Jumpstart your genomics pipelines with genomepy

This manuscript (permalink) was automatically generated from vanheeringen-lab/genomepy_manuscript@3707ceb 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 genomics data, such as ATAC, ChIP and RNA-sequencing, requires multiple types of genomic data, such as genome sequence and gene annotations. These resources can generally be retrieved from multiple organizations, where they exist at multiple versions, and may have been generated with varying methods. Which set of genomic data to use depends on the context of the research, such as collaboration partners, data reuse or the quality of the genomic data. Many of the bioinformatic workflows and pipelines available to date require the user to make this informed decision and supply the genomic data manually. Obtaining this data can be a tedious and error-prone process and does not allow for full computational reproducibility.

Here we present genomepy, a quality-of-life enhancement tool, that can navigate the genome databases of Ensembl, UCSC and NCBI. Genomepy can search and install genome sequences and gene annotation data from these providers 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 downloads a specified genome with sensible defaults, while providing full control to advanced features. Additionally, gene annotations can be downloaded and converted to commonly used formats, with built-in checks for compatibility with the genome. Genomepy can optionally create genome indexes for commonly used aligners, including splice-aware aligners utilizing both genome and gene annotations. Genomes and gene annotations not available on supported databases can be processed by genomepy as well, providing a consistent workflow with any genome.

Genomepy provides this functionality and more via command line interface and Python application programming interface, aimed at easy of use and integration in automated pipelines.

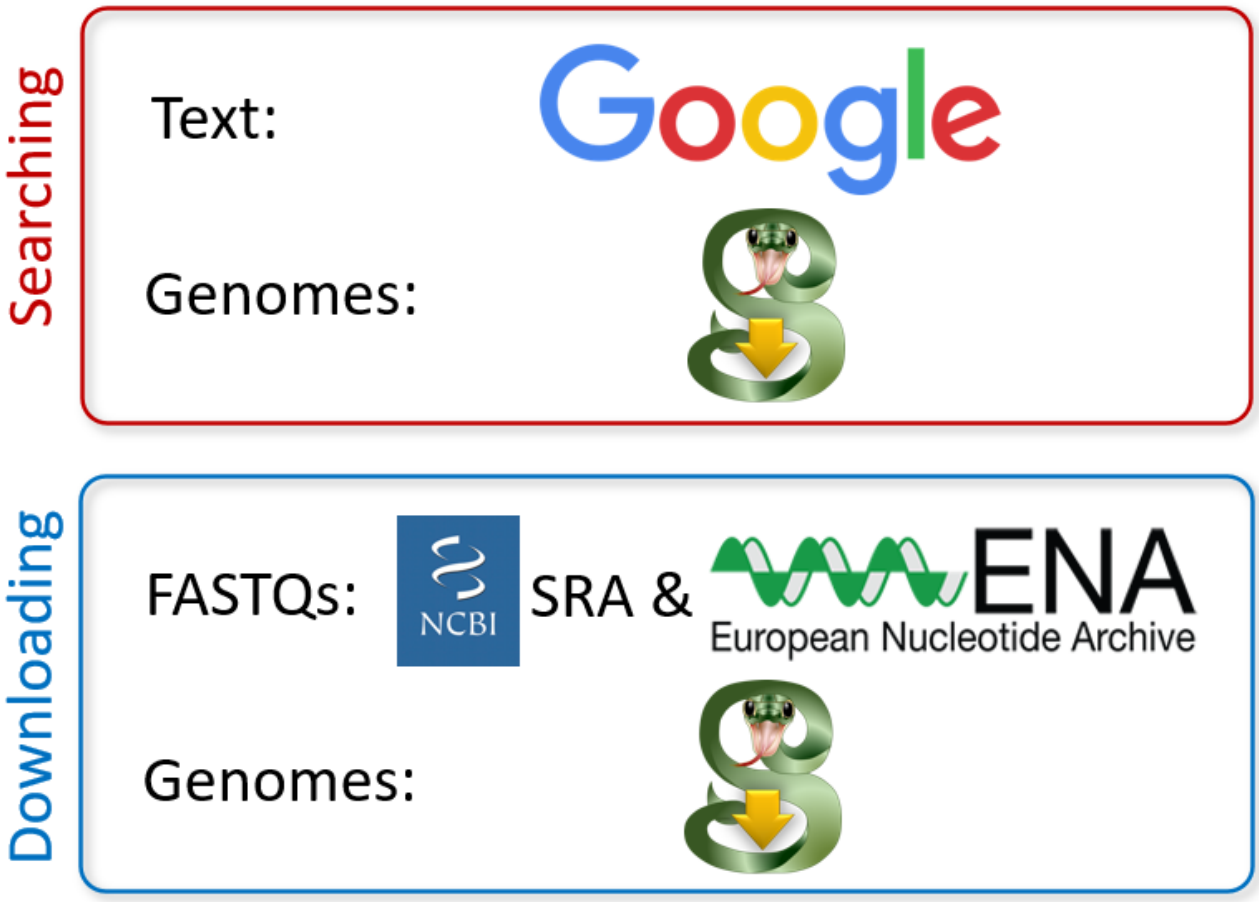

Figure 1: genomepy in a nutshell

Introduction

As high-throughput sequencing matured over the past decade and a half, the size and amount of sequencing data has exploded. Over the past five years, the number of datasets published on the NCBI GEO database increased by an average of 2000 per year, while the number of samples increased by 100.000 per year [1]. This explosion of data highlights the need for scalable, robust and automatable methods for data processing and analysis.

A large amount of bioinformatics tools have been created to facilitate this process, including (pre)processing tools, analysis tools and workflow managers to link these tools together (such as the snakemake wrapper [2]). However, not all input data can be obtained fully automatic. 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. These include three major genome providers, Ensembl [3], UCSC [4] and NCBI [5], and many niche providers, such as flybase [6], wormbase [7] or xenbase [8]. 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 impact the compatibility of the reference data with research tools [9], other reference databases, and other research. Therefore, the choice of provider and reference data is of significant importance.

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

and processed the output for downstream use. These methods are available via the command line interface and Python application programming interface for easy use and incorporation in automated workflows.

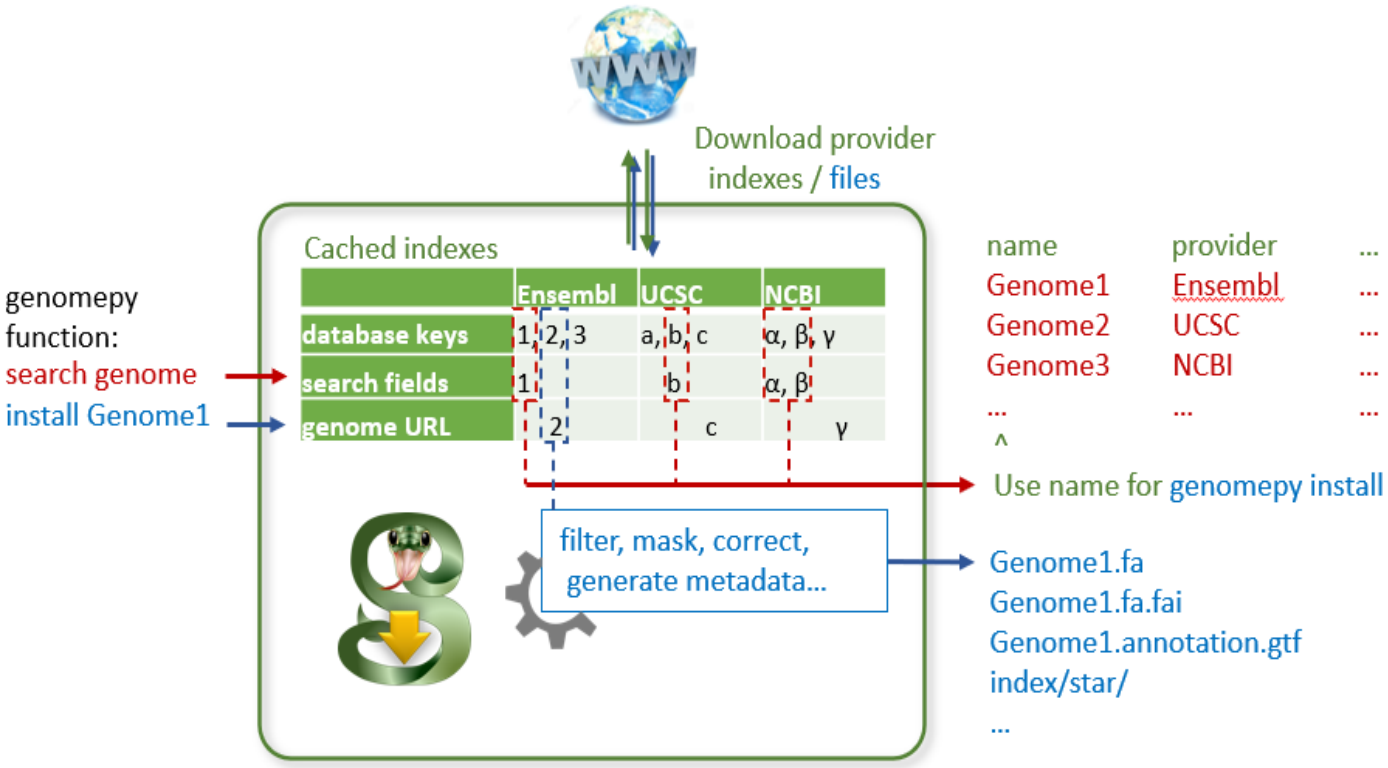

Figure 2: genomepy workflow

```
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 3: executive overview

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 [10] and ucsc-genomes-downloader [11]. 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. These data management tools accept reference data and derived assets such as aligner indexes from any source, such as iGenomes [12], refGenie [13] and Go Get Data [14]. 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 (such as non-model organisms), new assembly version (such as the latest patch of the human genome) or in some cases assets (such as an aligner index not present in the hosted data).

We conclude that there is a need for a tool that can assist in obtaining and preparing genomic data for downstream analysis. This includes downloading the data, error checking (and if possible correcting) genome and gene annotations, and outputting data types commonly used in bioinformatics tools. Genomepy was created to fit this need, while also providing an overview of available choices, and a framework that facilitates the downstream use of the data.

genomepy

The core functionality of genomepy is to search, download and prepare genomes and gene annotations. These functions are split over two command line functions: `genomepy search` and `genomepy install`.

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

When an assembly has been selected, the name (as returned from the search function) can be passed to the install function. The files will be downloaded from the major provider that hosts it. Users can optionally specify a 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 [15,16]. By default soft masked genomes (repetitive sequences written in lower case) are downloaded, but unmasked or hard masked (repetitive sequences written as Ns) can be obtained as well. If the provider does not have the genome at this masking level, genomepy will edit the FASTA to match.

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. For instance to filter out (or filter for) chromosomes, unplaced-, unknown- or mitochondrial sequences.

Once processing is performed, genomepy generates several commonly used support files. The genome is indexed using pyfaidx [17], and contig sizes and contig gap sizes are stored in separate files.

If specified, genomepy will attempt to download a gene annotation: If available, the annotation is processed to the commonly used GTF and BED output formats. Afterward, contig names between the genome and gene annotation are checked for compatibility. Should these mismatch, genomepy will attempt to match the names in the annotations to those in the genome.

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

Genomepy facilitates several optional processing steps via plugins. Using the command line, these options can be inspected and toggled with the `genomepy plugin` function.

The blacklist plugin downloads blacklists by the Kundaje lab [18] for the supported UCSC genomes (and GRCh38). The other plugins currently all support the generation of aligner indexes. These include DNA aligner indexes for Bowtie2 [19], BWA [20], GMAP [21] or Minimap2 [22], and splice-aware aligners such as STAR [23] and HISAT2 [24].

Once the install function has run with one or more aligner plugins, the genomic data is ready for alignment and analysis.

Conclusion

Obtaining suitable genomic data is a principal step in any genomics project. Genomepy was developed to provide a consistent overview of genomes on the three major providers, and reproducibly download and process genomic data ready for downstream use. Using genomepy, a project can utilize genomes from any provider and expect consistent output with data from a major provider or otherwise. This allows genomepy to automate a step in genomic data preprocessing that was required to be performed by hand. Additionally, it facilitates downstream analysis, by setting up paths to the genome FASTA file, providing genome metadata within the Python Genome object and generating support files. Combined, these features make genomepy ideal for integration in automated sequencing workflows, as demonstrated in seq2science [25], paving the way for robust and reproducible analysis.

Code availability

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

References

1. **NCBI GEO: archive for functional genomics data sets–update**
   Tanya Barrett, Stephen E Wilhite, Pierre Ledoux, Carlos Evangelista, Irene F Kim, Maxim Tomashevsky, Kimberly A Marshall, Katherine H Phillippy, Patti M Sherman, Michelle Holko, … Alexandra Soboleva
   Nucleic acids research (2013-01) https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3531084/
   DOI: 10.1093/nar/gks1193 · PMID: 23193258 · PMCID: PMC3531084

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

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

4. **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

5. **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

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

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

8. **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

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

10. **kblin/ncbi-genome-download**
    Kai Blin
    (2021-03-24) 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

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

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

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
bioRxiv (2020-09-11) https://www.biorxiv.org/content/10.1101/2020.09.10.291377v2
DOI: 10.1101/2020.09.10.291377

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

16. **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

17. **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

18. **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

19. **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

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

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

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

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

24. **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

25. **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