Jumpstart your genomics pipelines with genomepy

This manuscript (permalink) was automatically generated from vanheeringen-lab/genomepy_manuscript@07ed204 on November 25, 2020.

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

Summary

Analyzing genomics data, including RNA-, ATAC- and ChIP-sequencing, requires multiple types of support 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 data to use depends on the context of the research, such as collaboration partners, data reuse or data quality. Many of the bioinformatic workflows and pipelines available to date require the user to make this informed decision and supply the support data. 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.

Availability and implementation

Genomepy can be installed using Bioconda and Pip, and the code is available at https://github.com/vanheeringen-lab/genomepy.

Introduction

As high-throughput sequencing matured over the past decade and a half, the size and amount of genomic data hes 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.

Much of the genomics analysis preprocessing has already been made automatable, with multiple tools providing standardized output formats. One such step is the alignment of sequence read data to a genome, such as Bowtie2 [2], BWA [3], GMAP [4] or Minimap2 [5], and splice-aware aligners such as STAR [6] and HISAT2 [7]. Several steps, mainly including alignment, require additional input to the sequence read data in the form of a reference genome, and in the case of splice-aware aligners, gene annotations. These data can be obtained from a variety of different sources. There are the three major genome providers, which act as a general hub for reference data: Ensembl [8], UCSC [9] and NCBI [10].

Ensembl uses their in-house workflow to add or update genomes and gene annotations in a three-month production cycle 11. As a results, Ensembl provides detailed gene annotations on mature genome assemblies that update infrequently. One downside to Ensembl data is their chromsome naming scheme (e.g. “1” for chromosome 1) which clashes with several common bioinformatical tools.

UCSC hosts and maintains a modest set of reference genomes. Gene annotations for these genomes are generated through a variety of methods, including the Ensembl and NCBI workflow, as well as their own. However, not every version of these gene annotations conforms to the output format.

The NCBI database accepts submissions from the Genome Reference Consortium as well as individuals. In addition to reference assemblies by the reference consortium, uploads by individual groups often provide a trove of different strains per species. For instance, 946 different strains of Homo sapiens and 848 strains of Saccharomyces cerevisiae are available from NCBI. As a result of the open submission system, NCBI updates frequently, and often provides the latest version of an assembly before the other providers do. However, as an upload may contain either genome assembly, gene annotation, or both, the assembly data can be incomplete, and of varying levels of maturity.

In addition to the three major providers, there are many species specific providers, such as flybase 12, wormbase 13 or xenbase 14.

Table 1: Available genome assemblies per provider. Estimated by querying the provider REST API (assembly summaries for NCBI) for all unique assembly names. Ensembl genomes are excluding 43778 bacteria genomes not available programmatically.

Provider	Assemblies
UCSC	213
Ensembl	1741
NCBI	878821

Each provider has a different method of generating genome assemblies and gene annotations, which can affect available output (formats), naming schema, information density, as well as availability (see table 1), accessibility (archive and retrieval methods) and relevance (update frequency). These differences impact the compatibility of the reference data with research tools 15, other reference databases and other research. Therefore, the choice of provider and reference data is significant importance. To make an informed decision requires an overview of available options. To get this overview, one could check each website separately, then download and process the data manually. Such a method creates room for human error in the finding, processing and remembering these steps as well as the reasoning behind them. For the sake of sanity and reproducibility, it would be better if that could be done in a standardized system.

To this end we developed genomepy. Genopmepy is a tool with both command line interface and Python application programming interface, which can be called to search one or all three providers at once. Using the search function one can get an overview of all genomes containing the search term in their name, description or accession identifier, as well as all genomes matching a taxonomy identifier. Once a selection is made the genome and gene annotations can be downloaded and prepared for use with the install function. This includes automatic preparation for aligners (genome indexing with pyfaidx 16, generating support files (chromosome sizes and gaps), matching chromosome names between genome and gene annotation and optional aligner index generation). Which of these features is used is automatic logged for reproducibility. Because of the multiple interfaces, genomepy can be used in workflows to automate these steps.

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 [17] and ucsc-genomes-downloader [18]. However, to our knowledge no tool exists that can 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 assest such as aligner indexes from any source, such as iGenomes [19], refGenie [20] and Go Get Data [21]. These tools excel in their ability to reproducibly share data, a feature which is not present in genomepy, may therefore be used to obtain 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 path to the human genome) or asset (such as an aligner index not present in the hosted data). For these situations, data management tools would be an excellent extension to genomepy.

In several cases the reference data may not be ready for direct downstream use. For instance, many assemblies do not contain gene annotations in the correct format for splice-aware aligners. Furthermore, many gene annotations have contig (chromosomes and scaffolds) names that do not match the names in the reference genome. Additional steps, including compatibility checks and potentially processing, are required. Many tools exist to perform these actions, most noteably the UCSC gene annotation conversion tools. However, it should not bear mentioning that and automated checklist but would be more efficient that a manual one.

We conclude that there is a need for a tool that can provide an overview of the choices of reference data available, can obtain the specified data, and perform the processing required to utilize the data downstream. Genomepy was created to fit this need, and does so for both automated and human-supervised workflows.

genomepy

The core functionalities of genomepy are `search` and `install`.

search

Search will query a provider (or all three if none is specified) for the given search term. The search term normalized for case and whitspace, and input types is identified. For taxonomy identifiers, all assemblies with matching IDs are returned. For accession identifiers and text terms, all assemblies containing the term in their respective field are returned. Additionally text terms found in any other descriptive fields are also returned.

install

When an assembly has been selected, the name can be passed to the install function. This function downloads the genome assembly with soft masking, unless different masking is specified. The assembly is then filtered to exclude alternative regions (unless specified otherwise) and the presence or absence of any specified (regex search) term. The genome is then indexed using pyfaidx [16], and contig sizes and gaps are stored in separate files.

If specified, genomepy will attempt to download a gene annotation: genomepy will search the database for a GFF, GTF, BED or (for UCSC only) text format gene annotation. The annotation is then processed to output a consistent GTF and BED format gene annotation using the UCSC conversion tools 22/. If the genome was downloaded previously, the contig names are checked for compatibility, and matched to those in the genome if required and possible.

external providers

External provider often contain novel or more recent assemblies of organisms in their specialized field. These assemblies may be processed similarly by providing the direct link to the genome and/or gene annotation in the genomepy install command and specifying 'url' as provider.

plugins

Using the `plugin` function, the generation of aligner indexes can be toggled. The indexes will automatically generate upon the completion of the install function.

logging

Download sources, data and time, processing steps and requested filters are all logged in a README file which is stored in the same directory, and updated when further processing is performed with genomepy.

old

search, download, sensible defaults, reproducible, automatable. about those defaults...

Install via conda, pip or git.

basic steps in CLI

Repeat steps in API Extended steps, link to seq2science implementation?

Conclusions

Research is about making informed decisions. Choosing a reference assembly is no different. Genomepy offers an overview of the three largest genome providers, making this choice easier.

After choosing an assembly, data must be downloaded and processed for compatibility with downstream tools. Genomepy provides this functionality, while providing logging. Even if the required reference data is not available on the three largest genome providers, genomepy can process external data to provide a consistent output.

While genomepy makes choices during the processing, each of these can be tuned to the specific needs of a project using the CLI. More control can be achieved via the Python API. Combined, these features make genomepy ideal for integration in automated sequencing workflows, as demonstrated in seq2science 23.

Acknowledgements

We thank the Department of Molecular (Developmental) Biology, out github contributors, and issue posters for their patience, feedback and insight. We thank black, pytest, CodeCoverage and TravisCI for enduring our abuse and teaching us patience. And finally, we thank Manubot 24 for assisting with this manuscript.

Code availability

Genomepy can be installed using Bioconda and Pip. The code is available at https://github.com/vanheeringen-lab/genomepy.

Supplementary Information

The full genomepy documentation including examples can be viewed here

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. **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

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

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

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

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

7. **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

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

9. **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

10. **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

11. **Release Cycle** https://ensembl.org/info/about/release_cycle.html

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

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

14. **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

15. **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

16. **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

17. **kblin/ncbi-genome-download**
Kai Blin
(2020-11-19) https://github.com/kblin/ncbi-genome-download

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

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

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

21. **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

22. **Index of /admin/exe** http://hgdownload.cse.ucsc.edu/admin/exe/

23. **seq2science**
Maarten Van Der Sande, Siebren Frölich, Jos Smits, Simon Van Heeringen
Zenodo (2020-11-05) https://doi.org/ghktg7
DOI: 10.5281/zenodo.3921913

24. **Open collaborative writing with Manubot**
Daniel S. Himmelstein, Vincent Rubinetti, David R. Slochower, Dongbo Hu, Venkat S. Malladi, Casey S. Greene, Anthony Gitter
PLOS Computational Biology (2019-06-24) https://doi.org/c7np
DOI: 10.1371/journal.pcbi.1007128 · PMID: 31233491 · PMCID: PMC6611653