

# Cinderella User's Manual

P. Reegen

Institut für Astronomie, Türkenschanzstraße 17, 1180 Vienna, Austria

## Abstract

CINDERELLA is a software solution for the quantitative comparison of time series in the frequency domain. It assigns probabilities to coincident peaks in the DFT amplitude spectra of the datasets under consideration. Two different modes are available. In conditional mode, CINDERELLA examines target and comparison datasets on the assumption that the latter contain artifacts only, returning the conditional probability of a target signal, although there is a coincident signal in the comparison data within the frequency resolution. In composed mode, the probability of coincident signal components in both target and comparison data is evaluated. CINDERELLA permits to examine multiple target and comparison datasets at once.

## 1. What is CINDERELLA?

CINDERELLA is an abbreviation of “**C**omparison of **INDE**pendent **REL**ative **L**east-squares **A**mplitudes”. It provides a quantitative comparison between the DFT amplitude spectra of time-resolved astronomical measurements.

The SIGSPEC technique (Reegen 2005, 2007) allows to determine probabilities for coincident peaks in the DFT amplitude spectra of different datasets quantitatively and in a statistically unbiased way. The theoretical background of this procedure is introduced by Reegen et al. (2008).

CINDERELLA uses the standard output of the program SIGSPEC, which represents the results of a cascade of consecutive prewhitenings employing least-squares fits to obtain amplitudes and phases of signal components. Following the SIGSPEC terminology, CINDERELLA returns a spectral significances (hereafter abbreviated by ‘sig’) rather than a probability. This manual uses cross-references to SIGSPEC frequently. In these cases, the reader is referred to the SIGSPEC manual (Reegen 2011).

## 2. Projects

CINDERELLA is called by the command line

```
Cinderella <project>
```

where <project> is the name (or path, if desired) of the CINDERELLA project. The project structure is strictly consistent with SIGSPEC.

Before running the program, the user has to provide

1. at least two time series input files consistent with the SIGSPEC MultiFile mode (see SIGSPEC manual, p. 56, and "Time series input files", p. 100), and
2. a directory <project> containing the SIGSPEC result files corresponding to the time series input files (see "SIGSPEC result files", p. 101).

Furthermore, the user may pass a set of specifications to CINDERELLA by means of a file <project>.cnd (see "The .cnd file", p. 105). This file is expected in the same folder as the time series input files and the project directory. For specifications not given by the user, defaults are used.

CINDERELLA is designed to answer two different questions, depending on the problem it is applied to. By default, the program runs through both modes simultaneously and provides both conditional (p. 108) and composed sigs (p. 114).

## 3. Input

### 3.1. Time series input files

CINDERELLA expects at least two time series input files named according to #index#. <project>.dat. In this context, #index# is a placeholder for a six-digit index of the file. Note that only files with consecutive indices are appropriately recognised by CINDERELLA. The conventions are compatible with the SIGSPEC MultiFile mode (see SIGSPEC manual, p. 56), whence the most convenient preparation of data for CINDERELLA is the SIGSPEC MultiFile computation.

The only restrictions to the format of the time series input files are that the number of items per row has to be constant for all rows in the file and that the columns have to be separated by at least one whitespace character or tab. Dataset entries need not to be numeric, except for the columns specified as time, observable, and weights. The conventions for specifying these three column types are consistent with SIGSPEC. See SIGSPEC manual, p. 13 for details.

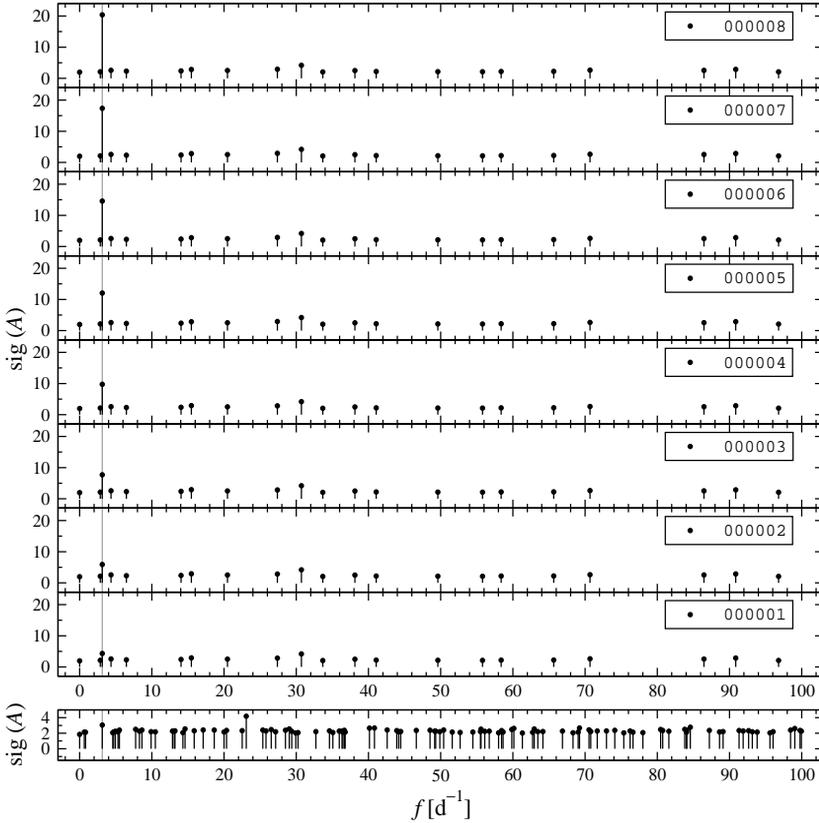


Figure 1: SIGSPEC result files used as input for the sample project CinderellaNative. The bottom panel refers to the file CinderellaNative/000000.result.dat, which is used as the comparison dataset. Above, the files CinderellaNative/000001.result.dat to CinderellaNative/000008.result.dat are displayed from bottom to top. The underlying time series represent Gaussian noise plus a single sinusoidal signal at a frequency of  $3.125 \text{ d}^{-1}$  (grey line) with different amplitudes.

### 3.2. SIGSPEC result files

The SIGSPEC result files `#index#.result.dat` are located in the project directory and contain a list of all sig maxima associated to each of the input time series `#index#.<project>.dat`. A SIGSPEC result file consists of seven columns. A full description is provided by the SIGSPEC manual, p.28. CINDERELLA uses only five columns:

- the frequency [inverse time units] (column 1),
- the sig (column 2),
- the amplitude [units of observable] (column 3),
- rms scatter of the time series before prewhitening (column 5), and
- point-to-point scatter of the time series before prewhitening (column 6).

The last line of a result file contains only two non-zero values in columns 5 and 6. These represent the rms and point-to-point scatter of the time series after the last prewhitening step, correspondingly, and are also used by CINDERELLA.

**Example.** *The sample project CinderellaNative provides a run without any additional options by typing Cinderella CinderellaNative. The files 000000.CinderellaNative.dat, ..., 000008.CinderellaNative.dat are the same as 000038.diffsig.dat to 000046.diffsig.dat in the SIGSPEC example diffsig (SIGSPEC manual, p. 60), correspondingly.*

*The SIGSPEC result files 000000.result.dat to 000008.result.dat are provided in the project directory CinderellaNative. They contain all signal components found with sig > 2 and are displayed in Fig. 1.*

*The screen output at runtime starts with a standard header.*

```

CCCC ii          dd          11 11
CC  CC          dd          11 11
CC  ii n nnn    dddd  eeee  r rrr  eeee 11 11  aaaa
CC  ii nn  nn dd  dd ee  ee rr  rr ee ee 11 11  aa  aa
CC  ii nn  nn dd  dd ee  ee rr  ee ee 11 11  aa
CC  ii nn  nn dd  dd eeeee  rr  eeeee 11 11  aaaaa
CC  ii nn  nn dd  dd ee  rr  ee  11 11  aa  aa
CC  CC ii nn  nn dd  dd ee  ee rr  ee ee 11 11  aa  aa
CCCCC ii nn  nn ddd d  eeee  rr  eeee  11 11  aaa a

```

Comparison of INDEpendent RELative Least-squares Amplitudes  
Version 1.0

```

*****
by Piet Reegen
Institute of Astronomy
University of Vienna
Tuerkenschanzstrasse 17
1180 Vienna, Austria
Release date: April 29, 2008

```

*The program starts with processing the command line, checking if a present directory CinderellaNative is present, and searching for a configuration file CinderellaNative.cnd (see "The .cnd file", p. 105). Since there is no such file present, CINDERELLA produces a warning message.*

```
*** start *****
```

```
Checking availability of project directory CinderellaNative...
project directory CinderellaNative ok.
```

```
Warning: CndFile_LoadCnd 001
Failed to open .cnd file.
```

*Now CINDERELLA counts the time series input files and checks for corresponding SIGSPEC result files.*

```
*** MultiFile count *****
```

```
Number of files          9
```

*The next step is to count the rows in each SIGSPEC result file.*

```
*** count rows in SigSpec result files *****
```

```
CinderellaNative/000000.result.dat: 106 rows
CinderellaNative/000001.result.dat: 21 rows
CinderellaNative/000002.result.dat: 21 rows
CinderellaNative/000003.result.dat: 21 rows
CinderellaNative/000004.result.dat: 21 rows
CinderellaNative/000005.result.dat: 21 rows
CinderellaNative/000006.result.dat: 21 rows
CinderellaNative/000007.result.dat: 21 rows
CinderellaNative/000008.result.dat: 21 rows
```

*Before reading the input files, CINDERELLA performs the assignment of target, comp and skip tags to the datasets (see "Dataset Types", p.106). Since there is no file CinderellaNative.cnd available, the defaults are used: the first file, 000000.CinderellaNative.dat is considered a comparison dataset, the eight remaining files are interpreted as targets.*

```
*** dataset type assignment *****
```

```
Warning: CndFile_Cind 001
Failed to open .cnd file.
Assigning default types.
```

```
000000.CinderellaNative.dat: comparison (default)
000001.CinderellaNative.dat: target (default)
000002.CinderellaNative.dat: target (default)
000003.CinderellaNative.dat: target (default)
000004.CinderellaNative.dat: target (default)
000005.CinderellaNative.dat: target (default)
000006.CinderellaNative.dat: target (default)
000007.CinderellaNative.dat: target (default)
000008.CinderellaNative.dat: target (default)
```

```
number of target datasets      8
number of comparison datasets  1
number of datasets to skip     0
```

*The time series and SIGSPEC result files are read, and CINDERELLA displays the frequency resolution and the mean observable for each time series.*

```
*** read input files *****
000000.CinderellaNative.dat: Rayleigh resolution 0.1089880382935977
000000.CinderellaNative.dat: mean observable 0.0074422789800987
CinderellaNative/000000.result.dat
000001.CinderellaNative.dat: Rayleigh resolution 0.0914470160931467
000001.CinderellaNative.dat: mean observable -0.0704869463068650
CinderellaNative/000001.result.dat
000002.CinderellaNative.dat: Rayleigh resolution 0.0914470160931467
000002.CinderellaNative.dat: mean observable -0.0875484339175066
CinderellaNative/000002.result.dat
000003.CinderellaNative.dat: Rayleigh resolution 0.0914470160931467
000003.CinderellaNative.dat: mean observable -0.1046099215281657
CinderellaNative/000003.result.dat
000004.CinderellaNative.dat: Rayleigh resolution 0.0914470160931467
000004.CinderellaNative.dat: mean observable -0.1216714091388107
CinderellaNative/000004.result.dat
000005.CinderellaNative.dat: Rayleigh resolution 0.0914470160931467
000005.CinderellaNative.dat: mean observable -0.1387328967494604
CinderellaNative/000005.result.dat
000006.CinderellaNative.dat: Rayleigh resolution 0.0914470160931467
000006.CinderellaNative.dat: mean observable -0.1557943843601256
CinderellaNative/000006.result.dat
000007.CinderellaNative.dat: Rayleigh resolution 0.0914470160931467
000007.CinderellaNative.dat: mean observable -0.1728558719707690
CinderellaNative/000007.result.dat
000008.CinderellaNative.dat: Rayleigh resolution 0.0914470160931467
000008.CinderellaNative.dat: mean observable -0.1899173595814251
CinderellaNative/000008.result.dat
```

*The amplitudes in the SIGSPEC result file of the comparison dataset (file CinderellaNative/000000.result.dat have to be transformed in order to be comparable to the target datasets (see “Amplitude transformation”, p. 112). CINDERELLA uses the rms residual as a measure for this transformation. This is the default setting.*

```
*** amplitude transformation *****
by rms error: file 0
```

*The core of CINDERELLA consists of three different analyses. The first part of these is the computation of conditional sigs for each pair of target vs. comparison datasets. In this case we have eight target datasets and one comparison dataset, which results in eight operations. In general, the total number of operations performed in this step is the product of the number of target datasets times the number of comparison datasets. Detailed information on the computation of pairwise conditional sigs is found in “Single-comparison output” (p. 109).*

```

*** pairwise Cinderella analysis *****
    1 vs.    0: conditional sig
    2 vs.    0: conditional sig
    3 vs.    0: conditional sig
    4 vs.    0: conditional sig
    5 vs.    0: conditional sig
    6 vs.    0: conditional sig
    7 vs.    0: conditional sig
    8 vs.    0: conditional sig

```

*The second part of the CINDERELLA analysis is a computation of mean conditional sigs for each target dataset over all comparison datasets (see “Multi-comparison output”, p. 109). Since there is only one comparison dataset available in this example, the output of this procedure is the same as of the previous pairwise analysis. Finally, CINDERELLA evaluates composed sigs for each target dataset. This calculation is applied to the “raw” SIGSPEC results and also to the conditional sigs obtained by the previous operation. Details on the computation of composed sigs are found in “Composed Mode” (p. 114).*

```

*** total Cinderella analysis *****
    1: conditional sig
    2: conditional sig
    3: conditional sig
    4: conditional sig
    5: conditional sig
    6: conditional sig
    7: conditional sig
    8: conditional sig
composed sig
composed sig of conditional sigs

```

*On exit, CINDERELLA displays a goodbye message.*

Finished.

```

*****
Thank you for using Cinderella!
Questions or comments?
Please contact Piet Reegen (reegen@astro.univie.ac.at)
Bye!

```

*The CINDERELLA output for this example is discussed in the subsequent chapters.*

### 3.3. The .cnd file

An optional file <project>.cnd consists of a set of keywords and arguments defining project-specific parameters for CINDERELLA. If this file is not present in the same folder as the time series input files, CINDERELLA uses a set of default parameters.

**Caution:** CINDERELLA demands a carriage-return character at the end of the file <project>.cnd, otherwise the program hangs!

Lines in the .cnd file starting with a # character are ignored by CINDERELLA. This provides the possibility to write comments into the file.

## 4. Indexing

By default, CINDERELLA expects the file index of time series and SIGSPEC results to start at zero. If the user wants the program to start at a different index, the keyword `mfstart` may be given in the .cnd file. This keyword is followed by an integer representing the desired start index. Furthermore, the software takes into account as many files with consecutive indices as available. The number of indices to process may be restricted by means of the keyword `multifile`, followed by an integer representing the last index to take into account. The use of the keywords `mfstart` and `multifile` is strictly consistent with the SIGSPEC conventions (SIGSPEC manual, p. 56).

**Example.** *The sample project index contains the same input as the project CinderellaNative (p.102), but both the time series input files and the SIGSPEC result files are now enumerated from 004847 to 004855 rather than from 000000 to 000008. The file index.cnd consists of a single line,*

```
mfstart 4847
```

*Since the keyword multifile is not specified in the file index.cnd, CINDERELLA uses all input files available through consecutive indices.*

*The computations are entirely the same as for CinderellaNative, but all indices are consistently incorporated by CINDERELLA. E. g., the screen output for the single-comparison computations is now:*

```
*** pairwise Cinderella analysis *****
4848 vs. 4847: conditional sig
4849 vs. 4847: conditional sig
4850 vs. 4847: conditional sig
4851 vs. 4847: conditional sig
4852 vs. 4847: conditional sig
4853 vs. 4847: conditional sig
4854 vs. 4847: conditional sig
4855 vs. 4847: conditional sig
```

## 5. Dataset Types

In order to avoid unnecessarily high computational effort and redundant output, if comparing all possible pairs of time series input files, CINDERELLA provides

the possibility to specify which pairs of target/comparison datasets to take into account. Moreover, the user has the opportunity to identify datasets to be ignored.

CINDERELLA will produce one so-called single-comparison output file (see “Single-comparison output”, p. 109) for each target-comparison pair. If there is more than one comparison dataset available, additional files are generated for each target. They contain summaries concerning all comparison datasets examined for the target (see “Multi-comparison output”, p. 109) and “Output for composed mode”, p. 114).

Contrary to the file nomenclature, the six-digit format is not required for file indices specified in the `.cnd` file.

### 5.1. Target datasets

The keyword `target` in the `.cnd` file is used for the specification of a target dataset. The keyword is followed by an integer value referring to the six-digit index of the desired time series input file. Multiple declarations of `target` are supported. If no `.cnd` file is available, CINDERELLA uses the first time series input file (i. e. the one with the start index) as the only target dataset.

### 5.2. Comparison datasets

The keyword `comp` in the `.cnd` file is used for the specification of a comparison dataset. The keyword is followed by an integer value referring to the six-digit index of the desired time series input file. Contrary to the file nomenclature, the six-digit format is not required for file indices specified in the `.cnd` file. If no `.cnd` file is available, CINDERELLA uses all time series input files – except for the first one, which is considered target data – as comparison datasets.

### 5.3. Datasets to ignore

The keyword `skip` in the `.cnd` file is used for the specification of a dataset not to be taken into account for computation. The keyword is followed by an integer value referring to the six-digit index of the desired time series input file. Contrary to the file nomenclature, the six-digit format is not required for file indices specified in the `.cnd` file.

### 5.4. Default type

To enhance the convenience for the user, not all files need to be specified by the keywords `target`, `comp` and `skip`. The keyword `deftype` may be used to assign a default dataset type.

1. Use `deftype target` to assign the target attribute by default. If no `deftype` keyword is provided, this setting is activated.
2. Use `deftype comp` to assign the comp attribute by default.
3. Use `deftype skip` to assign the skip attribute by default.

**Example.** *The sample project types contains the same input as the project CinderellaNative (p. 102), and the file types.cnd contains the two lines*

```
deftype target
comp 0
```

*This reproduces the default assignment of data types, and CINDERELLA performs the same calculations as for the project CinderellaNative. The only difference is the screen output:*

```
*** dataset type assignment *****
000000.types.dat: comparison
000001.types.dat: target
000002.types.dat: target
000003.types.dat: target
000004.types.dat: target
000005.types.dat: target
000006.types.dat: target
000007.types.dat: target
000008.types.dat: target
```

*The fact that (default) is not attached to the file list indicates that CINDERELLA uses the specifications given in the file types.cnd rather than the standard settings applied to the project CinderellaNative.*

## 6. Conditional Mode

The conditional sig is a measure of the probability that a signal component in a target star occurs, although a coincident signal component is found in a comparison star or sky background. It provides an answer to the question, "What is the probability that a signal component with given amplitude and sig in the target data is not due to the same process that produces a coincident signal component with given amplitude and sig in the comparison data?"

The conditional CINDERELLA mode is comparable to the differential sig (SIGSPEC manual, p. 58), although the numerical results are not the same.

- For the differential mode of SIGSPEC, the full spectral information is available. Thus SIGSPEC handles the DFT spectra as continuous functions. CINDERELLA accesses only a list of peaks detected by SIGSPEC.

Deviations of corresponding peak frequencies in comparison and target spectra cannot be handled as accurately as in the case of differential sig computation.

- The differential mode of SIGSPEC compares power integrals over the entire frequency range under consideration for the transformation of amplitudes from comparison into target data. Since CINDERELLA deals with a list of frequencies rather than the entire spectra, different strategies to transform amplitudes have to be employed. See “Amplitude transformation”, p. 112.

#### Single-comparison output

The single-comparison output files contain three columns:

1. target frequency [inverse time units],
2. conditional sig,
3. conditional csig.

Each file refers to the analysis of a single target-comparison dataset pair. Correspondingly, two six-digit indices `#target#`, `#comparison#` are used to form the filenames, `#target#.cd.#comparison#.dat` (conditional mode) and `#target#.cd.#comparison#.dat` (composed mode).

**Example.** *The output file 000004.cd.000026.dat contains conditional sigs for 000004.<project>.dat as target data and 000026.<project>.dat as comparison data.*

**Example.** *In the sample CinderellaNative, there are 8 single-comparison output files: 000001.cd.000000.dat to 000008.cd.000000.dat. Since there is only one comparison dataset available, these files are redundant, because #target#.cd.dat = #target#.cd.000000.dat.*

#### Multi-comparison output

For each target dataset, a multi-comparison output file `#target#.cd.dat` is generated for each target dataset. The three columns represent

1. target frequency [inverse time units],
2. mean conditional sig for all comparison datasets,
3. mean conditional csig for all comparison datasets.

If there is only one comparison dataset available, the single-comparison and multi-comparison output files are identical.

**Example.** *The sample CinderellaNative contains 8 multi-comparison output files in the project directory: 000001.cd.dat to 000008.cd.dat.*

### 6.1. Candidate selection

For each target dataset, CINDERELLA scans all comparison datasets, searching for coincident signal components. A pair of signal components is considered coincident, if the frequencies match to an accuracy that may be specified by the user, who may also define what CINDERELLA shall do if a comparison dataset does not contain a match for a given target frequency.

If more than one coincident frequency associated to a given target frequency is found, CINDERELLA chooses the candidate with the lowest conditional or composed sig in order to obtain the most conservative solution.

#### Frequency resolution

There are mainly two interpretations of the frequency resolution. It is either calculated as the inverse time interval width (Rayleigh frequency resolution),

$$\delta f := \frac{1}{T}, \quad (1)$$

or as

$$\delta f := \frac{1}{T \sqrt{\text{sig}(A)}}, \quad (2)$$

where  $\text{sig}(A)$  denotes the sig of an amplitude  $A$ . This definition is called Kallinger resolution (Kallinger, Reegen & Weiss 2008).

In order to enhance the flexibility of CINDERELLA, the frequency resolution is computed as

$$\delta f := \frac{1}{T \sqrt{\text{sig}(A)^\tau}}, \quad (3)$$

where the floating-point number  $\tau \in [0, 1]$  may be defined using the keyword `tol` in the `.cnd` file. The special value  $\tau = 0$  transforms eq. 3 into eq. 1, setting  $\tau = 1$  provides eq. 2. The default value is  $\tau = 0$ .

CINDERELLA checks for frequencies in the comparison datasets are within the frequency resolution around each frequency in the target dataset.

**Example.** *The sample cand contains the same input as CinderellaNative (p. 102), and the file cand.cnd contains the line*

tol 2

*The frequency tolerance parameter is increased compared to the default value 0, which means that the intervals taken into account to search for corresponding signal components are tendentially narrower. This setting is for demonstration only; in normal applications, only frequency tolerance parameters ranging from 0 to 1 will make sense.*

*The effect of this modification is visible, e.g., comparing the output files 000001.cd.000000.dat of the project CinderellaNative to the project cand. In the project CinderellaNative, this file contains the line*

```
58.3815412948909298 -8.8063413553403507 -5.6290446018702553
```

*whereas the corresponding line in the project cand is*

```
58.3815412948909298 2.1858505500938747 0.3972034177439077
```

*In the project CinderellaNative, the 14th component in the SIGSPEC result file 000001.result.dat in the project directory is related to the 47th component in the file 000000.result.dat. The two frequencies differ by 0.054, and the Rayleigh frequency resolution of the target dataset is 0.091, which is sufficient for a correspondence. In the project cand, the target sig of 2.387 becomes relevant. Eq.3 yields a frequency resolution 0.042 for this component, which is now too small for a coincidence. The corresponding line in the file 000001.cd.000000.dat consistently indicates that no coincident peak is found for this signal component. In this case, CINDERELLA uses a default sig threshold for the comparison data, see “Spectral significance threshold”, below.*

### Spectral significance threshold

If no coincidence in a comparison dataset is detected for a given target frequency, i.e., if CINDERELLA does not find a valid candidate for this target frequency, a default value is used for the sig in the comparison dataset. The user may specify this CINDERELLA threshold by means of the keyword `defsig` in the `.cnd` file. The same specification may be set for the default `csig`<sup>1</sup> using the keyword `defcsig`. If one of these keywords is not provided,  $\frac{\pi}{4} \log e \approx 0.341$  is used correspondingly by default. According to Reegen (2007), this is the expected value of the sig for white noise. The underlying assumption is that the residuals after prewhitening of all significant signal components in the comparison dataset represent pure noise, i.e. do not contain any further unresolved signal.

---

<sup>1</sup>abbreviation for cumulative sig

**Example.** *The sample cand contains the same input as CinderellaNative (p. 102), and the file cand.cnd contains the two lines*

```
defsig 0
defcsig 1
```

*The second row in the output file 000001.cd.000000.dat of the project CinderellaNative*

```
30.7091991449662061 3.8506295783390758 3.6090130812823817
```

*using the default sig and csig threshold  $\frac{\pi}{4} \log e$ , because the comparison dataset does not contain a coincident signal component. The second row in the corresponding file of the project cand is*

```
30.7091991449662061 4.1917236667995361 2.9501071697428420
```

*Since the default sig is lower in the project cand, the resulting conditional sig is higher. On the other hand, the default csig is higher, which causes the resulting conditional csig to drop down.*

## 6.2. Amplitude transformation

The assumption that instrumental and environmental artifacts use to be additive in terms of intensity may create needs to transform amplitudes in mag from the comparison into the target spectra, if the conditional CINDERELLA mode is applied. The amplitude transformation is only performed to obtain conditional sigs.

Three different strategies to adjust comparison amplitudes are offered, according to the specifications in the .cnd file.

- In photometry, the photon statistics may be employed to transform comparison into target amplitudes, if the mean magnitudes of the stars are known (Reegen et al. 2008). If the keyword `transam:mean` is provided in the .cnd file, CINDERELLA uses the mean observables  $\langle m_C \rangle$ ,  $\langle m_T \rangle$  of the comparison and target time series, respectively, to transform the comparison amplitude  $A_C$  into a target amplitude  $A_T$  according to

$$A_T = 2.5 \log \left[ 1 + \frac{10^{-0.4(\langle m_C \rangle - A_C)} - 10^{-0.4\langle m_C \rangle}}{10^{-0.4\langle m_T \rangle}} \right]. \quad (4)$$

- If the keyword `transam:rms` is specified in the .cnd file, CINDERELLA interprets the residual rms errors  $\sigma_C$ ,  $\sigma_T$  of the comparison and target

time series<sup>2</sup>, respectively, as measures of the photon noise levels and evaluates the transformed amplitude according to

$$A_T = 2.5 \log \left[ 1 + \frac{\sigma_C^2}{\sigma_T^2} (10^{0.4 A_C} - 1) \right]. \quad (5)$$

- The keyword `transam:ppsc` in the `.cnd` file causes CINDERELLA to use Eq. 5 employing residual point-to-point scatter instead of residual rms error for both  $\sigma_C$  and  $\sigma_T$ .
- If the keyword `transam:none` is specified in the `.cnd` file, no amplitude transformation is performed at all, i. e., CINDERELLA assumes  $A_T = A_C$ .

**Example.** *The sample project `transam-mean` contains the same input as the project `CinderellaNative` (p. 102). The time series data are considered to represent millimag photometry. The comparison dataset is assumed to refer to a 5 mag star, whereas the target datasets shall correspond to a 15 mag star. The resulting time series input files are `000000.transam-mean.dat` `000001.transam-mean.dat` to `000008.transam-mean.dat`. The keyword*

```
transam:mean
```

*in the file `transam-mean.cnd` forces CINDERELLA to employ the mean magnitudes of the datasets for the amplitude transformation.*

**Example.** *The sample project `CinderellaNative` contains an amplitude transformation based on the rms residual, which is the default method.*

**Example.** *The sample project `transam-ppsc` contains the same input as `CinderellaNative` (p. 102). The line*

```
transam:ppsc
```

*in the file `transam-ppsc.cnd` forces CINDERELLA to employ the residual point-to-point scatters of the datasets for the amplitude transformation.*

**Example.** *The sample project `transam-none` contains the same input as `CinderellaNative` (p. 102). The line*

```
transam:none
```

*in the file `transam-none.cnd` switches off the amplitude transformation.*

---

<sup>2</sup>with all significant signal prewhitened

## 7. Composed Mode

The composed sig is a measure of the probability that two coincident signal components occur in two different datasets. This implements a logical 'and', providing an answer to the question, "What is the probability that two different datasets show coincident signal components with given amplitudes and sigs?"

The composed mode is useful for, e. g., photometry of the same star in different filters, or if two short datasets of the same object obtained in different years are examined.

Note that the composed sig in the SIGSPEC result files (see SIGSPEC manual) is consistently defined, but applies to the set of significant signal components displayed in the file, whereas CINDERELLA refers to the composed sig of signal components found in two or more different datasets.

Contrary to the candidate selection procedure in conditional mode (p. 110), the frequency interval between the lowest and the highest frequency found in all target datasets is scanned in steps defined by half the frequency resolution (p. 110). For each of the frequencies under consideration, CINDERELLA computes a composed sig, basically following the introduction by Reegen et al. (2008). Since CINDERELLA's composed mode takes into account all signal components in all datasets, statistical weights have to be introduced that put more emphasis to signal frequencies closer to the frequency under consideration. Hence the composed sig  $\text{csig}(A_n)$  (annotation by Reegen et al. 2008) assigned to an arbitrary frequency  $f$  is evaluated according to

$$\log \left[ 1 - 10^{\text{csig}(A_n)} \right] = \frac{1}{N} \sum_{n=1}^N e^{-\frac{1}{2} \left[ \frac{f-f_n}{\min(\delta f_n)} \right]^2} \log \left[ 1 - 10^{-\text{sig}(A_n)} \right]. \quad (6)$$

In this context, the total number of signal components in all target datasets is denoted  $N$ ,  $f_n$  referring to the frequency of one of these signal components. The minimum frequency resolution  $\min(\delta f_n)$  incorporates the definition by Eq. 3.

An interpolation loop is used to exactly identify the maxima of this composed sig, which are written to the output file.

### 7.1. Output for composed mode

The calculation of the composed sig is applied to all target datasets at once. A file `cp.dat` is generated, which contains the composed sigs of all target datasets. The three columns refer to:

1. target frequency [inverse time units],
2. composed sig for all target datasets,

### 3. composed csig for all target datasets.

The composed sigs are also calculated for the conditional sigs, i. e., the files #target#.cd.dat (see “Multi-comparison output”, p.109), and written to a file cc.dat. The column format is the same as for the file cp.dat.

**Example.** *For the sample project CinderellaNative, the project directory contains a file cp.dat with the composed sigs (and csigs) for all target datasets (provided by the SIGSPEC result files 000001.result.dat to 000008.result.dat). Furthermore, a file cc.dat is found in the project directory. It contains the composed sigs (and csigs) applied to the conditional ones for all target datasets, i. e., the composed sigs are evaluated using the multi-comparison output files generated by the conditional mode, 000001.cd.dat to 000008.cd.dat.*

## 8. Keywords Reference

This section is a compilation of all keywords accepted by CINDERELLA. A brief description of arguments and default values is given. The type of argument is provided by either <int>, <double>, or <string>, and default values are given in parentheses, e. g. (2). Empty parentheses indicate that there is no default setting.

col:obs <int> (2)

observable column index (unique), starting with 1, SIGSPEC manual, p.13

col:time <int> (1)

time column index (unique), starting with 1, SIGSPEC manual, p.13

col:weights <int>

weights column index (also multiple), starting with 1, SIGSPEC manual, p.13

comp <int> (all except start index)

specification of time series input files to be regarded as comparison datasets, p.107

`defcsig <double> ( $\frac{\pi}{4} \log e \approx 0.341$ )`

threshold to be used for the csig, if no coincidence is found in a comparison dataset, p. 111

`defsig <double> ( $\frac{\pi}{4} \log e \approx 0.341$ )`

threshold to be used for the sig, if no coincidence is found in a comparison dataset, p. 111

`deftype <target/comp/skip> (target)`

specifies the type of dataset to be assigned to a time series by default, p. 107

`skip <int> ()`

specification of time series input files not to be taken into consideration, p. 107

`target <int> (start index)`

specification of time series input files to be regarded as target datasets, p. 107

`tol <double> (0)`

CINDERELLA frequency tolerance parameter, p. 110

`transam:mean`

amplitude transformation using the mean observable for photon statistics, p. 112

`transam:none`

no amplitude transformation at all, p. 113

`transam:ppsc`

amplitude transformation using the point-to-point scatter of the residual observable for photon statistics, p. 113

`transam:rms (default)`

amplitude transformation using the rms residual observable for photon statistics, p. 112

## 9. Online availability

The ANSI-C code is available online at <http://www.sigspec.org>. For further information, please contact P. Reegen, [peter.reegen@univie.ac.at](mailto:peter.reegen@univie.ac.at).\*

**Acknowledgments.** PR received financial support from the Fonds zur Förderung der wissenschaftlichen Forschung (FWF, projects P14546-PHY, P17580-N2) and the BM:BWK (project COROT). Furthermore, it is a pleasure to thank M. Gruberbauer (Univ. of Vienna), D. B. Guenther (St. Mary's Univ., Halifax), M. Hareter, D. Huber, T. Kallinger (Univ. of Vienna), R. Kuschnig (UBC, Vancouver), J. M. Matthews (UBC, Vancouver), A. F. J. Moffat (Univ. de Montreal), D. Punz (Univ. of Vienna), S. M. Rucinski (D. Dunlap Obs., Toronto), D. Sasselov (Harvard-Smithsonian Center, Cambridge, MA), L. Schneider (Univ. of Vienna), G. A. H. Walker (UBC, Vancouver), W. W. Weiss, and K. Zwintz (Univ. of Vienna) for valuable discussion and support with extensive software tests. I acknowledge the anonymous referee for a detailed examination of both this publication and the corresponding software, as well as for the constructive feedback that helped to improve the overall quality a lot. Finally, I address my very special thanks to J. D. Scargle for his valuable support.

## References

- Kallinger, T., Reegen, P., Weiss, W. W. 2008, *A&A*, 481, 571  
Reegen, P. 2005, in *The A-Star Puzzle*, Proceedings of IAUS 224, eds. J. Zverko, J. Ziznovsky, S. J. Adelman, W. W. Weiss (Cambridge: Cambridge Univ. Press), p. 791  
Reegen, P. 2007, *A&A*, 467, 1353  
Reegen, P. 2011, *CoAst* 163, 3  
Reegen, P., Gruberbauer, M., Schneider, L., Weiss, W. W. 2008, *A&A*, 484, 601  
Zwintz, K., Marconi, M., Kallinger, T., Weiss, W. W. 2004, in *The A-Star Puzzle*, Proceedings of IAUS 224, eds. J. Zverko, J. Ziznovsky, S. J. Adelman, W. W. Weiss (Cambridge: Cambridge Univ. Press), p. 353  
Zwintz, K., Weiss, W. W. 2006, *A&A*, 457, 237

---

\*Please contact Michael Gruberbauer, [mgruberbauer@ap.smu.ca](mailto:mgruberbauer@ap.smu.ca).



# Combine User's Manual

P. Reegen

Institut für Astronomie, Türkenschanzstrasse 17, 1180 Vienna, Austria

## Abstract

COMBINE is an add-on to SIGSPEC and CINDERELLA. A SIGSPEC result file or a file generated by CINDERELLA contains the significant sinusoidal signal components in a time series. In this file, COMBINE checks one frequency after the other for being a linear combination of previously examined frequencies. If this attempt fails, the corresponding frequency is considered "genuine". Only genuine frequencies are used to form linear combinations subsequently. A purely heuristic model is employed to assign a reliability to each linear combination and to justify whether to consider a frequency genuine or a linear combination.

## 1. What is COMBINE?

COMBINE performs an iterative analysis of the frequencies in a `result.dat` file generated by SIGSPEC (Reegen 2005, 2007, and this issue, p. 3 – 97) or one of the output files generated by CINDERELLA (Reegen et al. 2008; this issue, p. 99 – 117). The input file type is detected automatically.

If the attempt to interpret a given frequency as a linear combination fails, this frequency is considered genuine. Only genuine frequencies are used to form linear combinations in the subsequent iterations. The decision whether to accept a linear combination is drawn using a mathematical model to assign an equivalent spectral significance (hereafter abbreviated by 'sig') to a linear combination. This equivalent sig is compared to the sig of the given signal component, and only if it is high enough, the program adopts it.

If there is more than one linear combination available, COMBINE picks the one with the highest equivalent significance.

The underlying model leading to equivalent sigs and the reliabilities of linear combinations is purely heuristic and attempt to mimic the examination by an experienced person.

## 2. Input

COMBINE is called by the command line

```
combine <infile>
```

where <infile> is the name (or path, if desired) of a SIGSPEC result file or an output file generated by CINDERELLA.

**Caution:** COMBINE **overwrites existing output files!**

Furthermore, the user may pass a set of specifications to COMBINE by means of a file <infile>.ini in the same folder as <infile>. For specifications not given by the user, defaults are used.

**The file <infile>.ini has to be terminated by a carriage-return character, otherwise the program hangs!**

## 3. How COMBINE Works

For a peak with given frequency and significance, all possible combinations of previously detected genuine frequencies  $f_k$ ,  $k = 1, \dots, K$  are computed.  $K$  is the maximum number of frequencies in a linear combination. The resulting frequency for a linear combination is

$$f^l := \sum_{k=1}^K c_k f_k \quad (1)$$

and shall be compared to a frequency  $f$  in the input file.

### 3.1. Sig vs. csig

If the keyword csig is provided in the file <infile>.ini, the cumulative sig (Reegen 2007, 2011) is used instead of the sig. This keyword does not require any parameters.

### 3.2. Frequency resolution

The adjustment of the frequency resolution  $\delta f$  is consistent with Eq. 3 in the CINDERELLA manual (see this issue, p. 99 – 117), where the total time interval width  $T$  has to be provided by the user, because the time series is not incorporated by COMBINE. Moreover, the user is more flexible if allowed to specify a

different value for  $T$ . This interval width is provided by means of the keyword `dt` in the file `<infile>.ini`, followed by a floating-point number. The default setting is that `COMBINE` determines the closest pair of frequencies and uses its inverse frequency spacing as  $T$ .

The second parameter,  $\tau$ , is specified using the keyword `tol`, again followed by a floating-point number, in full consistency with `CINDERELLA`. The default value is  $\tau = 0$ , forcing `COMBINE` to employ the Rayleigh frequency resolution.

The frequency tolerance permits linear combinations where

$$\alpha := |f - f'| \leq \delta f \quad (2)$$

only. The quantity  $\alpha$  is the accuracy of a linear combination and provided in the output.

### 3.3. Limit of harmonic order

The range of harmonic orders is restricted by the parameter  $N$ , which is calculated according to

$$N = \text{ceil} \left( \sqrt{\Omega \frac{\text{sig}_k}{\text{sig}_K}} \right), \quad (3)$$

where  $\text{sig}_k$  denotes the `sig` associated to the frequency  $f_k$  and  $\text{sig}_K$  is the `sig` associated to the last frequency in the input file,  $f_K$ . If the keyword `csig` is set, the `csig` is consistently taken instead of the `sig`. The parameter  $\Omega$  is provided by the keyword `order` in the file `<infile>.ini`, followed by a floating-point number. The default value is 1. Given the limit  $N$ , the coefficients of a linear combinations are restricted to indices from  $-N$  to  $N$  according to

$$c_k = -N, \dots, N \quad (4)$$

### 3.4. Equivalent `sig`

Each linear combination is assigned an equivalent `sig`,

$$\text{sig}_{\text{eq}} := \min \left( |c_k|^{-\delta_k} \text{sig}_k \right) - \chi \log K, \quad (5)$$

where  $\delta_k$  denotes the decay parameter provided by the keyword `decay`, and  $\chi$  is the combination damping, specified using the keyword `cdamp`. Both keywords are followed by floating-point numbers. The default values for both parameters are 1.

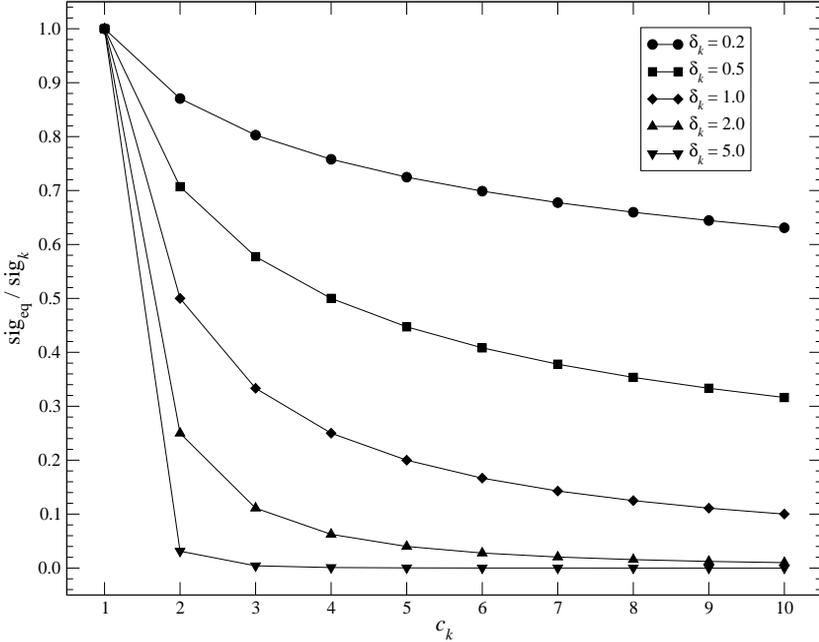


Figure 1: Ratio of equivalent sig over sig of an individual signal component vs. polynomial coefficient  $c_k$  associated to the signal component. Five graphs for different values of the decay parameter  $\delta_k$  are presented.

Fig. 1 displays the relative sig correction with increasing coefficient  $c_k$  for five different values of the decay parameter  $\delta_k$ . Fig. 2 illustrates the correction of equivalent sig with increasing number of components contributing to a linear combination  $K$  for five different values of the combination damping  $\chi$ .

### 3.5. Reliability and sensitivity

A linear combination is only accepted if the equivalent sig of the combination is high enough compared to the significance of the given peak according to

$$R := \frac{\text{sig}_{\text{eq}}}{\text{sig}_f} \geq S, \tag{6}$$

where  $S$  is the sensitivity, which can be adjusted by means of the keyword `sens` in the file `<infile>.ini`. The keyword is followed by a floating-point number, and the default value is 0.1. If all examined linear combinations have a reliability below  $S$ , the examined signal component is considered genuine.

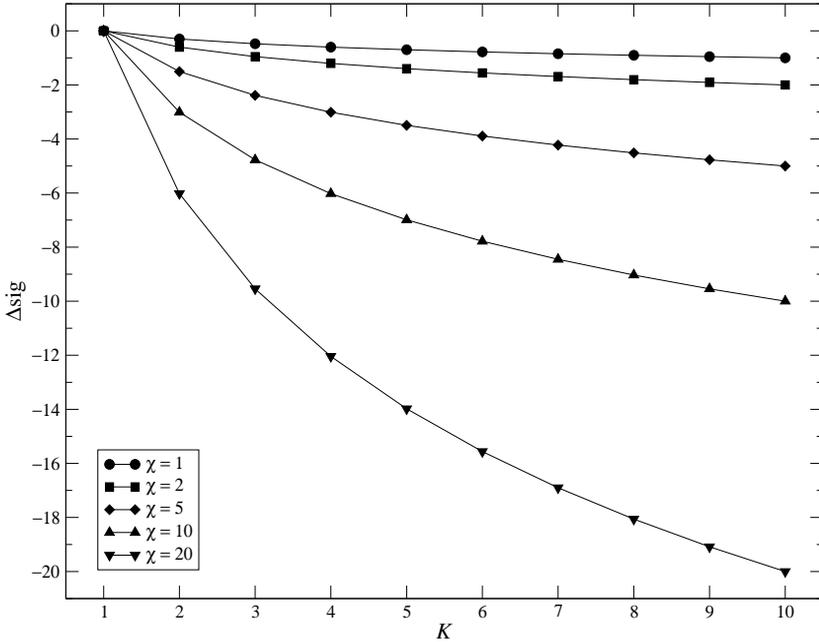


Figure 2: Additive significance correction for a linear combination employing  $K$  different signal components. Five graphs for different values of the combination damping  $\chi$  are presented.

Hence the sensitivity provided by the keyword `seas` permits to directly adjust the number of genuine components in a list of frequencies.

The ratio of sigs,  $R$ , is called the reliability of a linear combination and part of the COMBINE output. If multiple combinations are available, the reliability is used to decide which one to pick. This means, COMBINE picks the combination with the highest reliability.

#### 4. Output

Genuine frequencies are assigned identifiers `f#index#`, where `#index#` denotes an integer number starting at 1. According to the number of significant signals present in the file `<infile>`, COMBINE chooses a constant number of digits. For example, if the input file contains from 1 to 9 frequencies, the identifiers for genuine frequencies are `f1`, `f2`, ... If the input file contains from 10 to 99 frequencies, COMBINE enumerates the genuine components `f01`, `f02`, ..., and

so on. This format convention applies to the indexing of rows also.

Linear combinations are denoted by the frequency identifiers of the genuine components and appear as a formula: if the frequency under consideration is, e.g.,  $f_1 + 3f_3 - 2f_{10} - f_{14} - 0.00214$ , COMBINE displays it as `f01+3f02-2f10-f14-0.00214` both on the screen and in the output file. In this context, `-0.00214` is the frequency accuracy.

The screen output consists of a single line for each signal (i.e., for each row in the input file). COMBINE displays

1. the row index,
2. the linear combination including the frequency accuracy, and
3. the reliability  $R$  (Eq. 6).

For genuine frequencies, COMBINE displays only the row index and the frequency identifier. At runtime, the most reliable linear combination identified so far is displayed. If COMBINE finds a “better” solution, the line on the screen is updated.

By default, COMBINE generates an output file `<infile>.cmb`. It contains a row index in the first column, then all information of the input file in the further columns, plus three additional columns at the end:

1. reliability  $R$  (Eq. 6)<sup>1</sup>,
2. total number of linear combinations within the frequency resolution,
3. the linear combination itself, plus the frequency accuracy. If a frequency is considered genuine, only the frequency identifier is displayed.

For convenience, a second output file `<infile>.gen` is produced by COMBINE. It is truncated to the genuine frequencies only and contains the row index in the first column, then all the information provided in the input file, plus the frequency identifier in the last column. The columns for the reliability and the number of linear combinations within the frequency resolution are omitted. This file provides the opportunity to have all the genuine frequencies available at a glance.

**Example.**<sup>2</sup> *The sample project `CombineNative` contains a list of significant frequencies found in the MOST<sup>3</sup> (Microvariability & Oscillations of STars) pho-*

<sup>1</sup>Zero values indicate genuine frequencies

<sup>2</sup>The computation of the sample project `CombineNative` takes 40 minutes on an Intel Core2 CPU T5500 (1.66GHz) under Linux 2.6.18.8-0.9-default i686.

<sup>3</sup>MOST is a Canadian Space Agency mission, jointly operated by Dynacon Inc., the University of Toronto Institute of Aerospace Studies, the University of British Columbia, and with the assistance of the University of Vienna, Austria.

tometry of  $\zeta$  Oph (Walker et al. 2003, 2004, 2005). According to the input file `result.dat`, altogether 294 formally significant signal components ( $\text{sig} > 5$ ) were identified.

The file `result.dat.ini` contains five keywords:

```
order 0.2
dt 26
decay 1.5
cdamp 10
sens 0.2
```

The dataset is 26 days long, and the frequencies are provided in cycles per day. Thus COMBINE will assume a Rayleigh frequency resolution of 0.03846 cycles per day. There is no specification for the frequency tolerance parameter (keyword `tol`). Thus the default setting 0 is used.

Running COMBINE by typing the command line `Combine result.dat` yields a welcome message on the screen.

```
CCCCC          bb    ii
CC   CC          bb
CC      ooooo  m mm mm  bb bbb  ii n nnnn  eeeee
CC      oo   oo mm mm mm  bbb  bb ii nn  nn ee  ee
CC      oo   oo mm mm mm  bb   bb ii nn  nn ee  ee
CC      oo   oo mm mm mm  bb   bb ii nn  nn eeeeeee
CC      oo   oo mm mm mm  bb   bb ii nn  nn ee
CC   CC oo   oo mm mm mm  bb   bb ii nn  nn ee  ee
CCCCC  ooooo  mm mm mm  b bbbb  ii nn  nn  eeeee
```

Version 1.0

```
*****
by Piet Reegen
Institute of Astronomy
University of Vienna
Tuerkenschanzstrasse 17
1180 Vienna, Austria
Release date: August 18, 2009
```

The program finds out that the input file is a seven-column SIGSPEC result file, determines the number of rows and reads the input data. Note that 295 rows correspond to 294 significant signal components, because the last row in the SIGSPEC result file contains information on the residuals (see SIGSPEC manual, p. 28).

```
*** start *****
```

```
File result.dat: SigSpec format
rows                295
read input file
```

Then the search for linear combinations starts. For each row in the input file, COMBINE displays the most reliable combination detected so far.

*The first four signal components are found to be genuine. Since the number of signal components is 294, COMBINE uses a three-digit format for the row indices and frequency identifiers.*

```
row 001: f001
row 002: f002
row 003: f003
row 004: f004
```

*For rows 5 and 6 in the input data, the screen output contains the most reliable linear combination (including the frequency accuracy) and the reliability.*

```
row 005: 3f001-f002-2f003-f004+0.0284306 0.236585
row 006: 3f001+2f002-f004+0.0136421 0.35803
```

*An examination of the output file result.dat.cmb shows that rows 005 and 006 end with*

```
0.2365853347754522 1 3f001-f002-2f003-f004+0.0284306168856169
0.3580304203945811 2 3f001+2f002-f004+0.0136420746028509
```

*These entries refer to the columns added by COMBINE. The first value is the reliability, the second one is the number of examined linear combinations, and the last column represents the linear combination itself. For row 005, there is only one linear combination available within the frequency resolution, for row 006 the number of linear combinations taken into account is 2.*

*Subsequently, the screen output indicates a fifth genuine frequency.*

```
row 007: f005
```

*The frequency in row number 8 is 0.02783 cycles per day, which is below the frequency resolution. Thus the component is considered to refer to zero frequency, and in this case, no reliability is evaluated.*

```
row 008: 0+0.0278395
```

*In the further rows of the input files, no more genuine frequencies are detected.*

```
row 009: -f002+f005-0.025485 0.759005
row 010: f001-f002-f004+f005+0.0313392 0.490535
row 011: -f001+f004-0.00275538 1.26888
row 012: f001-f002-f004+f005-0.0295542 0.680494
row 013: -2f001+2f003+f004-0.00567519 0.523911
row 014: -f001+f005+0.024731 1.72772
row 015: 2f002+0.0249392 1.47442
row 016: 2f001-f004-0.0100088 1.70761
row 017: -f001+2f002-0.00217389 1.55951
row 018: f001-f002+0.00824894 3.95466
row 019: f002+f005-0.00668728 1.64167
row 020: 2f002+f003-f005-0.00199182 0.779607
```

*It is a remarkable matter of fact that COMBINE is able to compose all 294 frequencies contained by the input file as linear combinations of no more than five genuine frequencies. However, a different parameter constellation in the configuration file `result.dat.ini` can produce completely different output. Note that the time consumption by COMBINE dramatically increases with the number of genuine frequencies identified. This is because more genuine frequencies increase the number of possible linear combinations over-proportionally. A list of genuine frequencies only is found in the output file `result.dat.gen`.*

5 genuine frequencies found.

Finished.

\*\*\*\*\*

Thank you for using Combine!  
 Questions or comments?  
 Please contact Piet Reegen (reegen@astro.univie.ac.at)  
 Bye!

## 5. Order of Input Rows

Since COMBINE processes the input file row by row, the order of rows plays a crucial part in the way the analysis is performed. Changing the order of rows in the input file influences the base upon which the linear combinations are formed. Thus, if there are frequencies previously known to be genuine, it is advisable to ensure that they are on top of the input file, if all further frequencies are supposed to be checked for linear combinations of preferably these components.

**Example.**<sup>4</sup> *The input of the sample project order is essentially the same as for CombineNative. Only the order of rows is slightly modified: the 6th signal component of the file `result.dat` in the project CombineNative, which refers to the orbit frequency of the MOST spacecraft, appears now on top. This re-ordering forces COMBINE to consider  $14.188 \text{ d}^{-1}$  genuine. Also the configuration file `result.dat.ini` is the same as for the project CombineNative.*

*Again, there is a base of five genuine frequencies three of which are identical to the project CombineNative, namely 5.182, 2.675 and 3.055 cycles per day. The two genuine signal components at 6.722 and 7.193 cycles per day are replaced by 14.188 and 0.0697 cycles per day.*

---

<sup>4</sup>The computation of the sample project order takes 40 minutes on an Intel Core2 CPU T5500 (1.66GHz) under Linux 2.6.18.8-0.9-default i686.

## 6. Rejecting Unwanted Linear Combinations

Moreover, the user may indicate unwanted signal components in the input file <infile> by applying a minus sign to the corresponding frequencies. COMBINE reacts with a corresponding change of the sign for the reliability. If the user additionally provides the keyword `reject` in the file <infile>.ini, all rows are rejected from the output file <infile>.cmb for which the most reliable linear combination contains one or more unwanted frequencies.

The screen output contains linear combinations incorporating unwanted frequencies at runtime. To indicate such unwanted combinations, the reliability is displayed as a negative value. If the examination of an input line finishes with the “best” linear combination containing an unwanted frequency, the corresponding line is removed from the screen output.

**Example.**<sup>5</sup> *The input of the sample project reject is the same as for order, with a minus sign for the first frequency of 14.188 cycles per day, which represents the orbit of the MOST spacecraft. The file result.dat.ini contains an additional line,*

```
reject
```

*The combination of this keyword and the negative sign for the first signal component in the input file forces COMBINE to reject all linear combinations incorporating the frequency 14.188 cycles per day from the output file result.dat.cmb. In the screen output, such linear combinations are indicated by a negative reliability, e. g.*

```
row 005: f001+3f002+2f003+0.0136421 -0.325575
```

*This entry is visible at runtime, but vanishes from the screen output when the calculations for row 006 start.*

## 7. Keywords Reference

This section is a compilation of all keywords accepted by COMBINE. A brief description of arguments and default values is given. If an argument is required, it is indicated by <double>, and default values are given in parentheses, e. g. (1).

`cdamp` <double> (1)

combination damping, e. g. reduction of reliability of a linear combination with increasing number of components employed, p. 121

---

<sup>5</sup>The computation of the sample project `reject` takes 40 minutes on an Intel Core2 CPU T5500 (1.66GHz) under Linux 2.6.18.8-0.9-default i686.

`csig`

forces COMBINE to use `csig` instead of `sig`, p. 120

`decay <double> (1)`

decay of reliability assigned to a frequency multiple for increasing harmonic order, p. 121

`dt <double> (auto)`

total time interval of the time series, defining the Rayleigh frequency resolution. By default, COMBINE determines the Rayleigh frequency resolution as the frequency spacing of the closest pair of frequencies found in the input data, p. 121.

`order <double> (auto)`

parameter restricting the range of harmonics of individual frequency components to be employed to form linear combinations, p. 121

`reject`

activates the rejection of unwanted linear combinations. The user may indicate unwanted frequencies by a minus sign in the input file `<infile>`. If this keyword is set, COMBINE automatically suppresses the output of those signal components for which the most reliable linear combination incorporates such an unwanted frequency, p. 128.

`sens (0.1)`

reliability limit to be exceeded in order to accept a linear combination, adjusts the number of genuine components in a frequency list, p. 122

`tol <double> (0)`

COMBINE frequency tolerance parameter, p. 121

## 8. Online availability

The ANSI-C code is available online at <http://www.sigspec.org>. For further information, please contact P. Reegen, [peter.reegen@univie.ac.at](mailto:peter.reegen@univie.ac.at).\*

**Acknowledgments.** PR received financial support from the Fonds zur Förderung der wissenschaftlichen Forschung (FWF, projects P14546-PHY, P17580-N2) and the BM:BWK (project COROT). Furthermore, it is a pleasure to thank M. Gruberbauer, M. Hareter, D. Huber, D. Punz (Univ. Vienna), G. A. H. Walker (UBC, Vancouver), and W. W. Weiss (Univ. Vienna) for their help. I acknowledge the anonymous referee for a detailed examination of both this publication and the corresponding software, as well as for the constructive feedback that helped to improve the overall quality a lot. Finally, I address my very special thanks to J. D. Scargle for his valuable support.

## References

- Kallinger, T., Reegen, P., Weiss, W. W. 2008, *A&A*, 481, 571  
Reegen, P. 2005, in *The A-Star Puzzle*, Proceedings of IAUS 224, eds. J. Zverko, J. Ziznovsky, S.J. Adelman, W.W. Weiss (Cambridge: Cambridge Univ. Press), p. 791  
Reegen, P. 2007, *A&A*, 467, 1353  
Reegen, P. 2011, *CoAst* 163, 99  
Reegen, P., Gruberbauer, M., Schneider, L., Weiss, W. W. 2008, *A&A*, 484, 601  
Walker G., Matthews J., Kuschnig R., et al. 2003, *PASP*, 115, 1023  
Walker G. A. H., Matthews, J. M., Kuschnig, R., et al. 2004, *BAAS*, 36, 1361  
Walker G. A. H., Kuschnig, R., Matthews, J. M., et al. 2005, *ApJ*, 623, L145

---

\*Please contact Michael Gruberbauer, [mgruberbauer@ap.smu.ca](mailto:mgruberbauer@ap.smu.ca).