Outline of plyranges a grammar of genomic data manipulation

Overview of structure, references and layout for the plyranges package paper.

Abstract

The Bioconductor project has created many powerful abstractions for reasoning about genomics data, such as the *Ranges* data structures for representing genomic intervals. By recognising that these data structures follow 'tidy' data principles we have created a grammar of genomic data manipulation that defines verbs for performing actions on and between genomic interval data. This grammar simplifies performing common genomic data analysis tasks via method chaining, type consistency and results in creating human readable pipelines. We have implemented this grammar as an Bioconductor/R package plyranges.

Introduction

Putting the work into context.

What do we mean by genomic data?

- How does that relate to 'tidy' data?
- Why tidy data is a useful abstraction.
- Why *Ranges* follow this pattern.

Key references: Wickham (2014), Lawrence et al. (2013)

How can we analyse genomic data?

- the Ranges data structures
- other abstractions for analysing genomic data
  - the query language approaches
  - the bedtools approach
  - relational algebras

Key references: Quinlan and Hall (2010), Dale, Pedersen, and Quinlan (2011), Kaitoua et al. (2017), Ovaska et al. (2013), Mungall (2014) Kozanitis and Patterson (2016)

What does plyranges contribute?

`plyranges` API provides

- a consistent grammar for analysing genomic data.
- fluent interface enables readable pipelines
- reproducibility via code clarity, declaraitve programming, and return consistency

Key references:

Design and Implementation

Discussion of the API and its key features

- every action on a Range is a verb (functional composition, method chaining)
- fluent (human readable, code describes what to do rather than 'how')
- every function returns a class familiar to the user
- an expressive algebra for performing arithmetic (anchoring functions)
- split-apply-combine strategies (group_by + reduce/summarise/mutate/filter)
- recasting overlapping/nearest etc as joins

Results

A case study showing plyranges makes life easier... Perhaps could use long reads data from Matt and his post-doc Charity (interesting use case of looking for intron-exon junctions)

Conclusion (future directions)

- availability and maintenance of package
- extending to SummarisedExperiment class
- natural fit with ggbio2

References

Dale, Ryan K, Brent S Pedersen, and Aaron R Quinlan. 2011. "Pybedtools: A Flexible Python Library for Manipulating Genomic Datasets and Annotations." *Bioinformatics* 27 (24): 3423–4. doi:10.1093/bioinformatics/btr539.

Kaitoua, A, P Pinoli, M Bertoni, and S Ceri. 2017. "Framework for Supporting Genomic Operations." *IEEE Trans. Comput.* 66 (3): 443–57. doi:10.1109/TC.2016.2603980.

Kozanitis, Christos, and David A Patterson. 2016. "GenAp: A Distributed SQL Interface for Genomic Data." *BMC Bioinformatics* 17 (February): 63. doi:10.1186/s12859-016-0904-1.

Lawrence, Michael, Wolfgang Huber, Hervé Pagès, Patrick Aboyoun, Marc Carlson, Robert Gentleman, Martin Morgan, and Vincent Carey. 2013. "Software for Computing and Annotating Genomic Ranges." *PLoS Comput. Biol.* 9. doi:10.1371/journal.pcbi.1003118.

Mungall, Christopher John. 2014. "Formalization of Genome Interval Relations." *bioRxiv*. doi:10.1101/006650.

Ovaska, Kristian, Lauri Lyly, Biswajyoti Sahu, Olli A Jänne, and Sampsa Hautaniemi. 2013. "Genomic Region Operation Kit for Flexible Processing of Deep Sequencing Data." *IEEE/ACM Trans. Comput. Biol. Bioinform.* 10 (1): 200–206. doi:10.1109/TCBB.2012.170.

Quinlan, Aaron R, and Ira M Hall. 2010. "BEDTools: A Flexible Suite of Utilities for Comparing Genomic Features." *Bioinformatics* 26 (6): 841–42. doi:10.1093/bioinformatics/btq033.

Wickham, Hadley. 2014. "Tidy Data." *Journal of Statistical Software, Articles* 59 (10): 1–23. doi:10.18637/jss.v059.i10.