HOMER

Software for motif discovery and next-gen sequencing analysis

HOMER Motif Analysis

HOMER contains a novel motif discovery algorithm that was designed for regulatory element analysis in genomics applications (DNA only, no protein). It is a differential motif discovery algorithm, which means that it takes two sets of sequences and tries to identify the regulatory elements that are specifically enriched in on set relative to the other. It uses ZOOPS scoring (zero or one occurrence per sequence) coupled with the hypergeometric enrichment calculations (or binomial) to determine motif enrichment. HOMER also tries its best to account for sequenced bias in the dataset. It was designed with ChIP-Seq and promoter analysis in mind, but can be applied to pretty much any nucleic acids motif finding problem.

There are several ways to perform motif analysis with HOMER. The links below introduce the various workflows for running motif analysis. In a nutshell, HOMER contains two tools, **findMotifs.pl** and **findMotifsGenome.pl**, that manage all the steps for discovering motifs in promoter and genomic regions, respectively. These scripts attempt to make it easy for the user to analyze a list of genes or genomic positions for enriched motifs. However, if you already have the sequence files that you want to analyze (i.e. FASTA files), **findMotifs.pl** (and **homer2**) can process these directly.

Analyzing lists of genes with promoter motif analysis (**findMotifs.pl**)
Analyzing genomic positions (**findMotifsGenome.pl**)
Analyzing custom FASTA files (**findMotifs.pl**, **homer2**)
Analyzing data for RNA motifs (**findMotifs.pl/findMotifsGenome.pl**)

Tips for motif finding

Creating custom motif files

Regardless of how you invoke HOMER, the same basic steps are executed to discover regulatory elements:

Preprocessing:

1. Extraction of Sequences (findMotifs.pl/findMotifsGenome.pl)

If genomic regions are provided as input, the appropriate genomic DNA is extracted. If gene accession numbers are provided, the appropriate promoter regions are selected.

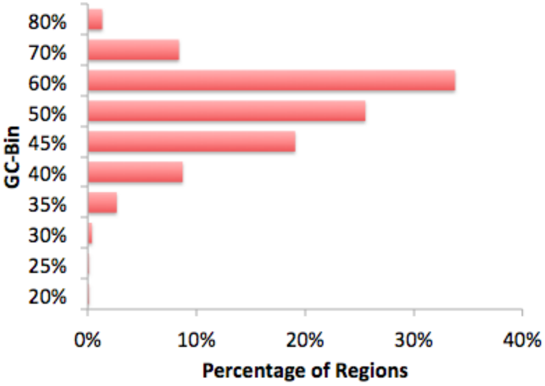
2. Background Selection (findMotifs.pl/findMotifsGenome.pl)

If the background sequences were not explicitly defined, HOMER will automatically select them for you. If you are using genomic positions, sequences will be randomly selected from the genome, matched for GC% content (to make GC normalization easier in the next step). If you are using promoter based analysis, all promoters (except those chosen for analysis) will be used as background. Custom backgrounds can be specified with "**-bg <file>**".

3. GC Normalization (findMotifs.pl/findMotifsGenome.pl)

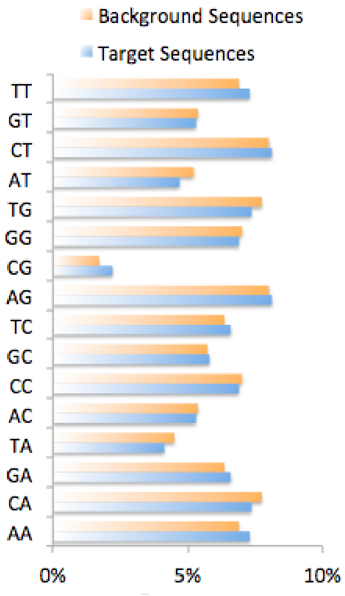
Sequences in the target and background sets are then binned based on their GC-content (5% intervals). Background sequences are weighted to resemble the same GC-content distribution observed in the target sequences. This helps avoid HOMER avoid simply finding motifs that are GC-rich when analyzing sequences from CpG Islands. To perform CpG% normalization instead of GC%(G+C) normalization, use "**-cpg**". An example of the GC%-distribution of regions from a

ChIP-Seq experiment:



4. Autonormalization (New with v3.0, homer2/findMotifs.pl/findMotifsGenome.pl)

Often the target sequences have an imbalance in the sequence content other than GC%. This can be caused by biological phenomenon, such as codon-bias in exons, or experimental bias caused by preferrential sequencing of A-rich stretches etc. If these sources of bias are strong enough, HOMER will lock on to them as features that significanly differentiate the target and background sequences. HOMER now offers autonormalization as a technique to remove (or partially remove) imblances in short oligo sequences (i.e. AA) by assigning weights to background sequences. The proceedure attempts to minimize the difference in short oligo frequency (summed over all oligos) between target and background data sets. It calculates the desired weights for each background sequence to help minimize the error. Due to the complexity of the problem, HOMER uses a simple hill-climbing approach by making small adjustment in background weight at a time. It also penalizes large changes in background weight to avoid trivial solutions that a increase or decrease the weights of outlier sequences to extreme values. The length of short oligos is controlled by the "**-nlen <#>**" option.



Discovering Motifs *de novo* (homer2)

By default, HOMER uses the new homer2 version of the program for motif finding. If you wish to use the old version when running any of the HOMER family of programs, add "**-homer1**" to the command line.

5. Parsing input sequences into an Oligo Table

Input sequences parsed in to oligos of desired motif length, and read into an Oligo Table. The Oligo Table hold each unique oligo in the data set, remembering how many times it occurs in the target and background sequences. This is done to make searching for motif (which are essentially collections of oligos) much more efficient. However, this also destroyes the relationship between individual oligos and their sequence of origin.

6. Oligo Autonormalization (optional)

While the Autonormalization described in step 4 above is applied to full sequences (i.e. ~200 bp), you can also apply the autonormalization concept to the Oligo Table. The idea is still to equalize the smaller oligos (i.e. 1,2,3 bp) within the larger motif lengthed oligos (i.e. 10,12,14 bp etc.). This is a little more dangerous since the total number of motif lengthed oligos can be very large (i.e. 500k for 10 bp, much more for longer motifs), meaning there are a lot of weights to "adjust". However, this can help if there is an extreme sequence bias that you might be having trouble scrubbing out of the data set (the "**-olen <#>**" option).

7. Global Search phase

After creating (and possibly normalizing) the Oligo Table, HOMER coducts a global search for enriched "oligos". The basic idea is that if a "Motif" is going to be enriched, then the oligos considered part of the motif should also be enriched. First, HOMER screens each possible oligo for enrichment. To increase sensitivity, HOMER then allows mismatches in the oligo when searching for enrichment. To speed up this process, which can be very resource consuming for longer oligos with a large number of possible mismatches, HOMER will skip oligos when allowing multiple mismatches if they were not promising, for example if they had more background instances than target instances, or if allowing more mismatches results in a lower enrichment value. The "**-mis <#>**" controls how many mismatches will be allowed.

Calculating Motif Enrichment:

Motif enrichment is calculated using either the cumulative hypergeometric or cumulative binomial distributions. These two statistics assume that the classification of input sequences (i.e. target vs. background) is independent of the occurence of motifs within them. The statistics consider the total number of target sequences, background sequences and how many of each type contains the motif that is being checked for enrichment. From these numbers we can calculate the probability of observing the given number (or more) of target sequences with the motif by chance if we assume there is no relationship between the target sequences and the motif. The hypergeometric and binomial distributions are similar, except that the hypergeometric assumes sampling without replacement, while the binomial assumes sampling with replacement. The motif enrichment problem is more accurately described by the hypergeometric, however, the binomial has advantages. The difference between them is usually minor if there are a large number of sequences and the background sequences >> target sequences. In these cases, the binomial is preferred since it is faster to calculate. As a result it is the default statistic for **findMotifsGenome.pl** where the number of sequences is typically higher. However, if you use your own background that has a limited number of sequences, it might be a good idea to switch to the hypergeometric (use "**-h**" to force use of the hypergeometric). **findMotifs.pl** exects smaller number for promoter analysis and uses the hypergeometric by default.

One important note: Since HOMER uses an Oligo Table for much of the internal calculations of motif enrichment, where it does not explicitly know how many of the original sequences contain the motif, it approximates this number using the total number of observed motif occurrences in background and target sequences. It assumes the occurrences were equally distributed among the target or background sequences with replacement, were some of the sequences are likely to have more than one occurence. It uses the expected number sequences to calculate the enrichment statistic (the final output reflects the actual enrichment based on the original sequences).

8. Matrix Optimization

HOMER takes the most enriched oligos from the global optimization step, transforms them into simple position specific probability matrices, and further optimizes them with a sensitive local optimization algorithm. This step is performed separately for each oligo, and will create the "motif probability matrix" as well as determine the optimal detection threshold to maximize the enrichment of the motif in the target vs. background sequences. The detection threshold is simply done by scoring each oligo in the data to the probability matrix, and then sorting the oligos by their similarity to the matrix. HOMER then steps down the list, effectively decreasing the detection threshold, including more and more oligos until an optimal enrichment is found. After this step, HOMER will create several new probability matrices based on the oligos found in different detection thresholds and check which one has the highest enrichment. This process is repeated until the enrichment can no longer be improved, producing a final motif.

9. Mask and Repeat

After the first "promising oligo" is optimized into a motif, the sequences bound by the motif to are removed from the analysis and the next promising oligo is optimized for the 2nd motif, and so on. This is repeated until the desired number of motifs are found ("**-S <#>**", default: 25). This is where the there is an important difference between the old (homer) and new (homer2) versions. The old version of homer would simply mask the oligos bound by the motif from the Oligo Table. For example if the motif was GAGGAW then GAGGAA and GAGGAT would be removed from the Oligo Table to avoid having the next motif find the same sequences. However, if GAGGAW was enriched in the data, there is a good chance that any 6-mer oligo like nGAGGA or AGGAWn would also be somewhat enriched in the data. This would cause homer to find multiple versions of the same motif and provide a little bit of confusion in the results.

To avoid this problem in the new version of HOMER (homer2), once a motif is optimized, HOMER revisits the original sequences and masks out the oligos making up the instance of the motif as well as well as oligos immediately adjacent to the site that overlap with at least one nucleotide. This helps provide much cleaner results, and allows greater sensitivity when co-enriched motifs. To make revert back to the old way of motif masking with homer2, specify "**-quickMask**" at the command line. You can also run the old version with "**-homer1**".

Screening for Enrichment of Known Motifs (homer2):

10. Load Motif Library

In order to search for Known Motifs in your data, HOMER loads a list of previously determined motifs from previous data. You can also add you own motifs by specifying them at the command line ("**-mknown <file>**") or by editing the primary file ("data/knownTFs/known.motifs"). HOMER doesn't screen all of TRANSFAC - partially due to motif quality (which can be low), and paritically due to the fact that we need a detection threshold.

11. Screen Each Motif

To find the enrichment for each motif, HOMER scans each sequence for instances of the motif and calculates the final enrichment by considering how many target vs. background sequences are considered "bound". ZOOPS (zero or one occurence per sequence) counting is used and the hypergeometric or binomial is used to calculate the significance.

Motif Analysis Output:

12. Motif Files (homer2, findMotifs.pl, findMotifsGenome.pl)

The true output of HOMER are "*.motif" files which contain the information necessary to identify future instance of motifs. They are reported in the output directories from findMotifs.pl and findMotifsGenome.pl. A typical motif file will look something like:

```
>ASTTCCTCTT     1-ASTTCCTCTT    8.059752       -23791.535714   0      T:17311.0(44 ...
0.726   0.002   0.170   0.103
0.002   0.494   0.354   0.151
0.016   0.017   0.014   0.954
0.005   0.006   0.027   0.963
0.002   0.995   0.002   0.002
0.002   0.989   0.008   0.002
0.004   0.311   0.148   0.538
0.002   0.757   0.233   0.009
0.276   0.153   0.030   0.542
0.189   0.214   0.055   0.543
```

The first row starts with a ">" followed by various information, and the other rows are the positions specific probabilities for each nucleotide (A/C/G/T). The header row is actually TAB delimited, and contains the following information:

1. ">" + Consensus sequence (not actually used for anything, can be blank) example: >ASTTCCTCTT
2. Motif name (should be unique if several motifs are in the same file) example: 1-ASTTCCTCTT or NFkB
3. Log odds detection threshold, used to determine bound vs. unbound sites (**mandatory**) example: 8.059752
4. log P-value of enrichment, example: -23791.535714
5. 0 (A place holder for backward compatibility, used to describe "gapped" motifs in old version, turns out it wasn't very useful :)
6. Occurence Information separated by commas, example: T:17311.0(44.36%),B:2181.5(5.80%),P:1e-10317
    1. T:#(%) - number of target sequences with motif, % of total of total targets
    2. B:#(%) - number of background sequences with motif, % of total background
    3. P:# - final enrichment p-value
7. Motif statistics separated by commas, example: Tpos:100.7,Tstd:32.6,Bpos:100.1,Bstd:64.6,StrandBias:0.0,Multiplicity:1.13
    1. Tpos: average position of motif in target sequences (0 = start of sequences)
    2. Tstd: standard deviation of position in target sequences
    3. Bpos: average position of motif in background sequences (0 = start of sequences)
    4. Bstd: standard deviation of position in background sequences
    5. StrandBias: log ratio of + strand occurrences to - strand occurrences.
    6. Multiplicity: The averge number of occurrences per sequence in sequences with 1 or more binding site.

You can easily create your own motif files, just remember that the **first 3 columns are required**!!!

13. De novo motif output (findMotifs.pl/findMotifsGenome.pl/compareMotifs.pl)

HOMER takes the motifs identified from *de novo* motif discovery step and tries to process and present them in a useful manner. An HTML page is created in the output directory named homerResults.html along with a directory named "homerResults/" that contains all of the image and other support files to create the page. These pages are explicitly created by running a subprogram called "**compareMotifs.pl**".

Comparison of Motif Matrices:

Motifs are first checked for redundancy to avoid presenting the same motifs over and over again. This is done by aligning each pair of motifs at each position (and their reverse opposites) and scoring their similarity to determine their best alignment. Starting with HOMER v3.3, matrices are compared using Pearson's correlation coefficient by converting each matrix into a vector of values. Neutral frequencies (0.25) are used in where the motif matrices do not overlap.

The old comparison was done by comparing the probability matrices using the formula below which manages the expectations of the calulations by scrambling the nuclotide identities as a control. (freq1 and freq2 are the matrices for motif1 and motif2)

$$SimilarityScore = \frac{1}{MotifLength} \sum_{i}^{MotifLength} -\frac{(Observed_i - Expect_i)}{Expect_i}$$

$$Observed_i = \sum_{j}^{A,C,G,T} -(freq1_{ij} - freq2_{ij})^2$$

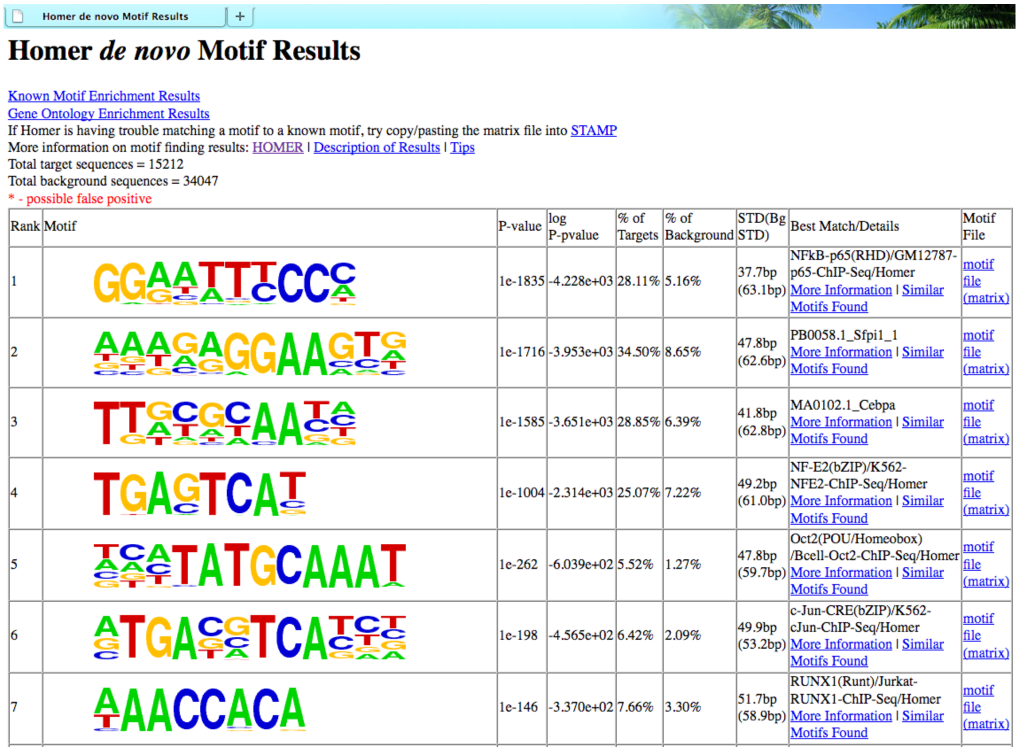
$$Expect_i = \sum_{j}^{A,C,G,T} \sum_{k}^{A,C,G,T} \frac{-(freq1_{ij} - freq2_{ik})^2}{4}$$

The output will be a score ranging from some lower bound (depending on the matrix frequencies) to 1, where 1 is complete similarity. By default the threshold for assigning similar motifs is 0.6, which is a reasonable cutoff in practice. This can be changed if you run **compareMotifs.pl** and change the "**-reduceThresh <#>**" parameter.

Motifs are next compared against a library of known motifs. For this step, all motifs in JASPAR and the "known" motifs are used for comparison. You can specify a custom motif library using "**-mcheck <motif library file>**" when using **findMotifs[Genome].pl** or "**-known <motif library file>**" when calling **compareMotifs.pl** directly.

By default, it looks for the file "/path-to-homer/data/knownTFs/all.motifs" to find the motif to compare with the de novo motifs. If "-rna" is specified, it will load the file "/path-to-homer/data/knownTFs/all.rna.motifs".

An example of the output HTML is show below:



Homer *de novo* Motif Results

Known Motif Enrichment Results
Gene Ontology Enrichment Results
If Homer is having trouble matching a motif to a known motif, try copy/pasting the matrix file into STAMP
More information on motif finding results: HOMER | Description of Results | Tips
Total target sequences = 15212
Total background sequences = 34047
* - possible false positive

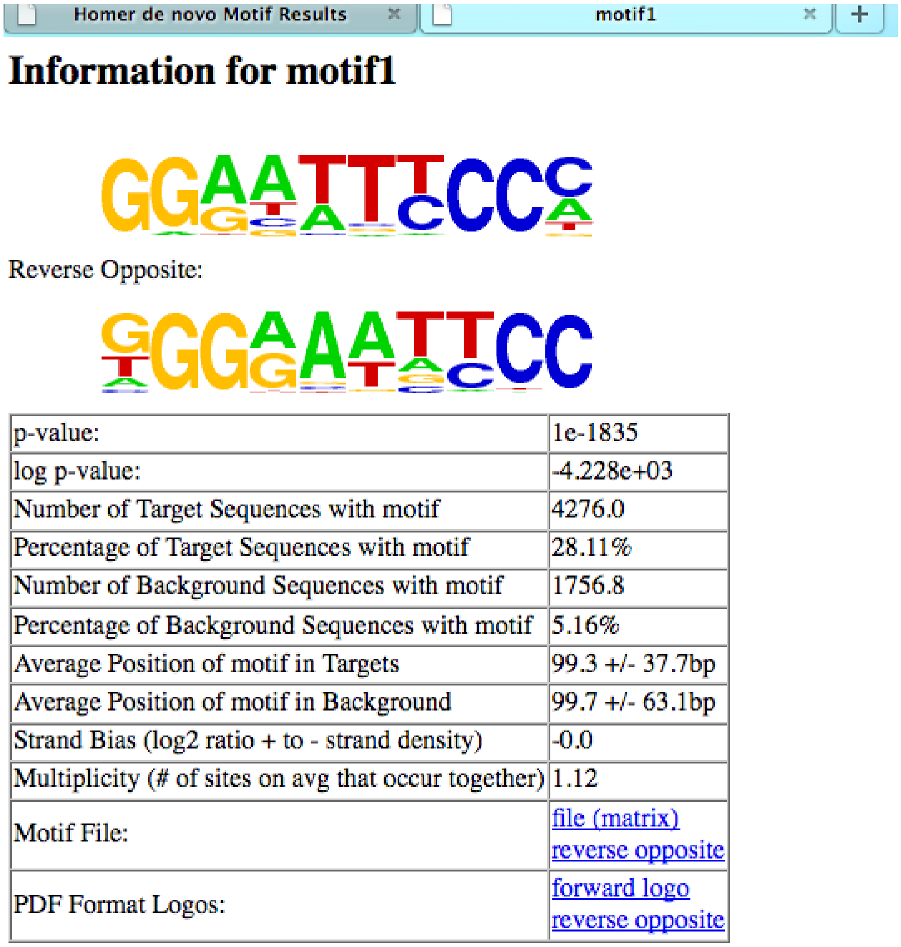
Rank	Motif	P-value	log P-pvalue	% of Targets	% of Background	STD(Bg STD)	Best Match/Details	Motif File
1		1e-1835	-4.228e+03	28.11%	5.16%	37.7bp (63.1bp)	NFkB-p65(RHD)/GM12787-p65-ChIP-Seq/Homer More Information | Similar Motifs Found	motif file (matrix)
2		1e-1716	-3.953e+03	34.50%	8.65%	47.8bp (62.6bp)	PB0058.1_Sfpi1_1 More Information | Similar Motifs Found	motif file (matrix)
3		1e-1585	-3.651e+03	28.85%	6.39%	41.8bp (62.8bp)	MA0102.1_Cebpa More Information | Similar Motifs Found	motif file (matrix)
4		1e-1004	-2.314e+03	25.07%	7.22%	49.2bp (61.0bp)	NF-E2(bZIP)/K562-NFE2-ChIP-Seq/Homer More Information | Similar Motifs Found	motif file (matrix)
5		1e-262	-6.039e+02	5.52%	1.27%	47.8bp (59.7bp)	Oct2(POU/Homeobox)/Bcell-Oct2-ChIP-Seq/Homer More Information | Similar Motifs Found	motif file (matrix)
6		1e-198	-4.565e+02	6.42%	2.09%	49.9bp (53.2bp)	c-Jun-CRE(bZIP)/K562-cJun-ChIP-Seq/Homer More Information | Similar Motifs Found	motif file (matrix)
7		1e-146	-3.370e+02	7.66%	3.30%	51.7bp (58.9bp)	RUNX1(Runt)/Jurkat-RUNX1-ChIP-Seq/Homer More Information | Similar Motifs Found	motif file (matrix)

Depending on how the **findMotifs[Genome].pl** program that was executed, the "Known Motif Enrichment Results" and "Gene Ontology Enrichment Results" may or may not link to anything. Motifs are sorted based on p-value, and basic statistics about the motif (present in the motif files) is displayed.

The final column contains a link to the "motif file", which is important if you want to search for the motif in other sequences.

In the Best Match/Details column, HOMER will display the known motif which most closely matched with the *de novo* motif. It is very important that you **TAKE THIS ASSIGNMENT WITH A GRAIN OF SALT!!!!!** Unfortunately, sometimes the best match still isn't any good. Also, it is common that the "known" motif isn't any good to begin with. To investigate the assignment further, click on the "More Information" link which provides a page that looks like this:

Basic Information: The section contains basic information, including links to the motif file (normal and reverse opposite) and the pdf version of the motif logo.

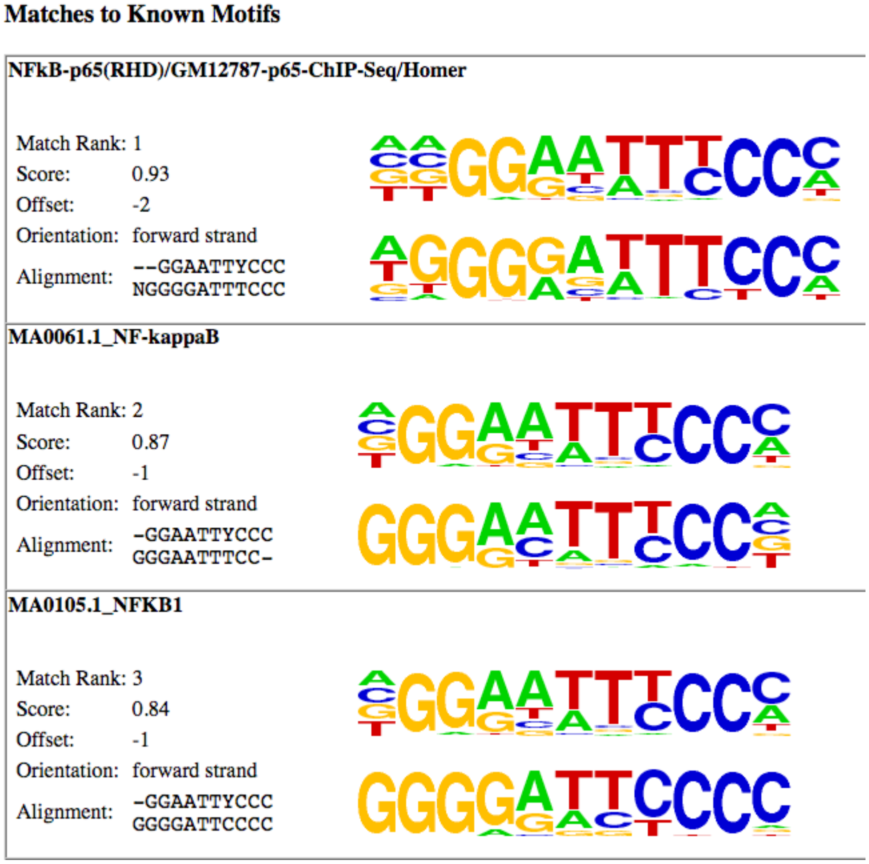


Information for motif1

Reverse Opposite:

p-value:	1e-1835
log p-value:	-4.228e+03
Number of Target Sequences with motif	4276.0
Percentage of Target Sequences with motif	28.11%
Number of Background Sequences with motif	1756.8
Percentage of Background Sequences with motif	5.16%
Average Position of motif in Targets	99.3 +/- 37.7bp
Average Position of motif in Background	99.7 +/- 63.1bp
Strand Bias (log2 ratio + to - strand density)	-0.0
Multiplicity (# of sites on avg that occur together)	1.12
Motif File:	file (matrix) reverse opposite
PDF Format Logos:	forward logo reverse opposite

Followed by matches to known motifs. This section shows the alignments between the de novo motif and known motifs. It's important to check and see if these alignments look reasonable:



Matches to Known Motifs

NFkB-p65(RHD)/GM12787-p65-ChIP-Seq/Homer

Match Rank: 1
Score: 0.93
Offset: -2
Orientation: forward strand
Alignment:
```
--GGAATTYCCC
NGGGGATTTCCC
```

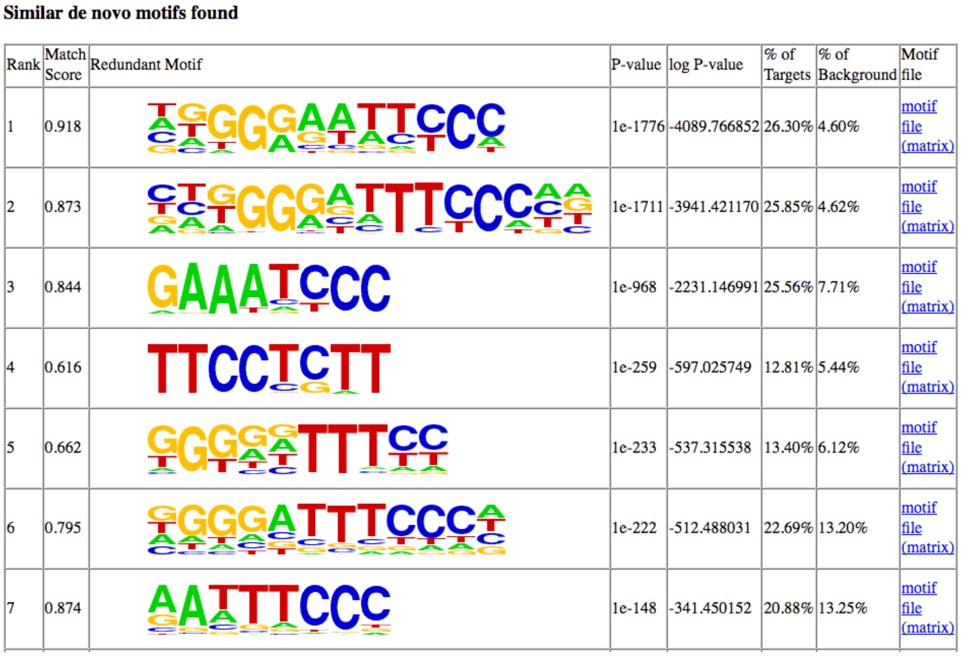
MA0061.1_NF-kappaB

Match Rank: 2
Score: 0.87
Offset: -1
Orientation: forward strand
Alignment:
```
-GGAATTYCCC
GGGAATTTCC-
```

MA0105.1_NFKB1

Match Rank: 3
Score: 0.84
Offset: -1
Orientation: forward strand
Alignment:
```
-GGAATTYCCC
GGGGATTCCCC
```

Clicking on the "similar motifs" will show the other de novo motifs found during motif finding that resemble the motif but had a lower enrichment value. It contains a similar "header" as the "More Information" link, but below it shows the motifs that were considered similar. It is usually a good idea to check this list over - sometimes a distinct motif will be grouped incorrectly in the list because it shares a couple residues.



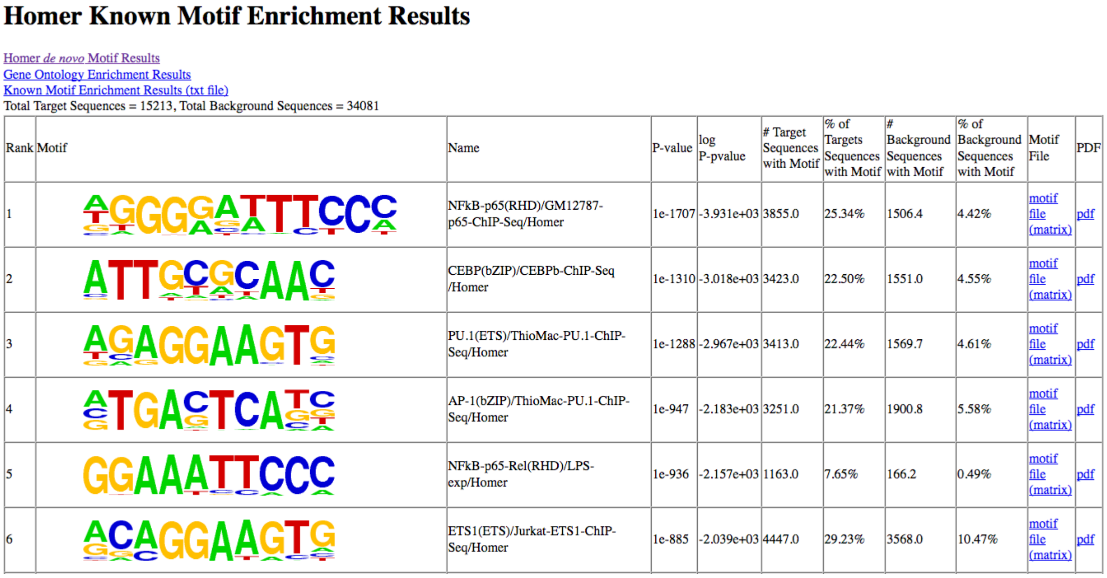
Similar de novo motifs found

Rank	Match Score	Redundant Motif	P-value	log P-value	% of Targets	% of Background	Motif file
1	0.918		1e-1776	-4089.766852	26.30%	4.60%	motif file (matrix)
2	0.873		1e-1711	-3941.421170	25.85%	4.62%	motif file (matrix)
3	0.844		1e-968	-2231.146991	25.56%	7.71%	motif file (matrix)
4	0.616		1e-259	-597.025749	12.81%	5.44%	motif file (matrix)
5	0.662		1e-233	-537.315538	13.40%	6.12%	motif file (matrix)
6	0.795		1e-222	-512.488031	22.69%	13.20%	motif file (matrix)
7	0.874		1e-148	-341.450152	20.88%	13.25%	motif file (matrix)

To rerun this part of the analysis on any arbitrary set of motifs, simply run the "**compareMotifs.pl**" command (use without any command line parameters to get the usage options).

14. Known motif output

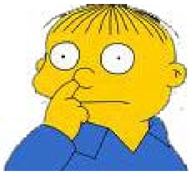
Known motif enrichment is displayed as a HTML file (knownResults.html). The page sorts the results based on enrichment and displays basic information:



Homer Known Motif Enrichment Results

Homer *de novo* Motif Results
Gene Ontology Enrichment Results
Known Motif Enrichment Results (txt file)
Total Target Sequences = 15213, Total Background Sequences = 34081

Rank	Motif	Name	P-value	log P-pvalue	# Target Sequences with Motif	% of Targets Sequences with Motif	# Background Sequences with Motif	% of Background Sequences with Motif	Motif File	PDF
1	AGGGGATTTCCC	NFkB-p65(RHD)/GM12787-p65-ChIP-Seq/Homer	1e-1707	-3.931e+03	3855.0	25.34%	1506.4	4.42%	motif file (matrix)	pdf
2	ATTGCGCAAC	CEBP(bZIP)/CEBPb-ChIP-Seq/Homer	1e-1310	-3.018e+03	3423.0	22.50%	1551.0	4.55%	motif file (matrix)	pdf
3	AGAGGAAGTG	PU.1(ETS)/ThioMac-PU.1-ChIP-Seq/Homer	1e-1288	-2.967e+03	3413.0	22.44%	1569.7	4.61%	motif file (matrix)	pdf
4	ATGACTCATC	AP-1(bZIP)/ThioMac-PU.1-ChIP-Seq/Homer	1e-947	-2.183e+03	3251.0	21.37%	1900.8	5.58%	motif file (matrix)	pdf
5	GGAAATTCCC	NFkB-p65-Rel(RHD)/LPS-exp/Homer	1e-936	-2.157e+03	1163.0	7.65%	166.2	0.49%	motif file (matrix)	pdf
6	ACAGGAAGTG	ETS1(ETS)/Jurkat-ETS1-ChIP-Seq/Homer	1e-885	-2.039e+03	4447.0	29.23%	3568.0	10.47%	motif file (matrix)	pdf



Can't figure something out? Questions, comments, concerns, or other feedback:
cbenner@ucsd.edu

HOMER

Software for motif discovery and ChIP-Seq analysis

Motif Finding with HOMER from FASTA files

Most of HOMER's functionality is built around either promoter or genomic position based analysis, and aims to manage the sequence manipulation, hiding it from the user. However, if you have some sequences that you would like HOMER to analyze, the program **findMotifs.pl** accepts **FASTA** formatted files for analysis. Alternatively you could use the **homer2** executable which also accepts FASTA files as input.

HOMER is designed to analyze high-throughput data using differential motif discovery, which means you **MUST** have both target and background sequences, and in each case you should have several (preferably thousands) of sequences in each set that are roughly the same length.

A quick note about FASTA files - Each sequence should have a unique identifier. In theory, HOMER should be flexible with what is in the header line, but if you're having trouble please just keep it simple with minimal quite-space, especially tabs. For example:

```
>NM_003456
AAGGCCTGAGATAGCTAGAGCTGAGAGTTTTCCACACG
```

Running findMotifs.pl with FASTA files:

To find motifs from FASTA files, run findMotifs.pl with the target sequence FASTA file as the first command-line argument, and use the option "**-fasta <file>**" to specify the background FASTA file. You CANNOT NOT specify a background file - that would defeat the purpose of differential motif finding.

findMotifs.pl <targetSequences.fa> fasta <output directory> -fasta <background.fa> [options]

NOTE: you must choose an "organism" for the 2nd argument to keep with the structure of the command, even though this isn't actually relevant for FASTA based analysis. Organism doesn't have to match the data in the FASTA files. You can use a valid organism or just put "**fasta**" as a place holder. i.e.:

findMotifs.pl chuckNorrisGenes.fa human analysis_output/ -fasta normalHumanGenes.fa

Many other options are available to control motif finding parameters.

findMotifs.pl will perform GC normalization and autonormalization be default (see here for more details).

Finding instances of motifs with FASTA files:

To find instance of a motif, run the same command used for motif discovery above but add the option "**-find <motif file>**". Motif results will be sent to stdout, so to capture the results in a file Add "**> outputfile**" to the end of the command.

```
findMotifs.pl <targetSequences.fa> fasta <output directory> -fasta
<background.fa> [options] -find motif1.motif > outputfile.txt
```

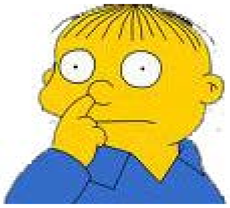
For more information on the output file format, see here.

Using homer2 directly with FASTA files:

homer2 is the motif finding executable, and it can choke down FASTA files if you want to avoid all the nonsense above. Running the homer2 command will also give you access to other options for optimizing the motif finding process. **homer2** works by first specifying a command, and then the appropriate options:

```
homer2 <command> [options]
i.e. homer2 denovo -i input.fa -b background.fa > outputfile.txt
```

To find instances of the output motifs, use "**homer2 find**". To see other commands, just type "**homer2**".



Can't figure something out? Questions, comments, concerns, or other feedback: cbenner@ucsd.edu

HOMER

Software for motif discovery and next-gen sequencing analysis

RNA Motif Analysis

HOMER was not originally designed with RNA in mind, but it can be used to successfully analyze data for RNA motifs. By RNA motifs, we mean short sequence elements in RNA sequences akin to DNA motifs, not structural elements such as hairpins and stuff like that. For example, HOMER can be used to successfully determine miRNA seeds in sets of co-regulated mRNAs, or RNA binding elements in CLIP-Seq data.

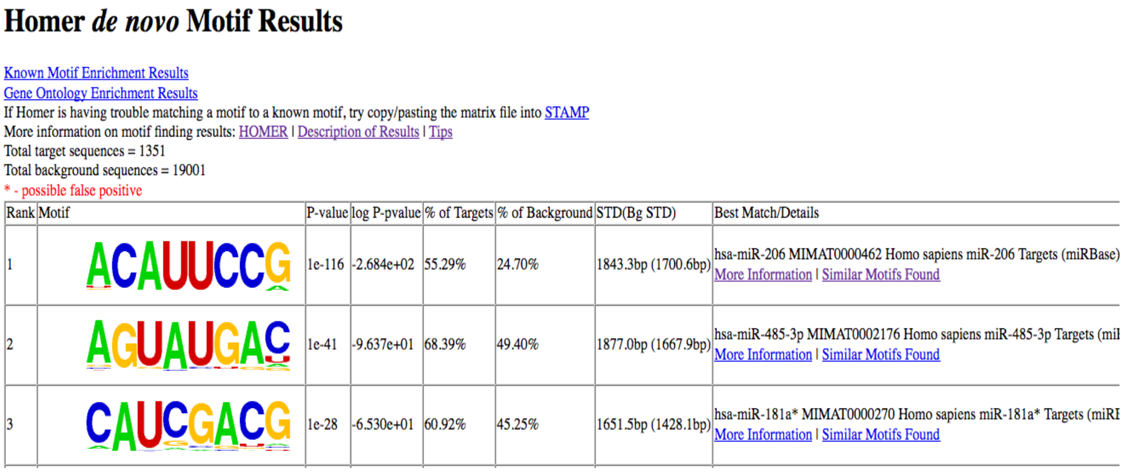
The "**-rna**" option can be used with findMotifs.pl and findMotifsGenome.pl, resulting in + strand only motif searching and motif display/matching with "U" instead of "T". HOMER does not contain a list of well known "RNA motifs" yet, so no "known motif" analysis is performed. If using FASTA files, please use "T" (normal DNA encoding) in the input files for now.

Analyzing Co-regulated Gene Lists for RNA motifs

HOMER contains preconfigured PROMOTER sets comprised of RefSeq mRNA sequences, or only the 5' and 3' UTRs. These are useful for analyzing gene lists for motifs in their mRNAs. To run the analysis, us findMotifs.pl with a mRNA PROMOTER set, and options for RNA motifs will be automatically set.

```
findMotifs.pl mir1-downregulated.genes.txt human-mRNA MotifOutput/ -rna -len 8
```

You don't actually need to specify -rna for this case since with the use of "human-mRNA" it's understood. Anyway, the output will look something like this:

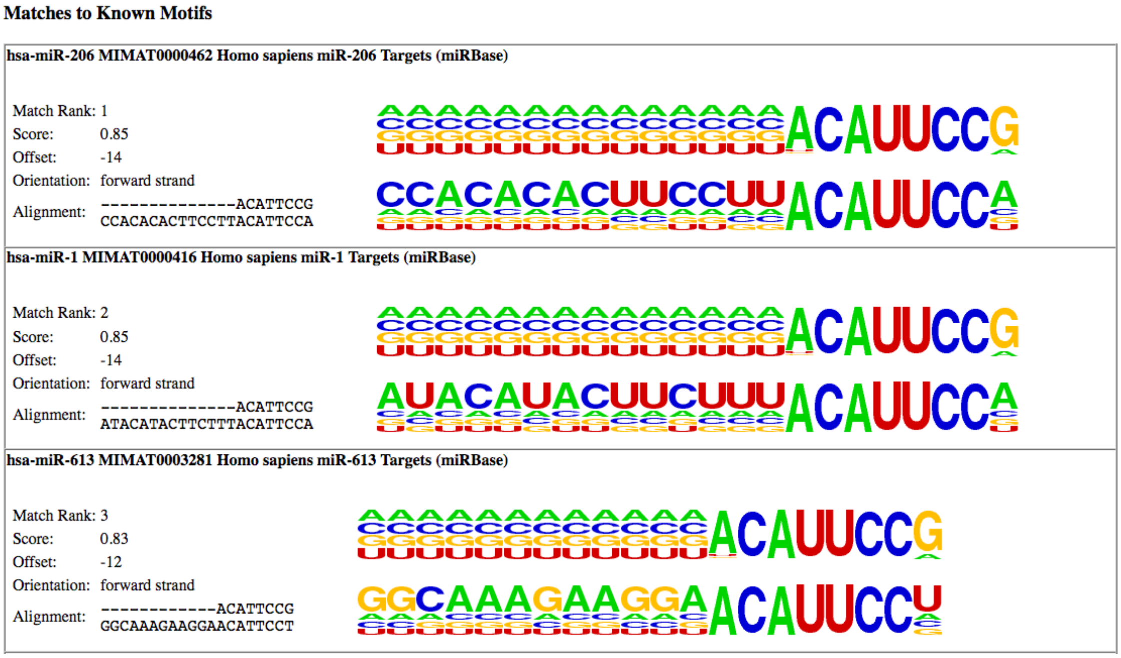


Homer *de novo* Motif Results

Known Motif Enrichment Results
Gene Ontology Enrichment Results
If Homer is having trouble matching a motif to a known motif, try copy/pasting the matrix file into STAMP
More information on motif finding results: HOMER | Description of Results | Tips
Total target sequences = 1351
Total background sequences = 19001
* - possible false positive

Rank	Motif	P-value	log P-pvalue	% of Targets	% of Background	STD(Bg STD)	Best Match/Details
1	ACAUUCCG	1e-116	-2.684e+02	55.29%	24.70%	1843.3bp (1700.6bp)	hsa-miR-206 MIMAT0000462 Homo sapiens miR-206 Targets (miRBase) More Information | Similar Motifs Found
2	AGUAUGAC	1e-41	-9.637e+01	68.39%	49.40%	1877.0bp (1667.9bp)	hsa-miR-485-3p MIMAT0002176 Homo sapiens miR-485-3p Targets (mil More Information | Similar Motifs Found
3	CAUCGACG	1e-28	-6.530e+01	60.92%	45.25%	1651.5bp (1428.1bp)	hsa-miR-181a* MIMAT0000270 Homo sapiens miR-181a* Targets (miRI More Information | Similar Motifs Found

For now, HOMER will try to match the results to the human list of miRNA seeds (from miRBase):



Matches to Known Motifs

hsa-miR-206 MIMAT0000462 Homo sapiens miR-206 Targets (miRBase)

Match Rank: 1
Score: 0.85
Offset: -14
Orientation: forward strand
Alignment:

```
--------------ACATTCCG
CCACACACTTCCTTACATTCCA
```

hsa-miR-1 MIMAT0000416 Homo sapiens miR-1 Targets (miRBase)

Match Rank: 2
Score: 0.85
Offset: -14
Orientation: forward strand
Alignment:

```
--------------ACATTCCG
ATACATACTTCTTTACATTCCA
```

hsa-miR-613 MIMAT0003281 Homo sapiens miR-613 Targets (miRBase)

Match Rank: 3
Score: 0.83
Offset: -12
Orientation: forward strand
Alignment:

```
------------ACATTCCG
GGCAAAGAAGGAACATTCCT
```

In this case, the motif matches the miR-1 consensus seed (which is shared by miR-206 and miR-613).

There are two RNA specific options for findMotifs.pl in rna mode:

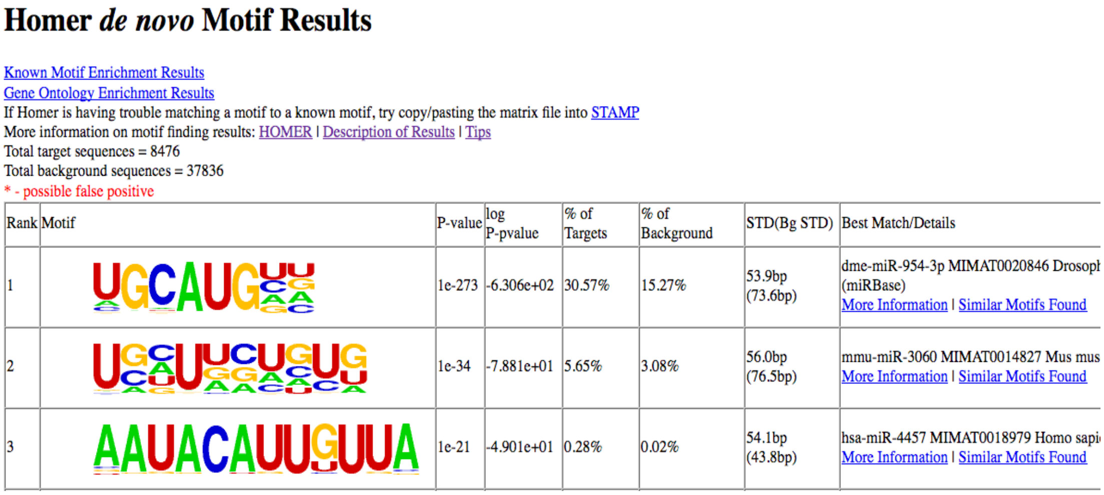
-min <#> : minimum length of mRNA to consider (basically removes extremely short mRNA sequences from the analysis)
-max <#> : maximum length of mRNA to consider (removes really long RNAs from the analysis)

Analyzing CLIP-Seq for RNA motifs

HOMER can analyze strand-specific genomic regions for motifs, such as the regions that would be defined from CLIP-Seq. To do this, just run findMotifsGenome.pl using the "**-rna**" flag (make sure your regions are strand specific!!). For now, HOMER just uses the same random genomic background used for ChIP-Seq motif finding. You could imagine that a better RNA motif finding background would be RNA, i.e. strand specific exon/intron sequences. You'd be right, but managing this with respect to the different experiments (i.e. intronic binding vs. mRNA binding vs. non-coding RNA binding) is tricky and for now left up to the user (you can specify your own strand specific background with "**-bg <peak/BED file>**"). Trying this with FOX CLIP-Seq data:

```
findMotifsGenome.pl fox2.clip.bed hg17 MotifOutput -rna
```

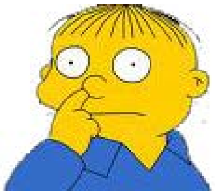
This will give the following results (which resembles a UGCAUG FOX motif):



Homer *de novo* Motif Results

Known Motif Enrichment Results
Gene Ontology Enrichment Results
If Homer is having trouble matching a motif to a known motif, try copy/pasting the matrix file into STAMP
More information on motif finding results: HOMER | Description of Results | Tips
Total target sequences = 8476
Total background sequences = 37836
* - possible false positive

Rank	Motif	P-value	log P-pvalue	% of Targets	% of Background	STD(Bg STD)	Best Match/Details
1	UGCAUGUU	1e-273	-6.306e+02	30.57%	15.27%	53.9bp (73.6bp)	dme-miR-954-3p MIMAT0020846 Drosoph (miRBase) More Information | Similar Motifs Found
2	UGCUUCUGUG	1e-34	-7.881e+01	5.65%	3.08%	56.0bp (76.5bp)	mmu-miR-3060 MIMAT0014827 Mus mus More Information | Similar Motifs Found
3	AAUACAUUGUUA	1e-21	-4.901e+01	0.28%	0.02%	54.1bp (43.8bp)	hsa-miR-4457 MIMAT0018979 Homo sapi More Information | Similar Motifs Found



Can't figure something out? Questions, comments, concerns, or other feedback: cbenner@ucsd.edu

HOMER

Software for motif discovery and next-sequencing analysis

Practical Tips to Motif Finding with HOMER

Below are some general tips for getting the most out of you motif analysis when using HOMER. Be sure to look over this section about judging motif quality!

What to do if motif finding takes too long...

Ctrl+C... If you are using reasonable parameters (see next section), it shouldn't take more than an hour or so, and in most cases much less.

Choosing the length of motifs to find

It's almost always a good idea to start with the default parameters. Resist the urge to find motifs larger than 12 bp the first time around. Longer motifs will show up as different short motifs when finding shorter motifs. If there aren't any truly significant motifs when looking at short motifs, it is unlikely that you will find good long motifs either. And it doesn't take much time to check for short motifs.

i.e. **-S 25 -len 8,10,12**

Once you do find motifs that look promising, try looking for longer motifs.

Finding Long Motifs

The new version of HOMER (v3.0+) is better at looking for long motifs. However, it can be tricky looking for long motifs because the search space gets very large. Also, the running time on longer motifs increases and may break your patience.

Since HOMER is an empirical motif finding program, it starts from actual oligos present in the sequence and attempts to figure out if they are enriched. If you are looking at 20 bp sequences, there is a good chance that they are all more-or-less unique in your data set with only 1 instance in either the target or background sequences. HOMER normally allows mismatches in the original oligo to see if the oligo together with similar oligos are collectively enriched. The problem is that this technique starts to break down at long lengths. It takes many mismatches to find enough related sequences to assess enrichment, and it is computationally expensive to find them.

To maintain sensitivity for longer motifs:

Increase the "-mis <#>" option to allow more mismatches. In practice, I would use at least "-mis 4" or "-mis 5" for sensitive detection of 20 bp motifs. If the data set is for a strong motif (i.e. CTCF ChIP-Seq peaks), then you don't have to worry about this so much since the motif signal is very strong.

To find longer version of a given motif:

The local optimization phase handles long motifs pretty well - long motifs cause more of a problem with the global search phase. Usually long motifs show enrichment for parts at shorter motif lengths. Another strategy is to first find a short version of the motif (i.e. -len 12), and then rerun HOMER and tell it to optimize the motif at a longer motif length with the "-opt <motif file>". To do this with a motif named "motif1.motif":

```
findMotifsGenome.pl peaks.txt hg18r OutputDirectory -opt motif1.motif -len 30
```

This will enlarge the motif(s) in the motif1.motif to 30 bp and optimize them.

Other things to try:

try to reduce the number of target sequences to include only high quality sequences (such as "focused" peaks or peak with the highest peak scores).
- try limiting the length of sequences used (i.e. "**-size 50**" when using **findMotifsGenome.pl**)
- try limiting the total number of background sequences (i.e. "**-N 20000**" when using **findMotifsGenome.pl**)

In a practical sense, you should be able to search for motifs of length 20 or 30 when analyzing ~10k peaks with parameters "-len 20,30 -size 50 -N 25000 -mis 5". HOMER wasn't really designed to find really long motifs; since it is an empirical motif finder, the sequence "space" gets a bit sparse at lengths >16, but in practice it still works.

How many sequences can HOMER handle?

In theory, a lot (i.e. millions). It has been designed to work well with ~10k target sequences and 50k background sequences. If you are using a large number of sequences with **findMotifs.pl**, you many want to use the "**-b**" option, which switches to the cumulative binomial distribution for motif scoring, which is faster to calculate and gives essentially the same results when using large numbers of sequences. The binomial is used by default in **findMotifsGenome.pl**. (I guess it should be called BOMER !?).

Choosing background sequences

Most of the methods in HOMER attempt to select the proper background for you, but in some cases this doesn't work. Normally, HOMER attempts to normalize the GC content in target and background sequences. If you believe normalizing the CpG content is better, use the option "**-cpg**" when performing motif finding with either **findMotifs.pl** or **findMotifsGenome.pl**.

In some cases the user may have a better idea of what the background should be, so HOMER offers the following options:

Promoters: When using analyzing promoters with **findMotifs.pl**, if you wish to use a specific set of promoters as background, place them in a text file (1st column is the ID) and use the "**-bg <background IDs file>**" option. Genes found in the target and background will be removed from the background set so that they don't cancel out each other. Examples:

- Use expressed genes from a microarray as background
- Use only genes represented on the microarray as background

Genomic Regions: When analyzing peaks/regions with **findMotifsGenome.pl**, you can specify the genomic regions of appropriate background regions by placing them in their own peak file and using the "**-bg <background peak file>**". Examples:

- Specify peaks common to two cell types as background when trying to find motifs specific to a set of cell-type specific peaks - this will help cancel out the primary motif and reveal the co-enriched motifs
- If peaks are near Exons, specify regions on Exons as background to remove triplet bias.

FASTA Files: Here you have (the necessary) freedom to specify whatever you want!

Please note, that if the number of background sequences is small, or similar in number to the number of target sequences, you should consider switching to the hypergeometric distribution to improve accuracy when using **findMotifsGenome.pl** ("**-h**").

You man also want to disable CpG/GC normalization depending on how you selected your background, which can be done with "**-noweight**".

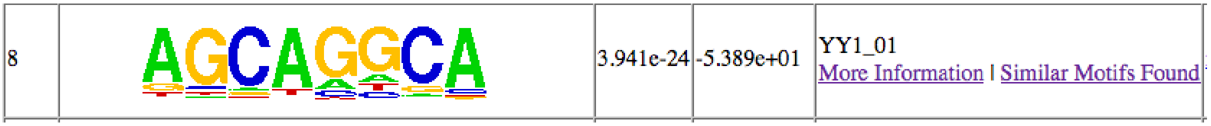
Sequence Bias, GC/CpG normalization, and Autonormalization

Be default, homer performs several normalization steps to make sure the sequences that are being analyzed look reasonable (details here). Since GC% differences are the largest source of bias, these are dealt with during the background selection stage to minimize any issues.

Other types of sequence bias may be present in your data. The purpose of the autonormalization routines ("**-nlen <#>**" and "**-olen <#>**") are there to help deal with this type of bias. If your results have strong enrichment for simple nucleotide repeats, you may want to try "-**olen <#>**" which will more aggressively normalize the data.

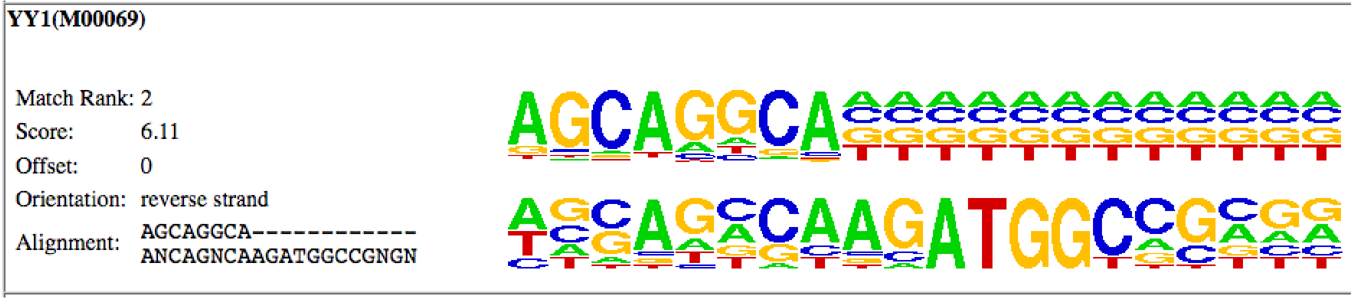
How to Judge the Quality of the Motifs Found

WARNING: Because this is the hardest thing for people to understand, I'll say it again here. HOMER will print the best guess for the motif next to the motif results, but before you tell your adviser that your factor is enriched for that motif, it is highly recommended that you look at the alignment!!! Here is an example of what might be going on:



8	AGCAGGCA	3.941e-24	-5.389e+01	YY1_01 More Information | Similar Motifs Found

In this case, HOMER has identified YY1 as the "best guess" match for this *de novo* motif. Well, lets click on "More Information" and see what's up:



YY1(M00069)

Match Rank:	2
Score:	6.11
Offset:	0
Orientation:	reverse strand
Alignment:	AGCAGGCA------------ ANCAGNCAAGATGGCCGNGN

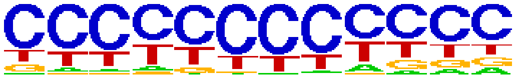
As you can see in this case, the motif aligns to the edge of the known YY1 motif, and not to the core of the YY1 motif (CAAGATGGC). This doesn't mean that the YY1 motif is not enriched in your data, but unless there are other motif results that show enrichment of the other parts of the YY1 motif, it is not likely that the YY1 motif is enriched in your data set.

And as always, remember that HOMER is a *de novo* motif tool!!! Even though HOMER will guess the best match, if it is a novel motif, your don't want to trust that match anyway. Hence, the you can see the importance of viewing the alignment and getting a feel for what evidence exists either for or against this assignment.

There are many cases where HOMER will find motifs with very low p-values, but the motifs might look "suspicious". Poor quality motifs can be loosely classified into the following groups:

Low Complexity Motifs:

(less of a problem with the v3.0+) These types of motifs tend to show preference for same collection of 1, 2, 3, or 4 nucleotides in each position and are typically very degenerate. For example:

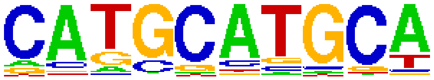


These motifs typically arise when a systematic bias exists between target and background sequence sets. Commonly they will be very high in GC-content, in which case you may want to try adding "**-gc**" to your motif finding command to normalize by total GC-content instead of CpG-content.

Other times this will come up when analyzing sequences for various genomic features that have not been controlled for in the background - for example, comparing sequences from promoters to random genomic background sequences in some organisms will show preferences for purines or pyrimidines. HOMER is very sensitive, so if there is a bias in the composition of the sequences, HOMER will likely pick it up. Autonormalization in the new version minimizes this problem.

Simple Repeat Motifs:

(less of a problem with the v3.0+) Some times motifs will show repeats of certain patterns:



Usually motifs like this will be accompanied by several other motifs looking highly similar. Unless there is a good reason to believe these may be real, it's best to assume there is likely a problem with the background. These can arise if your target sequences are highly enriched on exons (think triplets) and other types of sequences, and if "**-gc**" doesn't help, you may have to think hard about the types of sequences that you are trying to analyze and try to match them. (i.e. Promoters vs. Promoters, Exons vs. Exons etc.) You can also try upping the ante by using "**-olen <#>**" to autonormalize sequence bias at the oligo level.

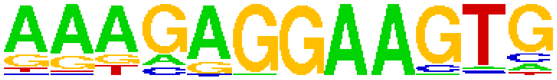
Small Quantity Motifs / Repeats:

These are a little harder to explain. These look like real motifs but are found in an incredibly low percentage of targets - i.e. like an oligo or part of a repeat that is in a couple of the target sequences that appears as a significant motif. Statistically speaking they are enriched, but likely not real. These are the biggest problem when looking for motifs in promoters from a small list of regulated genes. In principle, in a motif is present in **less than 5% of the targets sequences**, there may be a problem.

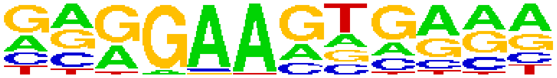
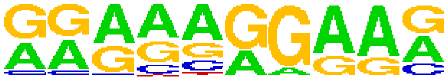
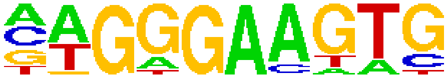
Leftover Junk:

These are motifs that appear in your lower in your results list after you've discovered high quality motifs. If an element is highly enriched in your sequences, HOMER will (hopefully) find it, mask it, and then continue to look for motifs. In this case, many of the other motifs that HOMER finds will be offsets or degenerate versions of highly enriched motif(s) found at the beginning. For example (another PU.1 example):

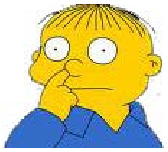
The top motif identified:

Rank	Motif	P-value	log P-pvalue	Best Match/Details
1		0.000e+00	-2.938e+04	PU.1/ThioMac-PU.1-ChIP-Seq/Homer More Information

Examples further down the list:

8		0.000e+00	-3.711e+03	PU.1/ThioMac-PU.1-ChIP-Seq/Homer More Information
22		0.000e+00	-1.692e+03	STAT1/Stat More Information
74		6.727e-190	-4.356e+02	MAF(M00648) More Information

This are not necessarily negative results, but they should be place in context. This commonly happens in ChIP-Seq data sets where the immunoprecipitated protein is highly expressed and binds strongly a ton of binding sites. These "other" motifs are likely also capable of binding PU.1 and probably represent low affinity binding sites, but giving them too much individual attention is not recommended in this context given they are motifs that have been constructed using leftover oligos in the motif finding process that didn't make it into the most highly enrichment motifs. A safer way to approach these elements is to repeat the motif finding procedure with regions lacking the top motif, or by adding "**-mask <motif file>**" to the motif finding command to cleanly mask the top motif from the motif finding procedure.



Can't figure something out? Questions, comments, concerns, or other feedback:
cbenner@ucsd.edu

HOMER

Software for motif discovery and next-gen sequencing analysis

Creating Custom Motif Matrices

A common task when doing regulatory element analysis is to scan for a specific sequence/motif - but not necessarily with one that is found using motif finding. Often you want to find a very specific sequence, or you want to load your own motif matrix derived from another source. This page will help explain how to get HOMER to play nice with your custom motifs.

Below is a description of the HOMER *.motif format for specifying motifs, as well as some tricks & tips on creating simple motif files.

*.motif format files

HOMER works exclusively with homer motif formatted files - so if you want to find a sequence or motif with HOMER, you must first create a "motif" file. A typical motif file will look something like:

```
>ASTTCCTCTT    1-ASTTCCTCTT    8.059752      -23791.535714   0
T:17311.0(44 ...
0.726   0.002   0.170   0.103
0.002   0.494   0.354   0.151
0.016   0.017   0.014   0.954
0.005   0.006   0.027   0.963
0.002   0.995   0.002   0.002
0.002   0.989   0.008   0.002
0.004   0.311   0.148   0.538
0.002   0.757   0.233   0.009
0.276   0.153   0.030   0.542
0.189   0.214   0.055   0.543
```

The first row starts with a ">" followed by various information, and the other rows are the positions specific probabilities for each nucleotide (A/C/G/T). These values do not need to be between 0-1. HOMER will automatically normalize whatever values are there, so interger counts are ok. The header row is actually TAB delimited, and contains the following information:

1. **">" + Consensus sequence (not actually used for anything, can be blank) example: >ASTTCCTCTT**
2. **Motif name (should be unique if several motifs are in the same file) example: 1-ASTTCCTCTT or NFkB**
3. **Log odds detection threshold, used to determine bound vs. unbound sites (mandatory) example: 8.059752**

4. (optional) log P-value of enrichment, example: -23791.535714
5. (optional) 0 (A place holder for backward compatibility, used to describe "gapped" motifs in old version, turns out it wasn't very useful :)
6. (optional) Occurence Information separated by commas, example: T:17311.0(44.36%),B:2181.5(5.80%),P:1e-10317
    1. T:#(%) - number of target sequences with motif, % of total of total targets
    2. B:#(%) - number of background sequences with motif, % of total background
    3. P:# - final enrichment p-value
7. (optional) Motif statistics separated by commas, example: Tpos:100.7,Tstd:32.6,Bpos:100.1,Bstd:64.6,StrandBias:0.0,Multiplicity:1.13
    1. Tpos: average position of motif in target sequences (0 = start of sequences)
    2. Tstd: standard deviation of position in target sequences
    3. Bpos: average position of motif in background sequences (0 = start of sequences)
    4. Bstd: standard deviation of position in background sequences
    5. StrandBias: log ratio of + strand occurrences to - strand occurrences.
    6. Multiplicity: The averge number of occurrences per sequence in sequences with 1 or more binding site.

Only the first 3 columns are needed. In fact, the rest of the columns are really just statistics from motif finding and aren't important when searching for instances of a motif.

The MOST IMPORTANT value is the 3rd column - this sets the detection threshold, which specifies whether a given sequence is enough of a "match" to be considered recognized by the motif. More on that below.

Creating Simple Motifs with seq2profile.pl

HOMER comes with a handly little tool called **seq2profile.pl**. This program automates the creation of motifs from consensus sequences from letters ACGTN.

seq2profile.pl <consensus> [# mismatches] [name] > output.motif
i.e. **seq2profile.pl GGAAGT 0 ets > output.motif**

Output from this example looks like this:

```
>GGAAGT ets     8.28973911259755
0.001   0.001   0.997   0.001
0.001   0.001   0.997   0.001
0.997   0.001   0.001   0.001
0.997   0.001   0.001   0.001
0.001   0.001   0.997   0.001
0.001   0.001   0.001   0.997
```

This script will automatically set the 3rd column to a threshold to only detect perfect matches. Using this file with tools like annotatePeaks.pl will only find sequences matching GGAAGT (or ACTTCC on the reverse strand) If we want to allow up to one mismatch:

i.e. **seq2profile.pl GGAAGT 1 ets > output.motif**

Output from this example looks like this:

```
>GGAAGT ets     1.38498834263571
0.001   0.001   0.997   0.001
0.001   0.001   0.997   0.001
0.997   0.001   0.001   0.001
0.997   0.001   0.001   0.001
0.001   0.001   0.997   0.001
0.001   0.001   0.001   0.997
```

The detection theshold now changed to allow up to one mismatch. This will now identify GGAAGT, GGAAGA, GGAAGC, GGAAGG, ... ettc.

Creating motif files manually

You can also create a motif using a text editor or Excel. The first line should contain 3 tab-separated columns with ">whatever", "name", and "threshold", followed by motif matrix. You can also start from an existing motif or use **seq2profile.pl** to start a motif for you.

Once you get the probabilities the way you want it (i.e. maybe you copied them from JASPAR, or from in vitro affinity selection, or a set of binding sites, or whatever), you need to pick the correct detection threshold. During motif discovery, HOMER optimizes this threshold as part of the algorithm. However, since you're making up your own motif, you need to figure out how degenerate you want your motif to be. This is easily the biggest hang-up for people new to the concept of probability matrices and motif finding.

Motif Scanning

In order to select the proper detection threshold, it helps to understand how motifs are "scored" by HOMER. Any given sequence can be scored using the probability matrix. HOMER calculates the score by adding the "log-odds probabilities" for the nucleotide found in each position.

Score for GGATGT

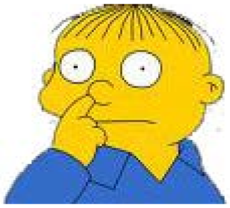
$$\text{score} = \log(pG_1/0.25) + \log(pG_2/0.25) + \log(pA_3/0.25) + \log(pT_4/0.25) + \log(pG_5/0.25) + \log(pT_6/0.25)$$

If the nucleotide in the given position has a high probability in the motif matrix (i.e. 0.9), a positive number is added to the score. If the nucleotide as a neutral probility (i.e. 0.25), then the score is unchanged, and if a nucleotide is disfavored in the probability matrix (i.e. 0.001), then a negative number is added to the score.

Homer fixes the log-odds expectation at 0.25 for each nucleotide. If an organism has a strong imbalance in nucleotide composition, this causes HOMER sacrifices some sensitivity by keeping the expectation at 0.25 instead of adjusting this to reflect the general sequence composition. However, the huge upside to fixing this at 0.25 is that you can use the motif with HOMER on any sequence in any context an always get recognize the same sequences, which makes life much easier.

If the score is above the threshold (from the 3rd column of the motif file), HOMER will identify the sequence as "recognized" by the motif. If the score is lower, the sequence will be ignored. Most motif thresholds are around 5.0-10.0

To select the correct threshold, you may need to "guess and check" you results to ensure your motif is recognizing the correct sequences. Usually, you can start with a threshold around 5-10. For example, run **findMotifs.pl**/**findMotifsGenome.pl** with "**-find motifFile.motif**". The score of each motif is found in the 6th column of the output. Looking through the file and the sequenes that were identified will give you a better idea of what to set the score at.



Can't figure something out? Questions, comments, concerns, or other feedback: cbenner@ucsd.edu