Package ‘hdcuremodels’

August 2, 2025

Title High-Dimensional Cure Models

Version 0.0.5

Date 2025-07-31

Description Provides functions for fitting various penalized parametric and semi-parametric mixture cure models with different penalty functions, testing for a significant cure fraction, and testing for sufficient follow-up as described in Fu et al (2022)<doi:10.1002/sim.9513> and Archer et al (2024)<doi:10.1186/s13045-024-01553-6>. False discovery rate controlled variable selection is provided using model-X knock-offs.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 4.2.0)

Imports doParallel, flexsurv, flexsurvcure, foreach, ggplot2, ggpubr, glmnet, knockoff, mvnfast, parallel, plyr, methods, survival, withr

RoxygenNote 7.3.2

Suggests knitr, Rdsdp, rmarkdown, roxygen2, testthat (>= 3.0.0)

VignetteBuilder knitr

LazyData true

URL `https://github.com/kelliejarcher/hdcuremodels`

BugReports `https://github.com/kelliejarcher/hdcuremodels/issues`

Config/testthat/edition 3

NeedsCompilation no

Author Han Fu [aut],
Kellie J. Archer [aut, cre] (ORCID:
<`https://orcid.org/0000-0003-1555-5781`>),
Tung Lam Nguyen [rev] (Reviewed the package for ROpenSci)

Maintainer Kellie J. Archer <`archer.43@osu.edu`>

Repository CRAN

Date/Publication 2025-07-31 21:02:06 UTC

Contents

`amltest` *AML test data*

Description

Duration of complete response for 40 cytogenetically normal AML patients and a subset of 320 transcript expression from RNA-sequencing.

Usage

```
amltest
```

Format

A data frame with 40 rows (subjects) and 322 columns:

cryr duration of complete response in years

relapse.death censoring indicator: 1 = relapsed or died; 0 = alive at last follow=up

ENSG00000001561 normalized expression for indicated transcript

ENSG00000005249 normalized expression for indicated transcript

ENSG00000006757	normalized expression for indicated transcript
ENSG00000007062	normalized expression for indicated transcript
ENSG00000007968	normalized expression for indicated transcript
ENSG00000008283	normalized expression for indicated transcript
ENSG00000008405	normalized expression for indicated transcript
ENSG00000008441	normalized expression for indicated transcript
ENSG00000010295	normalized expression for indicated transcript
ENSG00000011028	normalized expression for indicated transcript
ENSG00000011198	normalized expression for indicated transcript
ENSG00000012779	normalized expression for indicated transcript
ENSG00000012817	normalized expression for indicated transcript
ENSG00000013306	normalized expression for indicated transcript
ENSG00000013725	normalized expression for indicated transcript
ENSG00000018189	normalized expression for indicated transcript
ENSG00000022267	normalized expression for indicated transcript
ENSG00000023171	normalized expression for indicated transcript
ENSG00000023909	normalized expression for indicated transcript
ENSG00000029639	normalized expression for indicated transcript
ENSG00000047634	normalized expression for indicated transcript
ENSG00000049192	normalized expression for indicated transcript
ENSG00000053524	normalized expression for indicated transcript
ENSG00000058056	normalized expression for indicated transcript
ENSG00000060138	normalized expression for indicated transcript
ENSG00000061918	normalized expression for indicated transcript
ENSG00000065809	normalized expression for indicated transcript
ENSG00000065923	normalized expression for indicated transcript
ENSG00000068489	normalized expression for indicated transcript
ENSG00000069020	normalized expression for indicated transcript
ENSG00000070404	normalized expression for indicated transcript
ENSG00000071894	normalized expression for indicated transcript
ENSG00000072422	normalized expression for indicated transcript
ENSG00000073605	normalized expression for indicated transcript
ENSG00000076555	normalized expression for indicated transcript
ENSG00000080823	normalized expression for indicated transcript
ENSG00000089723	normalized expression for indicated transcript
ENSG00000090382	normalized expression for indicated transcript
ENSG00000090975	normalized expression for indicated transcript

ENSG00000100068	normalized expression for indicated transcript
ENSG00000100077	normalized expression for indicated transcript
ENSG00000100299	normalized expression for indicated transcript
ENSG00000100376	normalized expression for indicated transcript
ENSG00000100418	normalized expression for indicated transcript
ENSG00000100448	normalized expression for indicated transcript
ENSG00000100596	normalized expression for indicated transcript
ENSG00000100916	normalized expression for indicated transcript
ENSG00000102409	normalized expression for indicated transcript
ENSG00000102760	normalized expression for indicated transcript
ENSG00000104689	normalized expression for indicated transcript
ENSG00000104946	normalized expression for indicated transcript
ENSG00000105518	normalized expression for indicated transcript
ENSG00000105808	normalized expression for indicated transcript
ENSG00000106367	normalized expression for indicated transcript
ENSG00000106526	normalized expression for indicated transcript
ENSG00000106546	normalized expression for indicated transcript
ENSG00000106780	normalized expression for indicated transcript
ENSG00000107104	normalized expression for indicated transcript
ENSG00000107742	normalized expression for indicated transcript
ENSG00000107798	normalized expression for indicated transcript
ENSG00000107816	normalized expression for indicated transcript
ENSG00000107957	normalized expression for indicated transcript
ENSG00000109674	normalized expression for indicated transcript
ENSG00000110076	normalized expression for indicated transcript
ENSG00000110237	normalized expression for indicated transcript
ENSG00000110492	normalized expression for indicated transcript
ENSG00000110799	normalized expression for indicated transcript
ENSG00000111275	normalized expression for indicated transcript
ENSG00000112773	normalized expression for indicated transcript
ENSG00000113504	normalized expression for indicated transcript
ENSG00000114268	normalized expression for indicated transcript
ENSG00000114737	normalized expression for indicated transcript
ENSG00000115183	normalized expression for indicated transcript
ENSG00000115414	normalized expression for indicated transcript
ENSG00000115457	normalized expression for indicated transcript
ENSG00000115525	normalized expression for indicated transcript

ENSG00000116574	normalized expression for indicated transcript
ENSG00000117480	normalized expression for indicated transcript
ENSG00000119280	normalized expression for indicated transcript
ENSG00000120594	normalized expression for indicated transcript
ENSG00000120675	normalized expression for indicated transcript
ENSG00000120832	normalized expression for indicated transcript
ENSG00000120913	normalized expression for indicated transcript
ENSG00000121005	normalized expression for indicated transcript
ENSG00000121039	normalized expression for indicated transcript
ENSG00000121274	normalized expression for indicated transcript
ENSG00000123080	normalized expression for indicated transcript
ENSG00000123836	normalized expression for indicated transcript
ENSG00000124019	normalized expression for indicated transcript
ENSG00000124882	normalized expression for indicated transcript
ENSG00000126822	normalized expression for indicated transcript
ENSG00000127152	normalized expression for indicated transcript
ENSG00000129824	normalized expression for indicated transcript
ENSG00000130702	normalized expression for indicated transcript
ENSG00000131188	normalized expression for indicated transcript
ENSG00000131370	normalized expression for indicated transcript
ENSG00000132122	normalized expression for indicated transcript
ENSG00000132530	normalized expression for indicated transcript
ENSG00000132819	normalized expression for indicated transcript
ENSG00000132849	normalized expression for indicated transcript
ENSG00000133401	normalized expression for indicated transcript
ENSG00000133619	normalized expression for indicated transcript
ENSG00000134531	normalized expression for indicated transcript
ENSG00000134897	normalized expression for indicated transcript
ENSG00000135074	normalized expression for indicated transcript
ENSG00000135245	normalized expression for indicated transcript
ENSG00000135272	normalized expression for indicated transcript
ENSG00000135362	normalized expression for indicated transcript
ENSG00000135363	normalized expression for indicated transcript
ENSG00000135916	normalized expression for indicated transcript
ENSG00000136026	normalized expression for indicated transcript
ENSG00000136193	normalized expression for indicated transcript
ENSG00000136231	normalized expression for indicated transcript

ENSG00000136997	normalized expression for indicated transcript
ENSG00000137193	normalized expression for indicated transcript
ENSG00000137198	normalized expression for indicated transcript
ENSG00000138722	normalized expression for indicated transcript
ENSG00000139318	normalized expression for indicated transcript
ENSG00000140287	normalized expression for indicated transcript
ENSG00000144036	normalized expression for indicated transcript
ENSG00000144647	normalized expression for indicated transcript
ENSG00000144677	normalized expression for indicated transcript
ENSG00000145476	normalized expression for indicated transcript
ENSG00000145545	normalized expression for indicated transcript
ENSG00000146243	normalized expression for indicated transcript
ENSG00000146373	normalized expression for indicated transcript
ENSG00000147044	normalized expression for indicated transcript
ENSG00000147180	normalized expression for indicated transcript
ENSG00000148444	normalized expression for indicated transcript
ENSG00000148484	normalized expression for indicated transcript
ENSG00000149131	normalized expression for indicated transcript
ENSG00000150760	normalized expression for indicated transcript
ENSG00000150782	normalized expression for indicated transcript
ENSG00000151135	normalized expression for indicated transcript
ENSG00000151208	normalized expression for indicated transcript
ENSG00000151458	normalized expression for indicated transcript
ENSG00000152409	normalized expression for indicated transcript
ENSG00000152580	normalized expression for indicated transcript
ENSG00000152767	normalized expression for indicated transcript
ENSG00000152778	normalized expression for indicated transcript
ENSG00000153563	normalized expression for indicated transcript
ENSG00000154217	normalized expression for indicated transcript
ENSG00000154743	normalized expression for indicated transcript
ENSG00000154760	normalized expression for indicated transcript
ENSG00000154874	normalized expression for indicated transcript
ENSG00000156381	normalized expression for indicated transcript
ENSG00000157107	normalized expression for indicated transcript
ENSG00000157240	normalized expression for indicated transcript
ENSG00000157873	normalized expression for indicated transcript
ENSG00000157978	normalized expression for indicated transcript

ENSG00000158691	normalized expression for indicated transcript
ENSG00000159339	normalized expression for indicated transcript
ENSG00000159403	normalized expression for indicated transcript
ENSG00000159788	normalized expression for indicated transcript
ENSG00000160685	normalized expression for indicated transcript
ENSG00000160781	normalized expression for indicated transcript
ENSG00000161509	normalized expression for indicated transcript
ENSG00000162433	normalized expression for indicated transcript
ENSG00000162614	normalized expression for indicated transcript
ENSG00000162676	normalized expression for indicated transcript
ENSG00000163412	normalized expression for indicated transcript
ENSG00000163702	normalized expression for indicated transcript
ENSG00000163814	normalized expression for indicated transcript
ENSG00000164086	normalized expression for indicated transcript
ENSG00000164172	normalized expression for indicated transcript
ENSG00000164442	normalized expression for indicated transcript
ENSG00000165272	normalized expression for indicated transcript
ENSG00000166165	normalized expression for indicated transcript
ENSG00000166435	normalized expression for indicated transcript
ENSG00000166987	normalized expression for indicated transcript
ENSG00000167291	normalized expression for indicated transcript
ENSG00000167565	normalized expression for indicated transcript
ENSG00000167851	normalized expression for indicated transcript
ENSG00000168026	normalized expression for indicated transcript
ENSG00000168209	normalized expression for indicated transcript
ENSG00000168502	normalized expression for indicated transcript
ENSG00000168939	normalized expression for indicated transcript
ENSG00000169203	normalized expression for indicated transcript
ENSG00000169247	normalized expression for indicated transcript
ENSG00000169504	normalized expression for indicated transcript
ENSG00000169860	normalized expression for indicated transcript
ENSG00000169991	normalized expression for indicated transcript
ENSG00000170035	normalized expression for indicated transcript
ENSG00000170180	normalized expression for indicated transcript
ENSG00000170456	normalized expression for indicated transcript
ENSG00000170522	normalized expression for indicated transcript
ENSG00000170909	normalized expression for indicated transcript

ENSG00000171121	normalized expression for indicated transcript
ENSG00000171222	normalized expression for indicated transcript
ENSG00000171476	normalized expression for indicated transcript
ENSG00000171813	normalized expression for indicated transcript
ENSG00000171962	normalized expression for indicated transcript
ENSG00000172197	normalized expression for indicated transcript
ENSG00000172236	normalized expression for indicated transcript
ENSG00000173083	normalized expression for indicated transcript
ENSG00000173530	normalized expression for indicated transcript
ENSG00000173926	normalized expression for indicated transcript
ENSG00000174059	normalized expression for indicated transcript
ENSG00000174080	normalized expression for indicated transcript
ENSG00000174130	normalized expression for indicated transcript
ENSG00000174738	normalized expression for indicated transcript
ENSG00000175265	normalized expression for indicated transcript
ENSG00000175352	normalized expression for indicated transcript
ENSG00000176597	normalized expression for indicated transcript
ENSG00000179222	normalized expression for indicated transcript
ENSG00000179630	normalized expression for indicated transcript
ENSG00000179639	normalized expression for indicated transcript
ENSG00000179820	normalized expression for indicated transcript
ENSG00000180096	normalized expression for indicated transcript
ENSG00000180596	normalized expression for indicated transcript
ENSG00000180902	normalized expression for indicated transcript
ENSG00000181104	normalized expression for indicated transcript
ENSG00000182866	normalized expression for indicated transcript
ENSG00000182871	normalized expression for indicated transcript
ENSG00000183087	normalized expression for indicated transcript
ENSG00000183091	normalized expression for indicated transcript
ENSG00000184371	normalized expression for indicated transcript
ENSG00000185129	normalized expression for indicated transcript
ENSG00000185201	normalized expression for indicated transcript
ENSG00000185245	normalized expression for indicated transcript
ENSG00000185291	normalized expression for indicated transcript
ENSG00000185304	normalized expression for indicated transcript
ENSG00000185710	normalized expression for indicated transcript
ENSG00000185883	normalized expression for indicated transcript

ENSG00000185986	normalized expression for indicated transcript
ENSG00000186130	normalized expression for indicated transcript
ENSG00000186854	normalized expression for indicated transcript
ENSG00000187010	normalized expression for indicated transcript
ENSG00000187627	normalized expression for indicated transcript
ENSG00000187653	normalized expression for indicated transcript
ENSG00000187837	normalized expression for indicated transcript
ENSG00000188002	normalized expression for indicated transcript
ENSG00000188107	normalized expression for indicated transcript
ENSG00000188211	normalized expression for indicated transcript
ENSG00000188636	normalized expression for indicated transcript
ENSG00000188738	normalized expression for indicated transcript
ENSG00000188856	normalized expression for indicated transcript
ENSG00000189164	normalized expression for indicated transcript
ENSG00000189223	normalized expression for indicated transcript
ENSG00000196155	normalized expression for indicated transcript
ENSG00000196189	normalized expression for indicated transcript
ENSG00000196415	normalized expression for indicated transcript
ENSG00000196565	normalized expression for indicated transcript
ENSG00000197081	normalized expression for indicated transcript
ENSG00000197121	normalized expression for indicated transcript
ENSG00000197253	normalized expression for indicated transcript
ENSG00000197256	normalized expression for indicated transcript
ENSG00000197321	normalized expression for indicated transcript
ENSG00000197561	normalized expression for indicated transcript
ENSG00000197728	normalized expression for indicated transcript
ENSG00000197860	normalized expression for indicated transcript
ENSG00000197937	normalized expression for indicated transcript
ENSG00000197951	normalized expression for indicated transcript
ENSG00000198743	normalized expression for indicated transcript
ENSG00000198838	normalized expression for indicated transcript
ENSG00000199347	normalized expression for indicated transcript
ENSG00000200243	normalized expression for indicated transcript
ENSG00000201801	normalized expression for indicated transcript
ENSG00000203872	normalized expression for indicated transcript
ENSG00000204172	normalized expression for indicated transcript
ENSG00000205571	normalized expression for indicated transcript

ENSG00000205593	normalized expression for indicated transcript
ENSG00000208772	normalized expression for indicated transcript
ENSG00000213085	normalized expression for indicated transcript
ENSG00000213261	normalized expression for indicated transcript
ENSG00000213626	normalized expression for indicated transcript
ENSG00000213722	normalized expression for indicated transcript
ENSG00000213906	normalized expression for indicated transcript
ENSG00000213967	normalized expression for indicated transcript
ENSG00000214016	normalized expression for indicated transcript
ENSG00000214425	normalized expression for indicated transcript
ENSG00000216316	normalized expression for indicated transcript
ENSG00000220008	normalized expression for indicated transcript
ENSG00000223345	normalized expression for indicated transcript
ENSG00000224080	normalized expression for indicated transcript
ENSG00000225138	normalized expression for indicated transcript
ENSG00000226471	normalized expression for indicated transcript
ENSG00000227097	normalized expression for indicated transcript
ENSG00000227191	normalized expression for indicated transcript
ENSG00000227615	normalized expression for indicated transcript
ENSG00000228049	normalized expression for indicated transcript
ENSG00000229153	normalized expression for indicated transcript
ENSG00000230076	normalized expression for indicated transcript
ENSG00000231160	normalized expression for indicated transcript
ENSG00000231721	normalized expression for indicated transcript
ENSG00000233927	normalized expression for indicated transcript
ENSG00000233974	normalized expression for indicated transcript
ENSG00000234883	normalized expression for indicated transcript
ENSG00000236876	normalized expression for indicated transcript
ENSG00000237298	normalized expression for indicated transcript
ENSG00000237892	normalized expression for indicated transcript
ENSG00000238160	normalized expression for indicated transcript
ENSG00000239437	normalized expression for indicated transcript
ENSG00000241399	normalized expression for indicated transcript
ENSG00000241489	normalized expression for indicated transcript
ENSG00000241529	normalized expression for indicated transcript
ENSG00000244405	normalized expression for indicated transcript
ENSG00000247627	normalized expression for indicated transcript

ENSG00000249592	normalized expression for indicated transcript
ENSG00000250116	normalized expression for indicated transcript
ENSG00000250251	normalized expression for indicated transcript
ENSG00000251079	normalized expression for indicated transcript
ENSG00000253210	normalized expression for indicated transcript
ENSG00000253276	normalized expression for indicated transcript
ENSG00000254415	normalized expression for indicated transcript
ENSG00000259276	normalized expression for indicated transcript
ENSG00000260727	normalized expression for indicated transcript
ENSG00000261377	normalized expression for indicated transcript
ENSG00000264885	normalized expression for indicated transcript
ENSG00000264895	normalized expression for indicated transcript
ENSG00000267136	normalized expression for indicated transcript
ENSG00000267551	normalized expression for indicated transcript
ENSG00000267702	normalized expression for indicated transcript
ENSG00000268001	normalized expression for indicated transcript
ENSG00000268573	normalized expression for indicated transcript
ENSG00000270554	normalized expression for indicated transcript
ENSG00000270562	normalized expression for indicated transcript
ENSG00000271646	normalized expression for indicated transcript
ENSG00000273018	normalized expression for indicated transcript
ENSG00000273033	normalized expression for indicated transcript

Source

`doi:10.1186/s13045-024-01553-6`

`amltrain` *AML training data*

Description

Duration of complete response for 306 cytogenetically normal AML patients and a subset of 320 transcript expression from RNA-sequencing.

Usage

```
amltrain
```

Format

A data frame with 306 rows (subjects) and 322 columns:

cryr duration of complete response in years

relapse.death censoring indicator: 1 = relapsed or died; 0 = alive at last follow=up

ENSG00000001561 normalized expression for indicated transcript

ENSG00000005249 normalized expression for indicated transcript

ENSG00000006757 normalized expression for indicated transcript

ENSG00000007062 normalized expression for indicated transcript

ENSG00000007968 normalized expression for indicated transcript

ENSG00000008283 normalized expression for indicated transcript

ENSG00000008405 normalized expression for indicated transcript

ENSG00000008441 normalized expression for indicated transcript

ENSG00000010295 normalized expression for indicated transcript

ENSG00000011028 normalized expression for indicated transcript

ENSG00000011198 normalized expression for indicated transcript

ENSG00000012779 normalized expression for indicated transcript

ENSG00000012817 normalized expression for indicated transcript

ENSG00000013306 normalized expression for indicated transcript

ENSG00000013725 normalized expression for indicated transcript

ENSG00000018189 normalized expression for indicated transcript

ENSG00000022267 normalized expression for indicated transcript

ENSG00000023171 normalized expression for indicated transcript

ENSG00000023909 normalized expression for indicated transcript

ENSG00000029639 normalized expression for indicated transcript

ENSG00000047634 normalized expression for indicated transcript

ENSG00000049192 normalized expression for indicated transcript

ENSG00000053524 normalized expression for indicated transcript

ENSG00000058056 normalized expression for indicated transcript

ENSG00000060138 normalized expression for indicated transcript

ENSG00000061918 normalized expression for indicated transcript

ENSG00000065809 normalized expression for indicated transcript

ENSG00000065923 normalized expression for indicated transcript

ENSG00000068489 normalized expression for indicated transcript

ENSG00000069020 normalized expression for indicated transcript

ENSG00000070404 normalized expression for indicated transcript

ENSG00000071894 normalized expression for indicated transcript

ENSG00000072422 normalized expression for indicated transcript

ENSG00000073605	normalized expression for indicated transcript
ENSG00000076555	normalized expression for indicated transcript
ENSG00000080823	normalized expression for indicated transcript
ENSG00000089723	normalized expression for indicated transcript
ENSG00000090382	normalized expression for indicated transcript
ENSG00000090975	normalized expression for indicated transcript
ENSG00000100068	normalized expression for indicated transcript
ENSG00000100077	normalized expression for indicated transcript
ENSG00000100299	normalized expression for indicated transcript
ENSG00000100376	normalized expression for indicated transcript
ENSG00000100418	normalized expression for indicated transcript
ENSG00000100448	normalized expression for indicated transcript
ENSG00000100596	normalized expression for indicated transcript
ENSG00000100916	normalized expression for indicated transcript
ENSG00000102409	normalized expression for indicated transcript
ENSG00000102760	normalized expression for indicated transcript
ENSG00000104689	normalized expression for indicated transcript
ENSG00000104946	normalized expression for indicated transcript
ENSG00000105518	normalized expression for indicated transcript
ENSG00000105808	normalized expression for indicated transcript
ENSG00000106367	normalized expression for indicated transcript
ENSG00000106526	normalized expression for indicated transcript
ENSG00000106546	normalized expression for indicated transcript
ENSG00000106780	normalized expression for indicated transcript
ENSG00000107104	normalized expression for indicated transcript
ENSG00000107742	normalized expression for indicated transcript
ENSG00000107798	normalized expression for indicated transcript
ENSG00000107816	normalized expression for indicated transcript
ENSG00000107957	normalized expression for indicated transcript
ENSG00000109674	normalized expression for indicated transcript
ENSG00000110076	normalized expression for indicated transcript
ENSG00000110237	normalized expression for indicated transcript
ENSG00000110492	normalized expression for indicated transcript
ENSG00000110799	normalized expression for indicated transcript
ENSG00000111275	normalized expression for indicated transcript
ENSG00000112773	normalized expression for indicated transcript
ENSG00000113504	normalized expression for indicated transcript

ENSG00000114268	normalized expression for indicated transcript
ENSG00000114737	normalized expression for indicated transcript
ENSG00000115183	normalized expression for indicated transcript
ENSG00000115414	normalized expression for indicated transcript
ENSG00000115457	normalized expression for indicated transcript
ENSG00000115525	normalized expression for indicated transcript
ENSG00000116574	normalized expression for indicated transcript
ENSG00000117480	normalized expression for indicated transcript
ENSG00000119280	normalized expression for indicated transcript
ENSG00000120594	normalized expression for indicated transcript
ENSG00000120675	normalized expression for indicated transcript
ENSG00000120832	normalized expression for indicated transcript
ENSG00000120913	normalized expression for indicated transcript
ENSG00000121005	normalized expression for indicated transcript
ENSG00000121039	normalized expression for indicated transcript
ENSG00000121274	normalized expression for indicated transcript
ENSG00000123080	normalized expression for indicated transcript
ENSG00000123836	normalized expression for indicated transcript
ENSG00000124019	normalized expression for indicated transcript
ENSG00000124882	normalized expression for indicated transcript
ENSG00000126822	normalized expression for indicated transcript
ENSG00000127152	normalized expression for indicated transcript
ENSG00000129824	normalized expression for indicated transcript
ENSG00000130702	normalized expression for indicated transcript
ENSG00000131188	normalized expression for indicated transcript
ENSG00000131370	normalized expression for indicated transcript
ENSG00000132122	normalized expression for indicated transcript
ENSG00000132530	normalized expression for indicated transcript
ENSG00000132819	normalized expression for indicated transcript
ENSG00000132849	normalized expression for indicated transcript
ENSG00000133401	normalized expression for indicated transcript
ENSG00000133619	normalized expression for indicated transcript
ENSG00000134531	normalized expression for indicated transcript
ENSG00000134897	normalized expression for indicated transcript
ENSG00000135074	normalized expression for indicated transcript
ENSG00000135245	normalized expression for indicated transcript
ENSG00000135272	normalized expression for indicated transcript

ENSG00000135362	normalized expression for indicated transcript
ENSG00000135363	normalized expression for indicated transcript
ENSG00000135916	normalized expression for indicated transcript
ENSG00000136026	normalized expression for indicated transcript
ENSG00000136193	normalized expression for indicated transcript
ENSG00000136231	normalized expression for indicated transcript
ENSG00000136997	normalized expression for indicated transcript
ENSG00000137193	normalized expression for indicated transcript
ENSG00000137198	normalized expression for indicated transcript
ENSG00000138722	normalized expression for indicated transcript
ENSG00000139318	normalized expression for indicated transcript
ENSG00000140287	normalized expression for indicated transcript
ENSG00000144036	normalized expression for indicated transcript
ENSG00000144647	normalized expression for indicated transcript
ENSG00000144677	normalized expression for indicated transcript
ENSG00000145476	normalized expression for indicated transcript
ENSG00000145545	normalized expression for indicated transcript
ENSG00000146243	normalized expression for indicated transcript
ENSG00000146373	normalized expression for indicated transcript
ENSG00000147044	normalized expression for indicated transcript
ENSG00000147180	normalized expression for indicated transcript
ENSG00000148444	normalized expression for indicated transcript
ENSG00000148484	normalized expression for indicated transcript
ENSG00000149131	normalized expression for indicated transcript
ENSG00000150760	normalized expression for indicated transcript
ENSG00000150782	normalized expression for indicated transcript
ENSG00000151135	normalized expression for indicated transcript
ENSG00000151208	normalized expression for indicated transcript
ENSG00000151458	normalized expression for indicated transcript
ENSG00000152409	normalized expression for indicated transcript
ENSG00000152580	normalized expression for indicated transcript
ENSG00000152767	normalized expression for indicated transcript
ENSG00000152778	normalized expression for indicated transcript
ENSG00000153563	normalized expression for indicated transcript
ENSG00000154217	normalized expression for indicated transcript
ENSG00000154743	normalized expression for indicated transcript
ENSG00000154760	normalized expression for indicated transcript

ENSG00000154874	normalized expression for indicated transcript
ENSG00000156381	normalized expression for indicated transcript
ENSG00000157107	normalized expression for indicated transcript
ENSG00000157240	normalized expression for indicated transcript
ENSG00000157873	normalized expression for indicated transcript
ENSG00000157978	normalized expression for indicated transcript
ENSG00000158691	normalized expression for indicated transcript
ENSG00000159339	normalized expression for indicated transcript
ENSG00000159403	normalized expression for indicated transcript
ENSG00000159788	normalized expression for indicated transcript
ENSG00000160685	normalized expression for indicated transcript
ENSG00000160781	normalized expression for indicated transcript
ENSG00000161509	normalized expression for indicated transcript
ENSG00000162433	normalized expression for indicated transcript
ENSG00000162614	normalized expression for indicated transcript
ENSG00000162676	normalized expression for indicated transcript
ENSG00000163412	normalized expression for indicated transcript
ENSG00000163702	normalized expression for indicated transcript
ENSG00000163814	normalized expression for indicated transcript
ENSG00000164086	normalized expression for indicated transcript
ENSG00000164172	normalized expression for indicated transcript
ENSG00000164442	normalized expression for indicated transcript
ENSG00000165272	normalized expression for indicated transcript
ENSG00000166165	normalized expression for indicated transcript
ENSG00000166435	normalized expression for indicated transcript
ENSG00000166987	normalized expression for indicated transcript
ENSG00000167291	normalized expression for indicated transcript
ENSG00000167565	normalized expression for indicated transcript
ENSG00000167851	normalized expression for indicated transcript
ENSG00000168026	normalized expression for indicated transcript
ENSG00000168209	normalized expression for indicated transcript
ENSG00000168502	normalized expression for indicated transcript
ENSG00000168939	normalized expression for indicated transcript
ENSG00000169203	normalized expression for indicated transcript
ENSG00000169247	normalized expression for indicated transcript
ENSG00000169504	normalized expression for indicated transcript
ENSG00000169860	normalized expression for indicated transcript

ENSG00000169991	normalized expression for indicated transcript
ENSG00000170035	normalized expression for indicated transcript
ENSG00000170180	normalized expression for indicated transcript
ENSG00000170456	normalized expression for indicated transcript
ENSG00000170522	normalized expression for indicated transcript
ENSG00000170909	normalized expression for indicated transcript
ENSG00000171121	normalized expression for indicated transcript
ENSG00000171222	normalized expression for indicated transcript
ENSG00000171476	normalized expression for indicated transcript
ENSG00000171813	normalized expression for indicated transcript
ENSG00000171962	normalized expression for indicated transcript
ENSG00000172197	normalized expression for indicated transcript
ENSG00000172236	normalized expression for indicated transcript
ENSG00000173083	normalized expression for indicated transcript
ENSG00000173530	normalized expression for indicated transcript
ENSG00000173926	normalized expression for indicated transcript
ENSG00000174059	normalized expression for indicated transcript
ENSG00000174080	normalized expression for indicated transcript
ENSG00000174130	normalized expression for indicated transcript
ENSG00000174738	normalized expression for indicated transcript
ENSG00000175265	normalized expression for indicated transcript
ENSG00000175352	normalized expression for indicated transcript
ENSG00000176597	normalized expression for indicated transcript
ENSG00000179222	normalized expression for indicated transcript
ENSG00000179630	normalized expression for indicated transcript
ENSG00000179639	normalized expression for indicated transcript
ENSG00000179820	normalized expression for indicated transcript
ENSG00000180096	normalized expression for indicated transcript
ENSG00000180596	normalized expression for indicated transcript
ENSG00000180902	normalized expression for indicated transcript
ENSG00000181104	normalized expression for indicated transcript
ENSG00000182866	normalized expression for indicated transcript
ENSG00000182871	normalized expression for indicated transcript
ENSG00000183087	normalized expression for indicated transcript
ENSG00000183091	normalized expression for indicated transcript
ENSG00000184371	normalized expression for indicated transcript
ENSG00000185129	normalized expression for indicated transcript

ENSG00000185201	normalized expression for indicated transcript
ENSG00000185245	normalized expression for indicated transcript
ENSG00000185291	normalized expression for indicated transcript
ENSG00000185304	normalized expression for indicated transcript
ENSG00000185710	normalized expression for indicated transcript
ENSG00000185883	normalized expression for indicated transcript
ENSG00000185986	normalized expression for indicated transcript
ENSG00000186130	normalized expression for indicated transcript
ENSG00000186854	normalized expression for indicated transcript
ENSG00000187010	normalized expression for indicated transcript
ENSG00000187627	normalized expression for indicated transcript
ENSG00000187653	normalized expression for indicated transcript
ENSG00000187837	normalized expression for indicated transcript
ENSG00000188002	normalized expression for indicated transcript
ENSG00000188107	normalized expression for indicated transcript
ENSG00000188211	normalized expression for indicated transcript
ENSG00000188636	normalized expression for indicated transcript
ENSG00000188738	normalized expression for indicated transcript
ENSG00000188856	normalized expression for indicated transcript
ENSG00000189164	normalized expression for indicated transcript
ENSG00000189223	normalized expression for indicated transcript
ENSG00000196155	normalized expression for indicated transcript
ENSG00000196189	normalized expression for indicated transcript
ENSG00000196415	normalized expression for indicated transcript
ENSG00000196565	normalized expression for indicated transcript
ENSG00000197081	normalized expression for indicated transcript
ENSG00000197121	normalized expression for indicated transcript
ENSG00000197253	normalized expression for indicated transcript
ENSG00000197256	normalized expression for indicated transcript
ENSG00000197321	normalized expression for indicated transcript
ENSG00000197561	normalized expression for indicated transcript
ENSG00000197728	normalized expression for indicated transcript
ENSG00000197860	normalized expression for indicated transcript
ENSG00000197937	normalized expression for indicated transcript
ENSG00000197951	normalized expression for indicated transcript
ENSG00000198743	normalized expression for indicated transcript
ENSG00000198838	normalized expression for indicated transcript

ENSG00000199347	normalized expression for indicated transcript
ENSG00000200243	normalized expression for indicated transcript
ENSG00000201801	normalized expression for indicated transcript
ENSG00000203872	normalized expression for indicated transcript
ENSG00000204172	normalized expression for indicated transcript
ENSG00000205571	normalized expression for indicated transcript
ENSG00000205593	normalized expression for indicated transcript
ENSG00000208772	normalized expression for indicated transcript
ENSG00000213085	normalized expression for indicated transcript
ENSG00000213261	normalized expression for indicated transcript
ENSG00000213626	normalized expression for indicated transcript
ENSG00000213722	normalized expression for indicated transcript
ENSG00000213906	normalized expression for indicated transcript
ENSG00000213967	normalized expression for indicated transcript
ENSG00000214016	normalized expression for indicated transcript
ENSG00000214425	normalized expression for indicated transcript
ENSG00000216316	normalized expression for indicated transcript
ENSG00000220008	normalized expression for indicated transcript
ENSG00000223345	normalized expression for indicated transcript
ENSG00000224080	normalized expression for indicated transcript
ENSG00000225138	normalized expression for indicated transcript
ENSG00000226471	normalized expression for indicated transcript
ENSG00000227097	normalized expression for indicated transcript
ENSG00000227191	normalized expression for indicated transcript
ENSG00000227615	normalized expression for indicated transcript
ENSG00000228049	normalized expression for indicated transcript
ENSG00000229153	normalized expression for indicated transcript
ENSG00000230076	normalized expression for indicated transcript
ENSG00000231160	normalized expression for indicated transcript
ENSG00000231721	normalized expression for indicated transcript
ENSG00000233927	normalized expression for indicated transcript
ENSG00000233974	normalized expression for indicated transcript
ENSG00000234883	normalized expression for indicated transcript
ENSG00000236876	normalized expression for indicated transcript
ENSG00000237298	normalized expression for indicated transcript
ENSG00000237892	normalized expression for indicated transcript
ENSG00000238160	normalized expression for indicated transcript

ENSG00000239437	normalized expression for indicated transcript
ENSG00000241399	normalized expression for indicated transcript
ENSG00000241489	normalized expression for indicated transcript
ENSG00000241529	normalized expression for indicated transcript
ENSG00000244405	normalized expression for indicated transcript
ENSG00000247627	normalized expression for indicated transcript
ENSG00000249592	normalized expression for indicated transcript
ENSG00000250116	normalized expression for indicated transcript
ENSG00000250251	normalized expression for indicated transcript
ENSG00000251079	normalized expression for indicated transcript
ENSG00000253210	normalized expression for indicated transcript
ENSG00000253276	normalized expression for indicated transcript
ENSG00000254415	normalized expression for indicated transcript
ENSG00000259276	normalized expression for indicated transcript
ENSG00000260727	normalized expression for indicated transcript
ENSG00000261377	normalized expression for indicated transcript
ENSG00000264885	normalized expression for indicated transcript
ENSG00000264895	normalized expression for indicated transcript
ENSG00000267136	normalized expression for indicated transcript
ENSG00000267551	normalized expression for indicated transcript
ENSG00000267702	normalized expression for indicated transcript
ENSG00000268001	normalized expression for indicated transcript
ENSG00000268573	normalized expression for indicated transcript
ENSG00000270554	normalized expression for indicated transcript
ENSG00000270562	normalized expression for indicated transcript
ENSG00000271646	normalized expression for indicated transcript
ENSG00000273018	normalized expression for indicated transcript
ENSG00000273033	normalized expression for indicated transcript

Source

`doi:10.1186/s13045-024-01553-6`

`auc_mcm` *AUC for cure prediction using mean score imputation*

Description

This function calculates the AUC for cure prediction using the mean score imputation (MSI) method proposed by Asano et al (2014).

Usage

```
auc_mcm(object, newdata, cure_cutoff = 5, model_select = "AIC")
```

Arguments

`object`	a `mixturecure` object resulting from `curegmifs`, `cureem`, `cv_curegmifs`, or `cv_cureem`.
`newdata`	an optional data.frame that minimally includes the incidence and/or latency variables to use for predicting the response. If omitted, the training data are used.
`cure_cutoff`	cutoff value for cure, used to produce a proxy for the unobserved cure status (default is 5 representing 5 years). Users should be careful to note the time scale of their data and adjust this according to the time scale and clinical application.
`model_select`	either a case-sensitive parameter for models fit using `curegmifs` or `cureem` or any numeric step along the solution path can be selected. The default is `model_select = "AIC"` which calculates the predicted values using the coefficients from the model achieving the minimum AIC. The complete list of options are: • `"AIC"` for the minimum AIC (default). • `"mAIC"` for the minimum modified AIC. • `"cAIC"` for the minimum corrected AIC. • `"BIC"`, for the minimum BIC. • `"mBIC"` for the minimum modified BIC. • `"EBIC"` for the minimum extended BIC. • `"logLik"` for the step that maximizes the log-likelihood. • `n` where `n` is any numeric value from the solution path. This option has no effect for objects fit using `cv_curegmifs` or `cv_cureem`.

Value

Returns the AUC value for cure prediction using the mean score imputation (MSI) method.

References

Asano, J., Hirakawa, H., Hamada, C. (2014) Assessing the prediction accuracy of cure in the Cox proportional hazards cure model: an application to breast cancer data. *Pharmaceutical Statistics*, **13**:357–363.

See Also

concordance_mcm

Examples

```
library(survival)
withr::local_seed(1234)
temp <- generate_cure_data(n = 100, j = 10, n_true = 10, a = 1.8)
training <- temp$training
testing <- temp$testing
fit <- curegmifs(Surv(Time, Censor) ~ .,
  data = training, x_latency = training,
  model = "weibull", thresh = 1e-4, maxit = 2000,
  epsilon = 0.01, verbose = FALSE
)
auc_mcm(fit, model_select = "cAIC")
auc_mcm(fit, newdata = testing)
```

`coef.mixturecure` *Extract model coefficients from a fitted mixturecure object*

Description

`coef.mixturecure` is a generic function which extracts the model coefficients from a fitted `mixturecure` model object fit using `curegmifs`, `cureem`, `cv_curegmifs`, or `cv_cureem`.

Usage

```
## S3 method for class 'mixturecure'
coef(object, model_select = "AIC", ...)
```

Arguments

`object`	a `mixturecure` object resulting from `curegmifs`, `cureem`, `cv_curegmifs`, or `cv_cureem`.
`model_select`	either a case-sensitive parameter for models fit using `curegmifs` or `cureem` or any numeric step along the solution path can be selected. The default is `model_select = "AIC"` which calculates the predicted values using the coefficients from the model achieving the minimum AIC. The complete list of options are: • `"AIC"` for the minimum AIC (default). • `"mAIC"` for the minimum modified AIC. • `"cAIC"` for the minimum corrected AIC. • `"BIC"`, for the minimum BIC. • `"mBIC"` for the minimum modified BIC. • `"EBIC"` for the minimum extended BIC.

	- `"logLik"` for the step that maximizes the log-likelihood. - `n` where `n` is any numeric value from the solution path. This option has no effect for objects fit using `cv_curegmifs` or `cv_cureem`.
`...`	other arguments.

Value

`rate`	estimated rate parameter when fitting a Weibull or exponential mixture cure model.
`shape`	estimated shape parameter when fitting a Weibull mixture cure model.
`b0`	estimated intercept for the incidence portion of the mixture cure model.
`beta_inc`	the vector of coefficient estimates for the incidence portion of the mixture cure model.
`beta_lat`	the vector of coefficient estimates for the latency portion of the mixture cure model.
`p_uncured`	a vector of probabilities from the incidence portion of the fitted model representing the P(uncured).

See Also

`curegmifs`, `cureem`, `summary.mixturecure`, `plot.mixturecure`, `predict.mixturecure`

Examples

```
library(survival)
withr::local_seed(1234)
temp <- generate_cure_data(n = 100, j = 10, n_true = 10, a = 1.8)
training <- temp$training
fit <- curegmifs(Surv(Time, Censor) ~ .,
  data = training, x_latency = training,
  model = "weibull", thresh = 1e-4, maxit = 2000,
  epsilon = 0.01, verbose = FALSE
)
coef(fit)
```

`concordance_mcm` *C-statistic for mixture cure models*

Description

This function calculates the C-statistic using the cure status weighting (CSW) method proposed by Asano and Hirakawa (2017).

Usage

```
concordance_mcm(object, newdata, cure_cutoff = 5, model_select = "AIC")
```

Arguments

`object`	a `mixturecure` object resulting from `curegmifs`, `cureem`, `cv_curegmifs`, or `cv_cureem`.
`newdata`	an optional data.frame that minimally includes the incidence and/or latency variables to use for predicting the response. If omitted, the training data are used.
`cure_cutoff`	cutoff value for cure, used to produce a proxy for the unobserved cure status (default is 5 representing 5 years). Users should be careful to note the time scale of their data and adjust this according to the time scale and clinical application.
`model_select`	either a case-sensitive parameter for models fit using `curegmifs` or `cureem` or any numeric step along the solution path can be selected. The default is `model_select = "AIC"` which calculates the predicted values using the coefficients from the model achieving the minimum AIC. The complete list of options are: • `"AIC"` for the minimum AIC (default). • `"mAIC"` for the minimum modified AIC. • `"cAIC"` for the minimum corrected AIC. • `"BIC"`, for the minimum BIC. • `"mBIC"` for the minimum modified BIC. • `"EBIC"` for the minimum extended BIC. • `"logLik"` for the step that maximizes the log-likelihood. • `model_select = n` where `n` is any numeric value from the solution path. This option has no effect for objects fit using `cv_curegmifs` or `cv_cureem`.

Value

value of C-statistic for the cure models.

References

Asano, J. and Hirakawa, H. (2017) Assessing the prediction accuracy of a cure model for censored survival data with long-term survivors: Application to breast cancer data. *Journal of Biopharmaceutical Statistics*, **27**:6, 918–932.

See Also

`auc_mcm`

Examples

```
library(survival)
withr::local_seed(1234)
temp <- generate_cure_data(n = 100, j = 10, n_true = 10, a = 1.8)
training <- temp$training
testing <- temp$testing
fit <- curegmifs(Surv(Time, Censor) ~ .,
  data = training, x_latency = training,
  model = "weibull", thresh = 1e-4, maxit = 2000,
```

```
  epsilon = 0.01, verbose = FALSE
)
concordance_mcm(fit, model_select = "cAIC")
concordance_mcm(fit, newdata = testing, model_select = "cAIC")
```

`cureem` *Fit penalized mixture cure model using the E-M algorithm*

Description

Fits penalized parametric and semi-parametric mixture cure models (MCM) using the E-M algorithm with user-specified penalty parameters. The lasso (L1), MCP, and SCAD penalty are supported for the Cox MCM while only lasso is currently supported for parametric MCMs.

Usage

```
cureem(
  formula,
  data,
  subset,
  x_latency = NULL,
  model = c("cox", "weibull", "exponential"),
  penalty = c("lasso", "MCP", "SCAD"),
  penalty_factor_inc = NULL,
  penalty_factor_lat = NULL,
  thresh = 0.001,
  scale = TRUE,
  maxit = NULL,
  inits = NULL,
  lambda_inc = 0.1,
  lambda_lat = 0.1,
  gamma_inc = 3,
  gamma_lat = 3,
  na.action = na.omit,
  ...
)
```

Arguments

`formula`	an object of class "`formula`" (or one that can be coerced to that class): a symbolic description of the model to be fitted. The response must be a survival object as returned by the `Surv` function while the variables on the right side of the formula are the covariates that are included in the incidence portion of the model.
`data`	a data.frame in which to interpret the variables named in the `formula` or in the `subset` argument. Rows with missing data are omitted (only `na.action` = `na.omit` is operational) therefore users may want to impute missing data prior to calling this function.

`subset`	an optional expression indicating which subset of observations to be used in the fitting process, either a numeric or factor variable should be used in subset, not a character variable. All observations are included by default.
`x_latency`	specifies the variables to be included in the latency portion of the model and can be either a matrix of predictors, a model formula with the right hand side specifying the latency variables, or the same data.frame passed to the `data` parameter. Note that when using the model formula syntax for `x_latency` it cannot handle `x_latency = ~ .`.
`model`	type of regression model to use for the latency portion of mixture cure model. Can be "cox", "weibull", or "exponential" (default is "cox").
`penalty`	type of penalty function. Can be "lasso", "MCP", or "SCAD" (default is "lasso").
`penalty_factor_inc`	vector of binary indicators representing the penalty to apply to each incidence coefficient: 0 implies no shrinkage and 1 implies shrinkage. If not supplied, 1 is applied to all incidence variables.
`penalty_factor_lat`	vector of binary indicators representing the penalty to apply to each latency coefficient: 0 implies no shrinkage and 1 implies shrinkage. If not supplied, 1 is applied to all latency variables.
`thresh`	small numeric value. The iterative process stops when the differences between successive expected penalized complete-data log-likelihoods for both incidence and latency components are less than this specified level of tolerance (default is 10^-3).
`scale`	logical, if TRUE the predictors are centered and scaled.
`maxit`	integer specifying the maximum number of passes over the data for each lambda. If not specified, 100 is applied when `penalty = "lasso"` and 1000 is applied when `penalty = "MCP"` or `penalty = "SCAD"`.
`inits`	an optional list specifiying the initial values. This includes: • `itct` the incidence intercept. • `b_u` a numeric vector for the unpenalized incidence coefficients for the incidence portion of the model. • `beta_u` a numeric vector for unpenalized latency coefficients in the incidence portion of the model. • `lambda` a numeric value for the rate parameter when fitting either a Weibull or exponential MCM using `model = "weibull"` or `model = "exponential"`. • `alpha` a numeric value for the shape parameter when fitting a Weibull MCM using `model = "weibull"`. • `survprob` a numeric vector for the latency survival probabilities $S_u(t_i\|w_i)$ for i=1,...,N when fitting a Cox MCM `model = "cox"`. Penalized coefficients are initialized to zero. If `inits` is not specified or improperly specified, initialization is automatically provided by the function.
`lambda_inc`	numeric value for the penalization parameter λ for variables in the incidence portion of the model.
`lambda_lat`	numeric value for the penalization parameter λ for variables in the latency portion of the model.

`gamma_inc`	numeric value for the penalization parameter γ for variables in the incidence portion of the model when `penalty = "MCP"` or `penalty = "SCAD"` (default is 3).
`gamma_lat`	numeric value for the penalization parameter γ for variables in the latency portion of the model when `penalty = "MCP"` or `penalty = "SCAD"` (default is 3).
`na.action`	this function requires complete data so `"na.omit"` is invoked. Users can impute missing data as an alternative prior to model fitting.
`...`	additional arguments.

Value

`b_path`	Matrix representing the solution path of the coefficients in the incidence portion of the model. Row is step and column is variable.
`beta_path`	Matrix representing the solution path of the coefficients in the latency portion of the model. Row is step and column is variable.
`b0_path`	Vector representing the solution path of the intercept in the incidence portion of the model.
`logLik_inc`	Vector representing the expected penalized complete-data log-likelihood for the incidence portion of the model for each step in the solution path.
`logLik_lat`	Vector representing the expected penalized complete-data log-likelihood for the latency portion of the model for each step in the solution path.
`x_incidence`	Matrix representing the design matrix of the incidence predictors.
`x_latency`	Matrix representing the design matrix of the latency predictors.
`y`	Vector representing the survival object response as returned by the `Surv` function
`model`	Character string indicating the type of regression model used for the latency portion of mixture cure model ("weibull" or "exponential").
`scale`	Logical value indicating whether the predictors were centered and scaled.
`method`	Character string indicating the EM algorithm was used in fitting the mixture cure model.
`rate_path`	Vector representing the solution path of the rate parameter for the Weibull or exponential density in the latency portion of the model.
`alpha_path`	Vector representing the solution path of the shape parameter for the Weibull density in the latency portion of the model.
`call`	the matched call.

References

Archer, K. J., Fu, H., Mrozek, K., Nicolet, D., Mims, A. S., Uy, G. L., Stock, W., Byrd, J. C., Hiddemann, W., Braess, J., Spiekermann, K., Metzeler, K. H., Herold, T., Eisfeld, A.-K. (2024) Identifying long-term survivors and those at higher or lower risk of relapse among patients with cytogenetically normal acute myeloid leukemia using a high-dimensional mixture cure model. *Journal of Hematology & Oncology*, **17**:28.

See Also

cv_cureem

Examples

```
library(survival)
withr::local_seed(1234)
temp <- generate_cure_data(n = 80, j = 100, n_true = 10, a = 1.8)
training <- temp$training
fit <- cureem(Surv(Time, Censor) ~ .,
  data = training, x_latency = training,
  model = "cox", penalty = "lasso", lambda_inc = 0.1,
  lambda_lat = 0.1, gamma_inc = 6, gamma_lat = 10
)
```

---

curegmifs *Fit penalized parametric mixture cure model using the GMIFS algorithm*

Description

Fits a penalized Weibull or exponential mixture cure model using the generalized monotone incremental forward stagewise (GMIFS) algorithm (Hastie et al 2007) and yields solution paths for parameters in the incidence and latency portions of the model.

Usage

```
curegmifs(
  formula,
  data,
  subset,
  x_latency = NULL,
  model = c("weibull", "exponential"),
  penalty_factor_inc = NULL,
  penalty_factor_lat = NULL,
  epsilon = 0.001,
  thresh = 1e-05,
  scale = TRUE,
  maxit = 10000,
  inits = NULL,
  verbose = TRUE,
  suppress_warning = FALSE,
  na.action = na.omit,
  ...
)
```

Arguments

`formula`	an object of class "`formula`" (or one that can be coerced to that class): a symbolic description of the model to be fitted. The response must be a survival object as returned by the `Surv` function while the variables on the right side of the formula are the covariates that are included in the incidence portion of the model.
`data`	a data.frame in which to interpret the variables named in the `formula` or in the `subset` argument. Rows with missing data are omitted (only `na.action = na.omit` is operational) therefore users may want to impute missing data prior to calling this function.
`subset`	an optional expression indicating which subset of observations to be used in the fitting process, either a numeric or factor variable should be used in subset, not a character variable. All observations are included by default.
`x_latency`	specifies the variables to be included in the latency portion of the model and can be either a matrix of predictors, a model formula with the right hand side specifying the latency variables, or the same data.frame passed to the `data` parameter. Note that when using the model formula syntax for `x_latency` it cannot handle `x_latency = ~ .`.
`model`	type of regression model to use for the latency portion of mixture cure model. Can be "weibull" or "exponential"; default is "weibull".
`penalty_factor_inc`	vector of binary indicators representing the penalty to apply to each incidence coefficient: 0 implies no shrinkage and 1 implies shrinkage. If not supplied, 1 is applied to all incidence variables.
`penalty_factor_lat`	vector of binary indicators representing the penalty to apply to each latency coefficient: 0 implies no shrinkage and 1 implies shrinkage. If not supplied, 1 is applied to all latency variables.
`epsilon`	small numeric value reflecting the incremental value used to update a coefficient at a given step (default is 0.001).
`thresh`	small numeric value. The iterative process stops when the differences between successive expected penalized log-likelihoods for both incidence and latency components are less than this specified level of tolerance (default is 10^-5).
`scale`	logical, if TRUE the predictors are centered and scaled.
`maxit`	integer specifying the maximum number of steps to run in the iterative algorithm (default is 10^4).
`inits`	an optional list specifying the initial values as follows: • `itct` a numeric value for the unpenalized incidence intercept. • `b_u` a numeric vector for the unpenalized incidence coefficients. • `beta_u` a numeric vector for unpenalized latency coefficients. • `lambda` a numeric value for the rate parameter. • `alpha` a numeric value for the shape parameter when `model = "weibull"`. If not supplied or improperly supplied, initialization is automatically provided by the function.

`verbose`	logical, if TRUE running information is printed to the console (default is FALSE).
`suppress_warning`	logical, if TRUE, suppresses echoing the warning that the maximum number of iterations was reached so that the algorithm may not have converged. Instead, warning is returned as part of the output with this message.
`na.action`	this function requires complete data so `"na.omit"` is invoked. Users can impute missing data as an alternative prior to model fitting.
`...`	additional arguments.

Value

`b_path`	Matrix representing the solution path of the coefficients in the incidence portion of the model. Row is step and column is variable.
`beta_path`	Matrix representing the solution path of the coefficients in the latency portion of the model. Row is step and column is variable.
`b0_path`	Vector representing the solution path of the intercept in the incidence portion of the model.
`rate_path`	Vector representing the solution path of the rate parameter for the Weibull or exponential density in the latency portion of the model.
`logLik`	Vector representing the log-likelihood for each step in the solution path.
`x_incidence`	Matrix representing the design matrix of the incidence predictors.
`x_latency`	Matrix representing the design matrix of the latency predictors.
`y`	Vector representing the survival object response as returned by the `Surv` function
`model`	Character string indicating the type of regression model used for the latency portion of mixture cure model ("weibull" or "exponential").
`scale`	Logical value indicating whether the predictors were centered and scaled.
`alpha_path`	Vector representing the solution path of the shape parameter for the Weibull density in the latency portion of the model.
`call`	the matched call.
`warning`	message indicating whether the maximum number of iterations was achieved which may indicate the model did not converge.

References

Fu, H., Nicolet, D., Mrozek, K., Stone, R. M., Eisfeld, A. K., Byrd, J. C., Archer, K. J. (2022) Controlled variable selection in Weibull mixture cure models for high-dimensional data. *Statistics in Medicine*, **41**(22), 4340–4366.

Hastie, T., Taylor J., Tibshirani R., Walther G. (2007) Forward stagewise regression and the monotone lasso. *Electron J Stat*, **1**:1–29.

See Also

`cv_curegmifs`

Examples

```
library(survival)
withr::local_seed(1234)
temp <- generate_cure_data(n = 100, j = 10, n_true = 10, a = 1.8)
training <- temp$training

fit <- curegmifs(Surv(Time, Censor) ~ .,
  data = training, x_latency = training,
  model = "weibull", thresh = 1e-4, maxit = 2000, epsilon = 0.01,
  verbose = FALSE
)
```

`cure_estimate` *Estimate cured fraction*

Description

Estimates the cured fraction using a Kaplan-Meier fitted object.

Usage

```
cure_estimate(object)
```

Arguments

`object`	a `survfit` object.

Value

estimated proportion of cured observations

See Also

`survfit`, `sufficient_fu_test`, `nonzerocure_test`

Examples

```
library(survival)
withr::local_seed(1234)
temp <- generate_cure_data(n = 100, j = 10, n_true = 10, a = 1.8)
training <- temp$training
km_fit <- survfit(Surv(Time, Censor) ~ 1, data = training)
cure_estimate(km_fit)
```

`cv_cureem` *Fit penalized mixture cure model using the E-M algorithm with cross-validation for parameter tuning*

Description

Fits penalized parametric and semi-parametric mixture cure models (MCM) using the E-M algorithm with with k-fold cross-validation for parameter tuning. The lasso (L1), MCP and SCAD penalty are supported for the Cox MCM while only lasso is currently supported for parametric MCMs. When FDR controlled variable selection is used, the model-X knockoffs method is applied and indices of selected variables are returned.

Usage

```
cv_cureem(
  formula,
  data,
  subset,
  x_latency = NULL,
  model = c("cox", "weibull", "exponential"),
  penalty = c("lasso", "MCP", "SCAD"),
  penalty_factor_inc = NULL,
  penalty_factor_lat = NULL,
  fdr_control = FALSE,
  fdr = 0.2,
  grid_tuning = FALSE,
  thresh = 0.001,
  scale = TRUE,
  maxit = NULL,
  inits = NULL,
  lambda_inc_list = NULL,
  lambda_lat_list = NULL,
  nlambda_inc = NULL,
  nlambda_lat = NULL,
  gamma_inc = 3,
  gamma_lat = 3,
  lambda_min_ratio_inc = 0.1,
  lambda_min_ratio_lat = 0.1,
  n_folds = 5,
  measure_inc = c("c", "auc"),
  one_se = FALSE,
  cure_cutoff = 5,
  parallel = FALSE,
  seed = NULL,
  verbose = TRUE,
  na.action = na.omit,
  ...
```

```
)
```

Arguments

`formula`	an object of class "`formula`" (or one that can be coerced to that class): a symbolic description of the model to be fitted. The response must be a survival object as returned by the `Surv` function while the variables on the right side of the formula are the covariates that are included in the incidence portion of the model.
`data`	a data.frame in which to interpret the variables named in the `formula` or in the `subset` argument. Rows with missing data are omitted (only `na.action = na.omit` is operational) therefore users may want to impute missing data prior to calling this function.
`subset`	an optional expression indicating which subset of observations to be used in the fitting process, either a numeric or factor variable should be used in subset, not a character variable. All observations are included by default.
`x_latency`	specifies the variables to be included in the latency portion of the model and can be either a matrix of predictors, a model formula with the right hand side specifying the latency variables, or the same data.frame passed to the `data` parameter. Note that when using the model formula syntax for `x_latency` it cannot handle `x_latency = ~ .`.
`model`	type of regression model to use for the latency portion of mixture cure model. Can be "cox", "weibull", or "exponential" (default is "cox").
`penalty`	type of penalty function. Can be "lasso", "MCP", or "SCAD" (default is "lasso").
`penalty_factor_inc`	vector of binary indicators representing the penalty to apply to each incidence coefficient: 0 implies no shrinkage and 1 implies shrinkage. If not supplied, 1 is applied to all incidence variables.
`penalty_factor_lat`	vector of binary indicators representing the penalty to apply to each latency coefficient: 0 implies no shrinkage and 1 implies shrinkage. If not supplied, 1 is applied to all latency variables.
`fdr_control`	logical, if TRUE, model-X knockoffs are used for FDR-controlled variable selection and indices of selected variables are returned (default is FALSE).
`fdr`	numeric value in (0, 1) range specifying the target FDR level to use for variable selection when `fdr_control = TRUE` (default is 0.2).
`grid_tuning`	logical, if TRUE a 2-D grid tuning approach is used to select the optimal pair of λ_b and λ_β penalty parameters for the incidence and latency portions of the model, respectively. Otherwise the λ_b and λ_β are selected from a 1-D sequence and are equal to one another (default is FALSE).
`thresh`	small numeric value. The iterative process stops when the differences between successive expected penalized complete-data log-likelihoods for both incidence and latency components are less than this specified level of tolerance (default is 10^-3).
`scale`	logical, if TRUE the predictors are centered and scaled.

`maxit`	maximum number of passes over the data for each lambda. If not specified, 100 is applied when `penalty = "lasso"` and 1000 is applied when `penalty = "MCP"` or `penalty = "SCAD"`.
`inits`	an optional list specifiying the initial values to be used for model fitting as follows: - `itct` the incidence intercept. - `b_u` a numeric vector for the unpenalized. incidence coefficients for the incidence portion of the model. - `beta_u` a numeric vector for unpenalized latency coefficients in the incidence portion of the model. - `lambda` a numeric value for the rate parameter when fitting either a Weibull or exponential MCM using `model = "weibull"` or `model = "exponential"`. - `alpha` a numeric value for the shape parameter when fitting a Weibull MCM using `model = "weibull"`. - `survprob` a numeric vector for the latency survival probabilities $S_u(t_i\|w_i)$ for i=1,...,N when fitting a Cox MCM `model = "cox"`. Penalized coefficients are initialized to zero. If `inits` is not specified or improperly specified, initialization is automatically provided by the function.
`lambda_inc_list`	a numeric vector used to search for the optimal λ_b tuning parameter. If not supplied, the function computes a λ_b sequence based on `nlambda_inc` and `lambda_min_ratio_inc`. If `grid_tuning = FALSE`, the same sequence should be used for both λ_b and λ_β.
`lambda_lat_list`	a numeric vector used to search for the optimal λ_β tuning parameter. If not supplied, the function computes a λ_β sequence based on `nlambda_lat` and `lambda_min_ratio_lat`. If `grid_tuning = FALSE`, the same sequence should be used for both λ_b and λ_β.
`nlambda_inc`	an integer specifying the number of values to search for the optimal λ_b tuning parameter; default is 10 if `grid_tuning = TRUE` and 50 otherwise.
`nlambda_lat`	an integer specifying the number of values to search for the optimal λ_β tuning parameter; default is 10 if `grid_tuning = TRUE` and 50 otherwise.
`gamma_inc`	numeric value for the penalization parameter γ for variables in the incidence portion of the model when `penalty = "MCP"` or `penalty = "SCAD"` (default is 3).
`gamma_lat`	numeric value for the penalization parameter γ for variables in the latency portion of the model when `penalty = "MCP"` or `penalty = "SCAD"` (default is 3).
`lambda_min_ratio_inc`	numeric value in (0,1) representing the smallest value for λ_b as a fraction of `lambda.max_inc`, the data-derived entry value at which essentially all penalized variables in the incidence portion of the model have a coefficient estimate of 0 (default is 0.1).
`lambda_min_ratio_lat`	numeric value in (0.1) representing the smallest value for λ_β as a fraction of `lambda.max_lat`, the data-derived entry value at essentially all penalized variables in the latency portion of the model have a coefficient estimate of 0 (default is 0.1).

`n_folds`	an integer specifying the number of folds for the k-fold cross-valiation procedure (default is 5).
`measure_inc`	character string specifying the evaluation criterion used in selecting the optimal λ_b which can be either • `"c"` specifying to use the C-statistic for cure status weighting (CSW) method proposed by Asano and Hirakawa (2017) to select both λ_b and λ_β • `"auc"` specifying to use the AUC for cure prediction using the mean score imputation (MSI) method proposed by Asano et al. (2014) to select λ_b while the C-statistic with CSW is used for λ_β.
`one_se`	logical, if TRUE then the one standard error rule is applied for selecting the optimal parameters. The one standard error rule selects the most parsimonious model having evaluation criterion no more than one standard error worse than that of the best evaluation criterion (default is FALSE).
`cure_cutoff`	numeric value representing the cutoff time value that represents subjects not experiencing the event by this time are cured. This value is used to produce a proxy for the unobserved cure status when calculating C-statistic and AUC (default is 5 representing 5 years). Users should be careful to note the time scale of their data and adjust this according to the time scale and clinical application.
`parallel`	logical. If TRUE, parallel processing is performed for K-fold CV using `foreach` and the **doParallel** package is required.
`seed`	optional integer representing the random seed. Setting the random seed fosters reproducibility of the results.
`verbose`	logical, if TRUE running information is printed to the console (default is FALSE).
`na.action`	this function requires complete data so `"na.omit"` is invoked. Users can impute missing data as an alternative prior to model fitting.
`...`	additional arguments.

Value

`b0`	Estimated intercept for the incidence portion of the model.
`b`	Estimated coefficients for the incidence portion of the model.
`beta`	Estimated coefficients for the latency portion of the model.
`alpha`	Estimated shape parameter if the Weibull model is fit.
`rate`	Estimated rate parameter if the Weibull or exponential model is fit.
`logLik_inc`	Expected penalized complete-data log-likelihood for the incidence portion of the model.
`logLik_lat`	Expected penalized complete-data log-likelihood for the latency portion of the model.
`selected_lambda_inc`	Value of λ_b selected using cross-validation. NULL when fdr_control is TRUE.
`selected_lambda_lat`	Value of λ_β selected using cross-validation. NULL when fdr_control is TRUE.
`max_c`	Maximum C-statistic achieved.

`max_auc`	Maximum AUC for cure prediction achieved; only output when `measure_inc="auc"`.
`selected_index_inc`	Indices of selected variables for the incidence portion of the model when `fdr_control=TRUE`. If no variables are selected, `int(0)` will be returned.
`selected_index_lat`	Indices of selected variables for the latency portion of the model when `fdr_control=TRUE`. If no variables are selected, `int(0)` will be returned.
`call`	the matched call.

References

Archer, K. J., Fu, H., Mrozek, K., Nicolet, D., Mims, A. S., Uy, G. L., Stock, W., Byrd, J. C., Hiddemann, W., Braess, J., Spiekermann, K., Metzeler, K. H., Herold, T., Eisfeld, A.-K. (2024) Identifying long-term survivors and those at higher or lower risk of relapse among patients with cytogenetically normal acute myeloid leukemia using a high-dimensional mixture cure model. *Journal of Hematology & Oncology*, **17**:28.

See Also

`cureem`

Examples

```
library(survival)
withr::local_seed(1234)
temp <- generate_cure_data(n = 200, j = 25, n_true = 5, a = 1.8)
training <- temp$training
fit.cv <- cv_cureem(Surv(Time, Censor) ~ .,
  data = training,
  x_latency = training, fdr_control = FALSE,
  grid_tuning = FALSE, nlambda_inc = 10, nlambda_lat = 10,
  n_folds = 2, seed = 23, verbose = TRUE
)
fit.cv.fdr <- cv_cureem(Surv(Time, Censor) ~ .,
  data = training,
  x_latency = training, model = "weibull", penalty = "lasso",
  fdr_control = TRUE, grid_tuning = FALSE, nlambda_inc = 10,
  nlambda_lat = 10, n_folds = 2, seed = 23, verbose = TRUE)
```

`cv_curegmifs` *Fit a penalized parametric mixture cure model using the generalized monotone incremental forward stagewise (GMIFS) algorithm (Hastie et al 2007) with cross-validation for model selection*

Description

Fits a penalized Weibull or exponential mixture cure model using the generalized monotone incremental forward stagewise (GMIFS) algorithm with k-fold cross-validation to select the optimal iteration step along the solution path. When FDR controlled variable selection is used, the model-X knockoffs method is applied and indices of selected variables are returned.

Usage

```
cv_curegmifs(
  formula,
  data,
  subset,
  x_latency = NULL,
  model = c("weibull", "exponential"),
  penalty_factor_inc = NULL,
  penalty_factor_lat = NULL,
  fdr_control = FALSE,
  fdr = 0.2,
  epsilon = 0.001,
  thresh = 1e-05,
  scale = TRUE,
  maxit = 10000,
  inits = NULL,
  n_folds = 5,
  measure_inc = c("c", "auc"),
  one_se = FALSE,
  cure_cutoff = 5,
  parallel = FALSE,
  seed = NULL,
  verbose = TRUE,
  na.action = na.omit,
  ...
)
```

Arguments

`formula`	an object of class "`formula`" (or one that can be coerced to that class): a symbolic description of the model to be fitted. The response must be a survival object as returned by the `Surv` function while the variables on the right side of the formula are the covariates that are included in the incidence portion of the model.
`data`	a data.frame in which to interpret the variables named in the `formula` or in the `subset` argument. Rows with missing data are omitted (only `na.action = na.omit` is operational) therefore users may want to impute missing data prior to calling this function.
`subset`	an optional expression indicating which subset of observations to be used in the fitting process, either a numeric or factor variable should be used in subset, not a character variable. All observations are included by default.
`x_latency`	specifies the variables to be included in the latency portion of the model and can be either a matrix of predictors, a model formula with the right hand side specifying the latency variables, or the same data.frame passed to the `data` parameter. Note that when using the model formula syntax for `x_latency` it cannot handle `x_latency = ~ .`.

`model`	type of regression model to use for the latency portion of mixture cure model. Can be "weibull" or "exponential"; default is "weibull".
`penalty_factor_inc`	vector of binary indicators representing the penalty to apply to each incidence coefficient: 0 implies no shrinkage and 1 implies shrinkage. If not supplied, 1 is applied to all incidence variables.
`penalty_factor_lat`	vector of binary indicators representing the penalty to apply to each latency coefficient: 0 implies no shrinkage and 1 implies shrinkage. If not supplied, 1 is applied to all latency variables.
`fdr_control`	logical, if TRUE, model-X knockoffs are used for FDR-controlled variable selection and indices of selected variables are returned (default is FALSE).
`fdr`	numeric value in (0, 1) range specifying the target FDR level to use for variable selection when `fdr_control = TRUE` (default is 0.2).
`epsilon`	small numeric value reflecting incremental value used to update a coefficient at a given step (default is 0.001).
`thresh`	small numeric value. The iterative process stops when the differences between successive expected penalized complete-data log-likelihoods for both incidence and latency components are less than this specified level of tolerance (default is 10^-5).
`scale`	logical, if TRUE the predictors are centered and scaled.
`maxit`	integer specifying the maximum number of steps to run in the iterative algorithm (default is 10^4).
`inits`	an optional list specifying the initial values as follows: - `itct` a numeric value for the unpenalized incidence intercept. - `b_u` a numeric vector for the unpenalized incidence coefficients. - `beta_u` a numeric vector for unpenalized latency coefficients. - `lambda` a numeric value for the rate parameter. - `alpha` a numeric value for the shape parameter when `model = "weibull"`. If `inits` is not specified or improperly supplied, initialization is automatically provided by the function.
`n_folds`	an integer specifying the number of folds for the k-fold cross-validation procedure (default is 5).
`measure_inc`	character string specifying the evaluation criterion used in selecting the optimal λ_b which can be either - `"c"` specifying to use the C-statistic for cure status weighting (CSW) method proposed by Asano and Hirakawa (2017) to select both λ_b and λ_β - `"auc"` specifying to use the AUC for cure prediction using the mean score imputation (MSI) method proposed by Asano et al. (2014) to select λ_b while the C-statistic with CSW is used for λ_β.
`one_se`	logical, if TRUE then the one standard error rule is applied for selecting the optimal parameters. The one standard error rule selects the most parsimonious model having evaluation criterion no more than one standard error worse than that of the best evaluation criterion (default is FALSE).

`cure_cutoff`	numeric value representing the cutoff time value that represents subjects not experiencing the event by this time are cured. This value is used to produce a proxy for the unobserved cure status when calculating C-statistic and AUC (default is 5 representing 5 years). Users should be careful to note the time scale of their data and adjust this according to the time scale and clinical application.
`parallel`	logical. If TRUE, parallel processing is performed for K-fold CV using `foreach` and the **doParallel** package is required.
`seed`	optional integer representing the random seed. Setting the random seed fosters reproducibility of the results.
`verbose`	logical, if TRUE running information is printed to the console (default is FALSE).
`na.action`	this function requires complete data so `"na.omit"` is invoked. Users can impute missing data as an alternative prior to model fitting.
`...`	additional arguments.

Value

`b0`	Estimated intercept for the incidence portion of the model.
`b`	Estimated coefficients for the incidence portion of the model.
`beta`	Estimated coefficients for the latency portion of the model.
`alpha`	Estimated shape parameter if the Weibull model is fit.
`rate`	Estimated rate parameter if the Weibull or exponential model is fit.
`logLik`	Log-likelihood value.
`selected_step_inc`	Iteration step selected for the incidence portion of the model using cross-validation. NULL when fdr_control is TRUE.
`selected_step_lat`	Iteration step selected for the latency portion of the model using cross-validation. NULL when fdr_control is TRUE.
`max_c`	Maximum C-statistic achieved
`max_auc`	Maximum AUC for cure prediction achieved; only output when `measure_inc = "auc"`.
`selected_index_inc`	Indices of selected variables for the incidence portion of the model when `fdr_control = TRUE`. If none selected, `int(0)` will be returned.
`selected_index_lat`	Indices of selected variables for the latency portion of the model when `fdr_control = TRUE`. If none selected, `int(0)` will be returned.
`call`	the matched call.

References

Fu, H., Nicolet, D., Mrozek, K., Stone, R. M., Eisfeld, A. K., Byrd, J. C., Archer, K. J. (2022) Controlled variable selection in Weibull mixture cure models for high-dimensional data. *Statistics in Medicine*, **41**(22), 4340–4366.

Hastie, T., Taylor J., Tibshirani R., Walther G. (2007) Forward stagewise regression and the monotone lasso. *Electron J Stat*, **1**:1–29.

See Also

curegmifs

Examples

```
library(survival)
withr::local_seed(123)
temp <- generate_cure_data(n = 100, j = 15, n_true = 3, a = 1.8, rho = 0.2)
training <- temp$training

fit.cv <- cv_curegmifs(Surv(Time, Censor) ~ .,
  data = training,
  x_latency = training, fdr_control = FALSE,
  maxit = 450, epsilon = 0.01, n_folds = 2,
  seed = 23, verbose = TRUE
)
```

`dim.mixturecure` *Dimension method for mixturecure objects*

Description

Dimension method for `mixturecure` objects.

Usage

```
## S3 method for class 'mixturecure'
dim(x)
```

Arguments

`x`	An object of class `mixturecure`.

Value

`nobs`	number of subjects in the dataset.
`p_incidence`	number of variables in the incidence portion of the model.
`p_latency`	number of variables in the latency portion of the model.

Examples

```
library(survival)
withr::local_seed(1234)
temp <- generate_cure_data(n = 100, j = 10, n_true = 10, a = 1.8)
training <- temp$training
fit <- curegmifs(Surv(Time, Censor) ~ .,
  data = training, x_latency = training,
  model = "weibull", thresh = 1e-4, maxit = 2000,
```

```
  epsilon = 0.01, verbose = FALSE
)
dim(fit)
```

`family.mixturecure` *Return model family and fitting algorithm for mixturecure model fits*

Description

Return model family and fitting algorithm for`mixturecure` model fits.

Usage

```
## S3 method for class 'mixturecure'
family(object, ...)
```

Arguments

`object`	an object of class `mixturecure`
`...`	other arguments.

Value

the parametric or semi-parametric model fit and the fitting algorithm.

Examples

```
library(survival)
withr::local_seed(1234)
temp <- generate_cure_data(n = 100, j = 10, n_true = 10, a = 1.8)
training <- temp$training
fit <- curegmifs(Surv(Time, Censor) ~ .,
  data = training, x_latency = training,
  model = "weibull", thresh = 1e-4, maxit = 2000,
  epsilon = 0.01, verbose = FALSE
)
family(fit)
```

`formula.mixturecure` *Extract model formula for mixturecure object*

Description

Extract the model formula for `mixturecure` object

Usage

```
## S3 method for class 'mixturecure'
formula(x, ...)
```

Arguments

`x`	an object from class `mixturecure`.
`...`	other arguments.

Value

a formula representing the incidence and variables for the latency portion of the model

Examples

```
library(survival)
withr::local_seed(1234)
temp <- generate_cure_data(n = 100, j = 10, n_true = 10, a = 1.8)
training <- temp$training
fit <- curegmifs(Surv(Time, Censor) ~ .,
  data = training, x_latency = training,
  model = "weibull", thresh = 1e-4, maxit = 2000,
  epsilon = 0.01, verbose = FALSE
)
formula(fit)
```

`generate_cure_data` *Simulate data under a mixture cure model*

Description

Simulate data under a mixture cure model.

Usage

```
generate_cure_data(
  n = 400,
  j = 500,
  nonp = 2,
  train_prop = 0.75,
  n_true = 10,
  a = 1,
  rho = 0.5,
  itct_mean = 0.5,
  cens_ub = 20,
  alpha = 1,
  lambda = 2,
  same_signs = FALSE,
  model = "weibull"
)
```

Arguments

`n`	an integer denoting the total sample size.
`j`	an integer denoting the number of penalized predictors which is the same for both the incidence and latency portions of the model.
`nonp`	an integer denoting the number of unpenalized predictors (which is the same for both the incidence and latency portions of the model).
`train_prop`	a numeric value in [0, 1) representing the fraction of `n` to be used in forming the training dataset.
`n_true`	an integer less than `j` denoting the number of variables truly associated with the outcome (i.e., the number of covariates with nonzero parameter values) among the penalized predictors.
`a`	a numeric value denoting the effect size (signal amplitude) which is the same for both the incidence and latency portions of the model.
`rho`	a numeric value in [0, 1) representing the correlation between adjacent covariates in the same block.
`itct_mean`	a numeric value representing the expectation of the incidence intercept which controls the cure rate.
`cens_ub`	a numeric value representing the upper bound on the censoring time distribution which follows a uniform distribution on (0, `cens_ub`].
`alpha`	a numeric value representing the shape parameter in the Weibull density.
`lambda`	a numeric value representing the rate parameter in the Weibull density.
`same_signs`	logical, if TRUE the incidence and latency coefficients have the same signs.
`model`	type of regression model to use for the latency portion of mixture cure model. Can be one of the following: • `"weibull"` to generate times from a Weibull distribution. • `"GG"` to generate times from a generalized gamma distribution.

- `"Gompertz"` to generate times from a Gomertz distribution.
- `"nonparametric"` to generate times non-parametrically.
- `"GG_baseline"` to generate times from a generalized gamma baseline distribution.

Value

`training`	training data.frame which includes Time, Censor, and covariates. Variables prefixed with `"U"` indicates unpenalized covariates and is equal to the value passed to `nonp` (default is 2). Variables prefixed with `"X"` indicates penalized covariates and is equal to the value passed to `j`.
`testing`	testing data.frame which includes Time, Censor, and covariates. Variables prefixed with `"U"` indicates unpenalized covariates and is equal to the value passed to `nonp` (default is 2). Variables prefixed with `"X"` indicates penalized covariates and is equal to the value passed to `j`.
`parameters`	a list including: the indices of true incidence signals (`nonzero_b`), indices of true latency signals (`nonzero_beta`), unpenalized incidence parameter values (`b_u`), unpenalized latency parameter values (`beta_u`), parameter values for the true incidence signals among penalized covariates (`b_p_nz`), parameter values for the true latency signals among penalized covariates (`beta_p_nz`), parameter value for the incidence intercept (`itct`)

Examples

```
library(survival)
withr::local_seed(1234)
# This dataset has 2 penalized features associated with the outcome,
# 3 penalized features not associated with the outcome (noise features), and 1
# unpenalized noise feature.
data <- generate_cure_data(n = 1000, j = 5, n_true = 2, nonp = 1, a = 2)
# Extract the training data
training <- data$training
# Extract the testing data
testing <- data$testing
# To identify the features truly associated with incidence
names(training)[grep("^X", names(training))][data$parameters$nonzero_b]
# To identify the features truly associated with latency
names(training)[grep("^X", names(training))][data$parameters$nonzero_beta]
# Fit the model to the training data
fitem <- cureem(Surv(Time, Censor) ~ ., data = training,
  x_latency = training)
# Examine the estimated coefficients at the (default) minimum AIC
coef(fitem)
# As the penalty increases, the coefficients for the noise variables shrink
# to or remain at zero, while the truly associated features have coefficient
# paths that depart from zero. This shows the model's ability to distinguish
# signal from noise.
plot(fitem, label = TRUE)
```

`logLik.mixturecure` *Log-likelihood for fitted mixture cure model*

Description

This function returns the log-likelihood for a user-specified model criterion or step for a `curegmifs`, `cureem`, `cv_curegmifs` or `cv_cureem` fitted object.

Usage

```
## S3 method for class 'mixturecure'
logLik(object, model_select = "AIC", ...)
```

Arguments

`object`	a `mixturecure` object resulting from `curegmifs`, `cureem`, `cv_curegmifs`, `cv_cureem`.
`model_select`	either a case-sensitive parameter for models fit using `curegmifs` or `cureem` or any numeric step along the solution path can be selected. The default is `model_select = "AIC"` which calculates the predicted values using the coefficients from the model achieving the minimum AIC. The complete list of options are: • `"AIC"` for the minimum AIC (default). • `"mAIC"` for the minimum modified AIC. • `"cAIC"` for the minimum corrected AIC. • `"BIC"`, for the minimum BIC. • `"mBIC"` for the minimum modified BIC. • `"EBIC"` for the minimum extended BIC. • `"logLik"` for the step that maximizes the log-likelihood. • `n` where `n` is any numeric value from the solution path. This option has no effect for objects fit using `cv_curegmifs` or `cv_cureem`.
`...`	other arguments.

Value

log-likelihood of the fitted mixture cure model using the specified criteria.

Examples

```
library(survival)
withr::local_seed(1234)
temp <- generate_cure_data(n = 100, j = 10, n_true = 10, a = 1.8)
training <- temp$training
fit <- curegmifs(Surv(Time, Censor) ~ .,
  data = training, x_latency = training,
  model = "weibull", thresh = 1e-4, maxit = 2000,
  epsilon = 0.01, verbose = FALSE
```

```
)
logLik(fit, model_select = "AIC")
```

`nobs.mixturecure` *Number of observations in mixturecure object*

Description

Number of observations in fitted `mixturecure` object.

Usage

```
## S3 method for class 'mixturecure'
nobs(object, ...)
```

Arguments

`object`	An object of class `mixturecure`.
`...`	other arguments.

Value

number of subjects in the dataset.

Examples

```
library(survival)
withr::local_seed(1234)
temp <- generate_cure_data(n = 100, j = 10, n_true = 10, a = 1.8)
training <- temp$training
fit <- curegmifs(Surv(Time, Censor) ~ .,
  data = training, x_latency = training,
  model = "weibull", thresh = 1e-4, maxit = 2000,
  epsilon = 0.01, verbose = FALSE
)
nobs(fit)
```

`nonzerocure_test` *Non-parametric test for a non-zero cured fraction*

Description

Tests the null hypothesis that the proportion of observations susceptible to the event = 1 against the alternative that the proportion of observations susceptible to the event is < 1. If the null hypothesis is rejected, there is a significant cured fraction.

Usage

```
nonzerocure_test(object, reps = 1000, seed = NULL, plot = FALSE, b = NULL)
```

Arguments

`object`	a `survfit` object.
`reps`	number of simulations on which to base the p-value (default = 1000).
`seed`	optional random seed.
`plot`	logical. If TRUE a histogram of the estimated susceptible proportions over all simulations is produced.
`b`	optional. If specified the maximum observed time for the uniform distribution for generating the censoring times. If not specified, an exponential model is used for generating the censoring times (default).

Value

`proportion_susceptible`	estimated proportion of susceptibles
`proportion_cured`	estimated proportion of those cured
`p_value`	p-value testing the null hypothesis that the proportion of susceptibles = 1 (cured fraction = 0) against the alternative that the proportion of susceptibles < 1 (non-zero cured fraction)
`time_95_percent_of_events`	estimated time at which 95% of events should have occurred

References

Maller, R. A. and Zhou, X. (1996) *Survival Analysis with Long-Term Survivors*. John Wiley & Sons.

See Also

`survfit`, `cure_estimate`, `sufficient_fu_test`

Examples

```
library(survival)
withr::local_seed(1234)
temp <- generate_cure_data(n = 100, j = 10, n_true = 10, a = 1.8)
training <- temp$training
km_fit <- survfit(Surv(Time, Censor) ~ 1, data = training)
nonzerocure_test(km_fit)
```

`npar_mixturecure` *Number of parameters in fitted mixture cure model*

Description

This function returns the number of parameters in a user-specified model criterion or step for a `curegmifs`, `cureem`, `cv_curegmifs` or `cv_cureem` fitted object.

Usage

```
npar_mixturecure(object, model_select = "AIC")
```

Arguments

`object`	a `mixturecure` object resulting from `curegmifs`, `cureem`, `cv_curegmifs`, `cv_cureem`.
`model_select`	either a case-sensitive parameter for models fit using `curegmifs` or `cureem` or any numeric step along the solution path can be selected. The default is `model_select = "AIC"` which calculates the predicted values using the coefficients from the model achieving the minimum AIC. The complete list of options are: • `"AIC"` for the minimum AIC (default). • `"mAIC"` for the minimum modified AIC. • `"cAIC"` for the minimum corrected AIC. • `"BIC"`, for the minimum BIC. • `"mBIC"` for the minimum modified BIC. • `"EBIC"` for the minimum extended BIC. • `"logLik"` for the step that maximizes the log-likelihood. • `n` where `n` is any numeric value from the solution path. This option has no effect for objects fit using `cv_curegmifs` or `cv_cureem`.

Value

number of paramaters of the fitted mixture cure model using the specified criteria.

`plot.mixturecure` *Plot fitted mixture cure model*

Description

This function plots either the coefficient path, the AIC, the cAIC, the BIC, or the log-likelihood for a fitted `curegmifs` or `cureem` object. This function produces a lollipop plot of the coefficient estimates for a fitted `cv_curegmifs` or `cv_cureem` object.

Usage

```
## S3 method for class 'mixturecure'
plot(
  x,
  type = c("trace", "AIC", "BIC", "logLik", "cAIC", "mAIC", "mBIC", "EBIC"),
  xlab = NULL,
  ylab = NULL,
  label = FALSE,
  main = NULL,
  ...
)
```

Arguments

`x`	a `mixturecure` object resulting from `curegmifs` or `cureem`, `cv_curegmifs` or `cv_cureem`.
`type`	a case-sensitive parameter indicating what to plot on the y-axis. The complete list of options are: • `"trace"` plots the coefficient path for the fitted object (default). • `"AIC"` plots the AIC against step of model fit. • `"mAIC"` plots the modified AIC against step of model fit. • `"cAIC"` plots the corrected AIC against step of model fit. • `"BIC"`, plots the BIC against step of model fit. • `"mBIC"` plots the modified BIC against step of model fit. • `"EBIC"` plots the extended BIC against step of model fit. • `"logLik"` plots the log-likelihood against step of model fit. This option has no effect for objects fit using `cv_curegmifs` or `cv_cureem`.
`xlab`	a default x-axis label will be used which can be changed by specifying a user-defined x-axis label.
`ylab`	a default y-axis label will be used which can be changed by specifying a user-defined y-axis label.
`label`	logical. If TRUE the variable names will appear in a legend. Applicable only when `type = "trace"`. Be reminded that this works well only for small to moderate numbers of variables. For many predictors, the plot will be cluttered. The

	variables may be more easily identified using the `coef` function indicating the step of interest.
`main`	a default main title will be used which can be changed by specifying a user-defined main title. This option is not used for `cv_curegmifs` or `cv_cureem` fitted objects.
`...`	other arguments.

Value

this function has no returned value but is called for its side effects

See Also

`curegmifs`, `cureem`, `coef.mixturecure`, `summary.mixturecure`, `predict.mixturecure`

Examples

```
library(survival)
withr::local_seed(1234)
temp <- generate_cure_data(n = 100, j = 10, n_true = 10, a = 1.8)
training <- temp$training
fit <- curegmifs(Surv(Time, Censor) ~ .,
  data = training, x_latency = training,
  model = "weibull", thresh = 1e-4, maxit = 2000,
  epsilon = 0.01, verbose = FALSE
)
plot(fit)
```

`predict.mixturecure` *Predicted probabilities for susceptibles, linear predictor for latency, and risk class for latency for mixture cure fit*

Description

This function returns a list that includes the predicted probabilities for susceptibles as well as the linear predictor for the latency distribution and a dichotomous risk for latency for a `curegmifs`, `cureem`, `cv_curegmifs` or `cv_cureem` fitted object.

Usage

```
## S3 method for class 'mixturecure'
predict(object, newdata, model_select = "AIC", ...)
```

Arguments

`object`	a `mixturecure` object resulting from `curegmifs`, `cureem`, `cv_curegmifs`, or `cv_cureem`.
`newdata`	an optional data.frame that minimally includes the incidence and/or latency variables to use for predicting the response. If omitted, the training data are used.
`model_select`	either a case-sensitive parameter for models fit using `curegmifs` or `cureem` or any numeric step along the solution path can be selected. The default is `model_select = "AIC"` which calculates the predicted values using the coefficients from the model achieving the minimum AIC. The complete list of options are: • `"AIC"` for the minimum AIC (default). • `"mAIC"` for the minimum modified AIC. • `"cAIC"` for the minimum corrected AIC. • `"BIC"`, for the minimum BIC. • `"mBIC"` for the minimum modified BIC. • `"EBIC"` for the minimum extended BIC. • `"logLik"` for the step that maximizes the log-likelihood. • `n` where n is any numeric value from the solution path. This option has no effect for objects fit using `cv_curegmifs` or `cv_cureem`.
`...`	other arguments

Value

`p_uncured`	a vector of probabilities from the incidence portion of the fitted model representing the P(uncured).
`linear_latency`	a vector for the linear predictor from the latency portion of the model.
`latency_risk`	a dichotomous class representing low (below the median) versus high risk for the latency portion of the model.

See Also

`curegmifs`, `cureem`, `coef.mixturecure`, `summary.mixturecure`, `plot.mixturecure`

Examples

```
library(survival)
withr::local_seed(1234)
temp <- generate_cure_data(n = 100, j = 10, n_true = 10, a = 1.8)
training <- temp$training
fit <- curegmifs(Surv(Time, Censor) ~ .,
  data = training, x_latency = training,
  model = "weibull", thresh = 1e-4, maxit = 2000,
  epsilon = 0.01, verbose = FALSE
)
predict_train <- predict(fit)
names(predict_train)
testing <- temp$testing
predict_test <- predict(fit, newdata = testing)
```

`print.mixturecure` *Print the contents of a mixture cure fitted object*

Description

This function prints the first several incidence and latency coefficients, the rate (when fitting an exponential or Weibull mixture cure model), and alpha (when fitting a Weibull mixture cure model). This function returns the fitted object invisible to the user.

Usage

```
## S3 method for class 'mixturecure'
print(x, max = 6, ...)
```

Arguments

`x`	a `mixturecure` object resulting from `curegmifs`, `cureem`, `cv_cureem`, or `cv_curegmifs`.
`max`	maximum number of rows in a matrix or elements in a vector to display
`...`	other arguments.

Value

prints coefficient estimates for the incidence portion of the model and if included, prints the coefficient estimates for the latency portion of the model. Also prints rate for exponential and Weibull models and scale (alpha) for the Weibull mixture cure model. Returns all objects fit using `cureem`, `curegmifs`, `cv_cureem`, or `cv_curegmifs`.

Note

The contents of a `mixturecure` fitted object differ depending upon whether the EM (`cureem`) or GMIFS (`curegmifs`) algorithm is used for model fitting or if cross-validation is used. Also, the output differs depending upon whether `x_latency` is specified in the model (i.e., variables are included in the latency portion of the model fit) or only `terms` on the right hand side of the equation are included (i.e., variables are included in the incidence portion of the model).

See Also

`curegmifs`, `cureem`, `coef.mixturecure`, `summary.mixturecure`, `plot.mixturecure`, `predict.mixturecure`

Examples

```
library(survival)
withr::local_seed(1234)
temp <- generate_cure_data(n = 100, j = 10, n_true = 10, a = 1.8)
training <- temp$training
fit <- curegmifs(Surv(Time, Censor) ~ .,
  data = training, x_latency = training,
  model = "weibull", thresh = 1e-4, maxit = 2000,
```

```
  epsilon = 0.01, verbose = FALSE
)
print(fit)
```

`sufficient_fu_test` *Test for sufficient follow-up*

Description

Tests for sufficient follow-up using a Kaplan-Meier fitted object.

Usage

```
sufficient_fu_test(object)
```

Arguments

`object`	a `survfit` object.

Value

`p_value`	p-value from testing the null hypothesis that there was not sufficient follow-up against the alternative that there was sufficient follow-up
`n_n`	total number of events that occurred at time > pmax(0, 2*(last observed event time)-(last observed time)) and < the last observed event time
`N`	number of observations in the dataset

References

Maller, R. A. and Zhou, X. (1996) *Survival Analysis with Long-Term Survivors*. John Wiley & Sons.

See Also

`survfit`, `cure_estimate`, `nonzerocure_test`

Examples

```
library(survival)
withr::local_seed(1234)
temp <- generate_cure_data(n = 100, j = 10, n_true = 10, a = 1.8)
training <- temp$training
km_fit <- survfit(Surv(Time, Censor) ~ 1, data = training)
sufficient_fu_test(km_fit)
```

`summary.mixturecure` *Summarize a fitted mixture cure object*

Description

`summary` method for a mixturecure object fit using `curegmifs`, `cureem`, `cv_curegmifs`, or `cv_cureem`.

Usage

```
## S3 method for class 'mixturecure'
summary(object, ...)
```

Arguments

`object`	a `mixturecure` object resulting from `curegmifs`, `cureem`, `cv_curegmifs`, or `cv_cureem`.
`...`	other arguments.

Value

prints the number of non-zero coefficients from the incidence and latency portions of the fitted mixture cure model when using the minimum AIC to select the final model. When fitting a model using `curegmifs` or `cureem` the summary function additionally prints results associated with the following model selection methods: the step and value that maximizes the log-likelihood; the step and value that minimizes the AIC, modified AIC (mAIC), corrected AIC (cAIC), BIC, modified BIC (mBIC), and extended BIC (EBIC). This information can be used to guide the user in the selection of a final model from the solution path.

See Also

`curegmifs`, `cureem`, `coef.mixturecure`, `plot.mixturecure`, `predict.mixturecure`

Examples

```
library(survival)
withr::local_seed(1234)
temp <- generate_cure_data(n = 100, j = 10, n_true = 10, a = 1.8)
training <- temp$training
fit <- curegmifs(Surv(Time, Censor) ~ .,
  data = training, x_latency = training,
  model = "weibull", thresh = 1e-4, maxit = 2000,
  epsilon = 0.01, verbose = FALSE
)
summary(fit)
```

Index