HHS Public Access

Author manuscript

Curr Protoc Bioinformatics. Author manuscript; available in PMC 2017 August 21.

Published in final edited form as:

Curr Protoc Bioinformatics. 2013 March ; CHAPTER 8: Unit–8.17. doi:10.1002/0471250953.bi0817s41.

Using MEMo to Discover Mutual Exclusivity Modules in Cancer

Giovanni Ciriello[1], **Ethan Cerami**[1], **Bulent Arman Aksoy**[1], **Chris Sander**[1], and **Nikolaus Schultz**[1]

[1]Computational Biology Center, Memorial Sloan-Kettering Cancer Center, New York, New York 10065, USA

Abstract

Although individual tumors show surprisingly diverse genomic alterations, these events tend to occur in a limited number of pathways, and alterations that affect the same pathway tend to not co-occur in the same patient. While pathway analysis has been a powerful tool in cancer genomics, our knowledge of oncogenic pathway modules is incomplete. To systematically identify such modules, we have developed a novel method, Mutual Exclusivity Modules in Cancer (MEMo). The method searches and identifies modules characterized by three properties: (1) member genes are recurrently altered across a set of tumor samples; (2) member genes are known to or are likely to participate in the same biological process; and (3) alteration events within the modules are mutually exclusive. MEMo integrates multiple data types and maps genomic alterations to biological pathways. MEMo's mutual exclusivity uses a statistical model that preserves the number of alterations per gene and per sample. The MEMo software, source code and sample data sets are available for download at: http://cbio.mskcc.org/memo

Keywords

Mutual Exclusivity; Network Modules; Cancer Genomics

INTRODUCTION

The abundance of genomic data available today requires appropriate tools to make that data more informative and, thus, to guide functional validation studies. In cancer genomics, the highly altered genomes of by cancer cells makes it extremely challenging to predict which alterations are more likely to contribute to tumor initiation and progression. These alterations are commonly referred to as *driver* alterations. Researchers rely on prior biological knowledge and statistics to identify putative driver alterations, using involvement in specific cellular processes and high recurrence as key features for an alteration to be considered oncogenic. The study of functional interrelations of genes and, consequently, of alterations targeting sets of genes, shifts the focus from single proteins to intra-cellular pathways. This systematic perspective allows for the exploration of new features to identify tumor drivers.

In this context we developed a new method, MEMo (Mutual Exclusivity Modules), which integrates prior biological knowledge, recurrence of alterations, and interrelations between multiple alterations in the form of mutual exclusivity. MEMo identifies gene modules whose

components are frequently altered, likely to belong to the same pathway, and whose alterations tend to not co-occur in patients. Statistically significant mutual exclusivity between recurrent alterations strengthens the hypothesis of functional relatedness and can be explained by two scenarios. First, mutual exclusivity may highlight multiple ways to de-regulate the same pathway, suggesting that a second alteration of the same pathway won't confer a further selective advantage to the tumor. Second, mutual exclusivity may be explained by synthetic lethal interactions – in such a scenario a second alteration actually confers a disadvantage to the cell.

MEMo has so far been successfully used in multiple projects from the Cancer Genome Atlas and others [1–4]. In this unit we describe the functionalities MEMo and its required input data through three main protocols. We start by describing how to setup MEMo on your computer, then Basic Protocol 1 introduces you to the method by providing detailed instructions on how to use it on the example data provided with the software distribution. Within Basic Protocol 2, we expand the input data description to allow the user to run MEMo using standard copy number and somatic mutation data derived from commonly used tools in cancer genomics. Finally, Basic Protocol 3 provides an alternative way to import alteration data that does not require the use of external tools. MEMo is freely available for download at: http://cbio.mskcc.org/memo.

Support Protocol: SETUP ON LINUX or MAC OS

In this section we describe the few required steps to setup MEMo on your machine.

In order to run MEMo on a data set, you have to make sure that the following environment variables are set before running any of the scripts:

1. **`MEMO_HOME`**: This environment variable should point to where MEMo files were installed
2. **`PATH`**: This environment variable is used to run applications without specifying the full path. Before running MEMo, make sure this variable includes the MEMo bin directory so that memo scripts can be run without using full paths

Assuming that you have extracted the MEMo folder under/path/to, you can easily set these two variables on a Unix machine via the following steps:

```
> export MEMO_HOME="/path/to/memo"
> export PATH=$MEMO_HOME/bin:$PATH
```

To double-check whether you have defined these variables properly, the following commands should first print out the directory, then the paths to the executable files and finally the location of the main MEMo script:

```
> echo $MEMO_HOME
> echo $PATH
> which memo.py
```

Before you can execute MEMo, you must load the embedded database with background information, including information regarding human genes and human interaction networks. You have the option of loading HRN1 or HRN2 (see [5] for details), both the networks are available in the data folder.

To load HRN1:

```
> cd bin
> ./loadHrn1.sh
```

Loading of the database should take less than 3 minutes.

Setup on Windows

Windows users may need to install JAVA and Python unless they are already installed. Latest releases of these programming languages and detailed instructions for installation can be found here:

JAVA distributions: http://www.java.com/en/download/manual.jsp

Python distributions: http://www.python.org/download/releases

After that, you will need to:

- Add **`MEMO_HOME`** as an environment variable, and set it to your memo directory. In Windows, this is usually done via the Control Panel. Complete instructions are available at: http://www.cs.usask.ca/~wew036/latex/env.html.
- Add **`MEMO_HOME/bin`** to your global path. In Windows, this is also done via the Control Panel. See the URL above, if you need detailed instruction.

OPTIONAL SETUP: COMPILE MEMo FROM SOURCE CODE

The default MEMo distribution comes with pre-compiled binaries that are compatible with Java 1.6.x or higher. Most of the modern operating systems come with the newest version of Java Programming Language; therefore most users do not need to execute these steps. But if you are using an older version of Java or if you want to run MEMo on a custom environment, you can re-compile MEMo with **`ant`**.

In order to do this,

1. Change you working directory to the main MEMo directory

   ```
   > cd $MEMO_HOME
   ```

2. Run ant to compile from the sources

   ```
   > ant
   ```

These steps should replace the binary files with the ones that are compatible to the Java version that you are using on your system.

BASIC PROTOCOL 1: RUNNING MEMo ON TEMPLATE DATA

The main distribution of MEMo, for demonstration purposes, comes with two example data sets on which you can run MEMo out-of-the-box: TCGA Glioblastoma Multiforme (GBM) [1] and TCGA Ovarian Cancer [2]. Required material to run MEMo on these datasets can be found within the folders:

- `cancer_data/gbm`
- `cancer_data/ovarian`

MEMo requires a configuration file, in which you need to specify input file locations and parameter settings. In this section, we will first provide a detailed description of the configuration file and all its properties, and then we will show how to run MEMo on the GBM dataset.

Configuration File

A configuration file needs to be provided in the form <property>:<value>. A set of properties is provided by MEMo and needs to be specified as described in this sections. Details on the input file format are given in the next section. Main properties and their description are listed below:

`case_file` **# List of sample case IDs**

Name of the file containing identifiers for all cases or patients in your study. This file simply contains a list of case identifiers, one per line.

`mutation_file` **# List of Somatic Mutations**

Name of the file containing mutations observed within your study. File must be formatted in the Mutation Annotation Format (MAF) file format.

`cna_file` **# Copy Number Alterations (CNA) matrix**

Name of the file containing discretized copy number calls within your study. Copy number calls are determined by tools like GISTIC [6] or RAE [7]. Details are provided in the next section.

`copy_number_driven_genes_file` **# List of genes to use for CNA analysis**

Name of the file containing genes that pass an mRNA concordance filter test. This file simply contains a list of HUGO gene symbols, one per line.

`gistic_del_file` **# Regions of recurrent copy number loss**

Name of the file containing recurrently deleted regions of interest as determined by GISTIC or RAE. Details are provided in the next section.

`gistic_amp_file` **# Regions of recurrent copy number gain**

Name of the file containing recurrently amplified regions of interest as determined by GISTIC or RAE. Details are provided in the next section.

`mut_sig_file` **# Mutation recurrence analysis**

Name of the file containing results of mutation recurrence analysis provided by tools like MuSiC [8] or MutSig [9].

`mut_sig_q_value_threshold` **# Mutation significance threshold**

Threshold q-value for the mutation recurrence analysis. Genes with a q-value less than or equal to this threshold are included in the MEMo analysis.

sig_genes **# Recurrently mutated genes**

Name of the file listing recurrently mutated genes. To use if analyses of recurrence from either MutSig or MuSiC are not available. The user can specify a list of recurrently mutated genes in this file by listing the corresponding HUGO symbols one per line. This property is ignored if `mut_sig_file` and `mut_sig_q_value_threshold` are specified.

`min_number_of_alterations` **# Minimum number of alterations**

Only genes with the minimum number of alteration events are included for MEMo analysis.

`min_number_of_mutations` **# Minimum number of mutations**

Alteration threshold for mutations only. If omitted, the minimum number of mutations will be set equal to `min_number_of_alterations`.

`title` **# Descriptive title of your project**

The configuration file for the GBM project can be found in `cancer_data/gbm/` under the name of `memo_gbm.props` :

```
case_file=cases_all_three.txt

mutation_file=data_mutations_MAF.txt

cna_file=data_CNA_RAE.txt

copy_number_driven_genes_file=copy_number_driven_genes.txt

gistic_del_file=GBM_GISTIC_Del.txt

gistic_amp_file=GBM_GISTIC_Amp.txt

mut_sig_file=sig_genes_phase_1_2.txt

mut_sig_q_value_threshold=.05

min_number_of_alterations=4

title=TCGA Glioblastoma Multiforme, Based on Phase I and II
Sequencing Data
```

Running MEMo

Now that your configuration file has been set up, you can easily run MEMo on the GBM dataset. Command line examples are given for a UNIX-based environment.

1. On the console, change your current working directory to the example GBM data folder:

```
> cd $MEMO_HOME/cancer_data/gbm
```

2. Launch MEMo using the python script `memo.py` and the configuration file `memo_gbm.props`

```
> memo.py memo_gbm.props
```

You should then see output like this:

```
-------------------------------------------
MEMo Network Analysis.
Computational Biology Center, MSKCC.
-------------------------------------------
Initializing Database. This will take a few moments...
....
```

MEMo will automatically write its results to a file called `MemoReport.txt`, providing details of the extracted modules and statistical significance of mutual exclusivity. The report file is organized in the following tab-delimited columns:

`Module ID`	(Col 1)	Module numerical identifier.
`Genes`	(Col 2)	List of genes in the module and number of alterations per gene.
`Total Percent Altered Cases`	(Col 3)	Percentage of altered cases
`Total Altered Cases`	(Col 4)	Actual number of altered samples
`Multiple Alteration Samples`	(Col 5)	Number of samples with alterations in more than one gene of the module
`p-value`	(Col 6)	Nominal p-value
`p*-value`	(Col 7)	False Discovery Rate corrected p-value

Examples of MEMo report entries (from `cancer_data/gbm/MemoReport.txt`):

```
…
M1  CDKN2A [62], MDM2 [16], TP53 [44], 72.54%  103  19  0.0  0.01
M2  CDK4 [22], CDKN2A [62], TP53 [44], 76.06%  108  19  0.0  0.01
…
```

BASIC PROTOCOL 2: RUNNING MEMo INTEGRATING COPY NUMBER ALTERATIONS AND SOMATIC MUTATIONS

Now that you have MEMo set up and running, let's use it on your own data. MEMo integrates copy number alterations and somatic mutations by loading and processing the output generated by software like GISTIC [6] or RAE [7] for copy number changes, and MutSig or MuSiC [8] for mutations. These tools analyze the recurrence of each genomic alteration and determine its statistical significance. If you don't have data from either one of these tools, ***BASIC PROTOCOL 3*** describes how to create your own alteration table and how to use it with MEMo. In the next paragraphs, we will briefly describe the output from GISTIC and MutSig (a similar format is provided by RAE and MuSiC) and what is required to run MEMo with this type of data. For a complete description of these tools, we refer to the respective publications.

Materials

Copy Number Alteration Files (GISTIC)—As discussed in the previous section, MEMo uses a configuration file with multiple properties; precisely, MEMo loads copy number data through 3 properties: `cna_file`, `gistic_amp_file`, `gistic_del_file`.

`cna_file`: Here you need to specify the actual copy number discretized data. This file is given as output by GISTIC under the name `all.thresholded.by_genes.txt`.

Copy number data has to be provided in a matrix format with samples in columns (the first row contains the list of sample IDs) and genes in rows (the first column contains gene IDs, both Entrez Gene ID and HUGO official gene symbols are accepted). The actual matrix entries are discretized copy number calls following the standard adopted by both GISTIC and RAE:

`-2`	homozygous deletion
`-1`	hemizygous loss
`0`	diploid
`1`	low level copy number gain
`2`	high level amplification

An example of a valid copy number matrix is "data_CNA_RAE.txt" stored in the "cancer_data/gbm" folder from the MEMo distribution.

Somatic Mutation Files (MutSig)

`gistic_amp_file/gistic_del_file:` These files specify the genomic regions that are found to be recurrently gained or lost in the sample set analyzed by GISTIC. These files are included in the GISTIC output under the name of: `table_amp.conf_99.txt` for recurrently amplified regions, and `table_del.conf_99.txt` for recurrently deleted regions.

Each file consists of a table with an entry for each region. Each entry provides a full description of the region of interest (ROI) including: numerical index [Col. 1], chromosome location [Col. 2], three boundary definitions for the ROI (region [Col. 3–4], peak [Col. 5–6], and enlarged peak [Col. 7–8]), number and gene symbols of genes in the regions [Col. 9–10], number and symbols of genes in the peak [Col. 11–12]. Examples of valid ROI files are "GISTIC_GBM_Amp.txt" and "GISTIC_GBM_Del.txt" stored in the "cancer_data/gbm" folder from the MEMo distribution.

`mutation_file:` This file specifies the complete list of somatic mutations in your dataset. The test files provided with the MEMo distribution follow the Mutation Annotation Format (MAF) adopted by the TCGA projects. Even though this files contains several columns providing multiple details for each mutation, MEMo uses only 3 columns recognized by their headers:

`Tumor_sample_barcode:` This column indicates the sample identification, these IDs have to match those provided in the `case_file` file.

`Hugo_Symbol`: This column indicates the official HUGO gene symbol.

`Variant_Classification:` This column indicates the mutation type. MEMo automatically excludes mutations that are annotated as Silent, Intron, or LOH.

It is important to remember that MEMo interprets the MAF file by reading its header. If any of the three properties specified above are absent or identified by different column labels, MEMo will fail to load mutation data and return an error.

An example of a valid mutation data file is " `data_mutations_MAF.txt`" stored in the "cancer_data/gbm" folder from the MEMo distribution.

Finally, you need to specify for which set of genes you wish to include somatic mutation data. Users are allowed to do this in two ways:

`mut_sig_file:` If results from MutSig are available, the report file returned by MutSig can be used by MEMo and it has to be specified using this property. MutSig report includes an entry for each gene, and entries are ranked according to statistical significance of recurrence of mutations in the corresponding gene. MEMo uses two columns of this file: the gene column specifying the gene symbol (Col. 2), and the q column, where FDR corrected q-values are reported (Col. 11). Genes are included in the analysis if the corresponding q-values are below a threshold specified by the user using the mut_sig_q_value_threshold property (default = 0.05).

An example MutSig report is "sig_genes_phase_1_2.txt" stored in the "cancer_data/gbm" folder from the MEMo distribution.

`sig_genes:` If results from MutSig are not available or not in the proper format, the user may simply provide a list of genes of interest using this property. The user needs to create a text file with one gene per row identified by the corresponding HUGO gene symbol.

BASIC PROTOCOL 3: RUNNING MEMo WITH CUSTOMIZED ALTERATIONS

In the previous section we described MEMo input data as processed copy number alterations and mutation data coming from unbiased statistical analyses. Suppose now, we want to integrate these alterations with custom events of interest for the studied cancer type (e.g. *BRCA1* hyper-methylation in ovarian cancer). MEMo allows the user to define events in a simple and intuitive way using a binary matrix representation. Custom alterations may be used together with previously defined copy number changes and somatic mutations, or on their own. The latter case allows the user to run MEMo even if he does not have statistical results from other methods such as GISTIC, RAE, MuSiC, or MutSig.

Materials: Defining Customized Alterations for MEMo

Custom alterations have to be defined as follow:

1. Define the samples that are affected by your custom alteration, such that a sample either has or does not have the alteration (binary assignments).
2. Generate a binary matrix with samples listed along the rows and custom alterations in the columns. Each matrix entry is either 0 (not altered) or 1 (altered). For example:

```
Event/Sample  BRCA1:methylated
Case-1                0
Case-2                1
Case-3                1
Case-4                0
…
```

3. Save the matrix file as a text file (e.g. “my_custom_alterations.txt”).
4. Edit the .props file by adding the following property:

```
custom_alteration_file= my_custom_alterations.txt
```

Requirements to correctly format the custom alteration file:

1. Case IDs need to be consistent with those specified in the `case_file` specified file;
   a. case IDs that are in the `case_file` specified file, but not in the custom alteration file will be treated as not altered,
   b. case IDs that are in the custom alteration file, but not in the `case_file` specified file will be ignored.
2. Matrix entries have to be either 1 or 0; entries other than 0 or 1 will be treated as 0.
3. Column names need to follow this format:

```
<gene_symbol_A>,<gene_symbol_B>:<optional description>
```

a. If the custom alterations refer to multiple genes, these have to be listed using their gene symbols and separated by commas

b. Genes that are targets of the alteration need to be specified

c. Columns not referring to specific genes will not be used, as the corresponding node cannot be mapped onto the reference pathway network.

d. All comma-separated genes will be treated as equally altered unless other alterations are elsewhere specified.

e. The colon separates a gene list from an optional description of the alteration.

Valid examples of column headers are:

```
BRCA1:hyper-methylation
EGFR,ERBB2,ERBB3:high-phosphorylation
```

Protocol Steps

Running MEMo using customized alterations is done as described in Protocol 1 and 2, once the `custom_alteration_file` property has been added to the .props file, and the custom alteration file has been properly formatted as described in the previous section.

The user can run MEMo either integrating custom alterations with copy number changes and somatic mutations (A), or using only custom alterations (B).

A. As for copy number changes and somatic mutations, MEMo relies on the general abstraction of alteration per gene, thus, if multiple types of alteration target the same gene X, these will be all assigned to the same node, representing gene X, in the network and module representation (e.g. BRCA1 hyper-methylated cases and BRCA1 mutated cases will be considered together as BRCA1 altered cases).

B. If copy number regions of interest and recurrently mutated genes are not provided or not available, the user can specify only the custom alterations and only these will be used in the analysis.

COMMENTARY

Background Information

MEMo software is written in the Java programming language. It uses Hibernate and the Java HyperSQL embedded database to store the Human Reference Networks (HRNs) and Entrez Gene information, and uses the Java JUNG library for all graph operations.

Critical Parameters

Critical parameters when running MEMo mainly concern its input files. First, if you plan to use MEMo with standard output from tools like GISTIC, RAE, MutSig, or MuSiC beware that future versions of these tools may modify the output format. Before running MEMo carefully check that the input format matches those described in Basic protocol 2. Second, independently of the way you selected your input, MEMo relies on background pathway networks to estimate functional relatedness between genes. For altered genes to be successfully mapped onto these networks, you need to make sure to either use HUGO official gene names, or Entrez Gene IDs. The file `Homo_sapiens.gene_info`, in the data folder from the MEMo distribution, provides extensive information on more human coding sequences, including official gene names, Entrez Gene ID and multiple synonyms.

Troubleshooting

You can control the space of solutions explored by MEMo in two ways: by selecting the set of recurrently altered genes to use as input, by modifying the recurrence thresholds (`min_number_of_alterations/min_number_of_mutations`). These choices may seriously affect both the results and time performances.

If your selection criteria are too stringent, the network explored by MEMo will include only few genes, and even less (if any) fully connected modules. Remember that MEMo consider as a Module gene sets containing at least 3 genes. On the other hand, too permissive thresholds will generate incredibly an large network to explore. This will both increase the computation time and generate and incredibly large number of cliques to test. Correcting for multiple testing over a huge set of hypothesis will likely return you no statistically significant results.

For optimal results, we suggest to keep the list of recurrently altered genes in the order below 100.

Internet Resources

MEMo source code, required libraries, and sample data are available at: http://cbio.mskcc.org/tools/memo.html

Constantly updated background networks in simple interaction format (SIF) are available through the cBio Cancer Genomics Portal [10] by clicking on the Networks tab. http://www.cbioportal.org/public-portal/

Details on the Mutation Annotation Format (MAF) are available at: https://wiki.nci.nih.gov/display/TCGA/Mutation+Annotation+Format+(MAF)+Specification+-+v2.2

References

1. Comprehensive genomic characterization defines human glioblastoma genes and core pathways. Nature. 2008; 455(7216):1061–8. [PubMed: 18772890]
2. Bell D, et al. Integrated genomic analyses of ovarian carcinoma. Nature. 2011; 474(7353):609–615. [PubMed: 21720365]
3. TCGA. Comprehensive molecular portraits of human breast tumors. Nature. 2012 In Press.

4. Muzny DM, et al. Comprehensive molecular characterization of human colon and rectal cancer. Nature. 2012; 487(7407):330–337. [PubMed: 22810696]
5. Ciriello G, et al. Mutual exclusivity analysis identifies oncogenic network modules. Genome Res. 2012; 22(2):398–406. [PubMed: 21908773]
6. Beroukhim R, et al. Assessing the significance of chromosomal aberrations in cancer: methodology and application to glioma. Proc Natl Acad Sci U S A. 2007; 104(50):20007–12. [PubMed: 18077431]
7. Taylor BS, et al. Functional copy-number alterations in cancer. PLoS One. 2008; 3(9):e3179. [PubMed: 18784837]
8. Dees ND, et al. MuSiC: Identifying mutational significance in cancer genomes. Genome Res. 2012
9. Getz G, et al. Comment on “The consensus coding sequences of human breast and colorectal cancers”. Science. 2007; 317(5844)
10. Cerami E, et al. The cBio Cancer Genomics Portal: An Open Platform for Exploring Multidimensional Cancer Genomics Data. Cancer Discov. 2012; 2(5):401–404. [PubMed: 22588877]