Proportional Parallelism Explorer Program

User Manual

Alan Shepherd

Table of Contents

Chapter 1 Installing the Program

1.1 Download

The program itself can be downloaded from:
https://github.com/Goldberg53/PPExp
with no guarantees that it will work as expected.
Instructions are on that page.
If you are reading this in the book: the manual included with the program may contain updates made after the book was published.
If you are reading this from the program: this manual is an extract from the book "Let's Calculate Bach" and may have been updated since the book was published. Familiarity with the book is required in order to fully understand the program.

1.2 Prerequisite

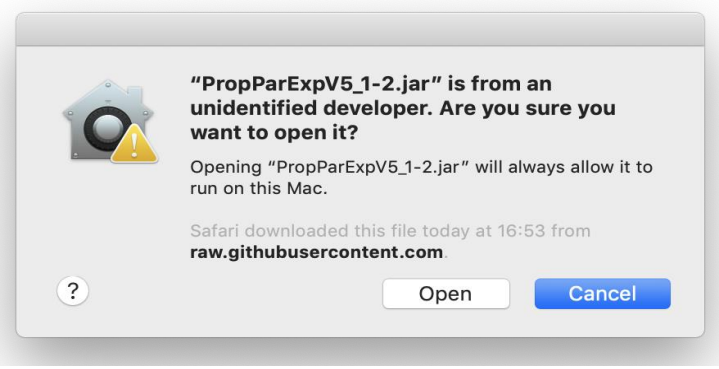
The program is written in the Java programming language. The Java runtime environment (JRE) version 8 (also known as 1.8) or later must be installed before the program can run.
Obtain the official free JRE from Oracle at www.java.com and install it as instructed.
The program was developed and tested with JRE version 8 update 201.

1.3 Caveats

The program has been tested on the following systems:
V6.0 on Apple MacOs Catalina 10.15.6
V5.0 on Microsoft Windows 10
V5.0 on Apple MacOS Mojave 10.14.3
V5.0 on Linux Ubuntu xenial 16.04, kernel 4.15.0-45-generic
V4.0 on Microsoft Windows 10, Windows 7
V4.0 on Apple macOS High Sierra Version 10.13.6
V3 on Linux Ubuntu xenial 16.04, kernel 4.10.0-42-generic
The program has been tested, but as with any software, there can be no guarantee that all results are correct, and no responsibility is taken for any harm caused by using the program.

The following differences between the systems are known:

- Apple MacOS has tighter security restrictions on downloading and running unsigned files (see Chapter 2). The program file is not signed, so the user must confirm download and execution.
- Apple MacOS does not implement the colouring of the progress bars for Pause and Cancel.



1.4 Installing the Program

The program does not need to be installed as an application. Download the .jar file to a convenient place on your computer. You can create a shortcut to the jar file and copy the shortcut to the desktop.
The user manual is included in the .jar file and can be extracted and opened with the "Extract Manual" function in the Help menu. Further information is in the Help function in the Help menu.

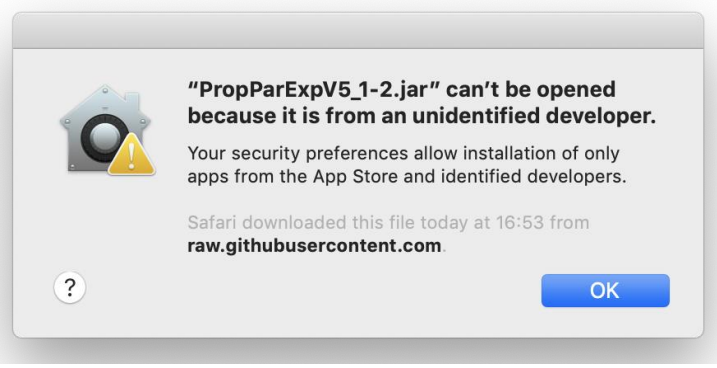
1.5 Removing the Program

To remove the program from your computer, simply delete the .jar file.
If you extracted the user manual, you can delete the extracted file in your home folder, called PropParExpManualV5.0.pdf or similar at any time.
If desired, uninstall the Java runtime environment in the usual way, e.g. on Windows with Programs and Features in the Control Panel.

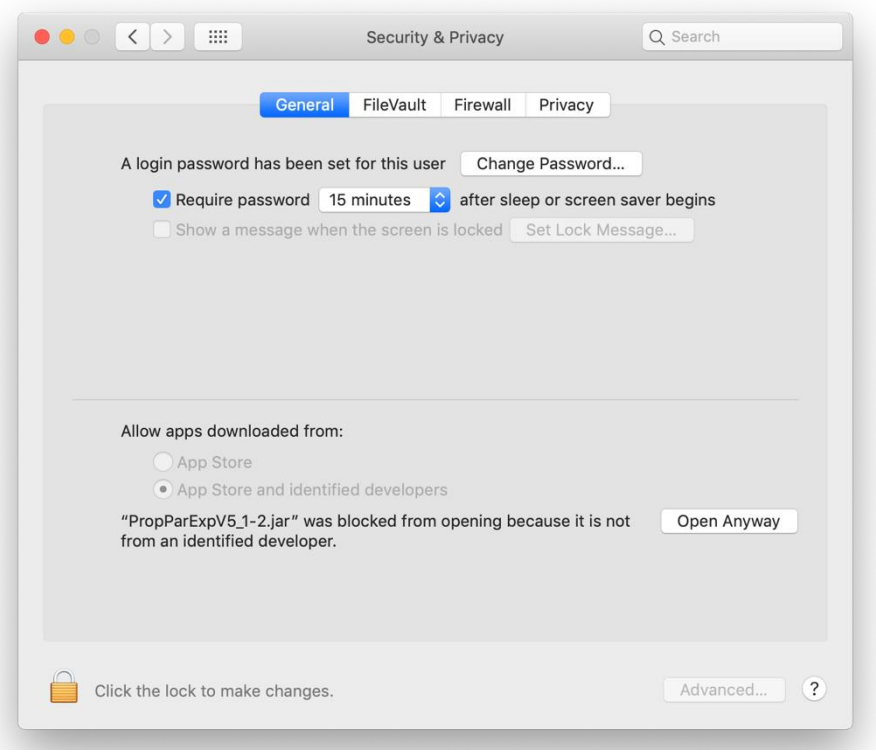
Chapter 2 Running the Program

2.1 From the Downloaded File

To run the program:
Windows: double click on the **.jar** file or the shortcut (see 1.4).
It can also be started from a command line - see 2.2.
Mac OS: double click on the **.jar** file or the shortcut (see 1.4).
The program is not signed, so the Mac system may not allow it to run straight away, giving the message below:
This can be circumvented by control-clicking on the jar file and selecting "Open", which will then ask if you want



to open it.
Alternatively, you can go to System Preferences – Security and Privacy and in "Allow apps downloaded from", select "Open Anyway" as shown below:



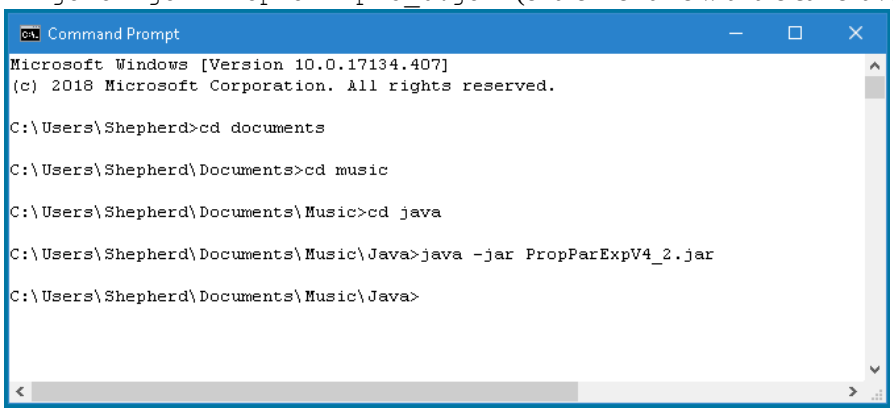
Unix: run from the command line (see 2.2).
Multiple instances of the program can be run simultaneously.

2.2 Running the Program from the Command Prompt

To run the program from a command prompt:

Start a Command Prompt, e.g. in the Windows start menu under Windows System, on Mac OS start a terminal window.

Change the directory to the folder[1] where the PropParExp.jar file is saved.
Type the following command:
`java –jar PropParExpV5_0.jar` (or the file name with the current version)



The program should run and give any error messages in the command line window.
The program does not support reading arguments from the command line, so it cannot be used for batch processing. (However multiple instances can be run simultaneously.)

2.3 Running Multiple Instances

The program can be run more than once to perform analysis of multiple file simultaneously or to run different Mont Carlo simulations at the same time. This is particularly useful if multiple runs are needed which take a long time and are to be run over night or even over several days.
If you are lucky enough to have a computer with multiple processor cores, each instance will run in a separate core, and will therefore genuinely run simultaneously. If you run more instances than you have cores, they will compete for the use of the available processors and take longer.
Note that running a file analysis and a Monte Carlo simulation in the same program instance will run these simultaneously on two processors if available. It is not necessary to start two instances of the program to do this.
Note also, that if you wish to do anything else while the program is running, such as reading mails or editing documents, this will be significantly slower unless you leave one processor core free for this work.
Windows 10 – simply double-click on the .jar file to start a new instance or from a command prompt type

```
start java –jar PropParExpVm_n.jar
```

(this will also start a new terminal window – if you close this it will terminate the program as well).
MacOS – from a terminal command line window

```
open -n PropParExpVm_n.jar
```

or open a new Terminal window to start each instance with the command

```
java –jar PropParExpV5_0.jar
```

or start them asynchronously from a single terminal window by adding an ampersand to the command:

```
java –jar PropParExpV5_0.jar &
```

The most convenient method is to create a script file – see 2.4.
Note that using the `open -n` command has the locale problem described in section 2.6.

2.4 Starter Script

To avoid having to open a command line or terminal to start the program in the above circumstances, you can make a script which will run with a double-click.

Windows

```
Start java -Duser.language=de -jar….
```

There is no simple way to avoid leaving the command line window open.

MacOS

Create a file named e.g. RunPPE.command with the line

```
java -jar /Users/<you>n/PropParExpV6.0b.jar &
```

Make the file executable with

```
chmod +x /Users/<you>/RunPPE.command
```

[1] The terms “directory” and “folder” are synonymous - the usage depends on the operating system.

This will leave the terminal window open. To automatically close the terminal window when the program exits, go to Terminal – Preferences – Profiles – Shell and set "When the shell exits" to "Close if the shell exited cleanly".

2.5 CSV Separator

When it is started, the program pre-sets the CSV-file separator depending on the decimal separator:

- If the decimal separator is set to comma, the CSV separator is set to semicolon.
- If the decimal separator is dot or anything else, the CSV separator is set to comma.

This is the same logic as is used by Excel, so the CSV files should open in Excel with no problem (unless the system settings have been changed since the file was produced). Note also the problems with mixed languages – see 2.6.
The separator can be changed in the preferences – see 7.1.

2.6 Mixed Languages

If you have your computer set to a different language than your region, e.g. the operating system is in English but you are in Germany, the decimal number formatting may not work as expected. Java seems to define its decimal separator from the system language rather than the region locale setting, so even though your regional settings show comma as the decimal separator, it may use dot. The workarounds for this, apart from changing the system language, are as follows:

Windows 10
Start from the command line and use
`java -Duser.language=de -jar….`

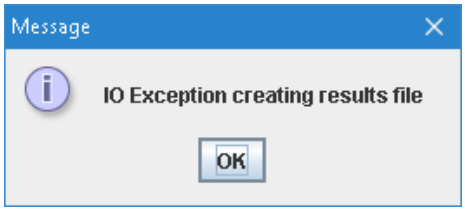
Mac OS Catalina
Start from the terminal with
`java -jar…`
and if you need multiple instances, start a new terminal window for each, as `open` will not use the correct setting.

CSV Separator
This also depends on the language settings – see 2.5.

Thousands
The thousands separator has been set to space to aid legibility and avoid any confusion with decimal and CSV separators.

2.7 Error Messages

Errors handled by the program will give a message on the screen e.g. if the output file could not be created:

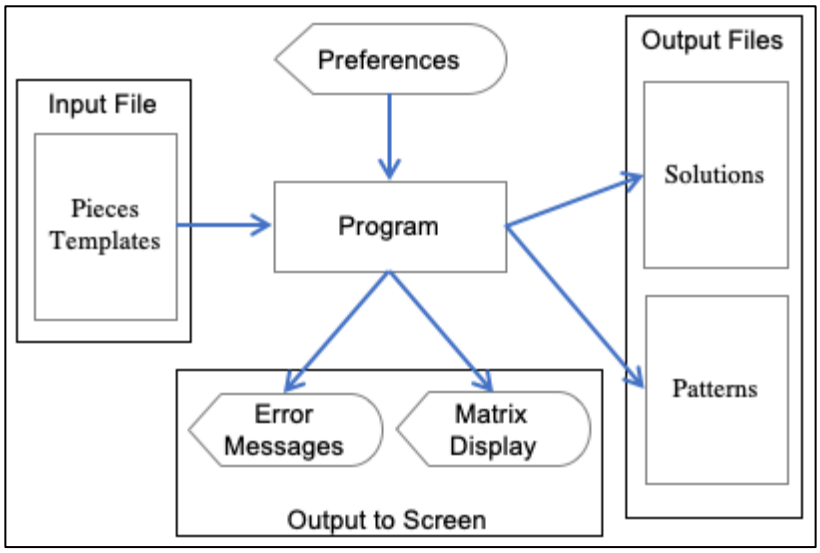


If you need assistance in solving the problem, e.g. if the message contains a cryptic stack dump, send a screenshot of the message to the developer.
If the program fails without an error message, or more information is required for the developer, start it from a command prompt (see 2.2) in order to see the console output, and repeat the action. This is useful as internal error messages are displayed in the console, which is not otherwise visible.
Various error messages may arise from checking the input file. These include the line number of the input file so that they can easily be localised.
An error "java.lang.UnsupportedClassVersionError" means that you have an older version of Java (see section 1.2).

Chapter 3 File Processing

3.1 File Processing Data Flow

The basic data flow for processing an input file to find the proportions is:



The program works as follows:

- In the Preferences dialog, the user sets the desired proportion, the contents of the output files and other parameters (see 7.1).
- When the user Opens an input file, read the file to get the titles and lengths (number of bars) of a set of pieces and the pattern matching criteria (templates) - see 3.2 - 3.7; Check the input for errors and output them to the screen – see 2.7.
- When the user starts the file processing, search for combinations of the pieces that will satisfy a given proportional split, i.e. Solutions; errors occurring during processing are output to the screen (see 2.7).
- For each solution found:
    - The solutions are optionally shown on the screen in real time in a matrix display (see 7.1)
    - Write the solution and optionally its complement to the solutions output file (see 3.8)
    - Test each solution pair against all the patterns given in the input file
    - Write solutions with matching patterns to the patterns output file (see 3.9)
- Write the summary to the output files and the screen.

3.2 Input File – General

The input file consists of different types of line, each with their own syntax, for defining a set of pieces and patterns to match.
Each line starts with a keyword denoting the type of line. The keywords are not case sensitive, i.e. capital letters can be used at will.
The elements of a line can be separated by any number of spaces or tabs.
Empty lines are ignored.
Comment lines starting with // are ignored and can be used for your own notes.
A line can conclude with a comment starting with //.
It is recommended to keep the name of the input file short, as various items are appended to it to form the output file names – see 7.1.
Input files can be created with a text editor (e.g. Notepad on Windows or TextEdit on Mac) or with a text processing program such as Microsoft Word by saving the file as plain text.

3.3 Input File - Pieces

The set of pieces must be provided in the input file with one piece per line.

The syntax of the lines is as follows (angled brackets denote a placeholder for variable content):

```
Piece <title>  <length>
```

Each line consists of the keyword "`Piece`", the title of the piece and its length in bars as an integer number.
The title cannot contain spaces, as these delimit the elements of the line – an underline or hyphen can be used instead. The title is used for column headers in the output files.
There cannot be more than 63 pieces in a file (this is limited by the size of the largest integer 2^{63} when counting the combinations).
Example for J. S. Bach's Well Tempered Clavier Book 1:

```
// J. S. Bach Well Tempered Clavier Book 1
Piece   01_Cmaj_a4           62
Piece   02_cmin_a3           69
Piece   03_C♯maj_a3          159
Piece   04_c♯min_a5          154
Piece   05_Dmaj_a4           62
Piece   06_dmin_a3           70
Piece   07_Ebmaj_a3          107
Piece   08_d♯min_a3          127
Piece   09_Emaj_a3           53
Piece   10_emin_a2           83
Piece   11_Fmaj_a3           90
Piece   12_fmin_a4           80
Piece   13_F♯maj_a3          65
Piece   14_f♯min_a4          64
Piece   15_Gmaj_a3           105
Piece   16_gmin_a4           53
Piece   17_Abmaj_a4          79
Piece   18_g♯min_a4          70
Piece   19_Amaj_a3           78
Piece   20_amin_a4           115
Piece   21_Bb(B)maj_a3       68
Piece   22_bb(b)min_a5       99
Piece   23_B(H)maj_a4        53
Piece   24_b(h)min_a4        123
```

3.4 Input File – Patterns: Overview

The pattern matching is described in more detail in the book Let's Calculate Bach.
There are three functions, Template, Count and BuiltIn, described in the following sections. The lines can be in any order. Each solution pair found, i.e. each solution and its complement, is tested against all the given patterns.

3.5 Input File – Patterns: Binary Template

Binary templates can be given to either exactly match the signature or match a part of the signature in any position. These are defined by lines in the input file with the following syntax:

```
Template      <name> <action><binary template>
```

The keyword "`Template`" denotes the type of line (as opposed to "`Piece`" – see 3.3).
<name> is a name for this template which the user can define. This is shown in the output file (see 3.9). It cannot contain spaces, as these delimit the elements of the line – an underline or hyphen can be used instead.
<action> defines how the pattern is to be matched. This can be "`Exact`" or "`Shift`".
<binary template> is a sequence of bits to be matched against the solution signatures.
If the template has the wrong number of bits, an error message is shown. Comment lines showing the bit positions are useful for checking that one has the correct number of bits and they are lined up as desired.

The following is a simple example of templates in an input file.

```
// Pattern Templates
// Bit Positions 10s          00000000011111111122222
// Bit Positions 1s           12345678901234567890123
Template    Test        Shift 1111111001
Template    Tatlow      Exact 11111000000100000011111
Template    12inARow    Shift 111111111111
```

3.6 Input File – Patterns: Count

This pattern will match a solution that consists of a given number of pieces. It is defined by lines in the input file with the syntax:

`Count` <name> <number>

The keyword "`Count`" denotes the type of line (as opposed to `Piece` or `Template`).
<name> is a name for the pattern as for binary templates.
<number> is an integer giving the number (or count) of pieces in a solution or its complement which will give a match.
The following example will give a match if a solution or its complement is made up of any 12 pieces or 8 pieces respectively.

```
Count Twelve      12    // Twelve anywhere
Count Eight       8     // Eight anywhere
```

Note that all solutions are checked for having the proportion in the number of pieces – see PropCount in section 3.8.

3.7 Input File – Patterns: Built In Functions

The "built in" pattern matching functions are denoted as follows by lines in the input file with the syntax:

`BuiltIn` <function>

The functions are predefined, and the following are currently available.

Left=Right

```
BuiltIn     Left=Right
```

- this finds any solutions where the right half of the signature is the same as the left half, e.g.

```
010100 010100
```

Left=thgiR

```
BuiltIn     Left=thgiR
```

- this finds any solutions where the right half of the signature is the mirror image or reverse of the left half, e.g.

```
010100 001010
```

The following example shows both of these added to the previous example:

```
Count Twelve      12    // Twelve anywhere
Count Eight       8     // Eight anywhere
// Pattern Templates
// Bit Positions 10s          00000000011111111122222
// Bit Positions 1s           12345678901234567890123
Template    Test        Shift 1111111001
Template    Tatlow      Exact 11111000000100000011111
Template    12inARow    Shift 111111111111
BuiltIn     Left=Right
BuiltIn     Left=thgiR
```

If there is an odd number of pieces, the middle piece is irrelevant for the above, e.g. Left=Right will match both

`010100 0 010100` and
`010100 1 010100`

3.8 Solutions Output File

The solutions output is delivered to a file either in the same folder as the input file or a different folder chosen by the user (for example if all output files are to be gathered in the same folder). The details are determined by the Preferences – see 7.1.

The first line consists of the tiles of the columns, with the titles of the pieces as given in the input file from column F (this can be omitted – see 7.1).
The basic output is a sequence of pairs of lines each of which lists a combination of lengths which will result in the given proportion, i.e. the solution pairs. The columns are as follows:
Index – the sequential index number of the solution pair. The two solutions of a pair have the same index, suffixed with an S for Solution and C for Complement, and this is referenced in the patterns output file (3.9). Each layer has an additional layer in the sequence number, separated by underlines. The complement lines are omitted if the preference "Include Complements" (see 7.1) is not checked – this is the initial setting.
Layer - this gives the layer number of the solution pair, which is also indicated in the Index.
Sum - this column gives the sums of the lengths of the solutions and complement.
Count – this column gives the number of pieces in each solution or complement row.
PropCount – this column indicates whether the number of pieces is in the same proportion as that chosen for the lengths, i.e. that it is a double proportion.
Note that for the 1:1 proportion there is a specific setting to determine whether the opposites are included.
Note that the Sum and Count columns can be excluded in the preferences.

	A	B	C	D	E	F	G	H	I	J	K
1	Index	Layer	Sum	Count	PropCount	U	V	W	X	Y	Z
2	1S	1	30	3	Y	5	10	15			
3	1C	1	30	3	Y				8	7	15
4	1S_1S	2	15	2		5	10				
5	1S_1C	2	15	1				15			
6	1C_1S	2	15	2					8	7	
7	1C_1C	2	15	1							15
8	2S	1	30	4		5	10		8	7	
9	2C	1	30	2				15			15
10	2S_1S	2	15	2	Y	5	10				
11	2S_1C	2	15	2	Y				8	7	
12	2C_1S	2	15	1	Y			15			
13	2C_1C	2	15	1	Y						15
14	3S	1	30	3	Y	5	10				15
15	3C	1	30	3	Y			15	8	7	
16	3S_1S	2	15	2		5	10				
17	3S_1C	2	15	1							15
18	3C_1S	2	15	1				15			
19	3C_1C	2	15	2					8	7	
20	Program Version: 6.0d										
21	Input File: /Users/alan/OneDrive/PropParResults/Z_Book/6test_for book.txt.										
22	Total Length: 60. Proportion: 1:1. Target: 30. Tried: 38.										
23	Include Complements: Yes. 1:1 Opposites: No.										
24	Layers: 2. Strict Layers: No. Strict PropCount: No. Only First Strict: No.										
25	Strict Proportion Solutions: 3.										
26	Strict PropCount Solutions: 0.										
27	Solutions Layer 1: 3. Layer 2: 6.										
28	Patterns: 1.										
29	Processing Time: 0,001 Seconds without Matrix.										

The file concludes with a summary of the results - see 3.10 for details.

3.9 Patterns Output File

A second output file is created which gives those solutions which match one or more of the patterns defined in the input file (see 3.4 - 3.7). This is written to the same folder as the solutions file.

	A	B	C	D	E	F	G	H	I	J
1	Index	Layer	Pattern	PropCount	U	V	W	X	Y	Z
2	2S	1	Left=Right+^		5	10		8	7	

These are shown as follows:

Index – a reference to the index number of the solution pair in the solutions output file (see 3.8).
Layer – the layer in which the pattern occurs.
Pattern – the name of the pattern as given in the input file. This is suffixed with an indicator for where the pattern occurred as shown in the table below.
PropCount – shows "Y" for yes if there is an additional proportion in the count of pieces.

Indicator	Meaning
>n	For shift template: the pattern was shifted right by n places to match
^	The pattern matches the complement of the given solution. Notes: This can occur in addition to the above as >n^, i.e. the pattern shifted n places matches the complement. The pattern output gives the lengths for the solution being tested (the first row of the pair in the solutions file) and not its complement where the match occurred.
+^	The pattern matches both the signature and its complement. For the built in patterns Left=Right and Left=thgiR the pattern that matches the signature will always match the complement as well, so this is always present.

A solution pair can match multiple patterns. This will be apparent as the same index will occur in consecutive rows with different patterns in the pattern output file.

3.10 File Processing Summary

The summary contains:

- The program version with which the results were obtained.
- Whether the run was cancelled before completion
- The input file name with path.
- The total length of all the pieces in bars, the chosen proportion, the target length for that proportion and the number of combinations actually tried[2].
- The preference settings of Include Complements and 1:1 Opposites.
- The number of layers selected and, if more than one, the settings for Strict Proportions and Only First and for Proportional Counts and Strict Proportional Count.
- The number of strict solutions found (if more than one layer).
- The number of strict proportional count solutions found (if more than one layer).
- The number of solutions and the number of PropCounts found in each layer.
- The number of patterns found.
- The time taken, indicating with or without matrix output, as this slows the program down significantly.

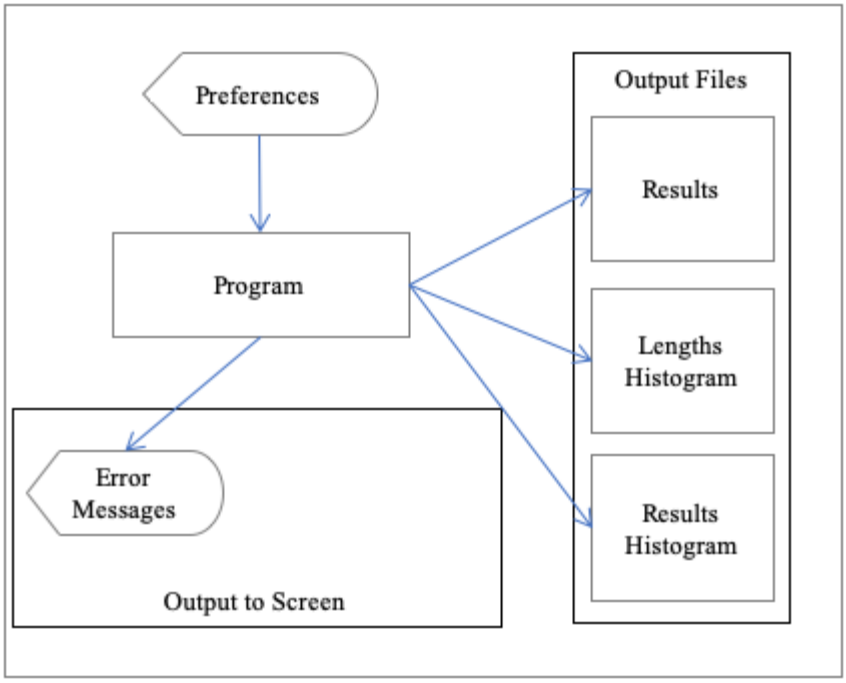
Note: The Proportional Counts show "-" if the number of pieces in the layer cannot be divided in the desired proportion.
The summary is shown in a window on the screen when the program completes, at the end of the matrix output and optionally at the end of the solutions output file.

[2] Note: this will usually be less than the theoretical number of possible combinations, as the program is optimised to stop trying a sequence when the target is exceeded – see the book for details.

Chapter 4 Monte Carlo Simulation

4.1 Monte Carlo Processing Data Flow

For Monte Carlo simulation the process is shown below and is as follows:



- The user sets the desired proportion, the number of pieces, number of samples, minimum and maximum length and the folder for the output file and other parameters in the Preferences dialog (see 7.1).
- When the user starts the simulation, generate pseudo-random sets of lengths.
- For each sample set:
  - Run the search routine to give the number of solutions
  - Write the lengths and number of solutions to the results output file.
- Write the summary to the output file and optionally output the histogram files.

The histogram files are intended to facilitate plotting graphs of the results, e.g. with Excel or the R statistics package.

4.2 Results Output File

The output file name is determined by the preferences – see section 7.1.
The output file has each sample set of pieces in a separate row. The columns are simply titled “R1”, “R2”, etc. for “Random 1” etc. The next columns are the number of solutions for each layer. The last column is the time taken to find the number of solutions.
The file includes a summary – see section 4.3 for details.

Note: the actual solutions for each sample are not output. If these are required, the user must take the lengths of a sample and create an input file as described above.

	A	B	C	D	E	F	G	H	I
1	R1	R2	R3	R4	R5	R6	SolsL1	SolsL2	Seconds
2	15	13	7	5	5	3	0	0	0
3	15	15	13	12	7	6	2	0	0
4	14	11	8	4	4	3	2	2	0
5	13	12	12	11	10	6	0	0	0
6	14	13	8	7	6	4	1	1	0
7	14	11	8	7	6	6	1	0	0
8	14	13	10	9	9	5	0	0	0
9	14	14	13	7	5	3	1	1	0
10	14	13	11	8	6	4	1	1	0
11	13	13	10	7	5	4	1	1	0
999	12	9	8	8	4	3	0	0	0
1000	13	12	12	11	10	6	0	0	0
1001	8	8	8	6	6	4	3	0	0
1002	Program Version: 6.0c								
1003	Tried: 1 000. Ignored: 3 077.								
1004	Pieces: 6. Samples: 1 000.								
1005	Minimum Length: 3. Maximum Length: 15.								
1006	Proportion: 1:1. Strict Layers: No. Layers: 2.								
1007	Layer 1:								
1008	Min. Solutions: 0. Max. Solutions: 6.								
1009	Mean: 1,10800. Std. Deviation: 1,01161.								
1010	Layer 2:								
1011	Min. Solutions: 0. Max. Solutions: 11.								
1012	Mean: 0,47700. Std. Deviation: 1,15793.								
1013	Processing Time: 0,044 Seconds.								

4.3 Monte Carlo Summary

The summary is written at the end of the samples output file and shown in a window on the screen when the program completes. It contains:

- The program version with which the results were obtained.
- Whether the run was cancelled before completion.
- The number of samples tried and the number of samples ignored (because they cannot give the desired proportion).
- The number of pieces and number of samples taken.
- The minimum and maximum lengths of the pieces.
- For each layer:
    - The minimum and maximum numbers of solutions.
    - The mean and standard deviation of the numbers of solutions.
- The time taken.

4.4 Lengths Histogram Output File

The output file name is determined by the preferences – see section 7.1.
The output file provides the data prepared for plotting a histogram of the lengths of all the pieces in all the samples. It consists of two columns, one for all the lengths in the range, and for each length, the number of pieces that have that length, i.e. the frequencies.
This can be used to validate that the pseudo-random numbers generated have an approximately uniform distribution.

4.5 Results Histogram Output File

The output file name is determined by the preferences – see section 7.1.
The output file provides the data prepared for plotting a histogram of the results. It consists of one column for the number of solutions and one for each layer with the number of samples that had that number of solutions, i.e. the frequencies.
Note: with Strict Proportions only one column is output for layer 1.

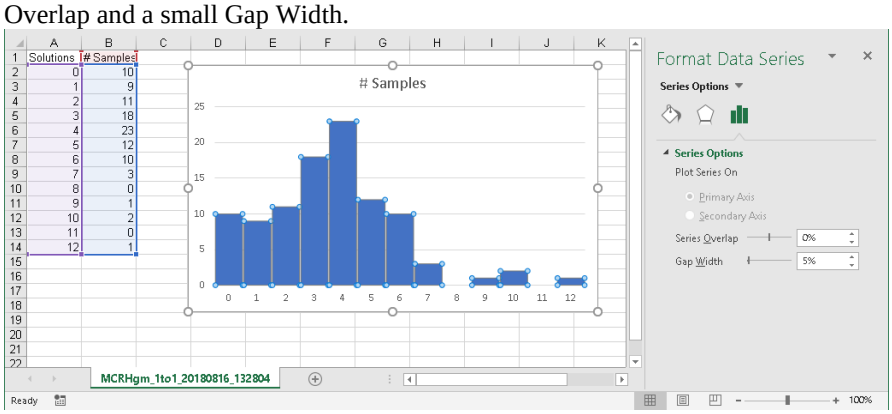
Chapter 5 Further Processing with Other Programs

5.1 Import to Excel

The output files can be used in Microsoft Excel for further exploration, e.g. by sorting or filtering, or making histogram graphs.
To open the output file directly in Excel, it must have the correct separator, which can be determined as follows. See also section 2.6.
For Windows: in the Windows Control Panel, under Region, Formats tab and Additional Settings, the List Separator is shown. This depends on the operating system language setting and is usually comma in English[3] or semicolon in German.
On MacOS the CSV separator is comma if the decimal separator is dot and is semicolon if the decimal separator is comma.
The program tries to select the appropriate separator as the default.
Use the same separator in the program preferences and set the output file extension to csv – see 7.1.
If the separator is set to Tab, you can open the file in a text editor, copy the content (ctrl-A, ctrl-C) and paste it into Excel.
For use with Excel you may wish to leave the "Include Summary" box unchecked, as the final summary lines do not fit the column structure and may be disturbing if sorting and filtering are used. However, it is recommended to leave it in if many files are being processed so that you can later verify the parameters that produced the file.

5.2 Formatting Histograms with Excel

To facilitate further analysis the data for histograms can be written to files (see 4.4 and 4.5).
For a simple plot in Excel, select the data and Insert a Column Chart. Format the data series, e.g. with no Series Overlap and a small Gap Width.



Note: the outputs of the histogram data are prepared for a direct x-y bar plot with x as the number of solutions and y number of samples that have x solutions (the frequency), and so are not suitable for use with histogram functions. We find this better for large numbers of samples because the standard histogram function combines the data into "bins", e.g. with 1-10 in the first column, 11-20 in the second column, etc. and the exact appearance of the diagram then depends on the bin size. (Excel is also limited to 1000 bins.) Using the data as provided by the program effectively has a bin size of 1 and shows the data accurately. To use an Excel histogram function on the results, copy the "Solutions" column of the main output file.

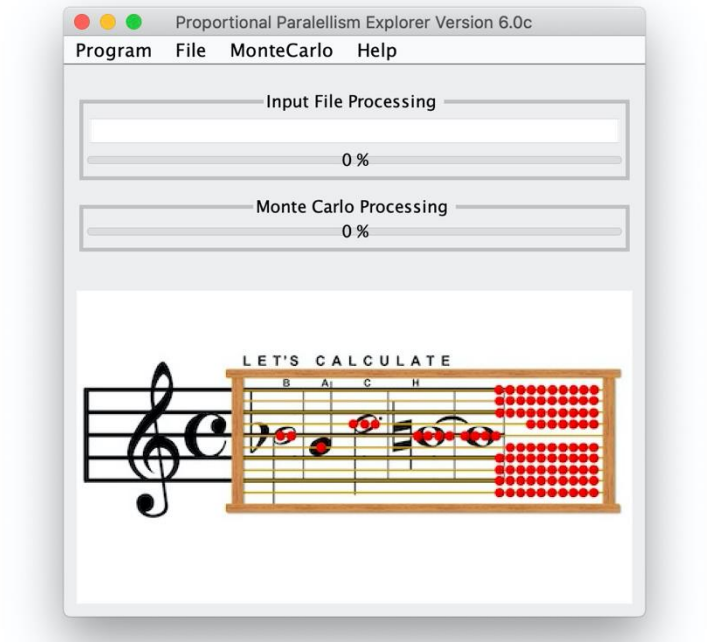
5.3 Text Processors

If the separator is set to Tab, you can copy the output from the file and paste it into a Word table.

[3] Hence the name "Comma Separated Variables" for the csv file extension.

Chapter 6 Main Window

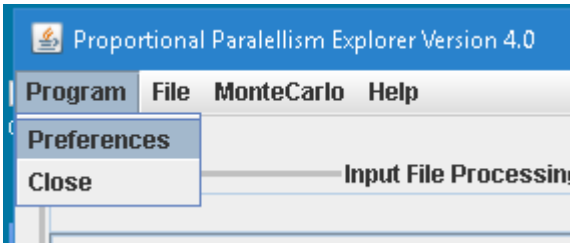
The main window appears as shown:



The program has four menus, Program, File, Monte Carlo and Help, with menu items described below.
Below this it shows for input file processing, the file that is open (if any) and the progress bar, and for Monte Carlo processing the progress bar.
The progress bars show a count of the number of solutions or samples tried and the maximum number, rather than a percentage. This is better for long running explorations which, taking many hours or even days to complete, remain at 0% for a long time. If the run is cancelled, the remaining part of the progress bar is red. If the run is paused, the remaining part is yellow. The progress bar titles also show the proportion when a process is started.
The file processing and Monte Carlo simulation can both be run at the same time.
The program can also be started multiple times, e.g. to leave several explorations running over night.

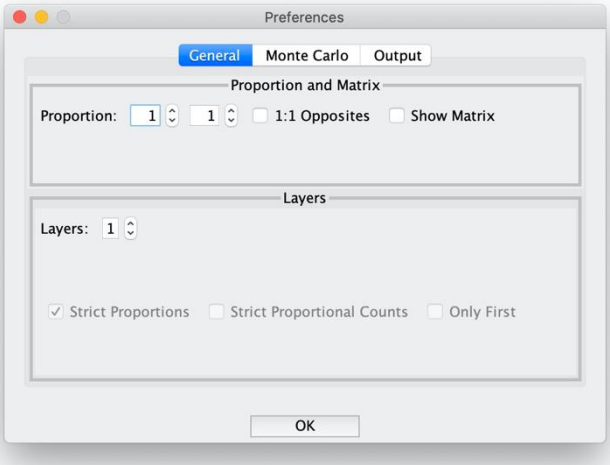
Chapter 7 Program Menu

The Program menu has:
Set the Preferences (see 7.1)
Close the program (see 7.2)



7.1 Preferences

The program is controlled by the preferences. This is divided into three tabs, General, Monte Carlo and Output. Note that the Close button (x) at the top right (or top left on Mac) is disabled. The preferences can only be closed with the OK button, to ensure that the error checking is performed.
If an error is found when the OK button is clicked, a message will appear on the screen and the tab containing the error will be shown. The preferences window will then not be closed, so that the error can be seen and corrected if necessary. If there are multiple errors, they will appear sequentially.
All the preferences are set to default values when the program is started.
The Preferences can be opened at any time, even while the program is running. However, any changes to the parameters will only take effect on the next run.



7.1.1 General Tab

Proportion and Matrix

The desired **proportion**, e.g. 1:1, 1:2 is set with the two numbers. They can be typed in or increased/decreased with the arrows beside the numbers.
If you choose a proportion which has common factors, it will be divided by the highest common factor, e.g. choosing 4:6 will use 2:3. This is indicated with a message on the screen when the processing is started.
If you choose a proportion which is not possible with the given set, e.g. 1:1 when the total number of bars is an odd number, a message will be displayed on the screen when the processing is started and no output file will be produced.
For proportions other than 1:1, putting the smaller number first (e.g. 1:2 rather than 2:1) will take less time to run, as it uses a smaller target value.
To obtain the opposite solution pairs (other than for 1:1), use the reverse proportion, e.g. 2:1 instead of 1:2. To obtain the opposites of a 1:1 proportion, check the checkbox "**1:1 Opposites**" - the program will output all the

solution pairs including the opposites. This checkbox is only visible when a 1:1 proportion is selected, as it is otherwise irrelevant.

Show Matrix

If this box is checked the results of file processing are scrolled into a new window on the screen[4]. This is useful if you wish to observe the search in real time. You may need to make the window wider to see all the columns. Note that the program will take significantly longer if this option is selected, and the time taken given in the summary shows whether the time is with or without the matrix.

Layers

For the layers, the number of layers can be set between 1 and 9.
The Strict settings are only available for more two or more layers.
Strict Proportions will only find solutions that are complete over all the given layers.
Strict Proportional Counts is only available if Strict Proportions is checked and will find any solutions where the number of pieces in the solution (or its complement) is equal to:

$$No.of\ pieces\ in\ solution = No.of\ pieces\ in\ collection * \frac{m}{m+n}$$

with a proportion of m:n.
It will match solutions where the number of pieces is in proportion, e.g.
for 24 pieces

and 1:1 – solutions with 12:12 pieces will match
and 1:2 – solutions with 8:16 pieces will match
and 1:3 – solutions with 6:18 pieces will match.

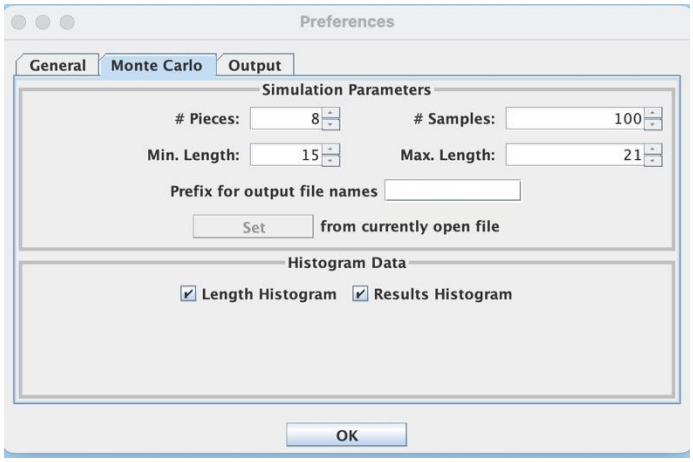
These are equivalent to using

```
Count Any12 12
Count Any8 8
Count Any6 6
```

respectively, but saves having to edit the input file when changing the proportion.
The pattern will also match the complement of a solution, which is the same as matching the 'n' part of the m:n proportion with the solution.
Only First is also only relevant if Strict Proportions is checked and with this the program will only output the first solution found in each sub-layer. This saves time and saves space in the output files.

7.1.2 Monte Carlo Tab

This tab is used to set the parameters for the Monte Carlo simulation. The numbers can be typed in or increased/decreased with the arrow buttons.



The entries are limited to:

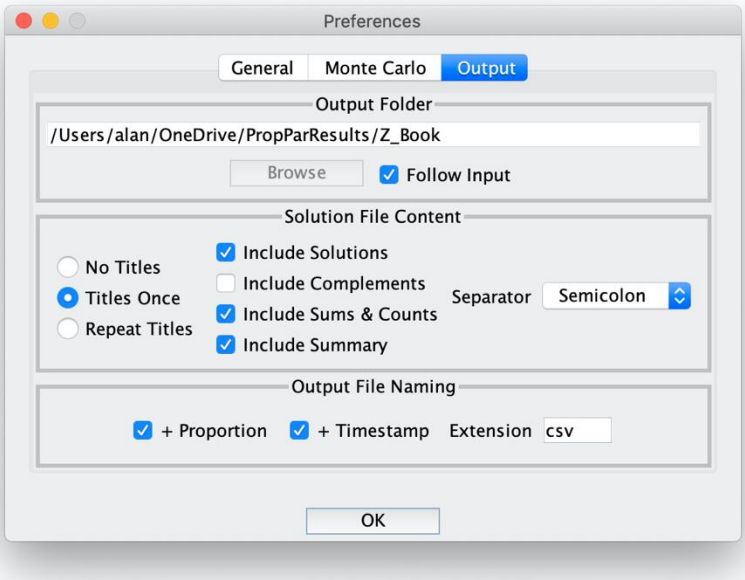
Number of pieces: 2 - 63
Number of samples: 1 - 2,147,483,647 (a 32-bit integer 2^{31}-1)
Minimum length: 1 – 1 000
Maximum length: 1 – 10 000

[4] The output is intentionally reminiscent of the science fiction film "The Matrix".", Warner Bros.

The Set button can be used to set the # Pieces, Minimum length and Maximum length and the Prefix for output file names to the values from the input file that is currently open. The button is disabled (greyed out) until a file has been opened. (See 7.1 for details of file naming.)
If the maximum length is less than the minimum length, when OK is clicked, the values will be swapped, the Preferences window will be switched to the Monte Carlo tab and a message "Monte Carlo Lengths were swapped as Max. was less than Min.!" shown on the screen.
The **Prefix for output file names** will be applied to the Monte Carlo output files if it is not empty. It is set from the currently open file name with the **Set** button and can be modified. For example, one could add "_1k" to indicate that only 1000 samples are being used. If the prefix is left empty, e.g. if no file has been opened or the Set function is not used, the Monte Carlo output files will only be distinguished by their time stamps, and if these are disabled, may overwrite each other.
To keep the program usable on multiple platforms, the file name prefix is restricted to letters, numbers and the underline. If any other characters are used, these will be removed, the Preferences window will be switched to the Monte Carlo tab, and a message "Illegal characters removed from Monte Carlo file name prefix!" shown on the screen.
The **Histogram Data** checkboxes determine whether the histogram data is written to files.

7.1.3 Output Tab

The output tab determines the location, contents and naming of the output files.



Output Folder
The folder for the output files is preset to the user's home folder.
This can be changed with the **Browse** button, but only if Follow Input is unchecked.
If the checkbox **Follow Input** is checked, the output file will be written to the same folder as the input file that is currently open for file processing. If a new input file is opened, the output folder will "follow" to this new location. Monte Carlo output will also be written to this folder (useful if you want to run a corresponding Monte Carlo after opening or processing an input file).
If the checkbox **Follow Input** is unchecked, the output folder will revert to the one last chosen with Browse (or the home folder if none has been chosen), and the Browse button is activated.

Solution File Content
The contents of the solutions output file can be determined by the checkboxes as follows:

No Titles – the solutions output file will not contain the titles of the pieces.
Titles Once – the solutions output file contains the titles of the pieces from the input file in the first row as headings.
Repeat Titles – the titles are repeated before each solution pair.
Include Solutions – the rows listing the solution pairs are included in the solutions output file.
Include Complements – for each solution row, the complement row is also included with the same index.

Include Sums & Counts – for each solution row, columns for the sum of the lengths and the number of pieces are included.

Include Summary – the summary is included at the end of the solutions output file – see 3.10.

The buttons and checkboxes can be set in any combination – see the following examples:

Titles	Include Solutions	Include Summary	Solutions Output File
No Titles	☐	☐	Empty
	☐	☑	Summary only
	☑	☐	Solutions rows
	☑	☑	Solutions rows and Summary
Titles Once	☑	☑	Titles in first line, Solutions rows, Summary
Repeat Titles	☑	☐	Titles and Solutions (3 rows for each solution)
	☐	☐	Only Title rows (not very useful!)

The above does not apply to the patterns output file – it always includes the titles once and the first solution of the matching pair, and it does not include the summary.

It also does not apply to the Monte Carlo output file – it always includes the titles once and always includes the summary.

For sets with a large number of solutions (e.g. the six Cello Suites with over 2 billion solutions) the solutions output file can be very large and may even exceed the capacity of the user's disc. For these cases the "Include Solutions" checkbox can be unchecked to leave only the summary. It is probably the patterns output which is more interesting.

Separator – the separator used between column entries in the output files. For visually examining the output, tab is recommended. (The matrix output always uses tab to keep the display in columns.) For use with Excel, see 5.1 and 2.6. The setting applies to the solutions and patterns output files as well as the Monte Carlo outputs.

Output File Names

The output file names are made up of several components, some of which can be selected in the preferences.

The file processing output files are initially named the same as the input file with an additional "_Solutions" or "_Patterns" for the solutions and patterns output respectively to avoid overwriting the input file, if it has the same extension.

The Monte Carlo simulation output files have "MC_" followed by "Result", "LenHgm" and "ResHgm".

Further components of the name can be added in the options in the Output File Naming area of the Output tab of the Preferences. These are:

+ Proportion – adds the chosen proportion to the output file name, e.g. "_1to1" for 1:1. "_Opp" is also added if 1:1 with opposites is selected (not applicable to Monte Carlo).

Layers – the layering parameters from the General tab are included next as **_nL** for the number of layers n, and if more than one layer, **S** or **nS** for Strict or not Strict. If strict, this is followed by **S** or **nS** again for **S**trict proportional count and **A** for all or **F** for only first of the sub-layer solutions.

+ Timestamp – adds a timestamp to the output file name with year, month, day and hour, minute, second, e.g. "_20171130_153059". The elements are ordered so that the names will sort in chronological order. (Note that in the example below the timestamps for the file processing and Monte Carlo are slightly different because it is impossible to start them both in the same second.)

The combination of Proportion and Timestamp can be used to avoid overwriting the output files when repeating the run with different inputs or preferences.

Extension – the extension of the output files. This should be chosen as needed for further processing, see 5.1.

To keep the program usable on multiple platforms, the extension is restricted to letters, numbers and the underline. If any other characters are used, these will be removed, the Preferences window will be switched to the Output tab, and a message "File Extension illegal characters removed!" shown on the screen.

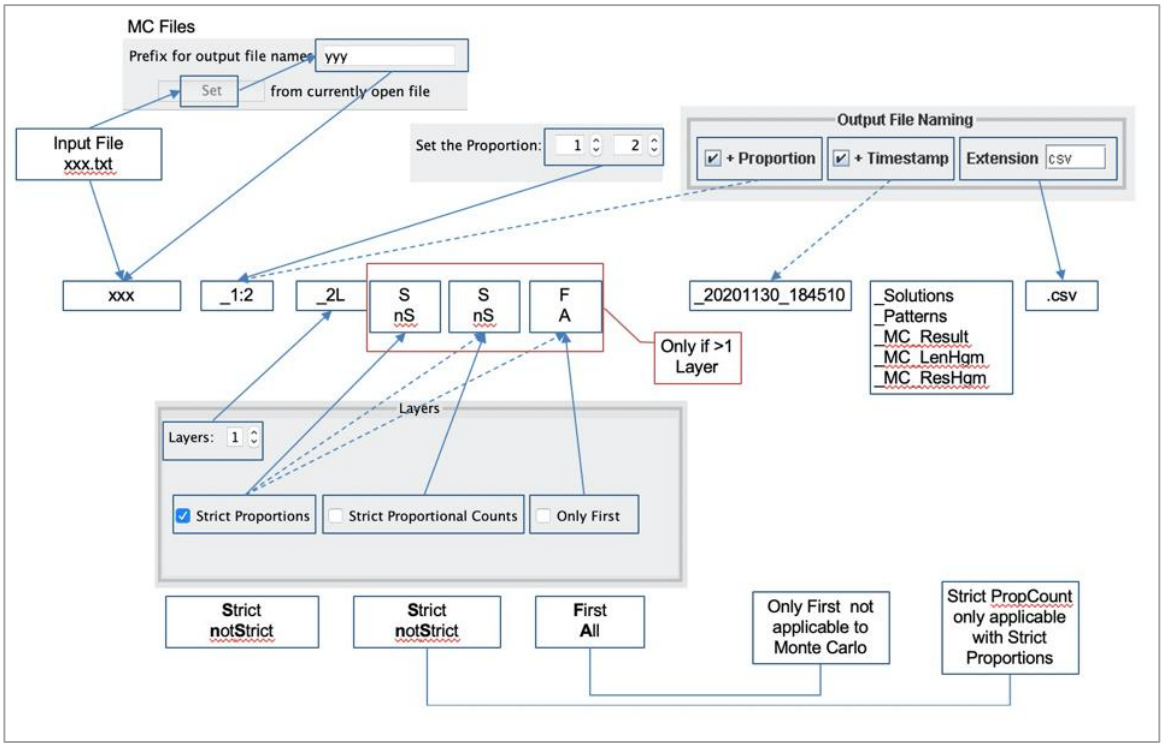
The full set of file names from an input file named WTC1 with all components, and the Monte Carlo simulation with the prefix set from the input file, appear as follows:

```
WTC1_1to2_2LSnSA_20180118_184715_Solutions.csv
WTC1_1to2_2LSnSA_20180118_184715_Patterns.csv
WTC1_1to2_2LSnSA_20180118_184755_MC_Result.csv
WTC1_1to2_2LSnSA_20180118_184755_MC_LenHgm.csv
WTC1_1to2_2LSnSA_20180118_184755_MC_ResHgm.csv
```

The file names with the minimum components appear as follows:

```
WTC1_Solutions.csv
WTC1_Patterns.csv
MC_Result.csv
MC_LenHgm.csv
MC_ResHgm.csv
```

This is summarised in the diagram below.



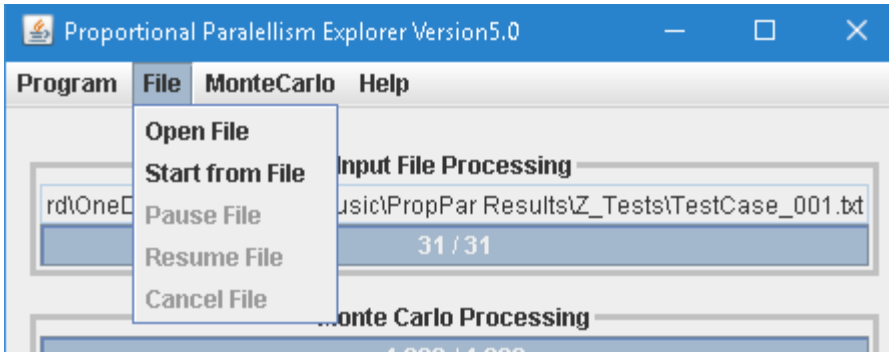
Note: if the program is run using the same output file name as a previous run, the file will be overwritten without any warning. Use the timestamp to avoid this. If the output file is open in another program, e.g. Excel, it cannot be overwritten and an IO[5] Exception error message will appear on the screen.
Note: avoid using extensions reserved for other specific purposes such as .xls, .exe, etc.

7.2 Close

This ends the program. A confirmation is required to avoid accidentally closing it during a run. (Note that no confirmation is required when closing with Command-Q or Quit (on MacOS). Next time you start the program, the preferences will be set to their default values and no input file will be open.

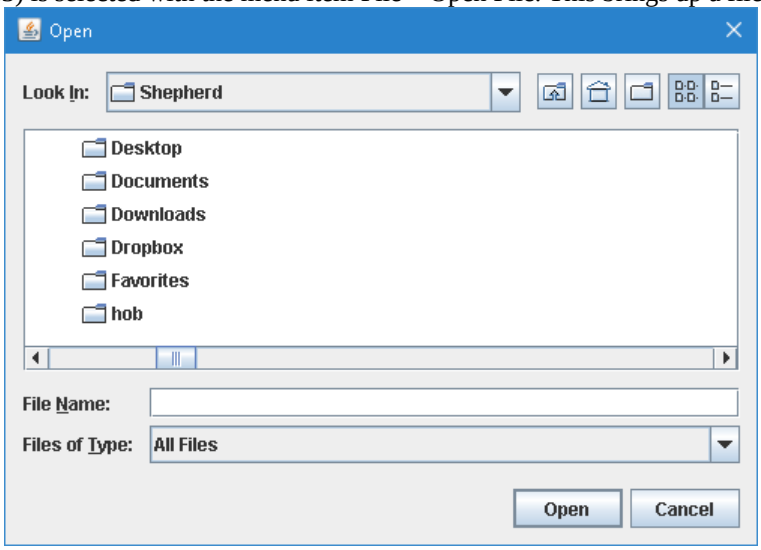
[5] IO = Input/Output

Chapter 8 File Menu

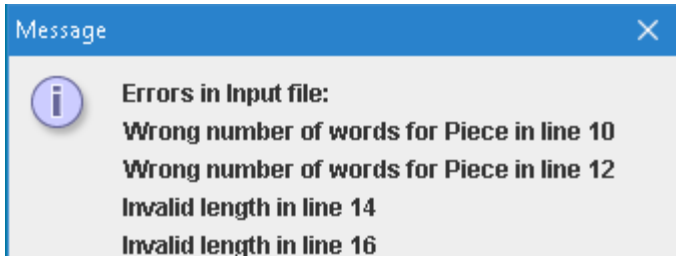


8.1 Open File

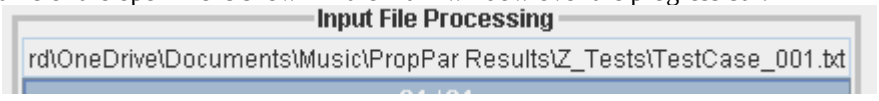
The input file (see 3.3) is selected with the menu item File – Open File. This brings up a file explorer, as usual.



When the program is started, the dialogue shows the user's home folder. Subsequently, it shows the folder last used.
When the file is opened the program reads it and checks it. The errors are shown on the screen, including the line number.



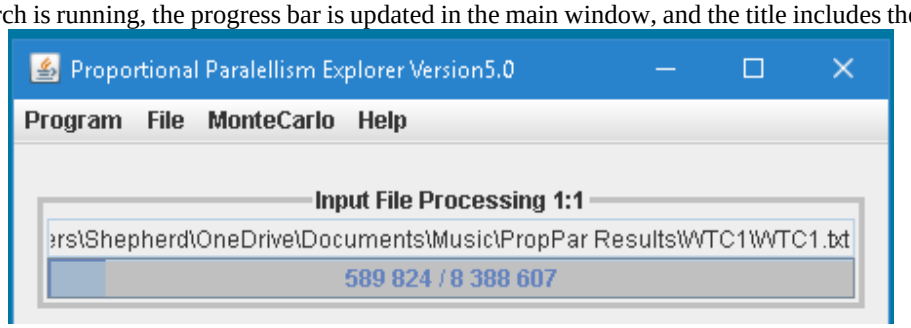
If an invalid command is found, the processing stops at that point. This is to prevent the program running out of memory and crashing if the wrong type of file is accidentally opened.
The path and name of the open file is shown in the main window over the progress bar:



8.2 Start from File

The item "Start from File" in the File menu is disabled until an input file has been selected.

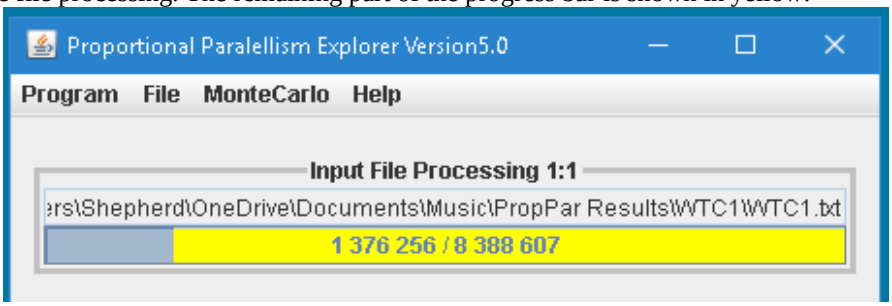
After that, selecting "Start from File" will start the search for solutions and patterns with the opened input file and with the chosen preferences (see 7.1), or with the preset default values if none have been changed.
The process can be run repeatedly with different preferences and/or with different input files.
The input file processing can be started, paused, resumed and cancelled independently of the Monte Carlo processing.
While the search is running, the progress bar is updated in the main window, and the title includes the proportion.:



The progress is pessimistic. It is not possible to predict the number of solutions and the 100% mark is the theoretical maximum number of combinations (2^N, or $2^{N/2}$ for 1:1 without opposites). Since the program does not continue to try sequences after the target has been exceeded, it will usually finish quicker than indicated.
For input file processing the progress bar is updated every second and the numerical progress counts the layer 1 tries. For multi-layer runs, this may appear to stop while lower layers are being processed.
When the processing is started, the program also checks that the input file has not been altered since it was opened. If it has, it is read again, and this may produce new error messages.

8.3 Pause File

This pauses the file processing. The remaining part of the progress bar is shown in yellow:



From this state you can Resume or Cancel the file processing.
Note: on Mac the colour does not appear[6]. To check if the processing is paused, see if the Resume item in the File menu is enabled.
This is useful if you wish to perform other work on the computer and the processing is making the computer too slow.

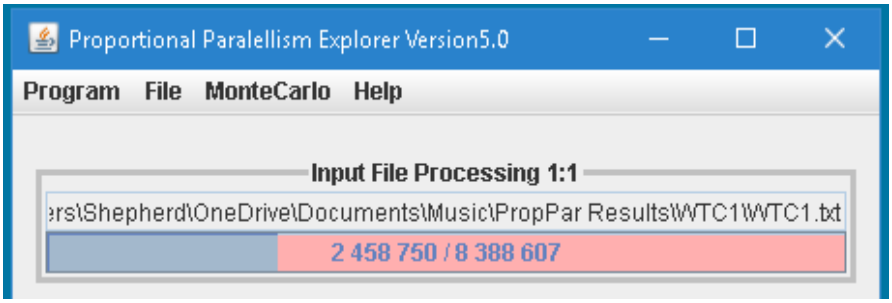
8.4 Resume File

This resumes the paused processing.

8.5 Cancel File

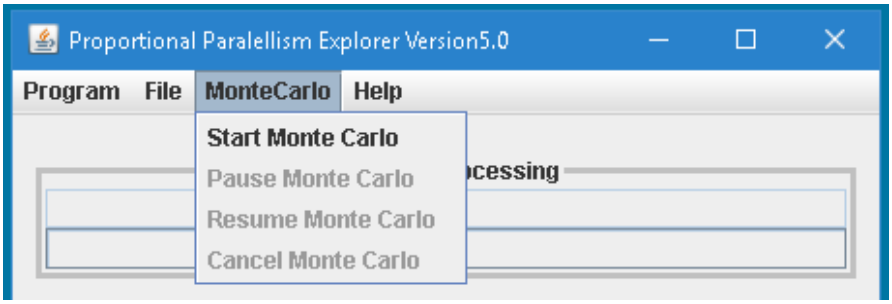
The Cancel File menu item is available while an input file is being processed or is paused and will stop the search. This will be indicated with a message on the screen and in the output file.
The remaining part of the progress bar is shown in red:

[6] This feature is not implemented in Java for MacOS.



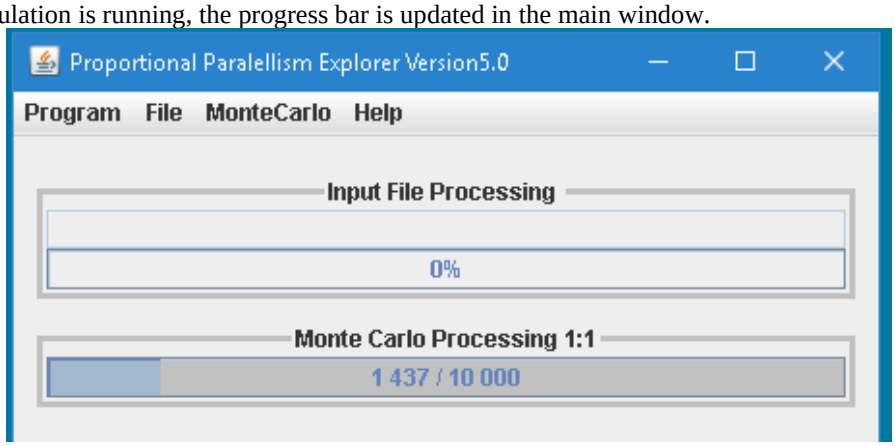
Note: on Mac the colour does not appear.
This can be used if the search is taking too long and you wish to restart it with other preference settings (e.g. with matrix output to observe progress – see 7.1).
A message is displayed with the summary – see 3.10.
The output files are produced with the results so far.

Chapter 9 Monte Carlo Menu



9.1 Start Monte Carlo

Selecting this starts the Monte Carlo simulation with the parameters given in the preferences.
While the simulation is running, the progress bar is updated in the main window.



The simulation can be started and cancelled independently of the file processing.

9.2 Pause Monte Carlo

This pauses the Monte Carlo simulation. The remaining part of the progress bar is shown in yellow (not on Mac).
From this state you can Resume or Cancel the Monte Carlo simulation.
This is useful if you wish to perform other work on the computer and the processing is making the computer too slow.

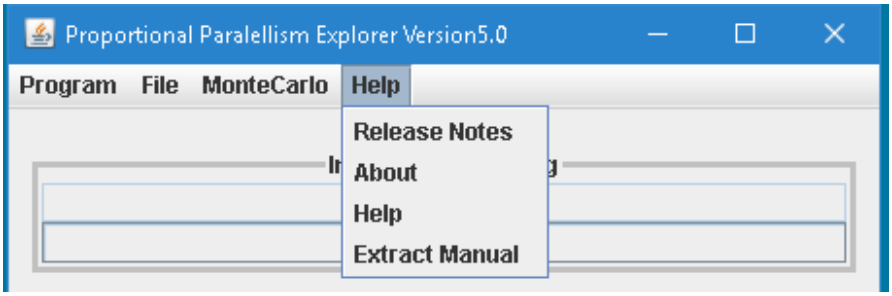
9.3 Resume Monte Carlo

This resumes the paused simulation.

9.4 Cancel Monte Carlo

The Cancel Monte Carlo item is available while the simulation is running or is paused and will stop the simulation. This will be indicated with a message on the screen and in the output file. The remaining part of the progress bar is shown in red (not on Mac).
This can be used if the simulation is taking too long and you wish to restart it with other preference settings (e.g. fewer samples).
A message is displayed with the summary – see 3.10.
The output files are produced with the results so far.

Chapter 10 Help Menu



The help menu contains the items Release Notes, About, Help and Extract Manual.
Release Notes shows the version history of the program with the changes made to each version.
About gives the origin and development information of the program.
Help gives assistance with extracting the manual.
Extract Manual will extract this user manual document from the .jar file and open it. It first tries to extract it to the directory in which the .jar file is located, and if that fails, tries the user's home directory. The location is shown in a confirmation message. The manual will then be opened in the PDF reader. If any of these operations fails, an appropriate message is given – see Help for alternatives.

Chapter 11 Use Cases

11.1 Try Various Proportions

If you wish to generate various proportions for the same input file, set the output file preferences to include the Proportion but not the Timestamp.
Set the proportion and run the program (click “Start from File” in the File menu) for all desired proportions. The output files will be differentiated by the proportion in their names.

11.2 Try Various Inputs

If you wish to alter the input file, e.g. by adjusting the lengths of the pieces, set the output file preferences to include the Timestamp.
You can repeatedly edit and save the input file and run the program. The output files will be differentiated by the timestamp in the file names and you will not risk overwriting one with the next.
Alternatively, use different names for the input files to indicate their purpose. This will be reflected in the output file names and the Timestamp is not necessary.

Chapter 12 Performance

Note that each additional piece doubles the number of combinations and so potentially doubles the time the program will take to run to completion. The time also depends on the number of ways the target can be reached with the given lengths as well as the processor speed of the user's computer.
It would be desirable to show an estimate of the time remaining with the progress bars in the main window, but from the nature of the search, which skips a number of combinations, it is clear that a useful estimate cannot be obtained. The situation for Monte Carlo simulation is similar, as it skips combinations which cannot give a solution, and also depends on the solution search algorithm.
The file processing and Monte Carlo simulation can be run simultaneously. Since they run in separate threads, they will use separate processors if your computer has multiple processor cores. Multiple instances of the program can also be run simultaneously to work on different input files or run with different parameters (see 2.3). Even if you do not have enough processor cores for these to run faster, they can all be left overnight or over several days to perform all the tasks without further intervention.

Chapter 13 Large Files

Some output files are too large to open in Excel. The following techniques can be used to access them.

Split the file

Split the file into a number of smaller files, e.g. of one million lines each. On MacOS the command is:

```
split -l 1000000 <file>.csv
```

On Windows there is no native command, so either install the Windows Subsystem for Unix to use the above command, or use a third-party utility or script.

Note that the file must be split by lines rather than file size so that lines are not split between files.

Search the file

To find a particular line use a search facility, e.g. on MacOS to find the pattern in layer 1:

```
grep -c "1;Left=Right" <file>.csv
```