Jumpstart your genomics pipelines with genomepy

This manuscript (permalink) was automatically generated from vanheeringen-lab/genomepy_manuscript@49d939f on November 24, 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 is available at https://github.com/vanheeringen-lab/genomepy.

Introduction

High-throughput sequencing is common practice (since/data volumes) automation required to process the large volumes consistently This is possible because of similarities in the sequencing methods and standardized formats Aligners map the read data to a reference genome or transcriptome. Both aligners and other steps in the various sequencing data analyses require as input a reference genome and gene annotation three major genome providers: Ensembl, UCSC and NCBI Ensembl: UCSC: NCBI: In addition, there are many species specific providers, such as flybase, wormbase, xenbase.

Each provider has a separate method of generating their reference genomes and gene annotations, which can affect data format, naming and density, as well as database versioning and update frequency. These differences impact the compatibility of the reference data with research tools 1, other reference databases and other research. Therefore, the choice of reference data is significant importance. To make this decision on an informed bases, you need an overview of the options. To get this overview, one could check each website, download the desired data, process and log these steps

manually. For the sake of sanity and reproducibility, it would be better if that could be done in a standardized system.

Here we present genomepy, which can do that. Search one or all 3 providers Install genome and gene annotations Automatic preparation for aligners (genome indexing with pyfaidx [**???**], generating support files (chromosome sizes and gaps), matching chromosome names between genome and gene annotation and optional aligner index generation) Automatic logging for reproducibility CLI and Python API can be used to automate this step in workflows.

Related Work

Ensembl, UCSC and NCBI support downloading from their databases via accessible FTP archives, web portals, and REST APIs. External tools have been developed to programmatically download from one or several databases, such as the ncbi-genome-download tool 2, and packages such as ucsc-genomes-downloader 3/ (Python) and metaseqR 4 (R). However, to our knowledge no tool exists that can search or download from all three major genome providers.

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/scaffolds) names that do not match the names in the reference genome. Additional processing steps are required to correct these issues. Tools exist to address some of these issues, but would be more effective when used in conjunction.

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 5, 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 6/. 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 7.

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 8 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.

References

1. **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

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

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

4. **Systematic integration of RNA-Seq statistical algorithms for accurate detection of differential gene expression patterns**
   Panagiotis Moulos, Pantelis Hatzis
   Nucleic Acids Research (2015-02-27) https://academic.oup.com/nar/article/43/4/e25/2411004
   DOI: 10.1093/nar/gku1273

5. **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

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

7. **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

8. **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