GimmeMotifs documentation

Simon van Heeringen

June 24, 2010

Contents

1 Introduction

GimmeMotifs is a *de novo* motif prediction pipeline, especially suited for ChIP-seq datasets. It incorporates several existing motif prediction algorithms in an ensemble method to predict motifs and clusters these motifs using the WIC similarity scoring metric. It is freely available for download and use under the MIT license. If you find GimmeMotifs useful, please cite:

- van Heeringen SJ and Veenstra GJC, GimmeMotifs: a *de novo* motif prediction pipeline for ChIP-sequencing experiments, *in preparation.*

This document describes how to install and use GimmeMotifs, for theoretical details, please see our publication [van Heeringen *et al.*, in preparation].

Hopefully this document explains at least the basics of installation and usage, but it's probably far from complete. If you have any further question, please don't hesistate to contact me: `s.vanheeringen@ncmls.ru.nl`.

2 Installation

I have tried to make installation of GimmeMotifs as easy as possible. However, as it depends on quite some external packages (motif prediction tools!), it's still

not quite a single-click install. Please make sure all prerequisites are installed before installing GimmeMotifs.

2.1 Prerequisites

GimmeMotifs runs on Linux. Definitely not on Windows, sorry. Mac OS X should work in theory, but as I don't have the option to test this, I'm not completely sure.

Before you can install GimmeMotifs you'll need:

- some Python modules and other packages
- genomic sequences
- motif prediction tools

2.1.1 Required packages (Python)

- Python 2.6 (not Python 3) `http://www.python.org`
- Scipy `http://www.scipy.org/`
  SciPy is the fundamental package needed for scientific computing with Python.
- matplotlib (0.99.1 or higher) `http://matplotlib.sourceforge.net/`
  A python 2D plotting library. All figures and plots produced by GimmeMotifs are made using matplotlib.
- parallel python `http://www.parallelpython.com/`
  A python module which provides mechanism for parallel execution of python code. This Python library is used for parallel execution of for instance the motif finding tools.
- kid `http://www.kid-templating.org/`
  A simple template language for XML based vocabularies; used to produce the HTML reports.

2.1.2 Other required packages

- gsl `http://www.gnu.org/software/gsl/`
  The GNU Scientific Library. Most likely this library is already installed on your system, but it can't hurt to check.
- WebLogo 2.8 (not version 3!) `http://weblogo.berkeley.edu/`
  To visualize sequence logos.

2.1.3 Motif prediction programs

In addition to all the basics you need to get GimmeMotifs up and running you will also need the motif prediction tools, which can be used by GimmeMotifs to predict motifs. You can use any or all of these according to your preference. The following tools are supported by GimmeMotifs:

- MEME [2] `http://meme.sdsc.edu/`
- MDmodule [6] (included in the MotifRegressor Package) `http://www.math.umass.edu/~conlon/mr.html`
- Weeder [7] `http://159.149.109.9/modtools/`
- trawler [3] `http://ani.embl.de/trawler/`
- MotifSampler [9] `http://homes.esat.kuleuven.be/~thijs/Work/MotifSampler.html` (Currently unavailable!)
- GADEM [4] `http://www.niehs.nih.gov/research/resources/software/gadem/index.cfm`
- BioProspector [5] `http://motif.stanford.edu/distributions/bioprospector/`
- Improbizer [1] `http://users.soe.ucsc.edu/~kent/`
- MoAn [10] `http://moan.binf.ku.dk/`

Please consult the respective manuals regarding installation of these tools. As mentioned, you can install any or all of them, according to your needs. However, we recommend installing at least several to leverage the ensemble approach of GimmeMotifs. The top performing tools in our ChIP-seq benchmarks were MEME, MotifSampler and Weeder, but all other tools find motifs that are not found by other tools. In fact, it's because there is no single all-round top performing method that GimmeMotifs exists.

It's always possible to install these programs after installation of GimmeMotifs and update the configuration files to include the new tools. However, during installation, GimmeMotifs will try to find any installed tools and add them automatically, so that's the easiest option.

After installation of MotifSampler, one additional configuration step is necessary, described in section 2.3.4.

2.1.4 Data sources

You will need some genome fasta files for any motif-prediction. Currently GimmeMotifs uses BED files as input (as that's the data-format ChIP-seq pipelines commonly produce), so the genomic fasta files are absolutely required to get the sequence information. These files should be organized in one directory with one file per chromosome or scaffold, with the filename being the chromosome name with an extension of `.fa`, `.fsa` or `.fasta`. No exceptions, no different layouts.

A good source is the UCSC Genome Browser database [8]. For instance, the human hg18 files needed to run the examples included with GimmeMotifs can be downloaded here:
`ftp://hgdownload.cse.ucsc.edu/goldenPath/hg18/bigZips/chromFa.zip`.

All fasta files need to be indexed before GimmeMotifs can use them, see section 2.3.1.

2.2 Building from source

You can download the lastest version of GimmeMotifs at:
`http://www.ncmls.eu/bioinfo/gimmemotifs/`.

Start by unpacking the source archive

```
tar xvzf gimmemotifs-1.00.tar.gz
cd gimmemotifs-1.00
```

You can build GimmeMotifs with the following command:

```
python setup.py build
```

Run the tests to check if the basics work correctly:

```
python run_tests.py
```

If you encounter no errors, go ahead with installing GimmeMotifs (root privileges required):

```
sudo python setup.py install
```

During installation GimmeMotifs will try to locate the tools you have installed. If you have recently installed them, running an `updatedb` will be necessary. Using this option GimmeMotifs will create a configuration file `/usr/share/gimmemotifs/gimmemotifs.cfg` by default. This is a system-wide configuration that can be used by all users.

It is also possible to run the `setup.py install` command with the `--prefix`, `--home`, or `--install-data` options, to install in GimmeMotifs in a different location (for instance, in your own home directory). In this case a configuration file `~/.gimmemotifs.cfg` will be created. However, these alternative methods of installing GimmeMotifs have not been extensively tested. Let me know if you run into problems.

2.3 Configuration

2.3.1 Indexing the genomes

All the genomes that you want to use with GimmeMotifs will need to be indexed for (relatively) fast retrieval of sequences. You can do this, once you have installed GimmeMotifs, by running the following command (as root or with sudo):

```
create_genome_index.py -f /dir/to/fasta/files/ -n genome_name
```

For instance, if I wanted to index the human genome (version hg18) on my computer, where all fasta files are located in the directory `/usr/share/genome/` I would run the following command:

```
sudo create_genome_index.py -f /usr/share/genome/hg18/ -n hg18
```

Repeat this step for every additional genome or organism that you want to use GimmeMotifs with.

Please note: for Weeder, currently only hg18, hg19, mm9, sacCer2 and xenTro2 are supported as organism names (following the UCSC naming convention). This will be fixed as a configuration file in a later release.

2.3.2 Adding gene files

When using the `genomic_matched` background setting (which is the default), there needs to be a file describing genes in BED format in the `/usr/share/gimmemotifs/genes/` directory. The file needs to be named `<index_name>.bed`, so for instance `hg18.bed`. By default `hg18.bed`, `mm9.bed` and `xenTro2.bed` are included.

2.3.3 The easy way: `add_organism.py`

The script `add_organism.py` combines the previous two steps (indexing the fasta files, and adding a gene file), and makes sure the gene BED file is in the correct place with the correct name. This is the easiest way to add a new genome/organism for use with GimmeMotifs.

2.3.4 MotifSampler configuration

If you want to use MotifSampler there is one more step that you'll have to take *after* installation of GimmeMotifs. For every organism, you'll need a MotifSampler background. These can be obtained from `http://homes.esat.kuleuven.be/~thijs/Work/BackgroundModel.html` or, alternatively, be created with `CreateBackgroundModel` (which can be downloaded from the same site as MotifSampler). The background model file needs to be saved in the directory `/usr/share/gimmemotifs/MotifSampler` and it should be named `<organism_index_name>.bg`. So, for instance, if I downloaded the human epd background (`epd_homo_sapiens_499_chromgenes_non_split_3.bg`), this file should be saved as `/usr/share/gimmemotifs/MotifSampler/hg18.bg`.

2.3.5 Other configuration options

All of GimmeMotifs configuration is stored in `/usr/share/gimmemotifs/gimmemotifs.cfg` or `~/.gimmemotifs.cfg`. If the file `~/.gimmemotifs.cfg` exists in your home directory this will always have precedence over the system-wide configuration. The configuraton file is created at installation time with all defaults set, but you can always edit it afterwards. It contains two sections `main` and `params` that

take care of paths, file locations, parameter settings etc. Additionally, every motif tool has it's own section. Let's have a look at the options.

```
[main]
index_dir = /usr/share/gimmemotifs/genome_index
template_dir = /usr/share/gimmemotifs/templates
seqlogo = /usr/local/bin/seqlogo
score_dir = /usr/share/gimmemotifs/score_dists
motif_databases = /usr/share/gimmemotifs/motif_databases
gene_dir = /usr/share/gimmemotifs/genes
```

- `index_dir` The location of the indeces of the genome fasta-files.
- `template_dir` The location of the KID html templates, used to generate the reports.
- `seqlogo` The seqlogo executable.
- `score_dir` To generate p-values, a pre-calculated file with mean and sd of score distributions is needed. These are located here.
- `motif_databases` For now contains only the JASPAR motifs.
- `gene_dir` A bed-file containing gene locations for every indexed organism. This is needed to create the matched genomic background.

```
[params]
background = genomic_matched,random
use_strand = False
tools = MDmodule,Weeder,MotifSampler
analysis = medium
pvalue = 0.001
width = 200
fraction = 0.2
genome = hg18
lwidth = 500
cluster_threshold = 0.95
available_tools = Weeder,MDmodule,MotifSampler,gadem,meme,trawler
abs_max = 1000
enrichment = 1.5
```

This section specifies all the default GimmeMotifs parameters. Most of these can also be specified at the command-line when running GimmeMotifs, in which case they will override the parameters specified here.

3 Usage

3.1 Quick GimmeMotifs example

You can try GimmeMotifs with two example datasets included in the examples directory. This example does require you to have hg18 present and indexed. Change to the examples directory and run the following command:

```
gimme_motifs.py -i TAp73alpha.fa -n p73
```

The `-n` or `--name` option defines the name of the output directory that is created. All output files are stored in this directory.

Depending on your computer you may have to wait some minutes for your results. Once GimmeMotifs is finished you can open `p73/p73_motif_report.html` in your browser.

3.2 Detailed options

- `-i` or `--inputfile`

  This is the only mandatory option. The inputfile needs to be in BED format: at least three tab-seperated columns describing chromosome name, start and end. The fourth column is optional, if specified it will be used by MDmodule to sort the features before motif prediction. GimmeMotifs will take the middle of these features, and subsequently extend those to the width specified by the `width` parameter (see below).

- `-n` or `--name`

  The name of your analysis. All outputfiles will be stored in a directory named as given by this parameter. By default this will be gimmemotifs_dd_mm_yyyy, where d,m and y are the current day, month and year respectively.

- `-a` or `--analysis`

  The size of motifs to look for: small (5-8), medium (5-12) or large (5-15). The larger the motifs, the longer GimmeMotifs will run.

- `-g` or `--genome`

  Name of the genome (index) to use. For instance, for the example in section 2.3.1 this would be `hg18`.

- `-s` or `--singlestrand`

  Only use the + strand for prediction (off by default).

- `-f` or `--fraction`

  This parameter controls the fraction of the sequences used for prediction. This 0.2 by default, so in this case a randomly chosen 20% of the sequences

will be used for prediction. The remaining sequences will be used for validation (enrichment, ROC curves etc.). If you have a large set of sequences (ie. most ChIP-seq peak sets), this is fine. However, if your set is smaller, it might be worthwile to increase this prediction fraction.

- `-w` or `--width`

  This is the width of the sequences used for motif prediction. Smaller sequences will result in a faster analysis, but you are of course limited by the accuracy of your data. For the tested ChIP-seq data sets 200 performs fine.

- `-e` or `--enrichment`

  All motifs should have an absolute enrichment of at least this parameter compared to background to be called significant.

- `-p` or `--pvalue`

  All motifs should have a pvalue of at most this parameter (hypergeometric enrichment compared to background) to be called significant.

- `-b` or `--background`

  Type of background to use. By default `random` (1st order Markov model, similar dinucleotide frequencies as your sequences) and `matched_genomic` (randomly chosen from the genome with a similar distribution respective to the TSS of genes) are used.

- `-l` or `--localization_width`

  Width used in the positional preference plots.

- `-t` or `--tools`

  A comma-seperated list of all the motif prediction tools to use. By default all installed tools that are specified in the GimmeMotifs configuration file are used.

3.3 Other scripts

In addition to `gimme_motifs.py` the GimmeMotifs package contains several other tools that can perform the various substeps of GimmeMotifs, as well as other useful tools. Run them to see the options.

```
closest_motif_match.py
create_genome_index.py
generate_sequences.py
motif_cluster.py
motif_localization_plots.py
motif_roc.py
motif_roc_metrics.py
```

```
pwm2logo.py
pwmscan.py
track2fasta.py
```

4 Acknowledgements

We are grateful to Waseem Akhtar, Robert Akkers, Max Koeppel, Evelyn Kouwenhoven, Leonie Smeenk and Jo Zhou for providing data and feedback during GimmeMotifs development. Also we would like to thank Stefanie Bartels, Adalberto Costessi, Joost Martens and Nagesha Rao for testing and helpful discussion. Of course GimmeMotifs by itself wouldn't be able to do anything, if there wasn't such a number of excellent tools available. Therefore, thanks to all the authors of the motif prediction programs for making their software publicly available!

References

[1] Wanyuan Ao, Jeb Gaudet, W. James Kent, Srikanth Muttumu, and Susan E. Mango. Environmentally induced foregut remodeling by PHA-4/FoxA and DAF-12/NHR. *Science*, 305(5691):1743–1746, September 2004. doi: 10.1126/science.1102216. URL `http://www.sciencemag.org.proxy.ubn.ru.nl:8080/cgi/content/abstract/305/5691/1743`.

[2] Timothy L. Bailey, Mikael Boden, Fabian A. Buske, Martin Frith, Charles E. Grant, Luca Clementi, Jingyuan Ren, Wilfred W. Li, and William S. Noble. MEME SUITE: tools for motif discovery and searching. *Nucl. Acids Res.*, 37(suppl_2):W202–208, July 2009. doi: 10.1093/nar/gkp335. URL `http://nar.oxfordjournals.org/cgi/content/abstract/37/suppl_2/W202`.

[3] Laurence Ettwiller, Benedict Paten, Mirana Ramialison, Ewan Birney, and Joachim Wittbrodt. Trawler: de novo regulatory motif discovery pipeline for chromatin immunoprecipitation. *Nat Meth*, 4(7):563–565, July 2007. ISSN 1548-7091. doi: 10.1038/nmeth1061. URL `http://dx.doi.org/10.1038/nmeth1061`.

[4] Leping Li. GADEM: a genetic algorithm guided formation of spaced dyads coupled with an EM algorithm for motif discovery. *Journal of computational biology : a journal of computational molecular cell biology*, 16(2):317–329, February 2009. ISSN 1066-5277. doi: 10.1089/cmb.2008.16TT. PMID: 19193149 PMCID: 2756050.

[5] X Liu, D L Brutlag, and J S Liu. BioProspector: discovering conserved DNA motifs in upstream regulatory regions of co-expressed genes. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, pages

127–138, 2001. ISSN 1793-5091. URL http://www.ncbi.nlm.nih.gov.proxy.ubn.ru.nl:8080/pubmed/11262934. PMID: 11262934.

[6] X. Shirley Liu, Douglas L. Brutlag, and Jun S. Liu. An algorithm for finding protein-DNA binding sites with applications to chromatin-immunoprecipitation microarray experiments. *Nat Biotech*, 20(8):835–839, 2002. ISSN 1087-0156. doi: 10.1038/nbt717. URL http://dx.doi.org/10.1038/nbt717.

[7] Giulio Pavesi, Paolo Mereghetti, Giancarlo Mauri, and Graziano Pesole. Weeder web: discovery of transcription factor binding sites in a set of sequences from co-regulated genes. *Nucl. Acids Res.*, 32(suppl_2):W199–203, July 2004. doi: 10.1093/nar/gkh465. URL http://nar.oxfordjournals.org/cgi/content/abstract/32/suppl_2/W199.

[8] Brooke Rhead, Donna Karolchik, Robert M. Kuhn, Angie S. Hinrichs, Ann S. Zweig, Pauline A. Fujita, Mark Diekhans, Kayla E. Smith, Kate R. Rosenbloom, Brian J. Raney, Andy Pohl, Michael Pheasant, Laurence R. Meyer, Katrina Learned, Fan Hsu, Jennifer Hillman-Jackson, Rachel A. Harte, Belinda Giardine, Timothy R. Dreszer, Hiram Clawson, Galt P. Barber, David Haussler, and W. James Kent. The UCSC genome browser database: update 2010. *Nucl. Acids Res.*, 38(suppl_1):D613–619, January 2010. doi: 10.1093/nar/gkp939.

[9] G Thijs, M Lescot, K Marchal, S Rombauts, B De Moor, P Rouz, and Y Moreau. A higher-order background model improves the detection of promoter regulatory elements by gibbs sampling. *Bioinformatics (Oxford, England)*, 17(12):1113–1122, December 2001. ISSN 1367-4803. URL http://www.ncbi.nlm.nih.gov/pubmed/11751219. PMID: 11751219.

[10] Eivind Valen, Albin Sandelin, Ole Winther, and Anders Krogh. Discovery of regulatory elements is improved by a discriminatory approach. *PLoS Comput Biol*, 5(11):e1000562, November 2009. doi: 10.1371/journal.pcbi.1000562. URL http://dx.doi.org/10.1371/journal.pcbi.1000562.