**RSNPper**: utilities for SNP data

VJ Carey stvjc@channing.harvard.edu

October 3, 2007

Contents

1 Introduction 1
2 How it works 1
3 Overview of the functions 2
4 Demonstrations 3
  4.1 Obtaining information on genes 3
  4.2 Obtaining information on SNPs 4
5 Application: SNP density on chr 1 7

1 Introduction

This document describes RSNPper version 1.0, added to Bioconductor in October of 2003. This first version focuses on SNP metadata, with functions that retrieve SNP-related data from the Boston Children’s Hospital Informatics Program SNPper web service.

Earlier non-released versions of this package included considerable code for working with prettybase format and for conducting other tasks in SNP discovery projects. That material has been moved to `inst/OLD` and may be re-introduced later. Users seeking legacy support should contact the author.

2 How it works

The core of this package is the XML-RPC service at CHIP accessible through the following URL stub:

> `print(.SNPperBaseURL)`
The useSNPper function allows you to work directly with the XML-RPC server by packing up appropriate command and argument strings.

```r
> dput(useSNPper)
function (cmd, parmstring)
{
  targ <- url(paste(.SNPperBaseURL, cmd, parmstring, sep = ""))
  open(targ)
  on.exit(close(targ))
  readLines(targ)
}
> print(useSNPper("geneinfo", ",name=CRP")[1:7])

[1] "<SNPPER-RPC SOURCE="*RPCSERV-NAME*" VERSION="$Revision: 1.1.1.1 $" GENOME="hg18" DBSNP="125">"
[2] " <GENEINFO>
[3] "   <GENE ID="1735">"
[4] "   <GENEID>1735</GENEID>"
[5] "   <NAME>CRP</NAME>"
[6] "   <CHROM>chr1</CHROM>"
[7] "   <STRAND>-</STRAND>"
```

The main functions of RSNPer attend to simplifying specification of parameters and parsing and packaging the XML results.

**Note on auditability.** All functions return textual information coupled with auditing information as a 'toolInfo' attribute, detailing the SNPper supplied information on the human genome sequence build, the dbSNP version, and the SNPper version from which the results are obtained. At present, there is one exception: when `itemsInRange` is invoked with `item='countsnps`, no toolInfo data is obtained. This will be corrected once the `countsnps` command at SNPper returns valid XML element tags.

### 3 Overview of the functions

The current set of functions intended for investigative use is:

- **geneInfo** – general information about location and nomenclature
- **geneLayout** – information about exon locations
- **geneSNPs** – all SNPs associated with a given gene
- **SNPinfo** – detailed information on a SNP
- **itemsInRange** – supports chromosome scanning for genes, SNPs, or counts of SNPs

An omission: for SNP information, I have not collected information on submitter.
4 Demonstrations

4.1 Obtaining information on genes

The geneInfo function will collect some basic information on a gene. The gene may be specified by HUGO name, mRNA accession number, or SNPper id.

> print(geneInfo("CRP"))

SNPper Gene metadata:
There are 1 entries.
Basic information:

```
GENEID  NAME  CHROM  STRAND  PRODUCT  NSNPS
1735    CRP    chr1    -        C-reactive protein, pentraxin-related 141
```

The geneLayout function provides information on exon locations.

> print(geneLayout("546"))

```
Information on all the genes catalogued in a certain chromosomal region can be obtained using `itemsInRange`.

```r
> print(itemsInRange("genes", "chr1", "156400000", "156500000"))
```

```
[[1]]
  NAME     CHROM
  "CD1D"   "chr1"
  PRODUCT NSNPS
  "CD1D antigen, d polypeptide" "114"

[[2]]
  NAME     CHROM                     PRODUCT
  "CD1A"   "chr1" "CD1A antigen precursor"
  NSNPS
  "53"
```

```r
$CHR
[1] "chr1"
```

```r
$START
[1] "156400000"
```

```r
$END
[1] "156500000"
```

```r
$COUNT
[1] "2"
```

```r
attr(,"toolInfo")
```

```
SOURCE VERSION GENOME
"*RPCSERV-NAME*" "$Revision: 1.1.1.1 $" "hg18"
DBSNP "125"
```

4.2 Obtaining information on SNPs

Suppose you want information on the SNP with dbSNP id rs25.

```r
> print(SNPinfo("25"))
```

```
SNP per SNP metadata:
DBSNPID CHROMOSOME POSITION ALLELES VALIDATED
```
There are details on 4 populations and 1 connections to gene features.

SNP per info:

<table>
<thead>
<tr>
<th>SOURCE</th>
<th>VERSION</th>
<th>GENOME</th>
<th>DBSNP</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>RPCSERV-NAME</em></td>
<td>$Revision: 1.1.1.1 $</td>
<td><em>hg18</em></td>
<td>125</td>
</tr>
</tbody>
</table>

Suppose instead you want information on all the SNPs cataloged in a certain chromosomal region.

```r
> ird <- itemsInRange("snps", "chr1", "156400000", "156500000")
> print(length(ird))

[1] 557

> print(ird[1:3])

[[1]]

<table>
<thead>
<tr>
<th>DBSNPID</th>
<th>TSCID</th>
<th>CHROMOSOME</th>
<th>POSITION</th>
<th>ALLELES</th>
<th>ROLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>rs16839876</td>
<td>&quot; &quot;</td>
<td>chr1</td>
<td>156400131</td>
<td>A/T</td>
<td>&quot; &quot;</td>
</tr>
</tbody>
</table>

RELPOS AMINO AMINOPOS

| " " | " " | " " |

[[2]]

<table>
<thead>
<tr>
<th>DBSNPID</th>
<th>TSCID</th>
<th>CHROMOSOME</th>
<th>POSITION</th>
<th>ALLELES</th>
<th>ROLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>rs12117055</td>
<td>&quot; &quot;</td>
<td>chr1</td>
<td>156400300</td>
<td>C/T</td>
<td>&quot; &quot;</td>
</tr>
</tbody>
</table>

RELPOS AMINO AMINOPOS

| " " | " " | " " |

[[3]]

<table>
<thead>
<tr>
<th>DBSNPID</th>
<th>TSCID</th>
<th>CHROMOSOME</th>
<th>POSITION</th>
<th>ALLELES</th>
<th>ROLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>rs17455763</td>
<td>&quot; &quot;</td>
<td>chr1</td>
<td>156400743</td>
<td>A/T</td>
<td>&quot; &quot;</td>
</tr>
</tbody>
</table>

RELPOS AMINO AMINOPOS

| " " | " " | " " |

Note that the start and end locations are supplied as strings. This is to avoid coercion to textual scientific notation.

Additional detail on the count of SNPs can be obtained more briefly:

```r
> print(itemsInRange("countsnps", "chr1", "156400000", "156500000"))

total exonic nonsyn

553 12 2
To see all the SNPs associated with a given gene, use the `geneSNPs` function. This requires knowledge of the SNPper gene id, which can be obtained using `geneInfo`.

```r
> gs <- geneSNPs("546")
> print(length(gs))

[1] 150

> print(gs[1:3])

[[1]]

<table>
<thead>
<tr>
<th>DBSNPID</th>
<th>TSCID</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;rs11809784&quot;</td>
<td>&quot; &quot;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CHROMOSOME</th>
<th>POSITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;chr1&quot;</td>
<td>&quot;33046120&quot;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ALLELES</th>
<th>ROLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;A/C&quot;</td>
<td>&quot;Promoter&quot;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RELPOS</th>
<th>AMINO</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;-18168&quot;</td>
<td>&quot; &quot;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>AMINOPOS</th>
<th>HUGO</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot; &quot;</td>
<td>&quot;S100PBP&quot;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LOCUSLINK</th>
<th>NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;64766&quot;</td>
<td>&quot;S100P binding protein Riken isoform a&quot;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>mRNA</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;NM_022753&quot;</td>
</tr>
</tbody>
</table>

[[2]]

<table>
<thead>
<tr>
<th>DBSNPID</th>
<th>TSCID</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;rs4422972&quot;</td>
<td>&quot; &quot;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CHROMOSOME</th>
<th>POSITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;chr1&quot;</td>
<td>&quot;33046500&quot;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ALLELES</th>
<th>ROLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;G/T&quot;</td>
<td>&quot;Promoter&quot;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RELPOS</th>
<th>AMINO</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;-17788&quot;</td>
<td>&quot; &quot;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>AMINOPOS</th>
<th>HUGO</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot; &quot;</td>
<td>&quot;S100PBP&quot;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LOCUSLINK</th>
<th>NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;64766&quot;</td>
<td>&quot;S100P binding protein Riken isoform a&quot;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>mRNA</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;NM_022753&quot;</td>
</tr>
</tbody>
</table>

[[3]]

6
5 Application: SNP density on chr 1

Human chromosome 1 is approximately 300Mb, and 142,629 SNPs have been recorded as of dbSNP build 106, according to NCBI SNP/maplists/maplist-newmap.html on 13 Sep 03. Let’s see if these facilities can recover this sort of information. Counting the number of SNPs on a long chromosomal region seems to take a long time for SNPper, so we will break up the task.

```r
> print(itemsInRange("countsnps", "chr1", "1", "100000"))

  total exonic nonsyn
     340  23  3

> system("sleep 2")
> print(itemsInRange("countsnps", "chr1", "100001", "200000"))

  total exonic nonsyn
     28  0  0

> system("sleep 2")
> print(itemsInRange("countsnps", "chr1", "200001", "300000"))

  total exonic nonsyn
     48  0  0

> system("sleep 2")
```

These runs complete in a reasonable amount of time. Here we will just look at the first 2Mb in intervals of .1Mb.
```r
> starts <- as.character(as.integer(seq(1, 2000001, 1e+05)))
> ends <- as.character(as.integer(as.integer(starts) + 99999))
> out <- matrix(NA, nr = 20, nc = 3)
> for (i in 1:20) {
+   cat(i)
+   out[i, ] <- itemsInRange("countsnps", "chr1", starts[i],
+                     ends[i])
+   system("sleep 2")
+ }

1234567891011121314151617181920

> print(out)

   [,1] [,2] [,3]
[1,] 340  23  3
[2,]  28  0  0
[3,]  48  0  0
[4,]  14  0  0
[5,]  4  0  0
[6,] 288  0  0
[7,]  49  0  0
[8,] 683  12  1
[9,] 454  39 14
[10,] 461  45 12
[11,] 359  13  0
[12,] 442  54 16
[13,] 400  75 25
[14,] 340  36  6
[15,] 431  38 12
[16,] 292  10  2
[17,] 385  37  9
[18,] 201  7  1
[19,] 405  20  4
[20,] 414  9  3
```