

# Exploring Biological Databases

---

Holger Dinkel

---

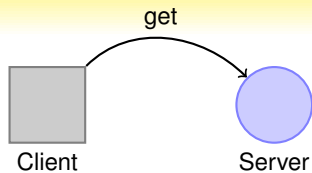
EMBO Practical Course Computational analysis of protein-protein interactions: From sequences to networks



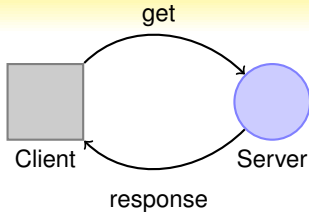
Client



Server



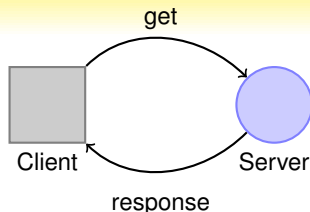
get: <http://www.uniprot.org/uniprot/P12931>



```
get: http://www.uniprot.org/uniprot/P12931
```

```
response: HTML
```

[illegible]



get: `http://www.uniprot.org/uniprot/P12931.txt`

response: **TEXT/TSV**

```
ID SRC_HUMAN Reviewed; 536 AA.  
AC P12931; E1P5V4; Q76P87; Q86VB9; Q9H5A8;  
DT 01-OCT-1989, integrated into UniProtKB/Swiss-Prot.  
DT 23-JAN-2007, sequence version 3.  
DT 03-SEP-2014, entry version 187.  
DE RecName: Full=Proto-oncogene tyrosine-protein kinase Src;  
...
```

## A RESTful application

is an application that exposes its state and functionality as a set of resources that the clients can manipulate and conforms to a certain set of principles:

- All resources are uniquely addressable, usually through URIs; other addressing can also be used, though.
- All resources can be manipulated through a constrained set of well-known actions, usually CRUD (create, read, update, delete), represented most often through the HTTP's POST, GET, PUT and DELETE; it can be a different set or a subset though - for example, some implementations limit that set to read and modify only (GET and PUT) for example
- The data for all resources is transferred through any of a constrained number of well-known representations, usually HTML, XML or JSON;
- The communication between the client and the application is performed over a \*stateless\* protocol that allows for multiple layered intermediaries that can reroute and cache the requests and response packets transparently for the client and the application.

## A RESTful application

is an application that exposes its state and functionality as a set of resources that the clients can manipulate and conforms to a certain set of principles:

- All resources are uniquely addressable, usually through URIs; other addressing can also be used, though.
- All resources can be manipulated through a constrained set of well-known actions, usually CRUD (create, read, update, delete), represented most often through the HTTP's POST, GET, PUT and DELETE; it can be a different set or a subset though - for example, some implementations limit that set to read and modify only (GET and PUT) for example
- The data for all resources is transferred through any of a constrained number of well-known representations, usually HTML, XML or JSON;
- The communication between the client and the application is performed over a \*stateless\* protocol that allows for multiple layered intermediaries that can reroute and cache the requests and response packets transparently for the client and the application.

## A RESTful application

is an application that exposes its state and functionality as a set of resources that the clients can manipulate and conforms to a certain set of principles:

- All resources are uniquely addressable, usually through URIs; other addressing can also be used, though.
- All resources can be manipulated through a constrained set of well-known actions, usually CRUD (create, read, update, delete), represented most often through the HTTP's POST, GET, PUT and DELETE; it can be a different set or a subset though - for example, some implementations limit that set to read and modify only (GET and PUT) for example
- The data for all resources is transferred through any of a constrained number of well-known representations, usually HTML, XML or JSON;
- The communication between the client and the application is performed over a \*stateless\* protocol that allows for multiple layered intermediaries that can reroute and cache the requests and response packets transparently for the client and the application.



## A RESTful application

is an application that exposes its state and functionality as a set of resources that the clients can manipulate and conforms to a certain set of principles:

- All resources are uniquely addressable, usually through URIs; other addressing can also be used, though.
- All resources can be manipulated through a constrained set of well-known actions, usually CRUD (create, read, update, delete), represented most often through the HTTP's POST, GET, PUT and DELETE; it can be a different set or a subset though - for example, some implementations limit that set to read and modify only (GET and PUT) for example
- The data for all resources is transferred through any of a constrained number of well-known representations, usually HTML, XML or JSON;
- The communication between the client and the application is performed over a \*stateless\* protocol that allows for multiple layered intermediaries that can reroute and cache the requests and response packets transparently for the client and the application.

**Method** defines what you want to do (**GET**=retrieve, **POST**=create/update, **DELETE**=remove).  
We'll be using just GET requests which can be thought of as read-only access.  
POST/DELETE are used to modify data on a server.

**URL** defines a path to a resource

**Parameters** additional arguments, filters etc. usually in the form *parameter = value*; the first parameter is separated from the url by '?' while subsequent ones use '&'.

**Example: searching for the term 'EMBO':**

**https://startpage.com/do/search?query=EMBO&with\_language=lang\_de**

**Method** defines what you want to do (**GET**=retrieve, **POST**=create/update, **DELETE**=remove).  
We'll be using just GET requests which can be thought of as read-only access.  
POST/DELETE are used to modify data on a server.

**URL** defines a path to a resource

**Parameters** additional arguments, filters etc. usually in the form *parameter = value*; the first parameter is separated from the url by '?' while subsequent ones use '&'.

**Example: searching for the term 'EMBO':**

[https://startpage.com/do/search?query=EMBO&with\\_language=lang\\_de](https://startpage.com/do/search?query=EMBO&with_language=lang_de)

**Method** defines what you want to do (**GET**=retrieve, **POST**=create/update, **DELETE**=remove).  
We'll be using just GET requests which can be thought of as read-only access.  
POST/DELETE are used to modify data on a server.

**URL** defines a path to a resource

**Parameters** additional arguments, filters etc. usually in the form *parameter = value*; the first parameter is separated from the url by '?' while subsequent ones use '&'.

**Example: searching for the term 'EMBO':**

[https://startpage.com/do/search?query=EMBO&with\\_language=lang\\_de](https://startpage.com/do/search?query=EMBO&with_language=lang_de)

**Method** defines what you want to do (**GET**=retrieve, **POST**=create/update, **DELETE**=remove).  
We'll be using just GET requests which can be thought of as read-only access.  
POST/DELETE are used to modify data on a server.

**URL** defines a path to a resource

**Parameters** additional arguments, filters etc. usually in the form *parameter = value*; the first parameter is separated from the url by '?' while subsequent ones use '&'.

**Example: searching for the term 'EMBO':**

[https://startpage.com/do/search?query=EMBO&with\\_language=lang\\_de](https://startpage.com/do/search?query=EMBO&with_language=lang_de)

**Note:**

For all these examples, any common browser can be used, however for proper 'programmatic' access tools such as 'curl' or 'wget' on the Linux/Mac commandline are much more efficient and can easily be incorporated into little scripts...

- Easy requests** The data can be requested with simple HTTP requests and returned in a variety of programatic and bioinformatical relevant formats such as JSON, XML, YAML and FASTA.
- Easy debugging** Debugging can be done in any browser. While some might not call this real programming, it surely is the first step towards programmatically querying resources.
- Reproducible** You can write all your queries into a simple script and repeat the same query later. Even just saving the URL as a bookmark in your browser helps!
- Powerful** Any data can be made available via a REST service.
- Bandwidth** An API allows programmatic access to some information if one does not want to download the entire dataset.
- Standards** By using existing protocols and best-methods (HTTP), all the existing knowledge can be reused (Caching, Redirecting, ...).
- Widespread** More and more resource providers change from fat/heavy webservices to this lightweight system, for obvious reasons.

- Easy requests** The data can be requested with simple HTTP requests and returned in a variety of programatic and bioinformatical relevant formats such as JSON, XML, YAML and FASTA.
- Easy debugging** Debugging can be done in any browser. While some might not call this real programming, it surely is the first step towards programmatically querying resources.
- Reproducible** You can write all your queries into a simple script and repeat the same query later. Even just saving the URL as a bookmark in your browser helps!
- Powerful** Any data can be made available via a REST service.
- Bandwidth** An API allows programmatic access to some information if one does not want to download the entire dataset.
- Standards** By using existing protocols and best-methods (HTTP), all the existing knowledge can be reused (Caching, Redirecting, ...).
- Widespread** More and more resource providers change from fat/heavy webservices to this lightweight system, for obvious reasons.

- Easy requests** The data can be requested with simple HTTP requests and returned in a variety of programatic and bioinformatical relevant formats such as JSON, XML, YAML and FASTA.
- Easy debugging** Debugging can be done in any browser. While some might not call this real programming, it surely is the first step towards programmatically querying resources.
- Reproducible** You can write all your queries into a simple script and repeat the same query later. Even just saving the URL as a bookmark in your browser helps!
- Powerful** Any data can be made available via a REST service.
- Bandwidth** An API allows programmatic access to some information if one does not want to download the entire dataset.
- Standards** By using existing protocols and best-methods (HTTP), all the existing knowledge can be reused (Caching, Redirecting, ...).
- Widespread** More and more resource providers change from fat/heavy webservices to this lightweight system, for obvious reasons.



- Easy requests** The data can be requested with simple HTTP requests and returned in a variety of programatic and bioinformatical relevant formats such as JSON, XML, YAML and FASTA.
- Easy debugging** Debugging can be done in any browser. While some might not call this real programming, it surely is the first step towards programmatically querying resources.
- Reproducible** You can write all your queries into a simple script and repeat the same query later. Even just saving the URL as a bookmark in your browser helps!
- Powerful** Any data can be made available via a REST service.
- Bandwidth** An API allows programmatic access to some information if one does not want to download the entire dataset.
- Standards** By using existing protocols and best-methods (HTTP), all the existing knowledge can be reused (Caching, Redirecting, ...).
- Widespread** More and more resource providers change from fat/heavy webservices to this lightweight system, for obvious reasons.

- Easy requests** The data can be requested with simple HTTP requests and returned in a variety of programatic and bioinformatical relevant formats such as JSON, XML, YAML and FASTA.
- Easy debugging** Debugging can be done in any browser. While some might not call this real programming, it surely is the first step towards programmatically querying resources.
- Reproducible** You can write all your queries into a simple script and repeat the same query later. Even just saving the URL as a bookmark in your browser helps!
- Powerful** Any data can be made available via a REST service.
- Bandwidth** An API allows programmatic access to some information if one does not want to download the entire dataset.
- Standards** By using existing protocols and best-methods (HTTP), all the existing knowledge can be reused (Caching, Redirecting, ...).
- Widespread** More and more resource providers change from fat/heavy webservices to this lightweight system, for obvious reasons.

- Easy requests** The data can be requested with simple HTTP requests and returned in a variety of programatic and bioinformatical relevant formats such as JSON, XML, YAML and FASTA.
- Easy debugging** Debugging can be done in any browser. While some might not call this real programming, it surely is the first step towards programmatically querying resources.
- Reproducible** You can write all your queries into a simple script and repeat the same query later. Even just saving the URL as a bookmark in your browser helps!
- Powerful** Any data can be made available via a REST service.
- Bandwidth** An API allows programmatic access to some information if one does not want to download the entire dataset.
- Standards** By using existing protocols and best-methods (HTTP), all the existing knowledge can be reused (Caching, Redirecting, ...).
- Widespread** More and more resource providers change from fat/heavy webservices to this lightweight system, for obvious reasons.

- Easy requests** The data can be requested with simple HTTP requests and returned in a variety of programatic and bioinformatical relevant formats such as JSON, XML, YAML and FASTA.
- Easy debugging** Debugging can be done in any browser. While some might not call this real programming, it surely is the first step towards programmatically querying resources.
- Reproducible** You can write all your queries into a simple script and repeat the same query later. Even just saving the URL as a bookmark in your browser helps!
- Powerful** Any data can be made available via a REST service.
- Bandwidth** An API allows programmatic access to some information if one does not want to download the entire dataset.
- Standards** By using existing protocols and best-methods (HTTP), all the existing knowledge can be reused (Caching, Redirecting, ...).
- Widespread** More and more resource providers change from fat/heavy webservices to this lightweight system, for obvious reasons.

- Easy requests** The data can be requested with simple HTTP requests and returned in a variety of programatic and bioinformatical relevant formats such as JSON, XML, YAML and FASTA.
- Easy debugging** Debugging can be done in any browser. While some might not call this real programming, it surely is the first step towards programmatically querying resources.
- Reproducible** You can write all your queries into a simple script and repeat the same query later. Even just saving the URL as a bookmark in your browser helps!
- Powerful** Any data can be made available via a REST service.
- Bandwidth** An API allows programmatic access to some information if one does not want to download the entire dataset.
- Standards** By using existing protocols and best-methods (HTTP), all the existing knowledge can be reused (Caching, Redirecting, ...).
- Widespread** More and more resource providers change from fat/heavy webservices to this lightweight system, for obvious reasons.

### Note:

Not meant to be a substitute for resources such as BioMART etc!

# EXAMPLE: PHOSPHO.ELM

**Phospho.ELM**  
a database of S/T/Y phosphorylation sites

**Statistics:**

Instances	42,575
Kinases	310
Reference	3,672
Sequences	11,223
Substrates	8,718

[Home](#) [PhosphoBlast](#) [Contribute](#) [Download](#) [Help](#) [Links](#) [About](#)

## SEARCH

- ☐ for phosphorylation sites in proteins using protein name or gene name  
(eg. Paxillin, Shc, MAPK)
- ☐ by UniPROT accession or Ensembl identifier:  
(eg. P12931 or P55211)
- ☐ by selected kinase (List):
- ☐ by selected phospho-peptide binding domain (List):
- ☐ Choose which organisms to include  

All

Caenorhabditis

Drosophila

Vertebrates
- ☐ Do not show high throughput data
- ☐ Output as Comma-Separated-Values (.csv)

[Search](#) [Reset](#)

<http://phospho.elm.eu.org/index.html>

### Access:

The PhosphoELM database can also be accessed via URL as follows:

- by **substrate name**:  
<http://phospho.elm.eu.org/bySubstrate/Paxillin.html>
- by **Uniprot ID**:  
<http://phospho.elm.eu.org/byAccession/P12931.html>
- by **Uniprot ID** and **Position**:  
<http://phospho.elm.eu.org/byAccession/P12931/Pos17.html>
- by **ENSEMBL ID** and multiple **Positions**:  
<http://phospho.elm.eu.org/byAccession/ENSP00000265709/Pos216,231.html>
- by **Uniprot name**:  
[http://phospho.elm.eu.org/byAccession/src\\_human.html](http://phospho.elm.eu.org/byAccession/src_human.html)
- by **Kinase**:  
<http://phospho.elm.eu.org/byKinase/Abl2.html>
- by **Binding domain**:  
[http://phospho.elm.eu.org/byDomain/CBL\\_SH2.html](http://phospho.elm.eu.org/byDomain/CBL_SH2.html)
- retrieve a **stored Sequence**:  
<http://phospho.elm.eu.org/P12931.fasta>
- retrieve data **as CSV**:  
<http://phospho.elm.eu.org/byAccession/P12931.csv>
- retrieve data for a single position **as CSV**:  
<http://phospho.elm.eu.org/byAccession/P12931/Pos12.csv>
- retrieve data for **multiple** IDs **as CSV**:  
<http://phospho.elm.eu.org/byAccession/P12931,P55211.csv>
- using **web-services**:  
<http://phospho.elm.eu.org/webservice/phosphoELMdb.wsdl>

<http://phospho.elm.eu.org/byAccession/P55211.html>

### Access:

The PhosphoELM database can also be accessed via URL as follows:

- by **substrate name**:  
<http://phospho.elm.eu.org/bySubstrate/Paxillin.html>
- by **Uniprot ID**:  
<http://phospho.elm.eu.org/byAccession/P12931.html>
- by **Uniprot ID** and **Position**:  
<http://phospho.elm.eu.org/byAccession/P12931/Pos17.html>
- by **ENSEMBL ID** and multiple **Positions**:  
<http://phospho.elm.eu.org/byAccession/ENSP00000265709/Pos216,231.html>
- by **Uniprot name**:  
[http://phospho.elm.eu.org/byAccession/src\\_human.html](http://phospho.elm.eu.org/byAccession/src_human.html)
- by **Kinase**:  
<http://phospho.elm.eu.org/byKinase/Abl2.html>
- by **Binding domain**:  
[http://phospho.elm.eu.org/byDomain/CBL\\_SH2.html](http://phospho.elm.eu.org/byDomain/CBL_SH2.html)
- retrieve a **stored Sequence**:  
<http://phospho.elm.eu.org/P12931.fasta>
- retrieve data **as CSV**:  
<http://phospho.elm.eu.org/byAccession/P12931.csv>
- retrieve data for a single position **as CSV**:  
<http://phospho.elm.eu.org/byAccession/P12931/Pos12.csv>
- retrieve data for **multiple** IDs **as CSV**:  
<http://phospho.elm.eu.org/byAccession/P12931,P55211.csv>
- using **web-services**:  
<http://phospho.elm.eu.org/webservice/phosphoELMdb.wsdl>

<http://phospho.elm.eu.org/byAccession/P55211.csv>



# EXAMPLE: PHOSPHO.ELM

## Query

<http://phospho.elm.eu.org/bySubstrate/cd66.html>

## Output:


Substrate:

CD66 [Immunoglobulin]

Seq-ID:


P13688 [*Homo sapiens*]

Interaction



Network(s):

External Source(s):



MINT Interaction(s):

-

GO-Terms:

[\[show\]](#)

Conservation:

Click on table headers for sorting

Res. ▴	Pos. ▴	Sequence ▴	Kinase ▴	PMID ▴	Src ▴	Cons. ▴	ELM ▴	Binding Domain ▴	SMART/Pfam ▴	IUPRED score ▴	PDB ▴	P3D Acc. ▴
Y	493	DEPHOSSEVY <sup>Y</sup> SLNFEAQQP	-	<a href="#">9967848</a>	LTP	1.00		-	-	0.65	-	low
S	508	FEAQPTQPT <sup>S</sup> ASPLATATEI	-	<a href="#">11850617</a>	LTP	1.00		-	-	0.65	-	low
Y	520	SPSLATATEI <sup>Y</sup> SEVWQ	-	<a href="#">9967848</a>	LTP	1.00		-	-	0.38	-	low

Substrate:

CD66 (Immunoglobulin)

Seq-ID:


P31809 [*Mus musculus*]

Interaction

-

Network(s):

External Source(s):

PHOSIDA


MINT Interaction(s):

-

GO-Terms:

[\[show\]](#)

Conservation:

Click on table headers for sorting

Res. ▴	Pos. ▴	Sequence ▴	Kinase ▴	PMID ▴	Src ▴	Cons. ▴	ELM ▴	Binding Domain ▴	SMART/Pfam ▴	IUPRED score ▴	PDB ▴	P3D Acc. ▴
--------	--------	------------	----------	--------	-------	---------	-------	------------------	--------------	----------------	-------	------------

# EXAMPLE: PHOSPHO.ELM

## Query

<http://phospho.elm.eu.org/bySubstrate/cd66.html>

### ■ Query by Substrate name

■ Substrate name

■ Output as HTML

## Output:


Substrate:

CD66 [Immunoglobulin]

Seq-ID:


P13688 [*Homo sapiens*]

Interaction



Network(s):

External Source(s):



MINT Interaction(s):

-

GO-Terms:

[\[show\]](#)

Conservation:

Click on table headers for sorting

Res. ▴	Pos. ▴	Sequence ▴	Kinase ▴	PMID ▴	Src ▴	Cons. ▴	ELM ▴	Binding Domain ▴	SMART/Pfam ▴	IUPRED score ▴	PDB ▴	P3D Acc. ▴
Y	493	DEPHRSEVTVYSLNFEAQQP	-	<a href="#">9967848</a>	LTP	1.00		-	-	0.65	-	low
S	508	FEAQQPTQPTSAASPLATATEI	-	<a href="#">11850617</a>	LTP	1.00		-	-	0.65	-	low
Y	520	SPSLATATEIIVYSEVWQ	-	<a href="#">9967848</a>	LTP	1.00		-	-	0.38	-	low

Substrate:

CD66 (Immunoglobulin)

Seq-ID:


P31809 [*Mus musculus*]

Interaction

-

Network(s):

External Source(s):

PHOSIDA 

MINT Interaction(s):

-

GO-Terms:

[\[show\]](#)

Conservation:

Click on table headers for sorting

Res. ▴	Pos. ▴	Sequence ▴	Kinase ▴	PMID ▴	Src ▴	Cons. ▴	ELM ▴	Binding Domain ▴	SMART/Pfam ▴	IUPRED score ▴	PDB ▴	P3D Acc. ▴
--------	--------	------------	----------	--------	-------	---------	-------	------------------	--------------	----------------	-------	------------

# EXAMPLE: PHOSPHO.ELM

## Query

http://phospho.elm.eu.org/bySubstrate/cd66.html

- Query by Substrate name
- **Substrate name**
- Output as HTML

## Output:


Substrate:

CD66 [Immunoglobulin]

Seq-ID:


P13688 [*Homo sapiens*]

Interaction



Network(s):

External Source(s):



MINT Interaction(s):

-

GO-Terms:

[\[show\]](#)

Conservation:

Click on table headers for sorting

Res. ▴	Pos. ▴	Sequence ▴	Kinase ▴	PMID ▴	Src ▴	Cons. ▴	ELM ▴	Binding Domain ▴	SMART/Pfam ▴	IUPRED score ▴	PDB ▴	P3D Acc. ▴
Y	493	DEPHRSEVTVYSLNFEAQQP	-	<a href="#">9967848</a>	LTP	1.00		-	-	0.65	-	low
S	508	FEAQQPTQPTSAASPLATATEI	-	<a href="#">11850617</a>	LTP	1.00		-	-	0.65	-	low
Y	520	SPSLATATEIIVYSEVWQ	-	<a href="#">9967848</a>	LTP	1.00		-	-	0.38	-	low

Substrate:

CD66 (Immunoglobulin)

Seq-ID:


P31809 [*Mus musculus*]

Interaction

-

Network(s):

External Source(s):

PHOSIDA


MINT Interaction(s):

-

GO-Terms:

[\[show\]](#)

Conservation:

Click on table headers for sorting

Res. ▴	Pos. ▴	Sequence ▴	Kinase ▴	PMID ▴	Src ▴	Cons. ▴	ELM ▴	Binding Domain ▴	SMART/Pfam ▴	IUPRED score ▴	PDB ▴	P3D Acc. ▴
--------	--------	------------	----------	--------	-------	---------	-------	------------------	--------------	----------------	-------	------------

# EXAMPLE: PHOSPHO.ELM

## Query

<http://phospho.elm.eu.org/bySubstrate/cd66.html>

- Query by Substrate name
- Substrate name
- Output as HTML

## Output:

Substrate:

Seq-ID:

Interaction

Network(s):

External Source(s):


MINT Interaction(s):


GO-Terms:

Conservation:

CD66 (Immunoglobulin)

P13688 [*Homo sapiens*]





-

[\[show\]](#)

Click on table headers for sorting

Res. ▴	Pos. ▴	Sequence ▴	Kinase ▴	PMID ▴	Src ▴	Cons. ▴	ELM ▴	Binding Domain ▴	SMART/Pfam ▴	IUPRED score ▴	PDB ▴	P3D Acc. ▴
Y	493	DEPHRSEVTVYSLNFEAQQP	-	<a href="#">9967848</a>	LTP	1.00		-	-	0.65	-	low
S	508	FEAQQPTQPTSAASPLATETI	-	<a href="#">11850617</a>	LTP	1.00		-	-	0.65	-	low
Y	520	SPSLATETIIVYSEVWQ	-	<a href="#">9967848</a>	LTP	1.00		-	-	0.38	-	low

Substrate:

Seq-ID:

Interaction

Network(s):

External Source(s):

MINT Interaction(s):


GO-Terms:

Conservation:

CD66 (Immunoglobulin)

P31809 [*Mus musculus*]

-

[PHOSIDA](#) 

-

[\[show\]](#)

Click on table headers for sorting

Res. ▴	Pos. ▴	Sequence ▴	Kinase ▴	PMID ▴	Src ▴	Cons. ▴	ELM ▴	Binding Domain ▴	SMART/Pfam ▴	IUPRED score ▴	PDB ▴	P3D Acc. ▴
--------	--------	------------	----------	--------	-------	---------	-------	------------------	--------------	----------------	-------	------------

## EXAMPLE: PHOSPHO.ELM

### Query

<http://phospho.elm.eu.org/byAccession/P12931/Pos12,17.csv>

### Output:

```
Acc.; Res.; Pos.; Context; Kinase; PMID; Source; ConScore; ELM; Domain; SMART; IUPRED; PDB; P3D-  
P12931; S; 12; SNKSKPKDASQRRRSLEPAE; none; 2136766; 1; 0.21; ; -; ; 0.9168; -; ;  
P12931; S; 17; PKDASQRRRSLEPAENVHGA; none; 18088087; 2; 0.24; MOD_PKA_1; -; ; 0.8828; -; ;  
P12931; S; 17; PKDASQRRRSLEPAENVHGA; none; 17192257; 2; 0.24; MOD_PKA_1; -; ; 0.8828; -; ;  
P12931; S; 17; PKDASQRRRSLEPAENVHGA; none; 17081983; 2; 0.24; MOD_PKA_1; -; ; 0.8828; -; ;  
P12931; S; 17; PKDASQRRRSLEPAENVHGA; PKA_group; 11804588; 1; 0.24; MOD_PKA_1; -; ; 0.8828; -; ;  
...
```

## EXAMPLE: PHOSPHO.ELM

### Query

<http://phospho.elm.eu.org/>**byAccession**/P12931/Pos12,17.csv

- **query by Uniprot Accession**
  - Protein Sequence Accession/ID
  - Position / multiple Positions
  - Output as CSV (character separated values)

### Output:

```
Acc.; Res.; Pos.; Context; Kinase; PMID; Source; ConScore; ELM; Domain; SMART; IUPRED; PDB; P3D-  
P12931; S; 12; SNKSKPKDASQRRRSLEPAE; none; 2136766; 1; 0.21; ; -; ; 0.9168; -; ;  
P12931; S; 17; PKDASQRRRSLEPAENVHGA; none; 18088087; 2; 0.24; MOD_PKA_1; -; ; 0.8828; -; ;  
P12931; S; 17; PKDASQRRRSLEPAENVHGA; none; 17192257; 2; 0.24; MOD_PKA_1; -; ; 0.8828; -; ;  
P12931; S; 17; PKDASQRRRSLEPAENVHGA; none; 17081983; 2; 0.24; MOD_PKA_1; -; ; 0.8828; -; ;  
P12931; S; 17; PKDASQRRRSLEPAENVHGA; PKA_group; 11804588; 1; 0.24; MOD_PKA_1; -; ; 0.8828; -; ;  
...
```

## EXAMPLE: PHOSPHO.ELM

### Query

<http://phospho.elm.eu.org/byAccession/P12931/Pos12,17.csv>

- query by Uniprot Accession
- **Protein Sequence Accession/ID**
- Position / multiple Positions
- Output as CSV (character separated values)

### Output:

```
Acc.; Res.; Pos.; Context; Kinase; PMID; Source; ConScore; ELM; Domain; SMART; IUPRED; PDB; P3D-
P12931; S; 12; SNKSKPKDASQRRRSLEPAE; none; 2136766; 1; 0.21; ; -; ; 0.9168; -; ;
P12931; S; 17; PKDASQRRRSLEPAENVHGA; none; 18088087; 2; 0.24; MOD_PKA_1; -; ; 0.8828; -; ;
P12931; S; 17; PKDASQRRRSLEPAENVHGA; none; 17192257; 2; 0.24; MOD_PKA_1; -; ; 0.8828; -; ;
P12931; S; 17; PKDASQRRRSLEPAENVHGA; none; 17081983; 2; 0.24; MOD_PKA_1; -; ; 0.8828; -; ;
P12931; S; 17; PKDASQRRRSLEPAENVHGA; PKA_group; 11804588; 1; 0.24; MOD_PKA_1; -; ; 0.8828; -; ;
...
```

## EXAMPLE: PHOSPHO.ELM

### Query

<http://phospho.elm.eu.org/byAccession/P12931/Pos12,17.csv>

- query by Uniprot Accession
- Protein Sequence Accession/ID
- **Position / multiple Positions**
- Output as CSV (character separated values)

### Output:

```
Acc.; Res.; Pos.; Context; Kinase; PMID; Source; ConScore; ELM; Domain; SMART; IUPRED; PDB; P3D-
P12931; S; 12; SNKSKPKDASQRRRSLEPAE; none; 2136766; 1; 0.21; ; -; ; 0.9168; -; ;
P12931; S; 17; PKDASQRRRSLEPAENVHGA; none; 18088087; 2; 0.24; MOD_PKA_1; -; ; 0.8828; -; ;
P12931; S; 17; PKDASQRRRSLEPAENVHGA; none; 17192257; 2; 0.24; MOD_PKA_1; -; ; 0.8828; -; ;
P12931; S; 17; PKDASQRRRSLEPAENVHGA; none; 17081983; 2; 0.24; MOD_PKA_1; -; ; 0.8828; -; ;
P12931; S; 17; PKDASQRRRSLEPAENVHGA; PKA_group; 11804588; 1; 0.24; MOD_PKA_1; -; ; 0.8828; -; ;
...
```



## EXAMPLE: PHOSPHO.ELM

### Query

<http://phospho.elm.eu.org/byAccession/P12931/Pos12,17.csv>

- query by Uniprot Accession
- Protein Sequence Accession/ID
- Position / multiple Positions
- **Output as CSV (character separated values)**

### Output:

```
Acc.; Res.; Pos.; Context; Kinase; PMID; Source; ConScore; ELM; Domain; SMART; IUPRED; PDB; P3D-  
P12931; S; 12; SNKSKPKDASQRRRSLEPAE; none; 2136766; 1; 0.21; ; -; ; 0.9168; -; ;  
P12931; S; 17; PKDASQRRRSLEPAENVHGA; none; 18088087; 2; 0.24; MOD_PKA_1; -; ; 0.8828; -; ;  
P12931; S; 17; PKDASQRRRSLEPAENVHGA; none; 17192257; 2; 0.24; MOD_PKA_1; -; ; 0.8828; -; ;  
P12931; S; 17; PKDASQRRRSLEPAENVHGA; none; 17081983; 2; 0.24; MOD_PKA_1; -; ; 0.8828; -; ;  
P12931; S; 17; PKDASQRRRSLEPAENVHGA; PKA_group; 11804588; 1; 0.24; MOD_PKA_1; -; ; 0.8828; -; ;  
...
```

# EXAMPLE: ELM



The Eukaryotic Linear Motif resource for  
*Functional Sites in Proteins*

Search ELMs Instances Candidates Links About News Help Downloads Diseases Viruses

## Search ELM Instances

Full-Text Search (use "\*" to get all instances)

Filter by instance Logic

Filter by organism

submit

Reset

export 5 instances as:

[gff](#) [pir](#) [fasta](#) [tsv](#)

5 instances for search term 'P12931':

(click table headers for sorting; Notes column: =Number of Switches, =Number of Interactions)

ELM identifier	Protein name	Gene name	Start	End	Subsequence	Logic	#Ev.	Organism	Notes
LIG_SH2_SRC	SRC_HUMAN	SRC	530	533	AFLEDYFTSTEPQQPGKEL	TP	1	Homo sapiens (Human)	1
LIG_SH3_4	SRC_HUMAN	SRC	252	259	FVCPTSQQTGLAKDAMEI	TP	0	Homo sapiens (Human)	
MOD_CDK_1	SRC_HUMAN	SRC	72	78	GFNSSDVTSRQAGPLAGG	TP	1	Homo sapiens (Human)	
MOD_NMyristoyl	SRC_HUMAN	SRC	1	7	MGENKSKPKDASQRRRSLEP	TP	0	Homo sapiens (Human)	
MOD_TYR_CSK	SRC_HUMAN	SRC	526	534	AFLEDYFTSPQTQPGKEL	TP	1	Homo sapiens (Human)	

# EXAMPLE: ELM



The Eukaryotic Linear Motif resource for  
*Functional Sites in Proteins*

Search ELMs Instances Candidates Links About News Help Downloads Diseases Viruses

## Search ELM Instances

Full-Text Search (use "\*" to get all instances)

Filter by instance Logic

Filter by organism

submit

Reset

export 5 instances as:

[gff](#) [pir](#) [fasta](#) [tsv](#)

5 instances for search term 'P12931':

(click table headers for sorting; Notes column: =Number of Switches, =Number of Interactions)

ELM identifier	Protein name	Gene name	Start	End	Subsequence	Logic	#Ev.	Organism	Notes
LIG_SH2_SRC	SRC_HUMAN	SRC	530	533	AFLEDFPTSTPQ <b>TPQ</b> SEKL	TP	1	Homo sapiens (Human)	1
LIG_SH3_4	SRC_HUMAN	SRC	252	259	EVCPT <b>EPQ</b> QGLAKDAKEI	TP	0	Homo sapiens (Human)	
MOD_CDK_1	SRC_HUMAN	SRC	72	78	GWRSE <b>TVTSPQ</b> AGPLAG	TP	1	Homo sapiens (Human)	
MOD_NMyristoyl	SRC_HUMAN	SRC	1	7	HGRNKK <b>PKDASQ</b> RRSLSP	TP	0	Homo sapiens (Human)	
MOD_TYR_CSK	SRC_HUMAN	SRC	526	534	AFLEDFPT <b>TPQ</b> TPQSEKL	TP	1	Homo sapiens (Human)	



## ELM Downloads

Below you'll find examples of the different ways that can be used to query ELM programmatically. No special client is needed for this just a browser or maybe "curl"/"wget" for scripted access. This data can be used and distributed according to the [ELM Software License Agreement](#).

### Classes

Last modified on: May 21, 2014, 4:43 p.m.

Here you can download a list of ELM classes, either all at once or limit the list by providing a query term "q".

Name	Example	URL
all	<a href="#">html</a>	/elms/browse_elms.html
all	<a href="#">tsv</a>	/elms/browse_elms.tsv
by query term	<a href="#">tsv</a>	/elms/browse_elms.tsv?q=PCSK
by ELM accession	<a href="#">html</a>	/ELME000012.html

### Instances

Last modified on: May 22, 2014, 5:38 p.m.

Annotated ELM instances can be queried in a variety of ways. You are encouraged to use the [search form](#) to get a feeling for the parameters. Common examples include limiting the query by either instance logic or taxon.

Name	Example	URL
all	<a href="#">html</a>	/elms/browse_instances.html?q=*
by Uniprot acc	<a href="#">fasta</a>	/elms/browse_instances.fasta?q=P12931
by Uniprot name	<a href="#">gff</a>	/elms/browse_instances.gff?q=SRC_HUMAN
by Uniprot acc	<a href="#">tsv</a>	/elms/browse_instances.tsv?q=P12931
by query term	<a href="#">pir</a>	/elms/browse_instances.pir?q=PCSK

- [Classes](#)
- [Instances](#)
- [Interactions](#)
- [Interaction Domains](#)
- [Methods](#)
- [PDBs](#)
- [Renamed ELM classes](#)
- [Media](#)


[Search](#)
[ELMs](#)
[Instances](#)
[Candidates](#)
[Links](#)
[About](#)
[News](#)
[Help](#)
[Downloads](#)
[Diseases](#)
[Viruses](#)

## ELM Downloads

Below you'll find examples of the different ways that can be used to query ELM programmatically. No special client is needed for this just a browser or maybe "curl"/"wget" for scripted access. This data can be used and distributed according to the [ELM Software License Agreement](#).

### Classes

Last modified on: May 21, 2014, 4:43 p.m.

Here you can download a list of ELM classes, either all at once or limit the list by providing a query term "q".

Name	Example	URL
all	<a href="#">html</a>	/elms/browse_elms.html
all	<a href="#">tsv</a>	/elms/browse_elms.tsv
by query term	<a href="#">tsv</a>	/elms/browse_elms.tsv?q=PCSK
by ELM accession	<a href="#">html</a>	/ELME000012.html

### Instances

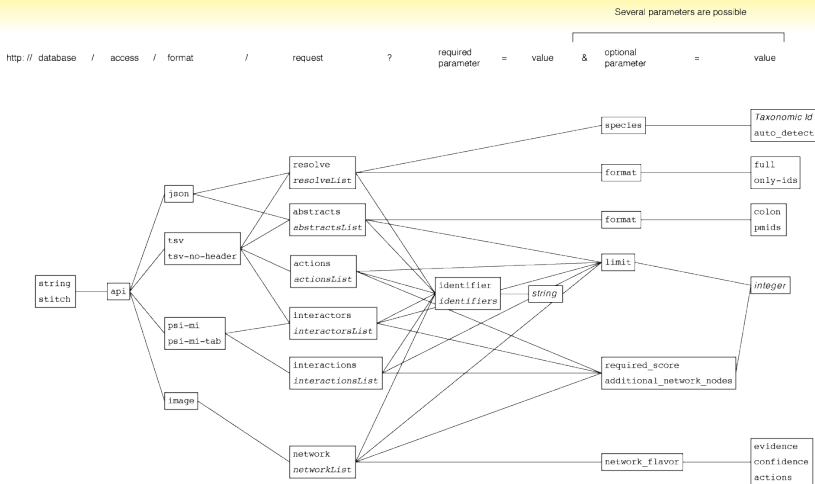
Last modified on: May 22, 2014, 5:38 p.m.

Annotated ELM instances can be queried in a variety of ways. You are encouraged to use the [search form](#) to get a feeling for the parameters. Common examples include limiting the query by either instance logic or taxon.

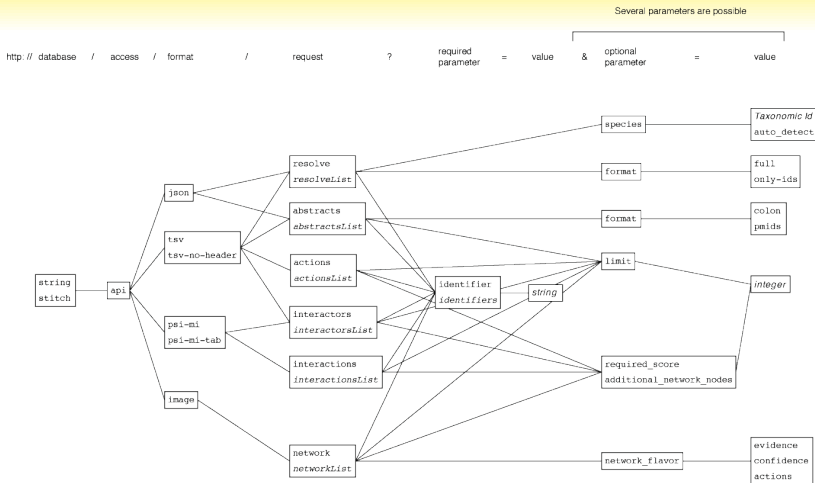
Name	Example	URL
all	<a href="#">html</a>	/elms/browse_instances.html?q=*
by Uniprot acc	<a href="#">fasta</a>	/elms/browse_instances.fasta?q=P12931
by Uniprot name	<a href="#">gff</a>	/elms/browse_instances.gff?q=SRC_HUMAN
by Uniprot acc	<a href="#">tsv</a>	/elms/browse_instances.tsv?q=P12931
by query term	<a href="#">pir</a>	/elms/browse_instances.pir?q=PCSK

- [Classes](#)
- [Instances](#)
- [Interactions](#)
- [Interaction Domains](#)
- [Methods](#)
- [PDBs](#)
- [Renamed ELM classes](#)
- [Media](#)

# EXAMPLE: STRING / STITCH



# EXAMPLE: STRING / STITCH



[http://string-db.org/api/psi-mi-tab/interactions?identifier=YOL086C&additional\\_network\\_nodes=2](http://string-db.org/api/psi-mi-tab/interactions?identifier=YOL086C&additional_network_nodes=2)

# EXAMPLE: UNIPROT

UniProtKB
abi
Advanced

BLAST Align Upload data

Help Contact
Show help for UniProtKB
Download

## Results

Filter by<sup>1</sup>

Reviewed (2,571)  
 Swiss-Prot  
 Unreviewed (50,014)  
 TrEMBL

Popular organisms

Human (365)  
 Fruit fly (180)  
 Mouse (167)  
 A. thaliana (96)  
 Rat (89)

Other organisms

Search terms

Pflur 367 (6)  
 domain (2)  
 domain (4)  
 gene name (417)  
 protein family (6)  
 protein name (444)  
 strain (32)  
 taxonomy (32)  
 tissue (6)

View by

Taxonomy  
 Keywords  
 Gene Ontology  
 Enzyme class  
 Pathway

UniRef

Your results in sequence clusters with identity of:  
 100%, 90% or 50%

Demo

Columns
BLAST
Align
Download
Add to history

Entry	Entry name	Protein names	Gene names	Organism	Length
<input type="checkbox"/> P00554	ABL1_HUMAN	Tyrosine-protein kinase ABL1	ABL1, ABL, TK7	Homo sapiens (Human)	1,130
<input type="checkbox"/> P00523	ABL1_MOUSE	Tyrosine-protein kinase ABL1	Abi1, Abi	Mus musculus (Mouse)	1,123
<input type="checkbox"/> P00522	ABL_DROME	Tyrosine-protein kinase Abi	Abi, ABL-1, Dash, CG4332	Drosophila melanogaster (Fruit fly)	1,620
<input type="checkbox"/> P00521	ABL_MVAB	Tyrosine-protein kinase	ABL	Abelson murine leukemia virus	746
<input type="checkbox"/> P10447	ABL_PSVNY	Tyrosine-protein kinase transformin...	ABL	Feline sarcoma virus (strain Hardy-Zuckerman 2)	439
<input type="checkbox"/> P03949	ABL1_CAEEL	Tyrosine-protein kinase abf-1	abf-1, MTS-1	Caenorhabditis elegans	1,224
<input type="checkbox"/> P11681	ABL1_CALVI	Tyrosine-protein kinase ABL	ABL	Calliphora vicina (Blue blowfly) (Calliphora erythrocephala)	74
<input type="checkbox"/> P11681	P11681_RAT	Abelson murine leukemia viral (V-ab...	Abi2, Abi2_mapped, ABL2, ABL2	Rattus norvegicus (Rat)	1,208
<input type="checkbox"/> P09915	P09915_DROME	Abi tyrosine kinase, isoform D	Abi, CG4332, Dmel_CG4032	Drosophila melanogaster (Fruit fly)	1,607
<input type="checkbox"/> P09918	P09918_DROME	Abi tyrosine kinase, isoform C	Abi, CG4332, Dmel_CG4032	Drosophila melanogaster (Fruit fly)	1,705
<input type="checkbox"/> P09916	P09916_DROME	Abi tyrosine kinase, isoform F	Abi, CG4332, Dmel_CG4032	Drosophila melanogaster (Fruit fly)	1,723
<input type="checkbox"/> ABLN22	ABLN22_DROME	Abi tyrosine kinase, isoform B	Abi, CG4332, Dmel_CG4032	Drosophila melanogaster (Fruit fly)	1,638
<input type="checkbox"/> P09918	P09918_DROME	Abi tyrosine kinase, isoform I	Abi, CG4332, Dmel_CG4032	Drosophila melanogaster (Fruit fly)	1,666
<input type="checkbox"/> P09913	P09913_DROME	Abi tyrosine kinase, isoform E	Abi, CG4332, Dmel_CG4032	Drosophila melanogaster (Fruit fly)	1,589
<input type="checkbox"/> P09911	P09911_DROME	Abi tyrosine kinase, isoform H	Abi, CG4332, Dmel_CG4032	Drosophila melanogaster (Fruit fly)	1,504
<input type="checkbox"/> P09903	P09903_DROME	Abi tyrosine kinase, isoform G	Abi, CG4332, Dmel_CG4032	Drosophila melanogaster (Fruit fly)	1,522
<input type="checkbox"/> X13084	X13084_DROME	Abi tyrosine kinase, isoform J	Abi, Dmel_CG4032	Drosophila melanogaster (Fruit fly)	1,620
<input type="checkbox"/> Q9H189	ABL2_HUMAN	Abi interactor 2	ABR2, ARHGAP5	Homo sapiens (Human)	513
<input type="checkbox"/> Q8ZD93	ABL1_HUMAN	Abi interactor 1	ABR1, SSHBP1	Homo sapiens (Human)	308
<input type="checkbox"/> Q73C96	Q73C96_ASHDO	ABL029Wp	ABL029W, AGDS_ABL029W	Ashleya gossypa (strain ATCC 10895 / CBS 109.31 / F05C 9923 / NVAL Y-1036) [Neurospora gossypa]	368
<input type="checkbox"/> Q73C98	Q73C98_ASHDO	ABL001Wp	ABL001W, AGDS_ABL001W	Ashleya gossypa (strain ATCC 10895 / CBS 109.31 / F05C 9923 / NVAL Y-1036) [Neurospora gossypa]	689



# EXAMPLE: UNIPROT

UniProt

BLAST Align Upload data

UniProtKB

abi

Advanced

Search

Help

Correct

Show help for UniProtKB

Submit

Filter by

Reviewed (2,971)

Swiss-Prot

Unreviewed (50,014)

TrEMBL

Popular organisms

Human (545)

Fruit fly (189)

Mouse (187)

A. thaliana (98)

Rat (89)

Other organism

Search

Search terms

Filter "abi" as:

disease (3)

disease (4)

gene name (417)

protein family (9)

protein name (644)

strain (32)

taxonomy (32)

tissue (6)

View by

Taxonomy

Keywords

Gene Ontology

Enzyme class

Pathway

UniRef

Your results in sequence clusters with identity of:

100%, 90% or 30%

Demo

Help video

Columns

BLAST

Align

Download

Add to basket

Download selected (0)

Download all (52,985)

Format:

Tab-delimited

FASTA (canonical)

FASTA (canonical & isoform)

TrEMBL

TS

Excel

TS/CSV

TS/XML

TS/JSON

TS/JSON

Tyrosine protein kinase

Abi-1

Entry	Entry name	Gene names	Organism	Length
P05519	ABL1_HUMAN	ABL1, ABL, TKC7	Homo sapiens (human)	1,130
P05520	ABL1_MOUSE	Abi1, Abi	Mus musculus (Mouse)	1,123
P05522	ABL1_DROME	Abi, ABL-1, Dsh, CG4032	Drosophila melanogaster (Fruit fly)	1,620
P05521	ABL1_MVAB	ABL	Abelson murine leukemia virus	746
P10447	ABL1_PSVHF	ABL	Feline sarcoma virus (strain Hardy-Zuckerman 2)	439
P03049	ABL1_CAEEL	abl-1, MY-1	Caenorhabditis elegans	1,224
P11681	ABL1_CAVIT	ABL	Calliphora vicina (Blue blowfly) (Calliphora erythrocephala)	74
P10901	P10901_RAT	Abelson murine leukemia viral (V-ab-...	Rattus norvegicus (Rat)	1,208
H09915	H09915_DROME	Abi tyrosine kinase, isoform D	Drosophila melanogaster (Fruit fly)	1,607
H09918	H09918_DROME	Abi tyrosine kinase, isoform C	Drosophila melanogaster (Fruit fly)	1,705
H09919	H09919_DROME	Abi tyrosine kinase, isoform F	Drosophila melanogaster (Fruit fly)	1,723
A03042	A03042_DROME	Abi tyrosine kinase, isoform B	Drosophila melanogaster (Fruit fly)	1,638
H09918	H09918_DROME	Abi tyrosine kinase, isoform I	Drosophila melanogaster (Fruit fly)	1,666
H09919	H09919_DROME	Abi tyrosine kinase, isoform E	Drosophila melanogaster (Fruit fly)	1,389
H09919	H09919_DROME	Abi tyrosine kinase, isoform H	Drosophila melanogaster (Fruit fly)	1,304
H09918	H09918_DROME	Abi tyrosine kinase, isoform G	Drosophila melanogaster (Fruit fly)	1,302
X12094	X12094_DROME	Abi tyrosine kinase, isoform J	Drosophila melanogaster (Fruit fly)	1,620
Q09V89	AB12_HUMAN	Abi interactor 2	Homo sapiens (human)	513
Q812P0	AB11_HUMAN	Abi interactor 1	Homo sapiens (human)	508
Q730P6	Q730P6_ASHGO	ABL029Wp	Ashbya gossypii (strain ATCC 10895 / CBS 109.51 / FGSC 9623 / NRRL Y-1058) (Yeast)	568
Q730X8	Q730X8_ASHGO	ABL001Wp	Ashbya gossypii (strain ATCC 10895 / CBS 109.51 / FGSC 9623 / NRRL Y-1058) (Yeast)	695

Exploring Biological Databases

11 / 12

# EXAMPLE: UNIPROT

UniProt

BLAST Align Upload data

Filter by

Reviewed (157)

Unreviewed (388)

Popular organisms

Proteomes

Search terms

View by

UniRef

Demo

Columns

BLAST

Align

Download

Add to basket

Entry	Entry name	Protein names			
P05019	ABL1_HUMAN	Tyrosine-protein kinase ABL1			
Q9N1B9	ABL2_HUMAN	Abl interactor 2	ABL2, ARCBFA	Homo sapiens (Human)	513
Q81ZP0	ABL1_HUMAN	Abl interactor 1	ABL1, SSH3BP1	Homo sapiens (Human)	506
ARUF05	ARUF05_HUMAN	BCR/ABL fusion protein isoform Y3	BCR/ABL fusion, BCR	Homo sapiens (Human)	467
P42684	ABL2_HUMAN	Abl tyrosine-protein kinase 2	ABL2, ABL, ARG	Homo sapiens (Human)	1,182
Q94875	SRB52_HUMAN	Sorbin and SH3 domain-containing pr...	SORBS2, ARCBP2, KIAA0777	Homo sapiens (Human)	1,100
Q8TDW4	CABL1_HUMAN	CDK5 and ABL1 enzyme substrate 1	CABLES1, CABLES	Homo sapiens (Human)	633
A2RQ07	A2RQ07_HUMAN	Bcr-abl1 e19a2 chimeric protein	BCR-ABL1 e19a2	Homo sapiens (Human)	496
ARUF02	ARUF02_HUMAN	BCR/ABL fusion protein isoform X9	BCR/ABL fusion	Homo sapiens (Human)	1,644
ARUF07	ARUF07_HUMAN	BCR/ABL fusion protein isoform Y5	BCR/ABL fusion	Homo sapiens (Human)	1,790
ARUE26	ARUE26_HUMAN	BCR/ABL fusion protein isoform X3	BCR/ABL fusion	Homo sapiens (Human)	1,633
Q98TV7	CABL2_HUMAN	CDK5 and ABL1 enzyme substrate 2	CABLES2, C2orf130	Homo sapiens (Human)	478
A2RQ04	A2RQ04_HUMAN	Bcr-abl1 e14a3 chimeric protein	BCR-ABL1	Homo sapiens (Human)	260
A2RQ03	A2RQ03_HUMAN	Bcr-abl1 e13a3 chimeric protein	BCR-ABL1	Homo sapiens (Human)	235
A2RQ05	A2RQ05_HUMAN	Bcr-abl1 e1a3 chimeric protein	BCR-ABL1	Homo sapiens (Human)	313
A2RQ06	A2RQ06_HUMAN	Bcr-abl1 e6a2 chimeric protein	BCR-ABL1	Homo sapiens (Human)	585
E7EB77	E7EB77_HUMAN	BCR-ABL1 e9a2 variant	BCR-ABL1	Homo sapiens (Human)	448
Q35837	Q35837_HUMAN	ABL1 protein	ABL1	Homo sapiens (Human)	396
Q35836	Q35836_HUMAN	ABL1 protein	ABL1	Homo sapiens (Human)	396
Q13914	Q13914_HUMAN	ABL1 protein	ABL1	Homo sapiens (Human)	64
ARUE27	ARUE27_HUMAN	BCR/ABL fusion protein isoform X4	BCR/ABL fusion	Homo sapiens (Human)	554

Searching in UniProtKB

Term

AB

Term

AB

AND

Organism [OS]

Homo sapiens (Human) [9606]

Term

AB

25

Contact

UniProtKB

About

# Questions?



EVERY TIME YOU ASK A STUPID QUESTION..

God kills a kitten.

[motifake.com](http://motifake.com)