UNIVERSITY OF Southampton

COMPARI MOTIF

CompariMotif: Motif-Motif comparison software

Richard J. Edwards © 2007.

Contents

Figures

Tables

1. Introduction

This manual gives an overview of CompariMotif as implemented in the comparimotif_V3.py module. Because there are many options, this manual will probably not be fully comprehensive but aims to cover the basics and the most useful of the more advanced stuff. If anything is missing or needs clarification, please contact me. The fundamentals are covered in Chapter 2, Fundamentals, including input and output details. Later sections give more details on how the methods work and statistics are generated.

Like the software itself, this manual is a 'work in progress' to some degree. If the version you are now reading does not make sense, then it may be worth checking the website to see if a more recent version is available, as indicated by the Version section of the manual. Check the readme on the website for up-to-date options etc. In particular, default values for options are subject to change and should be checked in the readme.

Good luck.

Rich Edwards, 2007.

1.1. Version

This manual was written to accompany CompariMotif version 3.8. The manual was last edited on 22 November 2012.

1.2. Using this Manual

As much as possible, I shall try to make a clear distinction between explanatory text (this) and text to be typed at the command-prompt etc. Command prompt text will be `written in Courier New` to make the distinction clearer. Program options, also called 'command-line parameters', will be **`written in bold Courier New`** (and coloured **`red`** for fixed portions or **`dark red`** for user-defined portions, such as file names etc.). Command-line examples will be given in (purple) *`italicised Courier New`*. Optional parameters will (if I remember) be `[`in square brackets`]`. Names of files will be marked in normal text by (blue-grey) Times New Roman.

1.3. Why use CompariMotif?

CompariMotif takes two lists of protein motifs and compares them to each other, identifying which motifs have some degree of overlap and describing the relationships between those motifs. It can be used to compare a list of motifs with themselves, their reversed selves, or to a second list of motifs. CompariMotif outputs a table of all pairs of matching motifs, along with their degree of similarity (information content) and their relationship to each other. Details can be found in 2.3.

Short linear motifs (SLiMs) in proteins are functional microdomains of fundamental importance in many biological systems (Neduva and Russell 2005). SLiMs typically consist of a 3 to 10 amino acid stretch of the primary protein sequence, of which as few as two sites may be important for activity. SLiM can usually tolerate a number of alternative amino acids at one or more positions, making precise definitions extremely difficult. CompariMotif can therefore be extremely useful when a new SLiM has been discovered, either by high throughput SLiM discovery (Neduva, Linding et al. 2005; Davey, Shields et al. 2006; Neduva and Russell 2006; Edwards, Davey et al. 2007; Edwards, Davey et al. 2012) or by low throughput experimental studies, by allowing similar motifs to be readily identified from published resources (e.g. ELM (Puntervoll, Linding et al. 2003; Dinkel, Michael et al. 2012) or MiniMotif (Balla, Thapar et al. 2006)). Alternatively, a list of motifs could be compared to itself to identify recurring motifs.

1.4. Getting Help

Much of the information here is also contained in the documentation of the Python modules themselves. A full list of command-line parameters can be printed to screen using the **`help`** option, with short descriptions for each one:

```
python comparimotif_V3.py help
```

General details about Command-line options can be found in the RJE Appendices document included with this download. Details of command-line options specific to CompariMotif can be found in the distributed readme.txt and readme.html files.

If still stuck, then please e-mail me (r.edwards@southampton.ac.uk) whatever question you have. If it is the results of an error message, then please send me that and/or the log file (see 2.3) too.

1.4.1. Something Missing?

As much as possible, the important parts of CompariMotif are described in detail in this manual. If something is not covered, it is generally not very important and/or still under development, and can therefore be safely ignored. If, however, curiosity gets the better of you, and/or you think that something important is missing (or badly explained), please contact me.

1.5. Citing CompariMotif

If using CompariMotif, please cite (Edwards, Davey et al. 2008):

- Edwards RJ, Davey NE & Shields DC (2008): CompariMotif: Quick and easy comparisons of sequence motifs. *Bioinformatics* **24(10):**1307-9.

For analyses on the webserver using specific motif databases, please cite the listed papers for those motif databases.

1.6. Availability and Local Installation

CompariMotif can be run from the CompariMotif webserver, available at http://bioware.ucd.ie/.

CompariMotif is also distributed as a number of open source Python modules as part of the SLiMSuite and SeqSuite packages available from http://bioware.soton.ac.uk/. It should work on any system with Python installed without any extra setup required. If you do not have Python, you can download it free from www.python.org at http://www.python.org/download/. The modules are written in Python 2.5. The Python website has good information about how to download and install Python but if you have any problems, please get in touch and I will help if I can.

2. Fundamentals

2.1. Running CompariMotif

2.1.1. The Basics

If you have python installed on your system, you should be able to run CompariMotif directly from the command line in the form:

```
python comparimotif_V3.py motifs=FILENAME
```

For the example provided in the distribution:

```
python comparimotif_V3.py motifs=comparimotif_eg.motifs
```

If the motif file is to be compared to itself, then no other commands are needed. If a second file is to be compared, however, this should be specified using the `searchdb=FILENAME` command:

```
python comparimotif_V3.py motifs=file1 searchdb=file2
```

IMPORTANT: If filenames contain spaces, they should be enclosed in double quotes: `motifs="example file"`. That said, it is recommended that files do not contain spaces as function cannot be guaranteed if they do.

A CompariMotif webserver is also available at http://bioware.ucd.ie (Edwards, Davey et al. 2008). See 2.5 for details.

2.1.2. Options

Command-line options are suggested in the following sections. General details about Command-line options can be found in the PEAT Appendices document included with this download. Details of command-line options specific to Slim Pickings can be found in the distributed readme.txt and readme.html files. These may be given after the run command, as above, or loaded from one or more *.ini files (see PEAT Appendices for details).

2.1.3. Running in Windows

If running in Windows, you can just double-click the comparimotif_V3.py file and use the menu prompts to navigate through the program. It is recommended to use the `win32=T` option. (Place this command in a file called comparimotif.ini.)

2.2. Input

The main input for CompariMotif is a motif file and an optional second motif file to compare to these motifs. If no second motif file is given (`searchdb=FILE`) then the first motif file (`motifs=FILE`) will be compared to itself.

2.2.1. Motif Input Formats

The recommended motif input format is PRESTO format. This should have a single line per motif, with the format:

```
Name Sequence # Comments
```

Comments are optional but anything after the # will be ignored.

Alternative allowed formats include: a fasta format file with motif/peptide names and sequences in the usual fasta format; a raw list of peptides/motifs (in this case the name and sequence will the same); ELM class downloads; SLiMFinder output; SLiMDisc output; TEIRESIAS output; Slim Pickings output. Any delimited file with 'Name' and 'Pattern' fields should also be recognised. Additional input formats can be added on request.

In all cases, the motif should be a peptide sequence using the standard single letter amino acid codes and the following regular expression rules:

- **A** = single fixed amino acid.

- `[AB]` = ambiguity, **A** or **B**. Any number of options may be given, e.g. `[ABC]` = **A** or **B** or **C**. `[^A]` = not **A**.
- `<R:m:n>` = At least **m** of a stretch of **n** residues must match **R**, where **R** is one of the above regular expression elements (single or ambiguity).
- `X` or `.` = Wildcard positions (any amino acid)
- `X{m,n}` or `.{m,n}` = At least **m** and up to **n** wildcards.
- `R{n}` = **n** repetitions of **R**, where **R** is any of the above regular expression elements.
- `(AB|CD)` = **AB** or **CD**.
- `(ABC)` = **ABC** in any order (**BAC**, **CAB** etc.).
- `^` = Beginning of sequence
- `$` = End of sequence
- `?` = Zero or one occurrence. Will be replaced with `{0,1}`.

E.g. `[IL][^P]X{3}RG` means: "leucine or isoleucine, followed by anything but proline, followed by three residues, followed by arginine followed by glycine".

E.g. (2) `^<KR:3:5>P` means: "three of the first five amino acids must be lysine or arginine; the sixth amino acid must be proline".

2.2.2. Motif Splitting

Version 3 saw a shift towards using the same underlying motif handling modules as the rest of SLiMSuite – particularly SLiMFinder and SLiMSearch. This has necessitated a slight simplification of the motifs that can be treated as a single entity. In particular, variable-length non-wildcards, "3of5" `<R:m:n>` elements and `(AB|CD)` ambiguities are now split into multiple motifs. This will produce a new files named after the input, FILE.split.motifs, and the split motifs will have a letter suffix a, b, c *etc.* with "variant 1", "variant 2" *etc.* appended to the motif descriptions.

E.g. `Motif [IL]{1,2}[^P].(RG|K)` will be split to form:

```
Motif_a [IL][^P].RG
Motif_b [IL][^P].K
Motif_c [IL][IL][^P].RG
Motif_d [IL][IL][^P].K
```

2.2.3. Advanced Input Options

If `reverse=T` is used, the first file only will be reversed before comparison. If a self-comparison is made using `reverse=T` then reversed motifs will be compared to the original "forward" motif file. Input motifs can be filtered to remove short or highly degenerate motifs (see 0 for details).

2.2.4. SLiM Database Files

CompariMotif provides a number of pre-defined motif datasets in the correct format for use. When using results from these datasets, please always cite the relevant paper. These datasets are given in Table 2.1 and are available from the CompariMotif webserver, where they can be selected via a drop-down list box and are available to download through the help pages.

If you have additional motif databases you would like to see available, please contact me. In addition, individual motifs may be submitted for inclusion in the miscellaneous literature motif file at: http://bioware.ucd.ie/~comparimotif/MotifBrowser/index.html.

Table 2.1. SLiM Databases provided for CompariMotif searches.

Database	Description	Reference	Motif File
ELM	The Eukaryotic Linear Motif database provides a number of high quality annotated SLiMs with known occurrences. **Note.** Some motifs have been split into _a and _b to be compatible with CompariMotif input formats. Such motifs are marked ***Modified*** in their descriptions.	(Puntervoll, Linding et al. 2003)	ELM.motifs
MiniMotif	Another database of SLiMs from all organisms. This has less annotation than ELM but more motifs. These motifs have been reformatted to conform to standard regular expressions.	(Balla, Thapar et al. 2006)	MnM.motifs
Phospho-MotifFinder	Motifs from the PhosphoMotif Finder database of HPRD. Motifs are labelled KIN for Kinase / Phosphatase motifs or BIND for binding motifs. _Y indicates a tyrosine motif, while _ST indicated serine/threonine. The number part of the motif identifier is arbitrary and has no link to the website. **Note**. All these motifs are phosphorylation motifs and, as such, have a key Ser, Thr or Tyr position. These are not given special treatment in CompariMotif and the user should pay special attention to whether the appropriate residue is included in the match.	(Amanchy, Periaswamy et al. 2007)	phosphomotif.motifs
Misc. Literature	Miscellaneous motifs collected from the literature. (These include pubmed links to the relevant paper but we cannot guarantee the accuracy of the motifs or their descriptions.)	See file.	literature.motifs
Combined Literature	A combined database of the above sources. The source database is indicated in the motif description: [ELM] for ELM, [MnM] for MiniMotif and [PMF] for PhosphoMotif Finder.	See above.	combined.motifs
Neduva & Russell	Predicted interaction SLiMs from the high-throughput study of Neduva *et al.* (2005). The motif name indicates what part of the study it is from. All names begin NR, followed by a two-letter code for the species and a one-letter code denoting Domain-level datasets or Protein-level datasets. (See paper for details.) Species codes are: Ce, *C. elegans*; Dm, *D. melanogaster*; Hs, *H. sapiens*; Sc, *S. cerevisiae*.	(Neduva, Linding et al. 2005)	Ned2005_Sig.motifs

2.2.5. Use of DNA Motifs

Although explicitly designed for protein motifs, there is no reason why DNA motifs cannot be compared with CompariMotif, as long as the correct regular expression notation is maintained. CompariMotif version 3.3 introduced a DNA motif option, `dna=T/F`. When `dna=T`, only the four DNA nucleotides G, A, T and C are used and information content (see 3.2) is adjusted accordingly. Uridines (U) are converted to thymidines (T). The following additional replacements (based on official IUB/IUPAC abbreviations) are made:

- N --> . (any)
- R --> G A (purine)
- Y --> T C (pyrimidine)
- K --> G T (keto)
- M --> A C (amino)
- S --> G C (strong)
- W --> A T (weak)
- B --> G T C
- D --> G A T
- H --> A C T
- V --> G C A

2.2.6. Amino acid frequencies

By default, all information content calculations (see 3.2) assume uniform amino acid frequencies (or nucleotide frequencies if dna=T.) This is because the motifs themselves are independent of amino acid bias. The chance of any given motif having a match to a database is more a reflection of the frequency of amino acids in motifs, rather than proteomes. (Indeed, rare amino acids might be more likely to occur in motifs because they are rare and therefore look unusual to discovery algorithms.) Nevertheless, it is sometimes desirable to weight results according to amino acid frequencies, and this can be done using the aafreq=FILE command. The FILE can be a FASTA sequence file from which frequencies are to be calculated, or a delimited file containing amino acid frequencies:

```
AA     FREQ
A      0.074
C      0.033
…
Y      0.033
```

The impact of weighted frequencies is discussed in 3.2.

2.3. Output

The main output for CompariMotif is delimited text file containing the following fields:

Table 2.2. Field headings for main CompariMotif output file.

Field	Description
File1	Name of motif file 1 (motifs=FILE). [outstyle=multi only]
File2	Name of file 2 (searchdb=FILE). [outstyle=multi only]
Name1	Name of motif from motif file 1.
Name2	Name of motif from motif file 2.
Motif1	Motif (pattern) from motif file 1.
Motif2	Motif (pattern) from motif file 2.
Sim1	Description of motif1's relationship to motif2.
Sim2	Description of motif2's relationship to motif1.
Match	Regular expression of match between motifs. Upper case positions indicate an exact match, while lower case positions have some degree of degeneracy difference between the two motifs. Mismatches are marked with an asterisk.
MatchPos	Number of matched positions between motif1 and motif2 (>=misahre=X).
MatchIC	Information content of matched positions.
NormIC	MatchIC as a proportion of the maximum possible MatchIC. If this is 1.0 then the match is a good as could possibly be achieved.
CoreIC	MatchIC as a proportion of the maximum possible IC in the matched region only (coreic=T).
Score	Heuristic score (MatchPos x NormIC) for ranking motif matches.
Info1	Information Content of motif1 (if motific=T).
Info2	Information Content of motif2 (if motific=T).
Desc1	Description of motif1 (if motdesc=1 or motdesc=3).
Desc2	Description of motif1 (if motdesc=1 or motdesc=2).

Details of relationship descriptions and information content calculations can be found in Chapter 3.

Note that Info1 and Info2 are only output if the motific=T option is used. Desc1 and Desc2 are controlled by motdesc=X. This outputs motif descriptions depending on the value of X, as follows: 0 = Neither; 1 = Motif1 only; 2 = Motif2 only; 3 = both. The default is 3 (both).

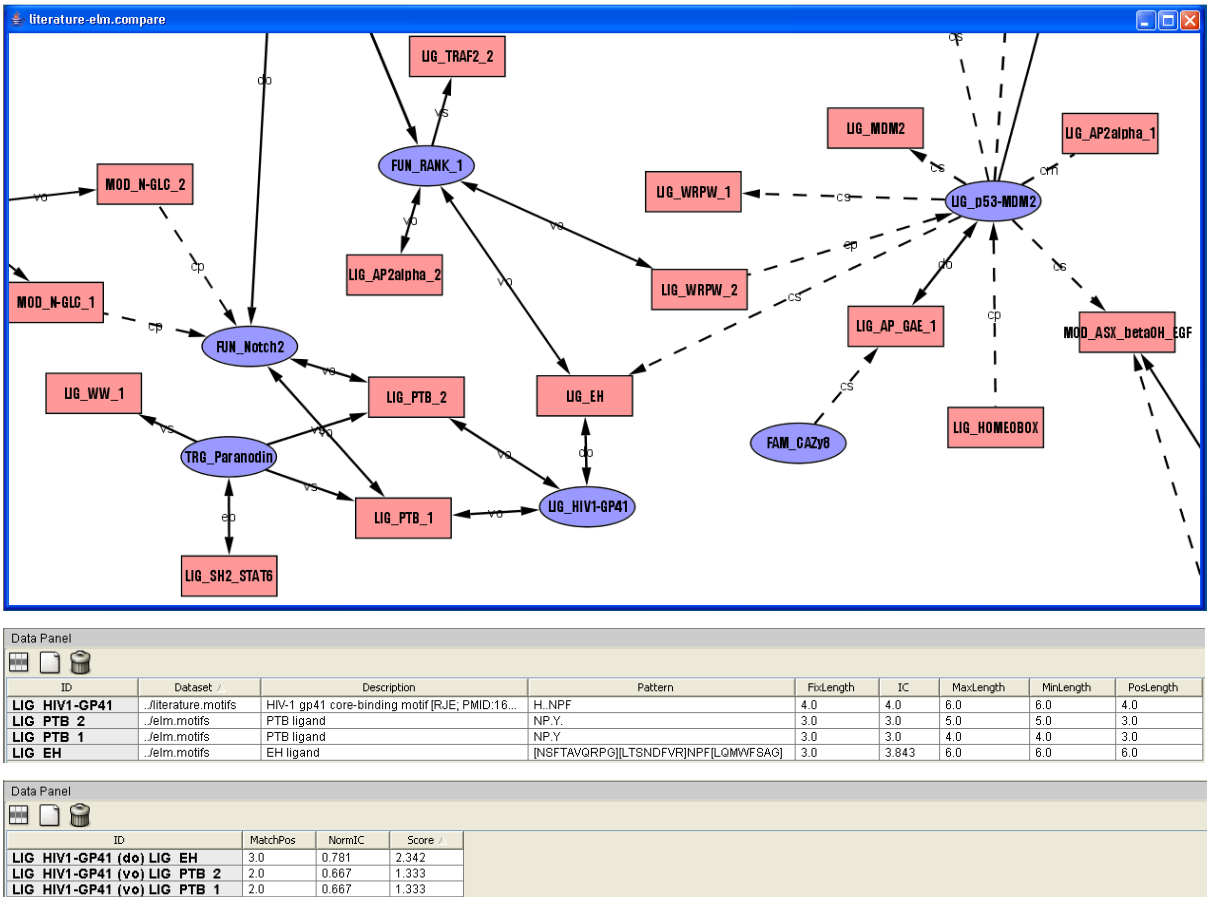
2.3.1. Output Styles

With the exception of the file names, which are only output if outstyle=multi, the above is the output for the default "normal" output style. If outstyle=single then only statistics for motif2 (the searchdb motif) are output as this is designed for searches using a single motif against a motif database. If outstyle=normalsplit or outstyle=multisplit then motif1 information is grouped together, followed by motif2 information, followed by the match statistics, e.g. **[File1,] Name1, Motif1, Sim1, [Info2,] [Desc1,] [File2,] Name2, Motif2, Sim2, [Info2,] [Desc2,] Match, MatchPos, MatchIC, NormIC, Score**

2.3.2. Optional Motif Information File

In addition to the main outputs, the motinfo=FILE command will output a motif summary table into the specified filename. This file consists of the following fields:

Field	Description
Motif	The name of the motif.
Pattern	The regular expression pattern of the motif (see **2.2.1Motif Input Formats**).
Description	The description of the motif.
MaxLength	Maximum length of the motif in terms of non-wildcard positions.
MinLength	Minimum length of the motif in terms of non-wildcard positions.
FixLength	Maximum Length of motif in terms of fixed positions.
FullLength	Maximum length of the motif, including wildcard positions.
Expect	The expected number of times the motif will occur in the search database given (searchdb=FILE).
IC	Information Content of motif (if motific=T).



ID	Dataset	Description	Pattern	FixLength	IC	MaxLength	MinLength	PosLength
LIG HIV1-GP41	../literature.motifs	HIV-1 gp41 core-binding motif [RJE; PMID:16...	H..NPF	4.0	4.0	6.0	6.0	4.0
LIG PTB 2	../elm.motifs	PTB ligand	NP.Y.	3.0	3.0	5.0	5.0	3.0
LIG PTB 1	../elm.motifs	PTB ligand	NP.Y	3.0	3.0	4.0	4.0	3.0
LIG EH	../elm.motifs	EH ligand	[NSFTAVQRPG][LTSNDFVR]NPF[LQMWFSAG]	3.0	3.843	6.0	6.0	6.0

ID	MatchPos	NormIC	Score
LIG HIV1-GP41 (do) LIG EH	3.0	0.781	2.342
LIG HIV1-GP41 (vo) LIG PTB 2	2.0	0.667	1.333
LIG HIV1-GP41 (vo) LIG PTB 1	2.0	0.667	1.333

Figure 2.1. Partial Cytoscape visualisation of CompariMotif relationships.
Motifs from the query and search database file are displayed as blue ellipses and red rectangles, respectively. Edges represent relationships identified by CompariMotif. Arrows indicate the direction of any parent/subsequence relationship. Relationship codes match those output by the webserver. Motif and match details are included in the XGMML file and can be viewed in the Cytoscape Data Panel.

2.3.3. Cytoscape XGMML File

CompariMotif also outputs an XGMML file (*.compare.xgmml) that can be imported directly into Cytoscape (Shannon, Markiel et al. 2003) (http://www.cytoscape.org) for visualisation of the results (Figure 2.1). This file contains all the necessary node (motif) and edge (match) data found in the results table, which can be viewed for selected nodes/edges using the Cytoscape Data Panel. The file can be uploaded into Cytoscape using the **File -> Import -> Network (Multiple File Types)** command (**CTRL+L**). When first loaded, nodes will be displayed in a simple, uninformative, grid. Use one of the Cytoscape Layouts (e.g. **Layout -> yFiles -> Organic**) to make it clearer. See Cytoscape documentation for details.

2.3.4. Unmatched motifs

If **unmatched=T** then a list of motifs from the input motif file (but **not** the searchdb file) that have no CompariMotif matches meeting the chosen scoring thresholds will be output as a list into a *.unmatched.txt file.

2.3.5. Sortable table output (webserver only)

The CompariMotif webserver has an additional output, which is a sortable table generated from the raw text delimited text output. See below for details.

2.4. Commandline Options

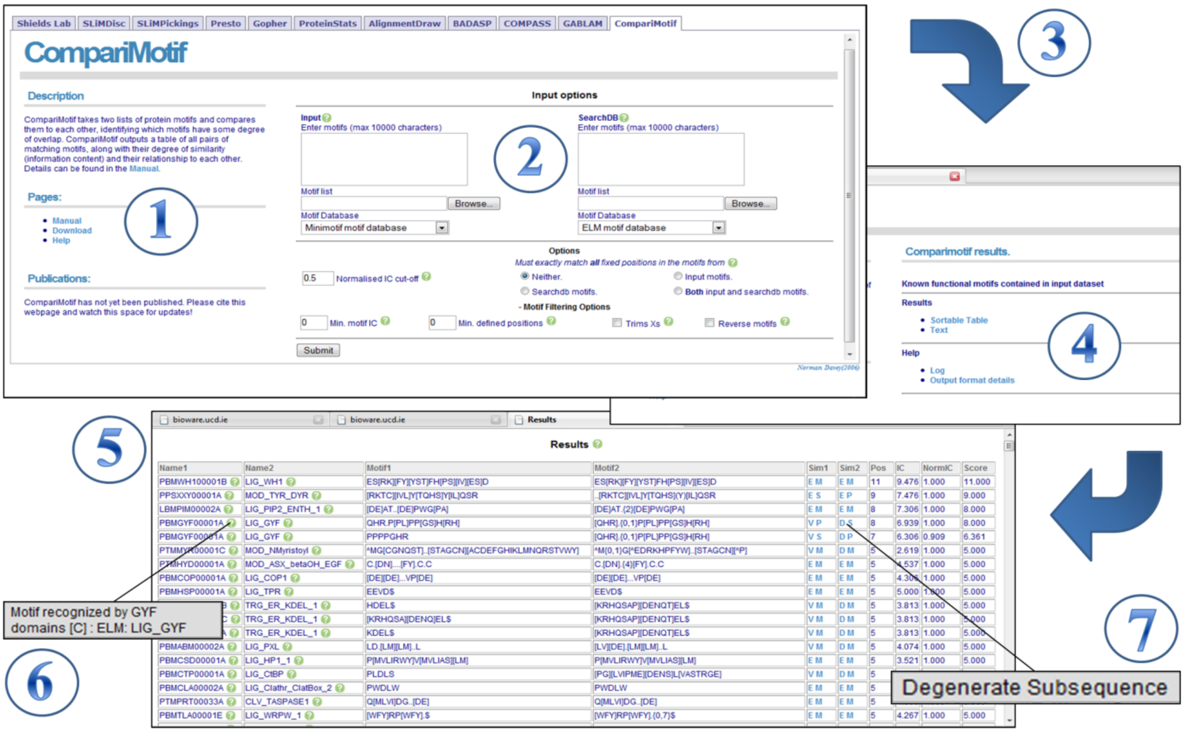
Table 2.3 lists the commandline options for CompariMotif. Please see also the PEAT Appendices document for additional general commandline options and the RJE_SEQ Manual for further input data options. Beginners will probably want to leave the default settings unchanged.

2.5. The CompariMotif Webserver

The CompariMotif webserver is available at bioware.ucd.ie. Click on the CompariMotif tab to access the main input user form (Figure 2.2). Note that a restricted set of options are available. These include all the recommended search parameters.

Table 2.3. CompariMotif Commandline Options.

Option	Description	Default
	Basic Input/Output Options	
motifs=FILE	Loads motifs from FILE	[None]
searchdb=FILE	Optional second motif file to compare.	[None]
dna=T/F	Whether motifs should be considered as DNA motifs	[False]
resfile=FILE	Name of results file, FILE.compare.tdt. *Default named after motifs and searchdb files.	[*]
motinfo=FILE	Filename for output of motif summary table	[None]
motific=T/F	Output Information Content for motifs.	[False]
coreic=T/F	Whether to output normalised Core IC	[True]
unmatched=T/F	Whether to output lists of unmatched motifs (not from searchdb) into *.unmatched.txt	[False]
	Motif Comparison Parameters	
minshare=X	Min. number of non-wildcard positions for motifs to share.	[2]
normcut=X	The minimum normalise IC allowed for a match.	[0.5]
matchfix=X	If >0 must exactly match *all* fixed positions in the motifs from: - 1: input (motifs=FILE) motifs - 2: searchdb motifs - 3: *both* input and searchdb motifs	[0]
ambcut=X	Max number of choices in ambiguous position before replaced with wildcard (0=use all)	[10]
overlaps=T/F	Whether to include overlapping ambiguities (e.g. [KR] vs [HK]) as match	[True]
	Advanced Input Parameters	
minic=X	Min information content for motif	[2.0]
minfix=X	Min number of fixed positions for a motif to contain	[0]
minpep=X	Min no. of defined positions	[2]
trimx=T/F	Whether to trim leading and trailing wildcards	[False]
nrmotif=T/F	Remove redundancy in input motifs (partial/full identities)	[False]
reverse=T/F	Reverse the first set of motifs.	[False]
mismatches=X	<= X mismatches of positions can be tolerated.	[0]
aafreq=FILE	Use FILE to replace uniform AAFreqs (FILE can be sequences or aafreq)	[None]
	Advanced Input Parameters	
xgmml=T/F	Whether to output XGMML format results	[True]
pickle=T/F	Whether to load/save pickle following motif loading/filtering	[False]
motdesc=X	Sets which motifs have description outputs (0-3 as matchfix option)	[3]
outstyle=X	Sets the output style for the resfile: - normal = all standard stats are output - multi = designed for multiple appended runs. File names also output - single = designed for searches of a single motif vs a database. Only motif2 stats are output - reduced = motifs do not have names or descriptions - normalsplit/multisplit = as normal/multi but stats are grouped by motif rather than by type	[normal]



CompariMotif

Description

CompariMotif takes two lists of protein motifs and compares them to each other, identifying which motifs have some degree of overlap. CompariMotif outputs a table of all pairs of matching motifs, along with their degree of similarity (information content) and their relationship to each other. Details can be found in the Manual.

Pages:

- Manual
- Download
- Help

Publications:

CompariMotif has not yet been published. Please cite this webpage and watch this space for updates!

Input options

Input ?
Enter motifs (max 10000 characters)
Motif list
Motif Database: Minimotif motif database

SearchDB ?
Enter motifs (max 10000 characters)
Motif list
Motif Database: ELM motif database

Options

0.5 Normalised IC cut-off ?

*Must exactly match **all** fixed positions in the motifs from* ?

● Neither. ○ Input motifs.
○ Searchdb motifs. ○ **Both** input and searchdb motifs.

- Motif Filtering Options

0 Min. motif IC ? 0 Min. defined positions ? □ Trims Xs ? □ Reverse motifs ?

Comparimotif results.

Known functional motifs contained in input dataset

Results

- Sortable Table
- Text

Help

- Log
- Output format details

Results ?

Name1	Name2	Motif1	Motif2	Sim1	Sim2	Pos	IC	NormIC	Score
PBMWH100001B ?	LIG_WH1 ?	ES[RK][FY][YST]FH[PS][IV][ES]D	ES[RK][FY][YST]FH[PS][IV][ES]D	E M	E M	11	9.476	1.000	11.000
PPSXXY00001A ?	MOD_TYR_DYR ?	[RKTC][IVL]Y[TQHS]Y[IL]QSR	..[RKTC][IVL]Y[TQHS](Y)[IL]QSR	E S	E P	9	7.476	1.000	9.000
LBMPIM00002A ?	LIG_PIP2_ENTH_1 ?	[DE]AT..[DE]PWG[PA]	[DE]AT.{2}[DE]PWG[PA]	E M	E M	8	7.306	1.000	8.000
PBMGYF00001A ?	LIG_GYF ?	QHR.P[PL]PP[GS]H[RH]	[QHR].{0,1}P[PL]PP[GS]H[RH]	V P	D S	8	6.939	1.000	8.000
PBMGYF00001A ?	LIG_GYF ?	PPPPGHR	[QHR].{0,1}P[PL]PP[GS]H[RH]	V S	D P	7	6.306	0.909	6.361
PTMMYR00001C ?	MOD_NMyristoyl ?	^MG[CGNQST]..[STAGCN][ACDEFGHIKLMNQRSTVWY]	^M{0,1}G[^EDRKHPFYW]..[STAGCN][^P]	V M	D M	5	2.619	1.000	5.000
PTMHYD00001A ?	MOD_ASX_betaOH_EGF ?	C.[DN]....[FY].C.C	C.[DN].{4}[FY].C.C	E M	E M	5	4.537	1.000	5.000
PBMCOP00001A ?	LIG_COP1 ?	[DE][DE]...VP[DE]	[DE][DE]...VP[DE]	E M	E M	5	4.306	1.000	5.000
PBMHSP00001A ?	LIG_TPR ?	EEVD$	EEVD$	E M	E M	5	5.000	1.000	5.000
[illegible] ?	TRG_ER_KDEL_1 ?	HDEL$	[KRHQSAP][DENQT]EL$	V M	D M	5	3.813	1.000	5.000
[illegible] ?	TRG_ER_KDEL_1 ?	[KRHQSA][DENQ]EL$	[KRHQSAP][DENQT]EL$	V M	D M	5	3.813	1.000	5.000
[illegible] ?	TRG_ER_KDEL_1 ?	KDEL$	[KRHQSAP][DENQT]EL$	V M	D M	5	3.813	1.000	5.000
PBMABM00002A ?	LIG_PXL ?	LD.[LM][LM]..L	[LV][DE].[LM][LM]..L	V M	D M	5	4.074	1.000	5.000
PBMCSD00001A ?	LIG_HP1_1 ?	P[MVLIRWY]V[MVLIAS][LM]	P[MVLIRWY]V[MVLIAS][LM]	E M	E M	5	3.521	1.000	5.000
PBMCTP00001A ?	LIG_CtBP ?	PLDLS	[PG][LVIPME][DENS]L[VASTRGE]	V M	D M	5	[illegible]	[illegible]	[illegible]
PBMCLA00002A ?	LIG_Clathr_ClatBox_2 ?	PWDLW	PWDLW	E M	E M	5	[illegible]	[illegible]	[illegible]
PTMPRT00033A ?	CLV_TASPASE1 ?	Q[MLVI]DG..[DE]	Q[MLVI]DG..[DE]	E M	E M	5	[illegible]	[illegible]	[illegible]
PBMTLA00001E ?	LIG_WRPW_1 ?	[WFY]RP[WFY].$	[WFY]RP[WFY].{0,7}$	E M	E M	5	4.267	1.000	5.000

Motif recognized by GYF domains [C] : ELM: LIG_GYF

Degenerate Subsequence

Figure 2.2. CompariMotif webserver.
1. Links to help pages and further information. 2. The main input form. Users can enter motifs directly into the text boxes, upload motif files, or use databases provided. In this example, Minimotif is being compared to ELM. Further options can be set to customise search parameters. The help buttons by each option give a brief description of its function. 3. Upon clicking “Submit”, a new page will open with the search progress. Upon completion, the CompariMotif results page (4) will open. 4. The webserver returns results as a simple unformatted text and also a sortable table (5). In addition, the program log is returned, which is useful for identifying formatting errors etc. 5. The main results table opens in a separate window and can be sorted on any column. By default, the “Score” column is used; matches with high Scores tend to be of better quality. 6. Mousing over the ? by each motif name will reveal the motif description (dependent on input format). 7. Match similarities are reduced to a code, which is expanded by positioning the mouse over them.

3. Motif Comparisons

This chapter gives more details on the inner workings of CompariMotif.

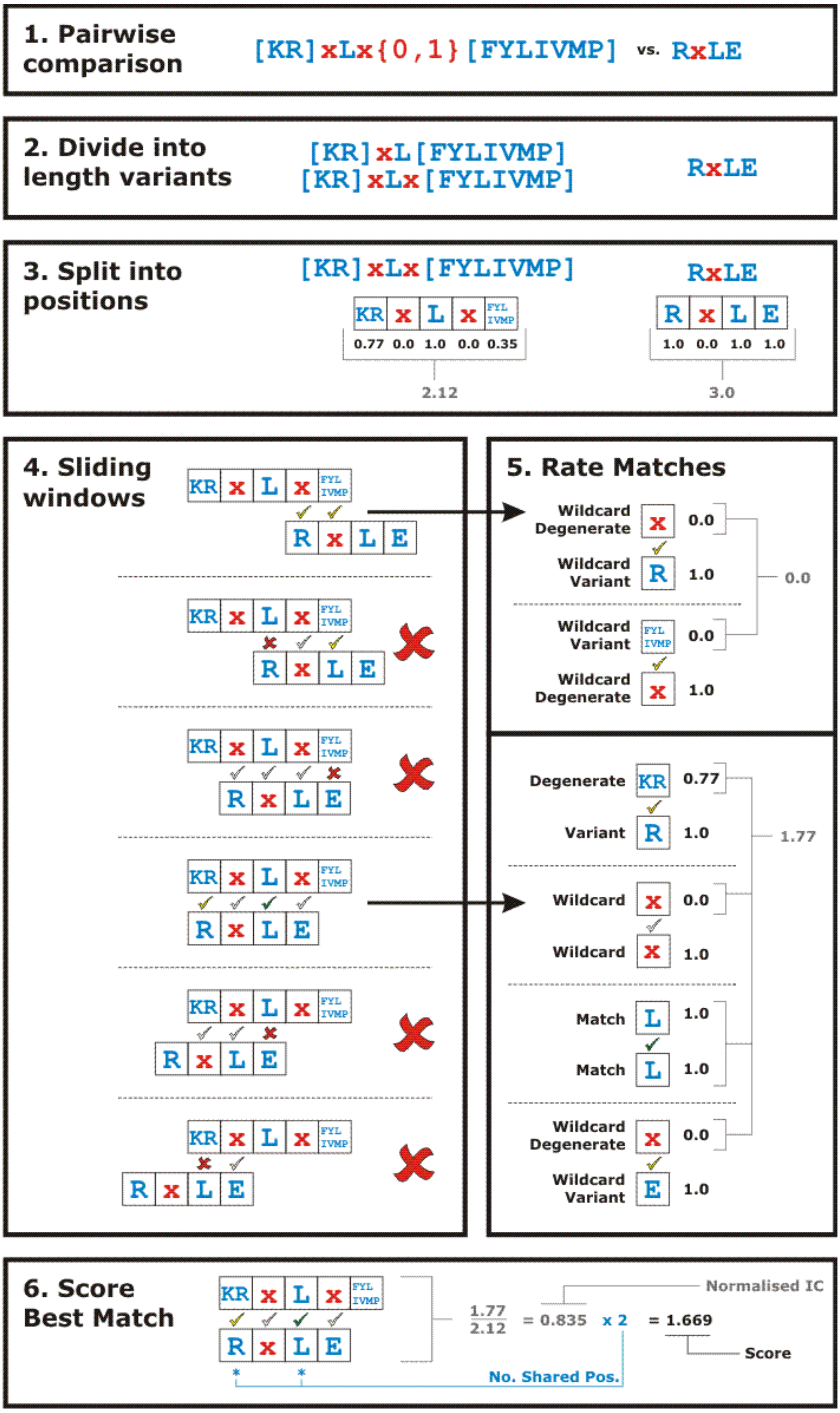


Figure 3.1. Overview of CompariMotif.
Details can be found in the text.

3.1. How CompariMotif Works

An overview of CompariMotif is given in Figure 3.1. Motifs are first compared for precise matches. If these are not found, then CompariMotif adopts a sliding window comparison in which every possible overlap between (variants of) the two motifs are compared against each other. Matches must meet a minimum match requirement determined by the **minshare=X**, **normcut=X** and **matchfix=X** options (see 3.1.2). Fixed positions in motifs are often more important that ambiguous ones, especially when the motif has been experimentally determined. For this reason, it is also possible to stipulate that all fixed positions in one or other motif (or both) match exactly to fixed positions in the compared motifs. This is controlled using the **matchfix=X** option.

3.1.1. Single Position Comparisons

For every comparison, each position in each motif is then rated according to its relationship with the compared position in the other motif:

- Match = perfect fixed position match
- Wildcard = wildcard in both motifs
- Wildcard variant = wildcard in compared motif but not in focal motif
- Wildcard degenerate = wildcard in focal motif but not in compared motif
- Ambiguous Match = ambiguities in both motifs comparing the same amino acids
- Degenerate Ambiguity = ambiguity in both motifs but the compared site is a subset of the ambiguity in the focal site
- Variant Ambiguity = ambiguity in both motifs but the focal site is a subset of the ambiguity in the compared site
- Degenerate = ambiguity in focal motif but fixed position in compared motif
- Variant = a fixed variant in the focal motif of a degenerate position in the compared motif
- Overlapping ambiguity = ambiguity in both motifs where 1+ amino acids overlap but each ambiguity also contains amino acids not in the other
- Bad ambiguity = ambiguity in focal motifs sharing no amino acids in common with compared motif
- Ambiguity mismatch = fixed position in focal motif that does not match ambiguity in compared motif
- Mismatch = different fixed positions in each motifs

Each positional comparison is then given an information content (IC) rating, if it is a “good” match. This is the lower IC out of the two positions compared. *E.g.* a fixed variant matching an ambiguity will take the IC of the ambiguity.

3.1.2. Selecting Pairwise Matches

The entire pairwise comparison is then rated for:

- Number of matching positions, allowing for degeneracy
- Number of exactly matching fixed positions
- Match Information content (IC), which is the sum of IC over all matched positions
- Number of incompatible positions (*e.g.* Bad ambiguities and mismatches)

The comparison is then rejected as a potential match if one of the following conditions is met:

- There are any incompatible positions. (If the **mismatches=X** option is used, this is relaxed.)
- The number of matched positions is less than that stipulated by **minshare=X**.
- The **matchfix=X** option is used and the relevant motif(s) in the comparison does not have exact matches at all its fixed positions.

- The normalised IC is below that set by `normcut=X`.

When a motif has multiple length variants and/or "NofM" elements, each possible variant is compared.

Multiple variants and/or sliding windows can produce multiple matches that meet the acceptance criteria. In this case, the match with the best information content is used. In the case of tied information content, matches are assessed by the number of matching positions and then the number of exactly matching fixed positions. The earlier comparison made is considered "best" if all these stats tie.

3.1.3. Defining Motif Relationships

The best match is then considered to define the relationship between the two motifs. These relationships are comprised of the following keywords:

- Match type keywords identify the type of relationship seen:
  - **Exact** = all the matches in the two motifs are precise
  - **Variant** = the focal motif contains only exact matches and subvariants of degenerate positions compared to the other motif
  - **Degenerate** = the focal motif contains only exact matches and degenerate versions of positions in the other motif
  - **Complex** = some positions in the focal motif are degenerate versions of positions in the compared motif, while others are subvariants of degenerate positions
  - **Ugly** = the two motifs match at partially (but not wholly) overlapping ambiguous positions (e.g. [AGS] vs [ST]). Such matches can be excluded using `overlaps=F`. (Version 3.8 onwards only.)
- Match length keywords identify the length relationships of the two motifs:
  - **Match** = both motifs are the same length and match across their entire length
  - **Parent** = the focal motif is longer and entirely contains the compared motif
  - **Subsequence** = the focal motif is shorter and entirely contained within the compared motif
  - **Overlap** = neither motif is entirely contained within the other

This gives twenty possible classifications for each motif's relationship to the compared motif (Figure 3.2).

3.2. Information Content

Information content (IC) is calculated for each position based on a modification of Shannon's classical Information Content (Shannon 1997):

$$IC_i = -\log_N(f_a)$$

where IC_i is the information content for position i, f_a is the summed frequency for the amino acids (or nucleotides) at position i and N is number of amino acids (or nucleotides) in the alphabet, *i.e.* N=4 for DNA and N=20 for proteins. The IC for the motif is simply this score summed over all positions.

This score is essentially a rescaling of Shannon's self-information such that a wildcard receives 0.0 and a fixed position scores 1.0 when a uniform frequency distribution is used. Ambiguous positions are given a value between 0.0 and 1.0. When non-uniform frequencies are used (see 2.2.6), fixed rare amino acids ($f_a < 1/N$) will score above 1.0, while fixed common amino acids ($f_a > 1/N$) will score less than 1.0.

For motif matches, the *lowest* information content for any compared pair of elements is used as the information content for that part of the match. As a result, "matches" at wildcard position contribute nothing to the IC for the match, while for fixed sites matching ambiguous positions, the IC for the degenerate (ambiguous) position is used. The match IC, IC_m, is the sum of the lower IC_i values for each position. The "best" IC for any given pair of motifs is therefore equal to the lower IC of the two motifs.

For the "Normalised IC" output, the IC of the match is divided by this number. This normalises the IC_m against the fact that longer and less degenerate motifs will tend to produce higher IC matches. Matches with a normalised IC below that set by `normcut=X` will not be returned.

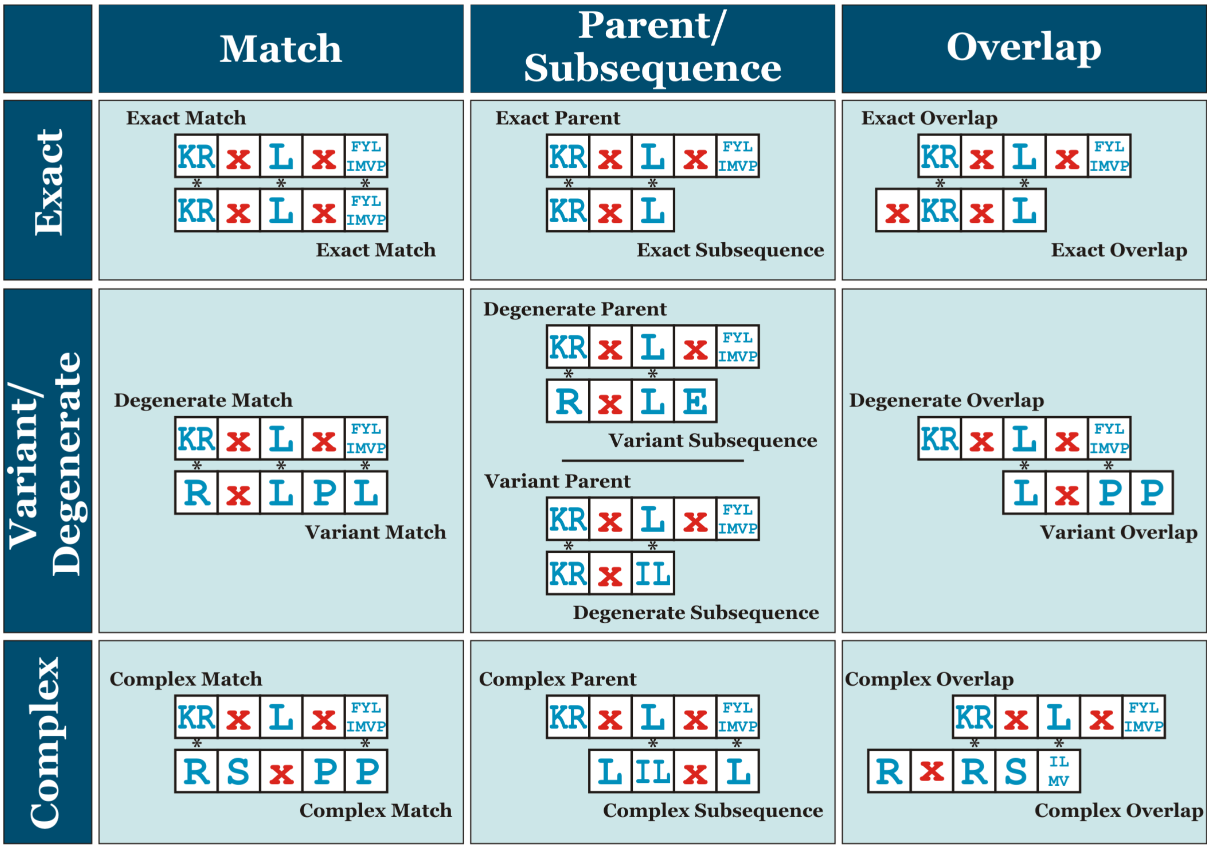


Figure 3.2. CompariMotif Match Type Examples.
Examples for each of the sixteen match types. In each case, the "query" motif [KR]xLx[FYLIMVP] is compared to an invented motif for illustration. Because of the natural relationship between parent/subsequence and variant/degenerate matches, these have been grouped in the figure. Matched positions that contribute towards the number of matched positions (i.e. those not involving a wildcard position) are marked with an asterisk. **Exact Match:** All positions are identical and the match spans the full length of both motifs; **Variant/Degenerate Match:** The match spans the full length of both motifs. All of the positions of the query are either the same as the match (X v. X and L v. L) or more degenerate ([KR] v. R, X v. P & [FYLIMVP] v. L) and so it is classed degenerate. Likewise, all positions in the other motif, RxLPL, are either identical to the query or variants of the query positions, so it is classed as variant; **Complex Match:** Again, the match spans the full length of both motifs. This time, each motif has some positions that are more degenerate than in the other motif. *i.e.* The query is a variant for the L v. X position but more degenerate at all other positions; **Exact Parent/Subsequence:** The [KR]xL motif is entirely and exactly contained within the query; **Degenerate Parent/Variant Subsequence:** The RxLE motif is entirely contained within the query. At two positions, however, ([KR] v. R & X v. E) the query is more degenerate and is hence a "degenerate parent", while RxLE is a "variant subsequence"; **Variant Parent/Degenerate Subsequence:** This time it is the query that is the variant in one position (L v. IL) and so the variant/degenerate labels are swapped; **Complex parent/subsequence:** the L[IL]xL motif is less degenerate at two positions (X v. L & [FYLIMVP] v. L) but more degenerate at one (L v. [IL]) and so the relationship is "complex"; **Exact overlap:** Neither motif is entirely contained within the other but the positions overlapping match exactly; **Degenerate/variant overlap:** Neither motif is entirely contained within the other. The first P of LxPP is a variant of the [FYLIMVP] in the query, while the other two matches (an L and an X) are exact, therefore the query is "degenerate" and LxPP is "variant"; **Complex overlap:** Neither motif is entirely contained within the other and both contain positions that are degenerate when compared to the matching position in the other motif ([KR] v. R & X v. S are degenerate in the query, L v. [ILMV] is degenerate in RxRS[ILMV]). **"Ugly"** matches are not show. These are like **degenerate/variant** or **complex** pairings except they feature 1+ positions where ambiguities partially overlap (see text).

3.3. Score

The Score assigned to a match is a simple heuristic of the match IC multiplied by the normalised IC. This is a useful metric for ranking matches, as the best matches tend to get the best scores.

3.4. Example Application

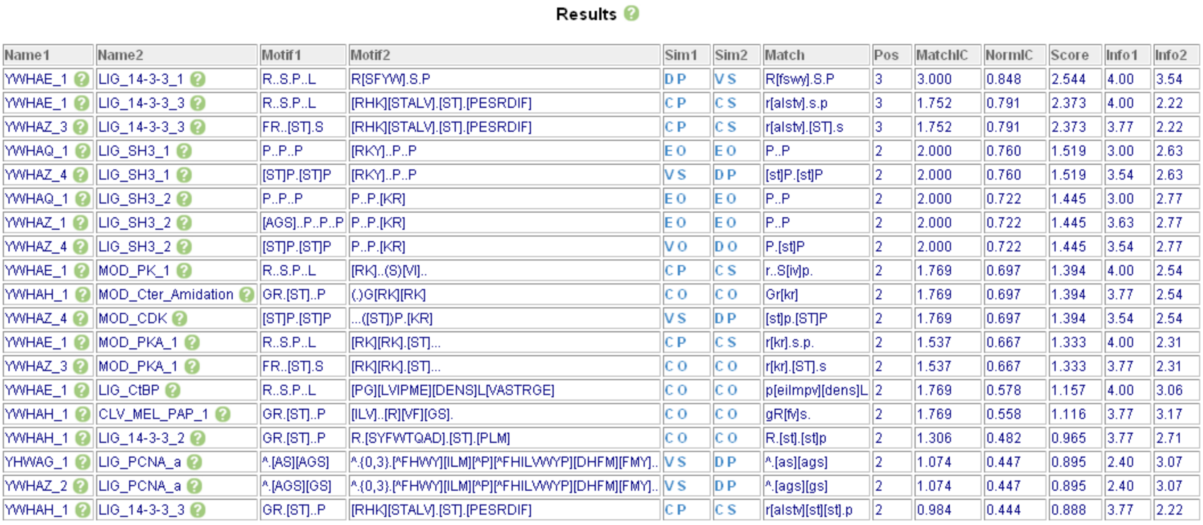
A typical application for CompariMotif is given in the SLiMFinder paper ((Edwards, Davey et al. 2007), Example 1), in which HPRD interaction datasets for 14-3-3 proteins (Mishra, Suresh et al. 2006) were analysed using SLiMFinder, returning several significant motifs (*p*<0.05, see Table 2 in (Edwards, Davey et al. 2007)). These motifs in the recommended CompariMotif input format would be as follows:

```
YWHAE_1  R..S.P..L       # Sig. SLiMFinder motif for HPRD 14-3-3 Epsilon interactors
YWHAH_1  GR.[ST]..P      # Sig. SLiMFinder motif for HPRD 14-3-3 Eta interactors
YHWAG_1  ^.[AS][AGS]     # Sig. SLiMFinder motif for HPRD 14-3-3 Gamma interactors
YHWAG_2  KE..K           # Sig. SLiMFinder motif for HPRD 14-3-3 Gamma interactors
YWHAQ_1  P..P..P         # Sig. SLiMFinder motif for HPRD 14-3-3 Theta interactors
YWHAZ_1  [AGS]..P..P..P  # Sig. SLiMFinder motif for HPRD 14-3-3 Zeta interactors
YWHAZ_2  ^.[AGS][GS]     # Sig. SLiMFinder motif for HPRD 14-3-3 Zeta interactors
YWHAZ_3  FR..[ST].S      # Sig. SLiMFinder motif for HPRD 14-3-3 Zeta interactors
YWHAZ_4  [ST]P.[ST]P     # Sig. SLiMFinder motif for HPRD 14-3-3 Zeta interactors
YWHAZ_5  Y.C.PG.L        # Sig. SLiMFinder motif for HPRD 14-3-3 Zeta interactors
```

Raw SLiMFinder and SLiMDisc delimited text results can be loaded directly into CompariMotif for analysis without any reformatting.

These motifs were compared to the ELM database (Puntervoll, Linding et al. 2003) using CompariMotif with a “normalized IC” cut-off of 0.4. (Figure 3.3, Figure 3.4). Results were constrained such that fixed positions in an ELM must match a fixed position in the SLiMFinder motif. In total, eight out of ten SLiMFinder motifs had matches with seventeen ELMs. Matches fell into three main clusters: (1) Three motifs matching known 14-3-3 motifs, (2) Three motifs matching SH3 binding motifs, and (3) Two motifs matching the highly degenerate LIG_PCNA_1 motif (Figure 3.4). In addition to the 14-3-3 and SH3 ELMs, matches to five phosphorylation ELMs were also identified; phosphorylation of the 14-3-3 motif is important for ligand recognition. These comparisons took less than two seconds to run on an Intel(R) Xeon(TM) dual 3.20GHz processor with 3Gb RAM.

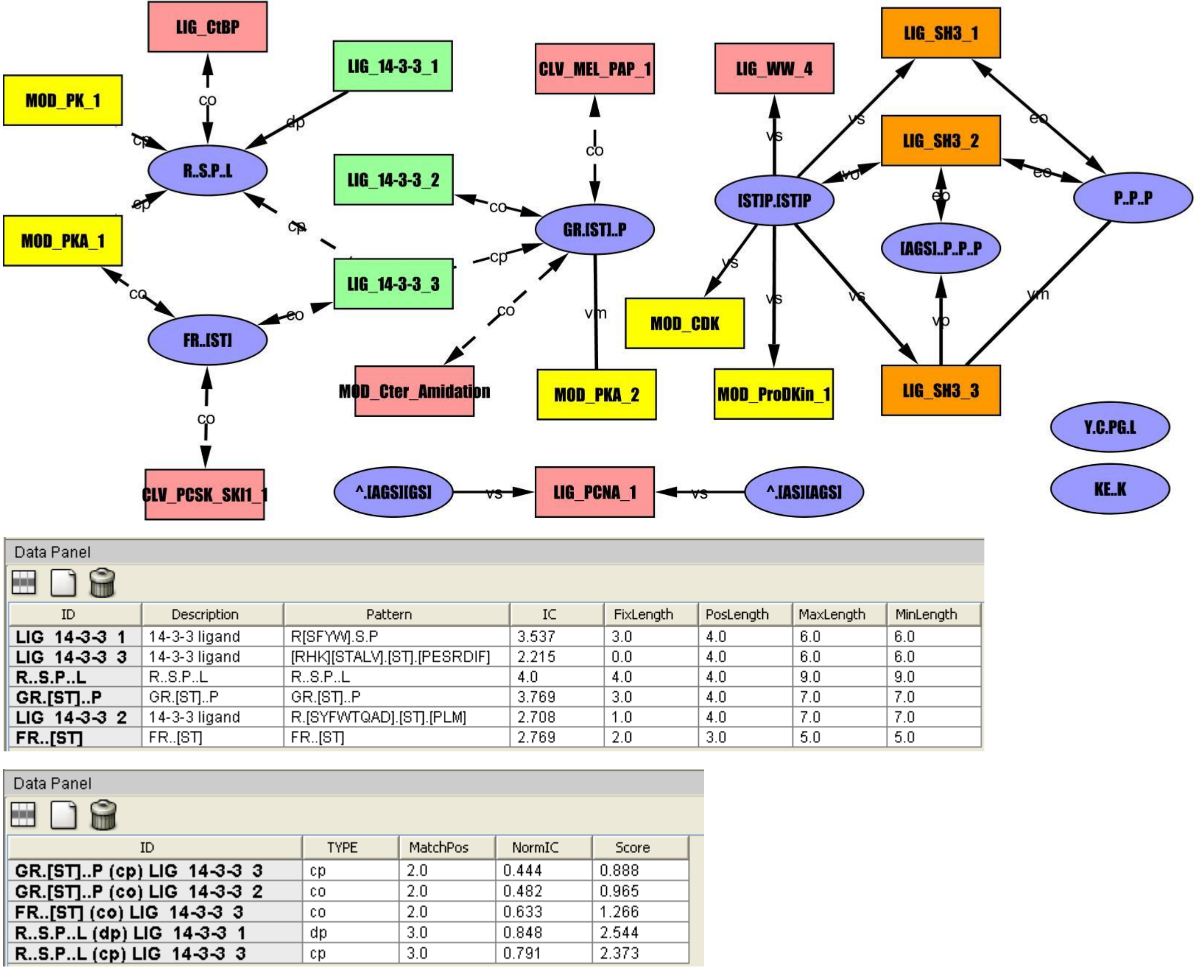
It is beyond the scope of this manual to discuss these results in detail. They do, however, highlight the ease with which CompariMotif can help to make sense of motif discovery results. As a simple, quick and high-throughput tool, CompariMotif can be an invaluable initial step in making sense of such data. Because of this, CompariMotif is now directly linked to both SLiMDisc and SLiMFinder web implementations (Davey, Edwards et al. 2007).



Results

Name1	Name2	Motif1	Motif2	Sim1	Sim2	Match	Pos	MatchIC	NormIC	Score	Info1	Info2
YWHAE_1	LIG_14-3-3_1	R..S.P..L	R[SFYW].S.P	D P	V S	R[fswy].S.P	3	3.000	0.848	2.544	4.00	3.54
YWHAE_1	LIG_14-3-3_3	R..S.P..L	[RHK][STALV].[ST].[PESRDIF]	C P	C S	r[alstv].s.p	3	1.752	0.791	2.373	4.00	2.22
YWHAZ_3	LIG_14-3-3_3	FR..[ST].S	[RHK][STALV].[ST].[PESRDIF]	C P	C S	r[alstv].[ST].s	3	1.752	0.791	2.373	3.77	2.22
YWHAQ_1	LIG_SH3_1	P..P..P	[RKY]..P..P	E O	E O	P..P	2	2.000	0.760	1.519	3.00	2.63
YWHAZ_4	LIG_SH3_1	[ST]P.[ST]P	[RKY]..P..P	V S	D P	[st]P.[st]P	2	2.000	0.760	1.519	3.54	2.63
YWHAQ_1	LIG_SH3_2	P..P..P	P..P.[KR]	E O	E O	P..P	2	2.000	0.722	1.445	3.00	2.77
YWHAZ_1	LIG_SH3_2	[AGS]..P..P..P	P..P.[KR]	E O	E O	P..P	2	2.000	0.722	1.445	3.63	2.77
YWHAZ_4	LIG_SH3_2	[ST]P.[ST]P	P..P.[KR]	V O	D O	P.[st]P	2	2.000	0.722	1.445	3.54	2.77
YWHAE_1	MOD_PK_1	R..S.P..L	[RK]..(S)[VI]..	C P	C S	r..S[iv]p.	2	1.769	0.697	1.394	4.00	2.54
YWHAH_1	MOD_Cter_Amidation	GR.[ST]..P	(.)G[RK][RK]	C O	C O	Gr[kr]	2	1.769	0.697	1.394	3.77	2.54
YWHAZ_4	MOD_CDK	[ST]P.[ST]P	...([ST])P.[KR]	V S	D P	[st]p.[ST]P	2	1.769	0.697	1.394	3.54	2.54
YWHAE_1	MOD_PKA_1	R..S.P..L	[RK][RK].[ST]...	C P	C S	r[kr].s.p.	2	1.537	0.667	1.333	4.00	2.31
YWHAZ_3	MOD_PKA_1	FR..[ST].S	[RK][RK].[ST]...	C O	C O	r[kr].[ST].s	2	1.537	0.667	1.333	3.77	2.31
YWHAE_1	LIG_CtBP	R..S.P..L	[PG][LVIPME][DENS]L[VASTRGE]	C O	C O	p[eilmpv][dens]L	2	1.769	0.578	1.157	4.00	3.06
YWHAH_1	CLV_MEL_PAP_1	GR.[ST]..P	[ILV]..[R][VF][GS].	C O	C O	gR[fv]s.	2	1.769	0.558	1.116	3.77	3.17
YWHAH_1	LIG_14-3-3_2	GR.[ST]..P	R.[SYFWTQAD].[ST].[PLM]	C O	C O	R.[st].[st]p	2	1.306	0.482	0.965	3.77	2.71
YHWAG_1	LIG_PCNA_a	^.[AS][AGS]	^.{0,3}.[^FHWY][ILM][^P][^FHILVWYP][DHFM][FMY]..	V S	D P	^.[as][ags]	2	1.074	0.447	0.895	2.40	3.07
YWHAZ_2	LIG_PCNA_a	^.[AGS][GS]	^.{0,3}.[^FHWY][ILM][^P][^FHILVWYP][DHFM][FMY]..	V S	D P	^.[ags][gs]	2	1.074	0.447	0.895	2.40	3.07
YWHAH_1	LIG_14-3-3_3	GR.[ST]..P	[RHK][STALV].[ST].[PESRDIF]	C P	C S	r[alstv][st][st].p	2	0.984	0.444	0.888	3.77	2.22

Figure 3.3. CompariMotif webserver output for 14-3-3 HPRD SLimFinder analysis vs. ELM.



Data Panel

ID	Description	Pattern	IC	FixLength	PosLength	MaxLength	MinLength
LIG 14-3-3 1	14-3-3 ligand	R[SFYW].S.P	3.537	3.0	4.0	6.0	6.0
LIG 14-3-3 3	14-3-3 ligand	[RHK][STALV].[ST].[PESRDIF]	2.215	0.0	4.0	6.0	6.0
R..S.P..L	R..S.P..L	R..S.P..L	4.0	4.0	4.0	9.0	9.0
GR.[ST]..P	GR.[ST]..P	GR.[ST]..P	3.769	3.0	4.0	7.0	7.0
LIG 14-3-3 2	14-3-3 ligand	R.[SYFWTQAD].[ST].[PLM]	2.708	1.0	4.0	7.0	7.0
FR..[ST]	FR..[ST]	FR..[ST]	2.769	2.0	3.0	5.0	5.0

Data Panel

ID	TYPE	MatchPos	NormIC	Score
GR.[ST]..P (cp) LIG 14-3-3 3	cp	2.0	0.444	0.888
GR.[ST]..P (co) LIG 14-3-3 2	co	2.0	0.482	0.965
FR..[ST] (co) LIG 14-3-3 3	co	2.0	0.633	1.266
R..S.P..L (dp) LIG 14-3-3 1	dp	3.0	0.848	2.544
R..S.P..L (cp) LIG 14-3-3 3	cp	3.0	0.791	2.373

Figure 3.4. CompariMotif XGMML output visualized with Cytoscape(Shannon, Markiel et al. 2003) (recoloured).

Motifs returned by SLiMFinder analysis of 14-3-3 interaction datasets are shown as blue ellipses. ELMs with CompariMotif matches are shown as rectangles. These are pale red by default but the following groups have been manually recoloured: 14-3-3 ligands, green; SH3 ligands, orange, phosphorylation motifs, yellow. Arrows proceed from parent to subsequence motifs (bidirectional where equal); Data Panel details for 14-3-3 ELMs and connected nodes and edges are shown.

4. Appendices

4.1. Troubleshooting & FAQ

There are currently no specific Troubleshooting issues arising with SLiMFinder. Please see general items in the PEAT Appendices document and contact me if you experience any problems not covered.

4.2. SLiM Definitions

This covers the basic definitions needed to understand this manual. The term "motif" can be used in a number of different contexts with different meanings. In this manual, I use motif to mean a short, linear motif (SLiM) in a protein. In biology, SLiMs are functional microdomains with three main properties:

- *Short* – generally less than 10aa long with five or less defined residues.
- *Linear* – comprised of adjacent amino acids in a protein's primary sequence. While three-dimensional conformation may be important for function, it is not necessary for definition.
- *Motifs* – there are some defined sequence patterns that are necessary for function and will therefore recur in the relevant proteins, allowing identification.

The basic anatomy of a SLiM is shown in Figure 4.1.

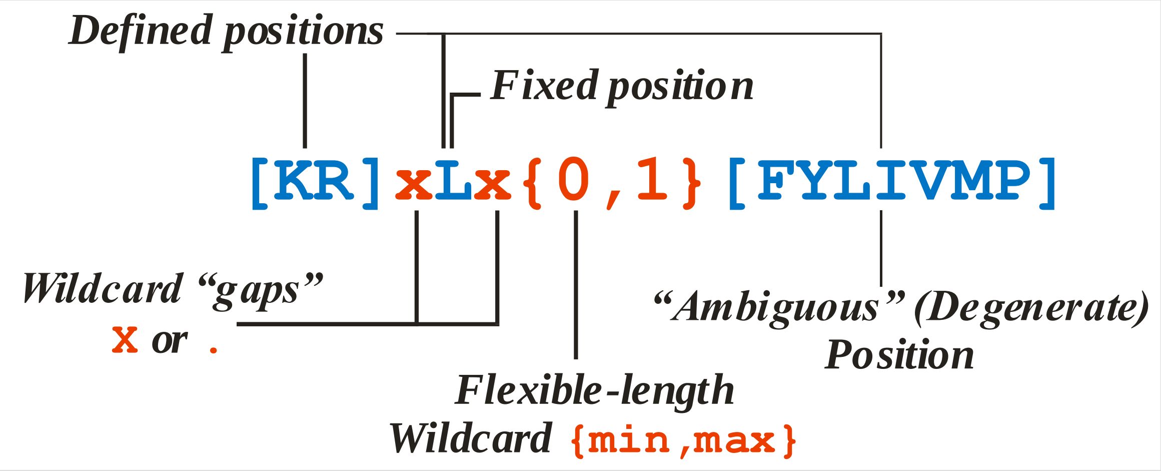


Figure 4.1. Anatomy of a SLiM.
Definitions of different properties of SLiM have been marked on the example ELM, LIG_CYCLIN_1 (Puntervoll, Linding et al. 2003). This motif has three defined positions (one fixed and two degenerate) and two wildcard spacers (one fixed, one flexible-length) for a total length of 4-5aa.

4.3. References

Amanchy R, Periaswamy B, Mathivanan S, Reddy R, Tattikota SG & Pandey A (2007). "A curated compendium of phosphorylation motifs." *Nat Biotechnol.* **25(3)**: 285-286.

Balla S, Thapar V, Verma S, Luong T, Faghri T, Huang CH, Rajasekaran S, del Campo JJ, Shinn JH, Mohler WA, Maciejewski MW, Gryk MR, Piccirillo B, Schiller SR & Schiller MR (2006). "Minimotif Miner: a tool for investigating protein function." *Nat Methods.* **3(3)**: 175-177.

Davey NE, Edwards RJ & Shields DC (2007). "The SLiMDisc server: short, linear motif discovery in proteins." *Nucleic Acids Res* **35(Web Server issue)**: W455-459.

Davey NE, Shields DC & Edwards RJ (2006). "SLiMDisc: short, linear motif discovery, correcting for common evolutionary descent." *Nucleic Acids Res.* **34(12)**: 3546-3554.

Dinkel H, Michael S, Weatheritt RJ, Davey NE, Van Roey K, Altenberg B, Toedt G, Uyar B, Seiler M, Budd A, Jodicke L, Dammert MA, Schroeter C, Hammer M, Schmidt T, Jehl P, McGuigan C, Dymecka M, Chica C, Luck K, Via A, Chatr-Aryamontri A, Haslam N, Grebnev G, Edwards RJ, Steinmetz MO, Meiselbach H, Diella F & Gibson TJ (2012). "ELM--the database of eukaryotic linear motifs." *Nucleic Acids Res* **40(Database issue)**: D242-251.

Edwards RJ, Davey NE, O'Brien K & Shields DC (2012). "Interactome-wide prediction of short, disordered protein interaction motifs in humans." *Mol Biosyst* **8(1)**: 282-295.

Edwards RJ, Davey NE & Shields DC (2007). "SLiMFinder: A Probabilistic Method for Identifying Over-Represented, Convergently Evolved, Short Linear Motifs in Proteins." *PLoS ONE* **2(10)**: e967.

Edwards RJ, Davey NE & Shields DC (2008). "CompariMotif: quick and easy comparisons of sequence motifs." *Bioinformatics* **24(10)**: 1307-1309.

Mishra GR, Suresh M, Kumaran K, Kannabiran N, Suresh S, Bala P, Shivakumar K, Anuradha N, Reddy R, Raghavan TM, Menon S, Hanumanthu G, Gupta M, Upendran S, Gupta S, Mahesh M, Jacob B, Mathew P, Chatterjee P, Arun KS, Sharma S, Chandrika KN, Deshpande N, Palvankar K, Raghavnath R, Krishnakanth R, Karathia H, Rekha B, Nayak R, Vishnupriya G, Kumar HG, Nagini M, Kumar GS, Jose R, Deepthi P, Mohan SS, Gandhi TK, Harsha HC, Deshpande KS, Sarker M, Prasad TS & Pandey A (2006). "Human protein reference database--2006 update." *Nucleic Acids Res.* **34(Database issue)**: D411-414.

Neduva V, Linding R, Su-Angrand I, Stark A, de Masi F, Gibson TJ, Lewis J, Serrano L & Russell RB (2005). "Systematic discovery of new recognition peptides mediating protein interaction networks." *PLoS Biol.* **3(12)**: e405.

Neduva V & Russell RB (2005). "Linear motifs: evolutionary interaction switches." *FEBS Lett.* **579(15)**: 3342-3345 Epub 2005 Apr 3318.

Neduva V & Russell RB (2006). "DILIMOT: discovery of linear motifs in proteins." *Nucleic Acids Res.* **34(Web Server issue)**: W350-355.

Puntervoll P, Linding R, Gemund C, Chabanis-Davidson S, Mattingsdal M, Cameron S, Martin DM, Ausiello G, Brannetti B, Costantini A, Ferre F, Maselli V, Via A, Cesareni G, Diella F, Superti-Furga G, Wyrwicz L, Ramu C, McGuigan C, Gudavalli R, Letunic I, Bork P, Rychlewski L, Kuster B, Helmer-Citterich M, Hunter WN, Aasland R & Gibson TJ (2003). "ELM server: A new resource for investigating short functional sites in modular eukaryotic proteins." *Nucleic Acids Res* **31(13)**: 3625-3630.

Puntervoll P, Linding R, Gemund C, Chabanis-Davidson S, Mattingsdal M, Cameron S, Martin DM, Ausiello G, Brannetti B, Costantini A, Ferre F, Maselli V, Via A, Cesareni G, Diella F, Superti-Furga G, Wyrwicz L, Ramu C, McGuigan C, Gudavalli R, Letunic I, Bork P, Rychlewski L, Kuster B, Helmer-Citterich M, Hunter WN, Aasland R & Gibson TJ (2003). "ELM server: A new resource for investigating short functional sites in modular eukaryotic proteins." *Nucleic Acids Res.* **31(13)**: 3625-3630.

Shannon CE (1997). "The mathematical theory of communication. 1963." *MD. Comput.* **14**: 306-317.

Shannon P, Markiel A, Ozier O, Baliga NS, Wang JT, Ramage D, Amin N, Schwikowski B & Ideker T (2003). "Cytoscape: a software environment for integrated models of biomolecular interaction networks." *Genome Res* **13(11)**: 2498-2504.