc_genomes_downloader)
12. **iGenomes** [https://support.illumina.com/sequencing/sequencing\\_software/igenome.html](https://support.illumina.com/sequencing/sequencing_software/igenome.html)
13. <https://academic.oup.com/gigascience/article/doi/10.1093/gigascience/giz149/5717403>
14. **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  
(2020-09-11) <https://www.biorxiv.org/content/10.1101/2020.09.10.291377v2>
15. **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>

16. **pandas-dev/pandas: Pandas 1.3.5**  
Jeff Reback, jbrockmendel, Wes McKinney, Joris Van den Bossche, Tom Augspurger, Phillip Cloud, Simon Hawkins, Matthew Roeschke, gfyong, Sinhrks, ... Skipper Seabold  
*Zenodo* (2021-12-12) <https://zenodo.org/record/5774815>
17. **High-performance web services for querying gene and variant annotation**  
Jiwen Xin, Adam Mark, Cyrus Afrasiabi, Ginger Tsueng, Moritz Juchler, Nikhil Gopal, Gregory S Stupp, Timothy E Putman, Benjamin J Ainscough, Obi L Griffith, ... Chunlei Wu  
*Genome Biology* (2016-05-06) <https://doi.org/10.1186/s13059-016-0953-9>  
DOI: [10.1186/s13059-016-0953-9](https://doi.org/10.1186/s13059-016-0953-9)
18. <https://academic.oup.com/bioinformatics/article-lookup/doi/10.1093/bioinformatics/btr539>
19. **CellOracle: Dissecting cell identity via network inference and in silico gene perturbation**  
Kenji Kamimoto, Christy M Hoffmann, Samantha A Morris  
(2020-04-21) <https://www.biorxiv.org/content/10.1101/2020.02.17.947416v3>
20. **Nextflow enables reproducible computational workflows**  
Paolo Di Tommaso, Maria Chatzou, Evan W Floden, Pablo Prieto Barja, Emilio Palumbo, Cedric Notredame  
*Nature Biotechnology* (2017-04) <https://www.nature.com/articles/nbt.3820>  
DOI: [10.1038/nbt.3820](https://doi.org/10.1038/nbt.3820)
21. <https://academic.oup.com/nar/article/46/W1/W537/5001157>
22. **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://f1000research.com/articles/10-33>
23. **RepeatMasker Home Page** <http://repeatmasker.org/>
24. **A Fast and Symmetric DUST Implementation to Mask Low-Complexity DNA Sequences**  
Aleksandr Morgulis, EMichael Gertz, Alejandro A Schäffer, Richa Agarwala  
*Journal of Computational Biology* (2006-06) <https://doi.org/fqs85g>  
DOI: [10.1089/cmb.2006.13.1028](https://doi.org/10.1089/cmb.2006.13.1028) · PMID: [16796549](https://pubmed.ncbi.nlm.nih.gov/16796549/)
25. **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](https://doi.org/10.1038/s41598-019-45839-z)
26. **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](https://doi.org/10.1038/nmeth.1923)
27. <https://academic.oup.com/bioinformatics/article-lookup/doi/10.1093/bioinformatics/btp324>
28. <https://academic.oup.com/craw/prevention/governor?content=%2fbioinformatics%2farticle-lookup%2fdoi%2f10.1093%2fbioinformatics%2fbti310>
29. <https://academic.oup.com/bioinformatics/article/34/18/3094/4994778>
30. <https://academic.oup.com/bioinformatics/article-lookup/doi/10.1093/bioinformatics/bts635>

31. **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](https://doi.org/10.1038/nmeth.3317)