Python 3.5 Compatibility Changes to AIMS

The following is a list of changes made to the AIMS program, including explanations for each change, that were made to enable compatibility with Python 3.5. Instructions on the difference in how the program must be run are also present where relevant.

- All `print "text"` statements were converted into `print("text")` function call. Python 3.5 no longer contains the print statement, which has been replaced by the print function.
- All occurrences of `xrange` were replaced with `range`. The Python 3.5 'range' generator functions as the Python 2.7 'xrange' generator. The 'xrange' generator has been removed in Python 3.5.
- Occurrences of `map()` and `zip()` where a function or assignment expects a list have been converted to `list(map())` and `list(zip())` respectively. Python 3.5 'map' and 'zip' functions return an iterator of type 'map' and 'zip' respectively rather than a list as was the case in Python 2.7.
- String formatting in `write_readme()` has been changed to convert non-string data types into their string representation explicitly via `str()`. Python 3.5 doesn't accept non-string data types when formatting a string, which worked in the Python 2.7 version.
- Integer division where the desired data type is 'int' had the '/' operator changed to the '//' operator. Python 3.5 '/' operator now returns a 'float' type even if both operands are of type 'int' and the '//' is required to return type 'int' instead.
- The function `add_constraint(self,(name,distribution))` in AIMS.py has been changed to `add_constraint(self,name_distribution)`. Python 3.5 does not permit tuple parameter unpacking so a single parameter was created instead. The tuple was then unpacked on the first line of the function via `name, distribution = name_distribution` to preserve the rest of the code in its original state.
- Numerous tab spacings that worked in Python 2.7 prevented code from running in Python 3.5 and these were resolved.
- Binary files are now read/written using "rb" and "wb" instead of "r" and "w" due to Python 3.5 opening files differently. This results in binary files created using Python 2.7 being incompatible with Python 3.5. As a result, all binary files to be used in Python 3.5 compatible AIMS must be generated using Python 3.5 compatible AIMS.
- The method `duplicate_ages` checks for duplicate ages within a track in pairs. For the method to return 'True' or 'False', the length of each track must be at least 2 otherwise the program returns 'None'. This raises an exception which doesn't explain the source of the problem and so an if statement that checks for track length has been included.