



# HORN MORPHO 3.0 Quick Reference

Michael Gasser  
Indiana University, School of Informatics and Computing  
[gasser@indiana.edu](mailto:gasser@indiana.edu)  
25 January, 2019

## Installation

1. Uncompress the file that you downloaded. This will yield a directory (folder) called `HornMorpho-3.0`, which contains all of the files that you need to run HORN MORPHO.
2. Go to the `HornMorpho-3.0` directory (folder), and enter the following, making sure that you are running some version of Python 3.

```
python setup.py install
```

## Use

### STARTING THE PROGRAM

Start up the Python interpreter, again making sure that you are running at least Python 3.0, and type the following to load the program.

```
import hm
```

### FUNCTIONS

Options for each function are shown with their default values.

```

anal(language, word)
Options: roman=False, root=True, gram=True, citation=True, raw=False,
nbest=100 [Amharic only]

Performs morphological analysis of the word. For ambiguous words returns the first nbest
analyses. For Amharic only, analyses are ordered by their estimated frequency.

>>> hm.anal('ti', 'ፍብ')
word: ፍብ

>>> hm.anal('ti', 'ፔፕሲ')
?word: ፔፕሲ

>>> hm.anal('am', 'የማያስፈልጋትስ')
word: የማያስፈልጋትስ
POS: verb, root: <fl_g>, citation: አስፈለገ
subject: 3, sing, masc
object: 3, sing, fem
grammar: imperfective, causative, relative, definite, negative
conjunctive suffix: s

>>> hm.anal('om', 'afeeramaniiru')
word: afeeramaniiru
POS: verb, root: <afeer>, citation: afeeramuu
subject: 3, plur
derivation: passive
TAM: perfect

>>> hm.anal('ti', 'በዘጋጥመና')
word: በዘጋጥመና
POS: verb, root: <gTm>, citation: አጋጠመ
subject: 3, sing, masc
object: 1, plur
grammar: imperfective, reciprocal, transitive, relative
preposition: bI

>>> hm.anal('am', 'አይደለችም')
word: አይደለችም
POS: copula, root: <ne>
subj: 3, sing, fem
negative

>>> hm.anal('ti', 'ዘየብለይ')
word: ዘየብለይ
POS: verb, root: <al_e>, citation: አለ
subject: 3, sing, masc
object: 1, sing
grammar: present, relative, negative

>>> hm.anal('om', 'dubbanne')
word: dubbanne
POS: verb, root: <dubbadh>, citation: dubbachuu
TAM: past, negative
POS: verb, root: <dubbadh>, citation: dubbachuu

```

```

subject: 1, plur
TAM: past

>>> hm.anal('am', 'lezemedocacnm', roman=True)
word: lezemedocacnm
POS: noun, stem: zemed
possessor: 1, plur
grammar: plural
preposition: le, conjunctive suffix: m

>>> hm.anal('am', 'ቢያስጨንቁአቸው', root=False, gram=False)
word: ቢያስጨንቁአቸው
POS: verb, citation: አስጨነቀ

>>> hm.anal('am', 'ለዘመዶቻችንም', raw=True)
[('zemed', [-acc, cnj='m', der=[-ass], -dis, +plr, pos='n',
poss=[+expl, +pl, -p2, +plr], pp='le', rl=[-acc, +p], v=None))]

>>> hm.anal('am', 'ይመጣሉ')
word: ይመጣሉ
POS: verb, root: <mT'>, citation: መጣ
subject: 3, plur
grammar: imperfective, aux:alle
POS: verb, root: <mTT>, citation: መጠጠ
subject: 3, plur
grammar: imperfective, aux:alle
POS: verb, root: <mT'>, citation: ተመጣ
subject: 3, plur
grammar: imperfective, aux:alle, passive

>>> hm.anal('am', 'ይመጣሉ', nbest=1)
word: ይመጣሉ
POS: verb, root: <mT'>, citation: መጣ
subject: 3, plur
grammar: imperfective, aux:alle

anal_file(language, input_file, output_file)
Options: root=True, gram=True, citation=True, raw=False
Runs anal on the words in a file.

>>> hm.anal_file('am', 'hm/languages/am/data/ag.txt',
'hm/languages/am/data/ag_out.txt')
Analyzing words in hm/languages/am/data/ag.txt
Writing to hm/languages/am/data/ag_out.txt

seg(language, word) [Amharic and Oromo verbs and nouns only]
Options: none
Performs morphological segmentation on the word.

Morphemes are separated by ‘-’; stems/roots appear within ‘{ }’. If raw is True, a list of analyses
is returned, each a part-of-speech, segmentation pair. For complex words, the part-of-speech is
that of the stem or root, not the whole word. If raw is False (the default), the segmentations are

```

printed out, with the part-of-speech appearing before the segmentation, separated by ':', and different analyses/segmentations separated by ';';

```
>>> hm.seg('am', 'ሲያጭበረብሩን')
ሲያጭበረብሩን -- v:s(cnj1)-y(sb=3sm|3p)-{Cbrbr+a12e3e4_5}(imprf,trans)-
u(sb=2p|3p)-n(ob=1p)
```

**seg\_file**(language, input\_file, output\_file)

Options: none

Runs seg on the words in a file and writes them to another file.

```
>>> hm.seg_file('am', 'hm/languages/am/data/ag.txt',
                'hm/languages/am/data/ag_out.txt')
Segmenting words in hm/languages/am/data/ag.txt
Writing to hm/languages/am/data/ag_out.txt
```

**phon**(language, word) [Amharic only]

Options: gram=True

Converts an Amharic word written in Ge'ez characters to a romanized form that shows consonant gemination and the epenthetic vowel (represented by 'I'). If multiple pronunciations are possible, they are ordered by estimated frequency.

```
>>> hm.phon('am', "ይመታሉ")
yImetal_u (132) yIm_et_al_u (61)

>>> hm.phon('am', "ይመታሉ", gram=True)
-- yImetal_u
POS: verb, root: <mt'>
subject: 3, plur
grammar: imperfective, aux:alle
-- yIm_et_al_u
POS: verb, root: <mt'>
subject: 3, plur
grammar: imperfective, aux:alle, passive

>>> hm.phon('am', 'እንደብር')
?IndIbIr (0)
```

**phon\_file**(language, input\_file, output\_file) [Amharic only]

Options: gram=True, print\_ortho=False, word\_sep='\n', anal\_sep=' '

Runs phon on the words in a file.

```
>>> hm.phon_file('am', 'hm/languages/am/data/ag.txt',
                'hm/languages/am/data/ag_phon.txt')
Analyzing words in hm/languages/am/data/ag.txt
Writing analysis to hm/languages/am/data/ag_phon.txt

>>> hm.phon_file('am', 'hm/languages/am/data/ag.txt',
                print_ortho=False, word_sep=':')
Analyzing words in hm/languages/am/data/ag.txt
yIh:meShaf:yezarE:
01:amet:gedema:bedenbu:mIrmera:alfo:tat_Imo:beweT_a:gizE:tal_aq_
tal_aq:cIg_Ir:feTrob_IN_:neb_er:.
```

**exit()**

Options: none

Exits HORN MORPHO, saving any analyses or generated forms in a file to be loaded next time you use the program. This will significantly speed up the performance of the program.

```
>>> hm.exit()
```