Automating Access to Data from HTML Forms with Applications in Genomics

March 26, 2009

Abstract

1 Introduction

Functional genomics field is gaining acceleration as more data sources from multiple organisms are becoming available. Statistical analysis of genomic data typically requires access to various online genomic databases either for main analysis or during downstream analysis for validation purposes or for both. A large collection of R annotation tools exists for using during analysis of DNA microarray gene expression experiments as a part of Bioconductor Project (give ref, e.g., AnnBuilder, annotate, and annaffy packages). Among these tools are R functions that provide querying support for different web services provided by National Library of Medicine (NLM) and the National Center for Biotechnology Information (NCBI).

Currently, analysis of important sources of genomic data such as ChIP-Chip data (**??**), comparative genomic data include downloading data via HTML forms. This can be tedious and error prone when done repeatedly. Since the data sources are constantly evolving, i.e., genome assemblies are being updated, there is a need to repeat the downloading process as new versions of the data become available. In the case of human ChIP-Chip data (**??**), for example, after having identified bound regions on the genome, i.e., potential targets of the transcription factor of interest, by analyzing the experimental data, the next step involves extracting DNA sequences that are in the immediate vicinity of these regions to search for regulatory motifs, i.e., DNA binding sites. Moreover, information as to where these regions are located, i.e., near CpG islands, within or around known (annotated) genes, also gain importance hence one needs access to the locations of these sequence features so that orientation of the identified bound regions with respect to these features can be identified. Another important example that typically requires downloading data via HTML forms arises in the field of comparative genomics. It is often interest to take a co-expressed group of genes and then analyze their sequence data together with corresponding orthologus sequences from related species. The orthologus information is typically downloadable via a HTML form from a genomic web site. In this article, we provide programmatic access to data that are available for downloading via HTML forms. We achieve this through a new R package that we developed. Using this package, users can create R functions corresponding to their specified HTML forms. These HTML form specific R functions allow users to access the corresponding online databases by inputting similar elements to the form, that they would have inputted to the HTML form, without ever leaving the R environment. Aside from automated access to data, this allows programmatic manipulation of the obtained data.

This paper is organized as follows. On the next section, we discuss the motivation of our approach in detail and outline general principles and components of it. Sections **??** and **??** focus on characteristics and implementation of our approach. In Section **??**, we provide several examples, we firstly described how the data described in each example are accessed via HTML forms and then illustrate our automated access to these data. This is followed by a summary and discussion of our approach.

2 Motivation

Motivation for programmatic access and why this very sensible and natural to do-extraction from Duncan's write up "Automating Access to Data from HTML Forms".

3 Software architecture

Describe components, characteristics, and philosophy.

4 Implementation details

Describe functions and tricks.

5 Examples

We chose our examples from UCSC Genome Bioinformatics Site at `http://genome.ucsc.edu/` and WormBase at `http://www.wormbase.org/`. UCSC Genome Bioinformatics Site contains the reference sequence for the human and *C. elegans* genomes and working drafts for several organisms including chimpanzee, mouse, Drosophila, *C. briggsae*, and yeast. WormBase is designed exclusively for nematodes and provides information concerning the genetics, genomics and biology of *C. elegans* and some related nematodes. Currently, WormBase has the reference sequence for *C. elegans* and draft sequence for *C.Briggsea*. As more and more nematodes are sequenced, they will be included in the WormBase.

Our first example in Section **??** concerns comparative genomics. Specifically, we focus on DNA sequence access for *C. elegans* and *C. briggsea* through WormBase. The second example in Section **??** focuses on UCSC Genome Bioinformatics Site and is especially useful for downstream analysis of human ChIP-Chip data (**??**).

5.1 Example I: WormBase

5.1.1 Extracting *C. elegans* DNA sequence using WormBase.

This is a rather simple task. The url `http://www.wormbase.org/db/searches/advanced/dumper` is a HTML form with fields that one can input specific gene names or chromosome numbers and specify the type of the sequence of interest, e.g., feature or flanking sequence. For example, in order to extract 100bp upstream of the transcription start site for gene unc-47, one has to input the following to the HTML form.

- Input “unc-47” to the `Input Options` box.
- Select the option “Gene Models” from the list `Select one feature to retrieve`.
- Select `output option` “flanking sequences only”.
- Input `flanking sequence length`, e.g., 100bp 5’ flank.
- Choose `coordinates relative to`, e.g., chromosome.
- Choose `sequence orientation`, e.g., relative to feature.
- Choose `output format`, e.g., plain text.
- Hit the `DUMP` button.

Explicit R comments illustrating how we do this by our functions.*

5.1.2 Extracting DNA sequences for *C. Briggsea* orthologs of *C. elegans* genes.

This is a more complicated task than the above example since *C. briggsea* genome is recently sequenced (**?**) and the information of which *C. briggsea* genes are orthologus to which *C. elegans* genes is not available as a separate look up table in WormBase. One way to extract DNA sequence of the *C. briggsea* ortholog of a *C. elegans* gene is as outlined below. This involves firstly extracting the name (sequence id) of the *C. briggsea* ortholog for the gene of interest and then repeating steps similar to that of section . For illustration purposes, we will use unc-47.

- Go to `Batch Genes` link in the menu bar of WormBase. This corresponds to the url `http://www.wormbase.org/db/searches/info_dump`.
- Input “unc-47” to the `Genes or Loci` box.
- Submitting the above query opens up a HTML page with a hyperlink to `locus` unc-47. Following this link, one obtains a very detailed page for unc-47. In particular, this page contains identification, location and function information for the gene of interest, e.g., unc-47 in this case. Under the identification title, one has a named hyperlinked marked as `Putative C. briggsae ortholog` for the specified gene unc-47. The name of the hyperlink, CBG09800, is a sequence id for `C. briggsea` ortholog of unc-47. This link leads to a detailed site for CBG09800 where access to the desired `C. briggsea` sequence id, cb25.fpc2234, that can be used for extracting sequences from `C. briggsea` genome, is possible.
- As mentioned previously, the above steps are just for the purpose of obtaining the proper sequence id of `C. briggsea` ortholog of a `C.elegans` gene. Next, one has to go to the `Batch Sequences` link of the WormBase and repeat the steps of Section **??** with the exception that *C. briggsea* should be selected as the species and option “Integrated (hybrid) briggsea gene set” should be chosen in the `Select one feature to retrieve` field. The output obtained from `Batch sequences` for `C. briggsea` is also slightly different than the one obtained for *C. elegans*. One typically obtains multiple queries for a given sequence id. The partial output obtained for the *C. briggsea* sequence id cb25.fpc2234 is as follows:

```
...
>CBG09797 (cb25.fpc2234:223440..227457)
TCTTTTCGCCTCTCTTTTCTTTTTTCCCCCACTATCGTTTTTTTCAGTTACTCCAGTTATCCAT
ATCCTATCTGTAATACTGGACCAGCACTACGGTACA
>CBG09798 (cb25.fpc2234:229315..254784)
CTTTTGCGGACTTGCTGCCAGCCTTCACACAGACCGGCATAACTATATACTACTTCGAGAGATA
GATGAGATTAATTTTTTTCACAACACCCGCATAACA
>CBG09799 (cb25.fpc2234:246992..246175)
AGAATTCATGCAAATCGAATTATGCACAGAAAAGGGAAAAGTATAAAAGAGCGAGTACGGAAGG
CTGAAAATCAGTTTCATTCTTGATTCTCCTCTCGAC
>CBG09800 (cb25.fpc2234:257081..255468)
ACGACGATGACGAGCGCCCAAGAGGTCTCCAGAGCTCTTTTCACAAATTCTCTTCTTTCAAAAC
CGGTGGTTCCTTTCAAGTTTGTGTTTCCTTACAGAC
>CBG09801 (cb25.fpc2234:259453..263924)
CTTTTCTTTTTGGTTTATTCTTCTTCTTTTTTATGTTTGCCACTCTATTTTTAAACTTGTTGCT
TCTATTTTAAACTTTACTACATATATTTCATTTCAG
...
```

Among these `>CBG09800 (cb25.fpc2234:257081..255468)` is the one that is orthologus to the unc-47 gene of *C.elegans*. Note that `CBG09800` matches the first sequence id that led us to `cb25.fpc2234`. *** As mentioned, extracting ortholog sequences from WormBase is pretty tricky. I am sure this will become easier as the quality of CB genome increases. For the time being, this is the way to go. As we see, even after extracting the data we have to do postprocessing to extract the actual one. Of course, once everything is in the R environment, this becomes a trivial task.***

Explicit R comments illustrating how we do this by our functions.*

5.2 Example II: UCSC Genome Browser

UCSC genome browser provides two main platforms for viewing sequence and related information on several organisms. The first one is the Human Genome Browser Gateway which provides graphical view of several features of the genomes. The second one is the Table Browser that mainly provides text output of various features. Here, we focus on tasks related to Table Browser since this provides output that can be programmatically manipulated in R. Specifically, as examples, we provide (1) extracting the locations of CpG islands and known (annotated) genes and (2) extracting sequences from specific coordinates on one or several chromosomes.

5.2.1 Extracting the locations of CpG islands and known genes from human genome

One proceeds as follows in the main page of UCSC Genome Bioinformatics Site:

- Select `Tables`. This takes one to url `http://www.genome.ucsc.edu/cgi-bin/hgText`. Here, select "Human" for `genome` and "Nov. 2002" for `assembly`, then hit the `submit` button.
- In the page that opens next,
  - Choose "CpGislands" in the `Positional Tables` pull down menu. Enter "chr21" in the `position` box. Submitting this query by hitting the `Get all fields` button creates a table of CpG island locations on chromosome 21.
  - Choose "KnownGenes" in the `Positional Tables` pull down menu. Enter "chr21" in the `position` box. Submitting this query by hitting the `Get all fields` button creates a list of known genes, including their intron, exon, and location information, on chromosome 21.

Explicit R comments illustrating how we do this by our functions.*

5.2.2 Extracting sequences of specific coordinates from human genome

Text files containing sequences of human chromosomes are available for downloading at the `download` page of UCSC Genome Bioinformatics Site. However, access to sequences of specific coordinates is also possible through the HTML form “Add you own custom track” at url
`http://genome.cse.ucsc.edu/cgi-bin/hgTracks?org=Human&db=hg16&position=&pix=620&hgsid=32119563&customTrackPage=Add+Your+Own+Custom+Tracks`.

Specifically, one proceeds as follows:

- Input the desired coordinates into the “Add you own custom track” window at the above url. This url also appears as a link on the main page of Human Genome Browser Gateway at `http://genome.ucsc.edu/cgi-bin/hgGateway`. The input should be in a specific format as described in the Genome Browser. An example is as follows:

  ```
  browser position chr22:40025000-40027001
  track name=example description=”blue ticks are mine” color=0,0,255
  chr22 40025000 40026000
  chr22 40026001 40027001
  ```

  These coordinates correspond to bases 40025000 to 40027001 on 22nd human chromosome.

- Submitting these coordinates, i.e., above text file, automatically takes one to a graphical display of this region at the url `http://genome.cse.ucsc.edu/cgi-bin/hgTracks`. From here, we follow the `Tables` link on the menu bar and this leads to the url

  `http://genome.ucsc.edu/cgi-bin/hgText?db=hg16&position=chr22:40025000-40027001&phase=table&tbPosOrKeys=pos&hgsid=34125104`.

  Note that this url is coordinate-specific whereas the graphical display url is not. Now, `custom tracks` pull down menu on this page has an option “ct_example”. After selecting this, we click on `Get sequence`. This leads to a form at the coordinate-specific url

  `http://genome.ucsc.edu/cgi-bin/hgText?hgsid=34125104&db=hg16&tbTrack=Choose+table&tbCustomTrac`
  `customTrack.ct_example&table0=Choose+table&table1=Choose+table&tbPosOrKeys=pos&position=chr22%3A40%2C025%2C000-40%2C027%2C001&origPhase=table&phase=Get+sequence...`

  where `Sequence Retrieval Region Options` and `Sequence Formatting Options` are presented. This forms allows one to require additional base pairs from upstream or downstream of the original coordinates. Finally, hitting the `Get sequence` button, one obtains the sequences of the regions specified in the custom track “ct_example”.

```
>hg16_ct_example_(null) range=chr22:40025001-40026000 5'pad=0 3'pad=0 revComp=FALSE strand=
TACAGGTGTGAGCCACCGCCCCCGGTACCTGGCTAATTTTTGTATTTTTA
GGAGAGACAGGGTTTCATCATGTTGGCCTGGCTGGTCTCAAACTCCTGAC
CTCGGGATCCACTCGCCTGGGCCTCTCAAAGTACTGGGATTGCAATCCTG
ACCCCCCCGCACCTGGCCAGTGCTACAGGCTATTCTGAATTAACCCTGTG
GCCAGGTGCAGTGGCTCACGCCTGTAATCCCATCACTTTGGGAGGCCAAG
GCAGGTGGATCACCTGAGGTCAGAAGTTCGAAACCAGTCTGGCCAACATG
TTGAAACCCCAGTCTCTACTAAATACAAAAAAAATTAATCAGGCGTGGTG
CCGCATGCCTATAATCCCAGCTACTTGGGAGGCTGAGGCAGGAGAATCGC
TTGAACCAGGGAGGCGGAGGTTGCAGTGAGCCTAGATTGTGCCATTGCAC
TCCAGCCTGGGCAACAGAGCGAAGCTCCATCTCAAAAAAAAAAAATAAAAC
AGGCTGGGCGTGGTGGCTCATGCCTGTAATCCCAGTACTTTGGGAGGCTG
AGGCGGGTGGATCACCTGAGGTCAGGAGTTTGAGACCAGTCTGGCCAACA
TGGTGAAACTCCGTCTCTACTAAAAATACAAAAAATTAGCTGGTCATGGT
GGCAGGCTCCTGTAATCCCAGTTTACTGGGGAGACTGAGACAGGAGAATT
GCTTGAACCCAGGAGGCAGAGGTTGCAGTGAGCCAAGATTGCGCCATTGT
AAGCCAGCCGAAGCAACAAAAGTGAAACTCTGTCTCAAATAAATAAATAA
AATAAAATAAAACAGGCCAGGCATAGTGGCTCATGCCTGTAATCCCAGCA
CTTTGGTGGGTGGATCACCTGAGCTCAGGAGTTCGAGACCAGCCTGGCCA
ACATAGTAAAACCCCATCTCTACTAAAAATACAAAAAATAGCTGGGCATG
GTGGTGCGTACCTGTAATCCCAGCTACTCGGGAGTCTGAGGCACAAGAAT
>hg16_ct_example_(null) range=chr22:40026002-40027001 5'pad=0 3'pad=0 revComp=FALSE strand=
GCTTGAACCCAGGAGGTGGAGGTTGCAGCGAGCCGAAATTGGGTCACTGC
ACTCCAGCCTGGGCGACGAGCAAAATACTGTCTCAAAATAATAAAAACTA
AAATAAAATTAAATAATGAATTAACCCTATGACCGCTTAAGGTCTCAGAG
TATGTGGTAAGCCTTCTGGGAAGGCCAGGCATGGATGAGGATGGGATGCC
GTGTCTAAAGGAAAACAGCCGGCCTGGTTCAAGTGCGAGATTCGAACTGG
AGAATAGAGTAGGTTCTTGAAATGCATGGGTTGGAATCGGGGCTCTGAGA
AAGATGGGCCAGGCCTGTGAGGCTCACTGCCCACTTCCTGGGATTGAGTT
ACTCTCCTGTGTGGTATTTCATCGCAGATTTTGTTTCTGCAGATAAGGAA
AAGGGGAAGGAAAAGCTGGAGGAGGACGAGGCCGCAGCCGCCAGCACCAT
GGCTGTCTCAGCCTCCCTCATGCCACCCATCTGGGACAAGACCATCCCAT
ATGATGGCGAATCTTTCCACCTGGAGTACATGGACCTGGATGAGTTCCTG
CTGGAGAATGGCATCCCCGCCAGCCCCACCCACCTGGCCCACAACCTGCT
GCTGCCTGTAGCAGAGCTAGAAGGGAAGGAGTCTGCCAGCTCTTCCACAG
CATCCCCACCATCCTCCTCCACTGCCATCTTTCAGCCCTCTGAAACCGTG
TCCAGCACAGGTTGGTGAAAGGCCATCGAGGAGGGCCACCTGTCCCATCC
AGGGAAGTCCATTTCCCACTGAGGGCTGCTTGTGTCCATGTACCCAGTGA
GTCCCTTCTCTGAGGGGAGGTCCTGGTGGGGCTGCAGTGTGGAGAAACTC
TTCACTCCCCACCCTTCCACACAGTTCCCCGAGGCTGGAGATAAAAATAG
TGGTCATGTCACTGGCAGTTGGAACCCTTCTGTTGGTTGCTGTGCTGCAA
ACAGTCAGGACAGGACCTTCCAGGCCACGGCATTACAGGATAGAAATCCC
```

Explicit R comments illustrating how we do this by our functions.*

6 Summary and discussion