

# Entrez To Probe (ETP) Package

Jeffrey D. Allen

February 17, 2011

We'll first need to setup our environment and establish a connection.

```
> library(rjson)
> source("ETP.R")
> conn <- etp.connect("http://qbridev.swmed.edu/etp/", "68BC9124A0A8BE32")
```

## 1 Getting Gene or Probe Information

The ETP package can be used to get information from the Entrez database about a Gene, or to convert between Vendor probe annotations and our internal probeIDs.

### 1.1 Get Information on a Gene

To lookup information on a gene, you must specify the EntrezID of that gene.

```
> gene780 <- etp.getGene(conn, entrezID = 780)
> gene780

$`780`
$`780`$name
[1] "discoidin domain receptor tyrosine kinase 1"

$`780`$description
[1] "discoidin domain receptor tyrosine kinase 1"

$`780`$symbols
[1] "DDR1,CAK,CD167,DDR,EDDR1,HGK2,MCK10,NEP,NTRK4,PTK3,PTK3A,RTK6,TRKE"

$`780`$accessions
[1] "AL805917.3,CR753093.2,CR936875.3,X99031.1,AAC50917.1,U48705.1,BAB63318.1,BA000025.2,B"
```

### 1.2 List the Available Platforms

All functions that reference a platform require that you reference the platform by its numerical ID. In order to lookup that ID, you can list out all the available platforms and find the one in which you're interested:

```
> platforms <- etp.getPlatforms(conn)
> platforms
```

	name	manufacturer
1	AffyU133A	Affymetrix
2	AffyU133Plus2	Affymetrix
3	AffyHG-FocusTargetArray	Affymetrix
4	AffyHumanExon1.0STArray	Affymetrix
5	AffyU95A	Affymetrix
6	AffyHu6800	Affymetrix
7	IlluminaHuman6v1	Illumina
8	IlluminaHuman6v2	Illumina
9	AgilentHumanG4112F	Agilent
10	AgilentHomoSapiens21.6Kcustomarray	Agilent
11	AgilentHumanG4112A	Agilent
12	Agilent44Kwholegenomelowdensityarray	Agilent
13	Agilent-UNC-custom-4X44K	Agilent
14	AffyU95Av2	Affymetrix
15	AffyU133B	Affymetrix
16	PRHU05-S1-0006(PCHumanOperonv2_21k)	
17	cDNAarray	
18	IlluminaHuman6v3	Illumina

### 1.3 Get Information on a Probe

To lookup information on a probe, you can either specify the probe ID which we've created internally:

```
> probe <- etp.getProbe(conn, probeID = 1000008)
> probe
```

```
$`1000008`
$`1000008`$name
[1] "1320_at"
```

```
$`1000008`$platform
[1] 1
```

Or you can reference it by its platform ID plus the name assigned to this probeset by the vendor.

```
> probe <- etp.getProbe(conn, platformID = 1, probeName = "1007_s_at")
> probe
```

```
$`1000001`
$`1000001`$name
[1] "1007_s_at"
```

```
$`1000001`$platform
[1] 1
```

## 2 Mapping Between Genes and Probes

### 2.1 Probe To Genes

You can use this package to find out which genes are associated with a certain probe. To see all of the associations, you can reference the probe by its probeID, or by the platform ID + probe name as specified earlier:

```
> genes <- etp.getGenesByProbe(conn, probeID = 2043812)
> genes <- etp.getGenesByProbe(conn, platformID = 2, probeName = "234562_x_at")
> genes
```

```
$probe
$probe$id
[1] 2043812
```

```
$probe$name
[1] "234562_x_at"
```

```
$probe$platform
[1] 2
```

```
$genes
```

	name
8647	ATP-binding cassette, sub-family B (MDR/TAP), member 11
57188	ADAMTS-like 3
63827	brevican
387535	NA
645644	NA
728678	NA

	description
8647	ATP-binding cassette, sub-family B (MDR/TAP), member 11
57188	ADAMTS-like 3
63827	brevican
387535	hepatocellular carcinoma-related HCRP1
645644	hypothetical LOC645644
728678	NA

	symbols
8647	ABCB11, ABCB16, BRIC2, BSEP, PFIC-2, PFIC2, PGY4, SPGP
57188	ADAMTSL3, KIAA1233, MGC150716, MGC150717
63827	BCAN, BEHAB, CSPG7, MGC13038
387535	HCRP1
645644	FLJ42627, FLJ12913, FLJ44722, MGC4278
728678	NA

```
8647
```

```
57188
```

```
63827 CAI13056.1, AL365181.24, CAI13057.1, AL365181.24, CAI13058.1, AL365181.24, CAI13059.1, AL3
```

```
387535
```

```
645644
```

	BLAST	Vendor	Bioconductor
8647	0.0302755	0	0
57188	0.0757342	0	0
63827	0.0909174	0	0
387535	0.712156	0	0
645644	0.0909174	0	0
728678	0	1	0

You can also filter to a certain authority. “Authority” here means the source of the association. For instance, our 3 current authorities are BLAST (=1), Vendor’s Annotations (=2), and Bioconductor (=3).

So to find the genes associated with this probe according to the vendor and BLAST, you would run:

```
> genes <- etp.getGenesByProbe(conn, probeID = 2043812, authorityID = c(1,
+ 2))
> genes

$probe
$probe$id
[1] 2043812

$probe$name
[1] "234562_x_at"

$probe$platform
[1] 2

$genes
      name
8647  ATP-binding cassette, sub-family B (MDR/TAP), member 11
57188 ADAMTS-like 3
63827  brevican
387535 NA
645644 NA
728678 NA
      description
8647  ATP-binding cassette, sub-family B (MDR/TAP), member 11
57188 ADAMTS-like 3
63827  brevican
387535 hepatocellular carcinoma-related HCRP1
645644 hypothetical LOC645644
728678 NA
      symbols
8647  ABCB11,ABCB16,BRIC2,BSEP,PFIC-2,PFIC2,PGY4,SPGP
57188 ADAMTSL3,KIAA1233,MGC150716,MGC150717
63827  BCAN,BEHAB,CSPG7,MGC13038
387535 HCRP1
645644 FLJ42627,FLJ12913,FLJ44722,MGC4278
```

```

728678                                     NA

8647
57188
63827  CAI13056.1,AL365181.24,CAI13057.1,AL365181.24,CAI13058.1,AL365181.24,CAI13059.1,AL3
387535
645644
728678

```

```

          BLAST Vendor
8647    0.0302755      0
57188   0.0757342      0
63827   0.0909174      0
387535   0.712156      0
645644   0.0909174      0
728678         0      1

```

You'll notice that the "Bioconductor" column disappears, since we're not interested in it according to the query we ran.

## 2.2 Gene to Probes

You can also specify a gene and find the relevant probes.

```

> probes <- etp.getProbesByGene(conn, entrezID = 780)
> probes

$gene
$gene$id
[1] 780

$gene$name
[1] "discoidin domain receptor tyrosine kinase 1"

$gene$description
[1] "discoidin domain receptor tyrosine kinase 1"

$gene$symbols
[1] "DDR1,CAK,CD167,DDR,EDDR1,HGK2,MCK10,NEP,NTRK4,PTK3,PTK3A,RTK6,TRKE"

$gene$accessions
[1] "AL805917.3,CR753093.2,CR936875.3,X99031.1,AAC50917.1,U48705.1,BAB63318.1,BA000025.2,B

$probes
      name platform BLAST Vendor Bioconductor
1000001  1007_s_at      1      1      1          1
1006695 207169_x_at      1      1      1          1
1008274 208779_x_at      1      1      1          1
1010210 210749_x_at      1      1      1          1
1030008  1007_s_at      5      1      1          1
1036697  36643_at      5      1      1          1

```

1123984	A_24_P367289	11	1	1	1
1132175	A_24_P123601	11	1	1	1
1142784	A_23_P93311	11	0	1	1
1220552	A_23_P93311	9	0	1	NA
1228705	A_24_P367289	9	1	1	NA
1233864	A_24_P123601	9	1	1	NA
1306237	ILMN_1812262	18	1	1	1
1312855	ILMN_2290547	18	1	1	1
1337400	ILMN_2360054	18	1	1	1
2000001	1007_s_at	2	1	1	1
2016616	207169_x_at	2	1	1	1
2018195	208779_x_at	2	1	1	1
2020131	210749_x_at	2	1	1	1
7001071	3360594	7	1	NA	0
7004647	6620193	7	1	NA	0
8020216	ILMN_1812262	8	1	1	1

Again you can filter by authority, but you can also filter by platform. To see only the probes associated with Affymetrix U133A (platform #1), you would run:

```
> probes <- etp.getProbesByGene(conn, entrezID = 780, authorityID = 2,
+   platformID = 1)$probes
> probes
```

	name	platform	Vendor
1000001	1007_s_at	1	1
1006695	207169_x_at	1	1
1008274	208779_x_at	1	1
1010210	210749_x_at	1	1

## 2.3 Missing Data

You may notice some values of NA while mapping between genes and probes. This is because not all information is available from every authority. For instance, Agilent G4112F is not available on Bioconductor, and Illumina did not give an Entrez ID in their v1 platform. Thus you'll find results like:

```
> etp.getProbesByGene(conn, 57188)$probes
```

	name	platform	BLAST	Vendor	Bioconductor
1013353	213974_at	1	1	1	1
1038932	38856_at	5	1	1	1
1130747	A_23_P43940	11	1	1	1
1143930	A_23_P308974	11	1	1	1
1155476	A_24_P400842	11	1	1	1
1201213	A_24_P400842	9	1	1	NA
1202372	A_23_P308974	9	1	1	NA
1212826	A_23_P43940	9	1	1	NA
1300959	ILMN_1798690	18	1	1	1
2004979	1559748_at	2	1	1	1

2023274	213974_at	2	1	1	1
2043812	234562_x_at	2	0.0757342	0	0
7031005	3130403	7	1	NA	0
8021599	ILMN_1798690	8	1	1	1

with rows such as:

```
> etp.getProbesByGene(conn, 57188)$probes[6, ]
```

	name	platform	BLAST	Vendor	Bioconductor
1201213	A_24_P400842	9	1	1	NA

which produces an NA value for Bioconductor because that information is not available.