mostest.js

Test-Suite for the MOSFECCS Structural Formula Editor

(updated for v6_240305)

mostest.js is an ES6 program for testing of the SMILES-generator and SMILES-parser of MOSFECCS under node.js.

In mostest.js, the code sections for calculating SMILES-codes from structural formulae (SMILES-generator; function getsmiles()) and for parsing SMILES-codes and reconstructing the structural formulae (SMILES-parser; function parse_m_Smiles() are identical with the corresponding sections of the MOSFECCS Editor. When the code of functions getsmiles() and/or parse_m_SMILES() in MOSFECCS has been changed, replace these functions in mostest.js accordingly to test the new version (if you use makeversion.pl, this is done automatically by the perl script).

mostest.js requires the name of a text file with SMILES-codes (one per line) as parameter. Each line of the input text file consists of an identifier (alphanumeric string) and a SMILES-code, separated by a tab character. Lines starting with # (comments) are ignored but copied to the SUMMARY file.

Example Input file *(tab separated text)*

```
Mol	SMILES
# example input file for mostest.js
BJ_1_1	CCCCCCC
BJ_1_3	CCCCCCCCCCCC
BJ_1_4	C
BJ_1_5	CCC&CCC
BJ_1_6	CC([SeH])C(N)C(C)Br
BJ_1_8	CNCSCCOC
BJ_1_9	COCCSCNC
BJ_13_10	OC[C@@]1(O)[C@@H]2O[C@@]3([O-])O[C@@H]1[C@@H]4[C@@H](O)NC(=[NH2+])N[C@]4([C@@H]2O)[C@H]3O
```

USAGE: the script mostest.js and the input text file must be in the same folder.

In terminal:

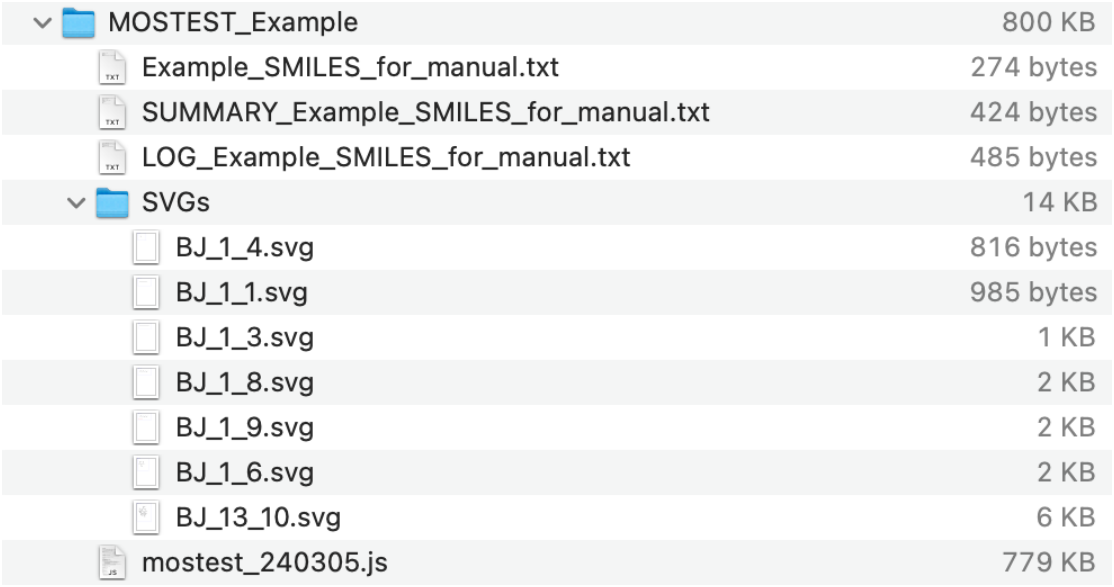
cd to this folder, then, at the terminal prompt, enter:

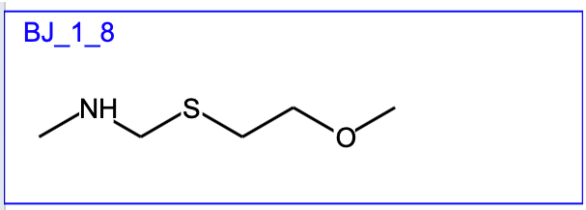
node mostest_<version>.js <inputFilename><return>.

Example console output (from run with version 231115 *of mostest.js):*

```
bj@alanin MOSTEST_Example % node mostest_240305.js Example_SMILES_for_manual.txt
05.03.2024 16:48:14 GMT+0100 (Central European Standard Time)
mostest.js run with txt file: Example_SMILES_for_manual.txt
BJ_1_1
BJ_1_3
BJ_1_4
BJ_1_5
mol:BJ_1_5 Parser ERROR for SMILES: CCC&CCC
     ERR: invalidSymbol in SMILES:"&"
BJ_1_6
BJ_1_8
BJ_1_9
BJ_13_10
16:48:14 GMT+0100 (Central European Standard Time):
/Users/bj/Desktop/MOSFECCS/GITHUB/MOSTEST_Testsuite for
MOSFECCS/MOSTEST_Example/mostest_240305.js run completed
Files SUMMARY_Example_SMILES_for_manual.txt and LOG_Example_SMILES_for_manual.txt
written.
bj@alanin MOSTEST_Example %
```

For each line in the input file, mostest.js parses the SMILES-code, reconstructs the structural formula as data objects (atoms, bonds etc.) in memory and generates an SVG graphic of the resulting structural formula as a file (with a 5-digit number constructed from the identifier as filename) in a subfolder "SVGs".





The SVG files can be visualized as shown above by most browsers.

Checks performed by mostest.js:
mostest.js recalculates the SMILES-codes from the molecular data objects and compares them with the SMILES-code in the input file. Errors and failures are recorded in a LOG file.

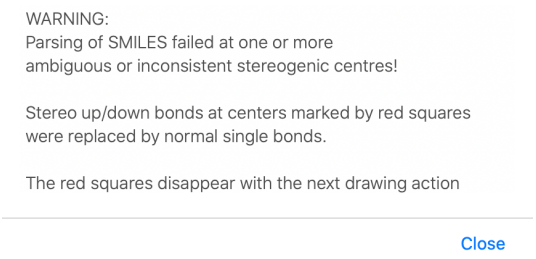
Example LOG file:

```
LOGFILE for run of /Users/bj/Desktop/MOSFECCS/GITHUB/MOSTEST_Testsuite for
MOSFECCS/MOSTEST_Example/mostest_240305.js with file: Example_SMILES_for_manual.txt
run by bj on node at 05.03.2024 16:48:14 GMT+0100 (Central European Standard Time)
=======================================
BJ_1_1 OK
BJ_1_3 OK
BJ_1_4 OK
BJ_1_5 Parser ERROR for SMILES: CCC&CCC
        ERR: invalidSymbol in SMILES:"&"
BJ_1_6 OK
BJ_1_8 OK
BJ_1_9 parsing failed for SMILES: COCCSCNC checkSMILES: CNCSCCOC
BJ_13_10 OK
```

An **ERROR** is recorded when the input SMILES cannot be parsed successfully (non-legal SMILES, illegal characters encountered, unpaired paranthesis etc.). No SVG graphics file is generated in this case.

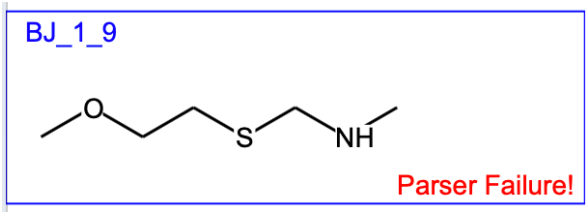
WarnAtoms are generated when the parser was unable to draw the structure with the correct stereo-configuration at certain atoms. No SVG graphics file is generated in this case.

If the same SMILES would be entered with "putSMILES" in the Editor MOSFECCS, the warning would be displayed as an alert:



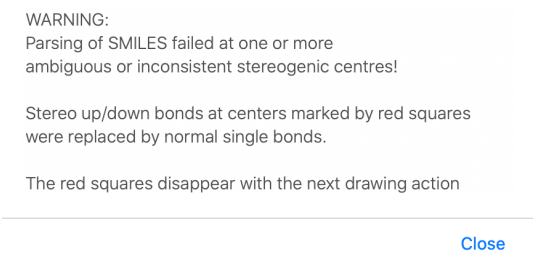
and the problematic atoms would be highlighted with red squares in the structural formula draw by the parser:

A **FAILURE** is recorded in the LOG whenever the SMILES recalculated from the structure generated by parsing the input-SMILES is not identical to the latter (i.e. if the input was a legal SMILES code but not the canonical one generated by MOSFECCS for this structure. An SVG graphic file is generated but annotated with "Parser Failure!" in this case.



WarnAtoms are generated when the parser was unable to draw the structure with the correct stereo-configuration at certain atoms. No SVG graphics file is generated in this case. *Note: No example of such a warning appears with the SMILES contained in the input file "Example_SMILES_for_manual.txt" shown above.*

If the same SMILES would be entered with "putSMILES" in the Editor MOSFECCS, the warning would be displayed as an alert:



and the problematic atoms would be highlighted with red squares in the structural formula drawn by the parser:

The statistics of the Test are summarized in the **SUMMARY_Example_SMILES_for_manual.txt** file:

Example SUMMARY file:

SUMMARY for run of /Users/bj/Desktop/MOSFECCS/GITHUB/MOSTEST_Testsuite for MOSFECCS/MOSTEST_Example/mostest_240305.js with file Example_SMILES_for_manual.txt
Comments:
example input file for mostest.js

6 of a total of 8 SMILES passed the test.

PARSER ERRORS:1
BJ_1_5: CCC&CCC Error:invalidSymbol in SMILES:"&"

PARSER FAILURES:1

BJ_1_9 SMILES entered:
COCCSCNC
CNCSCCOC
was returned by getsmiles() of parsed structure

B. Jaun March 2024