

# **Rcmd.pm – Perl Interface for R Language**

## **for G-language Genome Analysis Environment Version 1**

### **1. AUTHOR**

Kazuharu Arakawa, gaou@g-language.org

### **2. LAST UPDATE OF THIS DOCUMENT**

March 21, 2002

### **3. INTRODUCTION**

Rcmd.pm enables access to the powerful statistics package R from G-language GAE Version 1. We now recommend using R/S Perl (<http://www.omegahat.org/RSPerl/>) in a UNIX clone environment, but this module can be a preliminary interface. The interface provided by this module is very simple: all you need to do is to create an instance and pass R commands in an array. A notable feature of this interface is that all commands given in an instance is stored, so that the commands can be passed one at a time dynamically and not in a batch.

#R language is S-plus clone available with GPL at <http://www.r-project.org/>

### **4. SYNOPSIS**

```
use Rcmd;
```

```
$rcmd = new Rcmd;
```

```
@result = $rcmd->exec("<R commands>", "<next R command>");
```

### **5. DESCRIPTION**

Note: We recommend using R/S Perl (<http://www.omegahat.org/RSPerl/>) instead of this module in UNIX environment. Follow instructions of the above web site.

Rcmd enables Perl manipulation of the R language by simply executing them through \$rcmd->exec() function. Input is an array of R commands.

ex:

```
print $rcmd->exec(  
    "x_5",  
    "y_4",  
    "z_x*y",  
    "z"  
);
```

Returned values are always an array. Therefore, in case the returned value is only one, the value is accessible as:

```
@val = $rcmd->exec("y");
```

```
print $val[0];
```

All the values are saved in each session. Thus,

```
$val1 = $rcmd->exec( "x_5" , "x" );  
$val2 = $rcmd->exec( "x");
```

will output "5" for both \$val1 and \$val2.

Obviously, it is also possible to use perl variables, as:

```
$i = 3;
```

```
print $rcmd->exec("y_y*$i","x");
```

The strength of R graphing abilities can be accessed as:

```
@array = $rcmd->exec(  
    "postscript(¥"/tmp/out.ps¥)",  
    "x_c(1:10)",  
    "y_c(3,6,3,5,8,0,1,9,2,6)",  
    "plot(x,y)",  
    "z_lsfit(x,y)",  
    "abline(z)",  
    "y"  
);  
  
system("gs /tmp/out.ps");
```