GenomicDataCommons: a Bioconductor Interface to the NCI Genomic Data Commons

Martin T. Morgan*
Roswell Park Cancer Institute

Sean R. Davis†
Center for Cancer Research
National Cancer Institute
National Institutes of Health

March 15, 2017

Abstract

The National Cancer Institute (NCI) has established the Genomic Data Commons (Grossman et al. 2016, `https://gdc.cancer.gov/`). The GDC provides the cancer research community with an open and unified repository for sharing and accessing data across numerous cancer studies and projects via a high-performance data transfer and query infrastructure. The Bioconductor project (Huber et al. 2015) is an open source and open development software project built on the R statistical programming environment (R Core Team 2016). A major goal of the Bioconductor project is to facilitate the use, analysis, and comprehension of genomic data. The *GenomicDataCommons* Bioconductor package provides basic infrastructure for querying, accessing, and mining genomic datasets available from the GDC. We expect that the Bioconductor developer and the larger bioinformatics community will build on the *GenomicDataCommons* package to add higher-level functionality and expose cancer genomics data to the plethora of state-of-the-art bioinformatics methods available in Bioconductor.

Availability: `https://github.com/Bioconductor/GenomicDataCommons` (& soon in Bioconductor).

1 Introduction

Basic and translational cancer research projects–from large, multicenter consortia to individual labs–now often produce vast quantities of genomic data. Such datasets are ideally accompanied by extensive phenotypic and clinical information and valuable experimental metadata. To address the needs of the cancer community to access, query, and utilize these cancer-related data resources, the NCI has developed the Genomic Data Commons (GDC, Grossman et al. 2016). While similar in many respects to other large-scale genomics data repositories, the GDC differs in that data presented via the GDC have been "harmonized" using stardardized, publicly-available pipelines[1]. An additional and unusual new feature of the GDC relative to most genomic data repositories is that users can deposit and then apply GDC standardized pipelines to their own genomic data, provided they agree to share their data broadly (subject to data use agreements and participant consents).

The Bioconductor project is an open source and open development software project for the analysis and comprehension of genomic and molecular biology data (Huber et al. 2015). Based in the statistical programming language, R (R Core Team 2016), the project comprises 1296 interoperable software packages contributed by a diverse community of scientists. Bioconductor tools to access cancer genomics data, including RTCGA (Kosinski and Biecek 2016), TCGABiolinks (Colaprico et al. 2016), RTCGAToolbox (Samur 2014) facilitate access to cancer genomics data, but they are focused, by-and-large on TCGA data and in some cases rely on legacy data sharing platforms. The *GenomicDataCommons* software package provides

[*] martin.morgan@roswellpark.org
[†] seandavi@gmail.com, to whom correspondence should be addressed
[1] See `https://docs.gdc.cancer.gov/Data/Introduction/` for details.

infrastructure for finding, accessing, and downloading cancer genomics data from the NCI GDC. Our expectation is that the *GenomicDataCommons* package will facilitate use of GDC data by the larger Bioconductor community and increase the effiency of drawing biologically important conclusions from published cancer genomic data.

2 Design and Usage

The goal of the *GenomicDataCommons* Bioconductor package is to expose the GDC RESTful application programming interface (API[2]) via an R client interface. The *GenomicDataCommons* package supports both the metadata query capabilities and data download capabilities of the GDC API. The *GenomicDataCommons* package is implemented as a set of S3 classes and methods with accompanying utility functions.

2.1 Usage

Finding data in the GDC starts with constructing a query of available metadata. We model our R interface for querying GDC metadata on a now-common approach, using pipes to connect functional methods, similar in spirit to that used by packages such as dplyr (Wickham and Francois 2016). Querying the GDC API to get the unique file ids of all HTSeq quantified gene expression data from the TCGA-GBM project forms an illustrative example:

```r
library(GenomicDataCommons)
file_records = files() %>%
    filter(~ cases.project.project_id == 'TCGA-GBM' &
               data_type == 'Gene Expression Quantification' &
               analysis.workflow_type == 'HTSeq - Counts') %>%
    select(default_fields('files')) %>%
    response_all()
```

The `files()` call creates a `GDCquery`. The `filter()` verb allows R-like syntax to generate filter criteria (that are then translated to an R list); note that field names like "data_type" can be used as R names by capitalizing on so-called non-standard evaluation. The `select()` verb then sets the return fields, which, in this case are limited to the GDC-specified `default_fields`. Up to this point, no data have been retrieved from the GDC. The `response_all()` call performs the actual metadata retrieval, returning a `GDCResponse` object.

The GDC API also supports simple aggregation based on metadata fields, such as the number of cases summarized by GDC project.

```r
cases_by_project = cases() %>%
    facet('project.project_id') %>%
    aggregations()
head(cases_by_project)
```

2.2 Downloading data

Downloading single files or multiple small files is supported directly from R. For larger sets of files or single large files, we utilize the GDC data transfer tool and a "manifest" file. As an example, to create a manifest from the file records collected above, we simply apply the `manifest()` verb.

```r
manifest_df = file_records %>%
    manifest()
```

Once written to a file, the `transfer()` function will utilize the GDC transfer client (a separate command-line tool) to perform robust, performant download of the data.

[2] `https://docs.gdc.cancer.gov/API/Users_Guide/Getting_Started/`

2.3 BAM file slicing

The BAM file is the current standard for storing aligned reads against a reference. Many use cases for these alignments require only a particular genomic region rather than all alignments. BAM "slicing" is implemented as a very basic wrapper upon which developers can build functionality.

```
# get 10 RNA-seq BAM files
file_ids = files() %>%
    filter(~ data_format=='BAM' &
           experimental_strategy=='RNA-Seq') %>%
    response() %>%
    ids()
```

The file ids, all associated with aligned BAM files, can be used for region-based slicing.

```
# An access token from the GDC is needed here
# for the the controlled-access BAM files.
bamfile = slicing(file_ids[1], regions = 'chr17:74000000-76000000',
        token=gdc_token())
```

The resulting BAM file, with the file name `bamfile`, can be accessed using any of the standard Bioconductor tools for working with aligned reads in BAM format.

3 Conclusions

The GDC is the latest iteration of a genomic data repository from the NCI. While the GDC offers excellent web-based query interfaces and a growing number of web-based tools for interacting with data, many bioinformatics and genomic data scientists adhere to text-based analysis tools, version control, and principles of reproducible computational research (Sandve et al. 2013). The Bioconductor project enjoys a large, innovative community of such researchers and software developers, many of whom focus on cancer research. The reproducibility, interoperability, usability, sustainability, and continuous integration that are intrinsic to Bioconductor, combined with a plethora of existing tools, have produced a multiplier effect for software contributed to the project, making it an excellent environment for cancer genomic data science. The *GenomicDataCommons* package forms the foundation to build and rapidly iterate on innovative tools for the analysis and comprehension of cancer genomics datasets available at the NCI GDC.

References

[1] Antonio Colaprico, Tiago C. Silva, Catharina Olsen, Luciano Garofano, Claudia Cava, Davide Garolini, Thais S. Sabedot, Tathiane M. Malta, Stefano M. Pagnotta, Isabella Castiglioni, Michele Ceccarelli, Gianluca Bontempi, and Houtan Noushmehr. "¡i¿TCGAbiolinks¡/i¿ : an R/Bioconductor package for integrative analysis of TCGA data". In: *Nucleic Acids Research* 44.8 (May 2016), e71–e71. ISSN: 0305-1048. DOI: `10.1093/nar/gkv1507`. URL: `https://academic.oup.com/nar/article-lookup/doi/10.1093/nar/gkv1507`.

[2] Robert L. Grossman, Allison P. Heath, Vincent Ferretti, Harold E. Varmus, Douglas R. Lowy, Warren A. Kibbe, and Louis M. Staudt. "Toward a Shared Vision for Cancer Genomic Data". In: *New England Journal of Medicine* 375.12 (Sept. 2016), pp. 1109–1112. ISSN: 0028-4793. DOI: `10.1056/NEJMp1607591`. URL: `http://www.ncbi.nlm.nih.gov/pubmed/27653561%20http://www.nejm.org/doi/10.1056/NEJMp1607591`.

[3] Marcin Kosinski and Przemyslaw Biecek. *RTCGA: The Cancer Genome Atlas Data Integration*. 2016. URL: `https://rtcga.github.io/RTCGA`.

[4] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria, 2016. URL: `https://www.r-project.org/`.

[5] Hadley Wickham and Romain Francois. *dplyr: A Grammar of Data Manipulation.* R package version 0.5.0. 2016. URL: `https://CRAN.R-project.org/package=dplyr`.

[6] Wolfgang Huber, Vincent J Carey, Robert Gentleman, Simon Anders, Marc Carlson, Benilton S Carvalho, Hector Corrada Bravo, Sean Davis, Laurent Gatto, Thomas Girke, Raphael Gottardo, Florian Hahne, Kasper D Hansen, Rafael A Irizarry, Michael Lawrence, Michael I Love, James MacDonald, Valerie Obenchain, A K Oles, H Pages, Alejandro Reyes, Paul Shannon, Gordon K Smyth, Dan Tenenbaum, Levi Waldron, and Martin Morgan. "Orchestrating high-throughput genomic analysis with Bioconductor". In: *Nat Methods* 12.2 (Jan. 2015), pp. 115–121. ISSN: 1548-7091. DOI: `10.1038/Nmeth.3252`. arXiv: `9809069v1 [arXiv:gr-qc]`. URL: `http://www.nature.com/doifinder/10.1038/nmeth.3252`.

[7] Mehmet Kemal Samur. "RTCGAToolbox: A New Tool for Exporting TCGA firehose data". In: *PLoS ONE* 9.9 (Sept. 2014). Ed. by Yu Xue, e106397. ISSN: 19326203. DOI: `10.1371/journal.pone.0106397`. URL: `http://www.ncbi.nlm.nih.gov/pubmed/25181531%20http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC4152273%20http://dx.plos.org/10.1371/journal.pone.0106397`.

[8] Geir Kjetil Sandve, Anton Nekrutenko, James Taylor, and Eivind Hovig. "Ten simple rules for reproducible computational research." In: *PLoS computational biology* 9.10 (Oct. 2013), e1003285. ISSN: 1553-7358. DOI: `10.1371/journal.pcbi.1003285`. URL: `http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3812051%7B%5C&%7Dtool=pmcentrez%7B%5C&%7Drendertype=abstract`.