**Logging module**

**Warning**

# (function4gff.py, intra\_model.py,and inter\_model.py)

logger.warning('[Missing Attributes] Program failed….\n\t\t- Line {0:s}: {1:s}'.format(str(f['line\_index']+1), f['line\_raw']))

# duplicate key in fasta\_dict (gff3.py)

logger.warning('%s : Line %d : Duplicate %s [%s] : ID = [%s].', fasta\_file\_f.name, line\_num, '||'.join([i for I in keys if flags[i]]), key[:25] + (key[25:] and '..'), entry['id'])

# if exceptions occur when running add\_line\_error (gff3.py)

logger.warning('Fail to check the boundary relationship between parent and child features (Ema0003)...\n\t- Line {0:s}: {1:s}'.format(str(line['line\_index']+1), line['line\_raw']))

# (gff3.py)

logger.warning('[Parent feature missing] Cannot find the parents of this feature. Fail to check the boundary relationship (Ema0003).\n\t- Line {0:s}: {1:s}'.format(str(line['line\_index']+1), line['line\_raw']))

logger.warning('[Phase] check\_phase failed. \n\t\t- Line {0:s}: {1:s}'.format(str(cds\_list[0]['line\_index']+1), cds\_list[0]['line\_raw']))

logger.warning ('Sequence ID {0:s} not found in FASTA file.'.format(seqid))

**Error**

# (function4gff.py and gff3.py)

logger.error ('[Missing ID] A model needs to have a unique ID, but this feature does not. Please fix it before running the program.\n\t\t- Line {0:s}: {1:s}'.format(str(line['line\_index']+1), line['line\_raw']))

* sys.exit() # The program stops.

**sys.exit()**

# no gff or fasta input to gff3\_QC

sys.exit(1)

# Missing ID is detected by function check\_parent\_boundary in gff3\_QC

sys.exit()

# [intra\_model.py] in function check\_internal\_stop

sys.exit('[Error] Start/End is not a valid integer.\n\t\t- Line {0:s}: {1:s}'.format(str(line['line\_index']+1), line['line\_raw']))

**Raise an exception gff3.py**

if not reference:

 raise Exception('External or embedded fasta reference needed')

if len(args) > 1:

 raise TypeError('expected at most 1 arguments, got %d' % len(args))

if len(args) > 2:

 raise TypeError('update() takes at most 2 positional '

 'arguments (%d given)' % (len(args),))

elif not args:

 raise TypeError('update() takes at least 1 argument (0 given)')