

grafgen
User's Manual

May 15, 2006

Contents

1	Introduction	1
2	Technical Requirements and Installation	2
2.1	Getting Grafgen	2
2.2	System Requirements	2
2.3	Installation	2
3	Computations Principle	2
3.1	Genotypes	3
3.2	Breeding Scheme	3
3.3	Scanned Point	3
4	How to write input files	4
4.1	Writing a <code>codsys</code> file	4
4.2	Writing an <code>infile</code>	5
4.2.1	General Parameters	5
4.2.2	Mating Systems	5
4.2.3	Case of the additional locus	6
4.2.4	Describing individuals	6
4.2.5	Important Note	7
5	Outputs	7
5.1	Numerical Output	7
5.2	Graphical Output	8
5.2.1	Genotype Probability	8
5.2.2	Allele Dose	10
5.2.3	Allele Frequency in a Population	10
5.2.4	Zones of Highest Probability	11
6	Customizing the Behaviour of grafgen	12
6.1	Computations	12
6.2	Graphics	13

1 Introduction

grafgen is a program aimed at representing individuals in populations issued from known pedigrees. For each offspring of the pedigree, **grafgen** uses information on mapped marker data in the whole pedigree to produce a *Precision Graphical Genotype* (Figure 1).

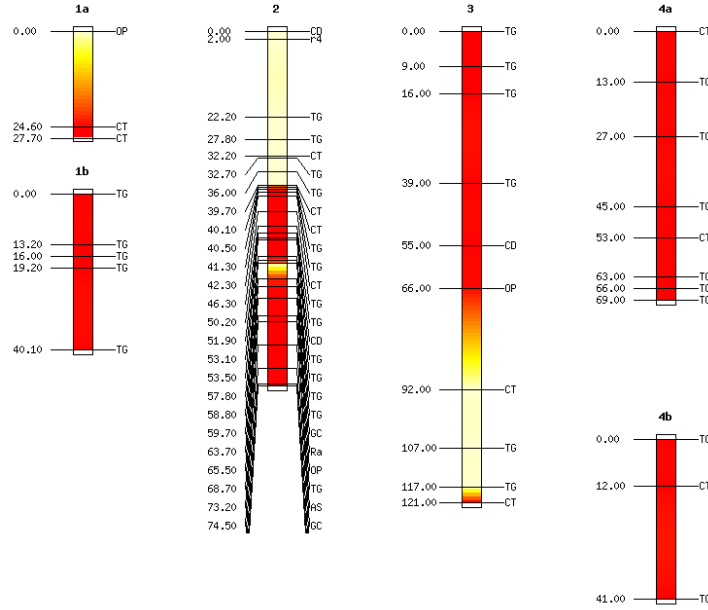


Figure 1: A precision graphical genotype

In order to draw the *Precision Graphical Genotype* of an individual, **grafgen** scans the genome at equally spaced points and infers the genotypes of these points using information on :

- – the *genotypes of markers* linked to the scanned point
- – the *breeding scheme* of the pedigree
- – the genotypes at markers of the individual's ancestors.

The computations performed by **grafgen** to infer the genotype of these scan points are obtained through the use of the core computation function of the program **mdm** (Servin *et al.* , 2002).

This User's Manual is aimed at describing the way **grafgen** works (section 3 – Computations Principle), how to make it work with your particular data (section 4 – How to Write Input Files) what kind of representation you can obtain using **grafgen** (section 5 – Outputs) and how to customize its behaviour (section 6 – Customizing **grafgen** behaviour).

2 Technical Requirements and Installation

2.1 Getting Grafgen

grafgen is freely available at <http://moulon.inra.fr/~fred/grafgen>. You can either download the C source code in order to compile **grafgen** for your system, or download an executable for your operating system (Linux or Windows).

2.2 System Requirements

grafgen uses the GD Library version 2 <http://www.boutell.com/gd> which must be installed on the system in order to run the program. The windows library is provided in the **grafgen** package. For linux, this library is included in all major Linux distributions and must be installed prior to running **grafgen**.

2.3 Installation

Executables The executables (pre-compiled binaries) are in the **binaries** directory included in the **grafgen** package. In this directory, chose the sub-directory corresponding to your operating system and follow the instructions in the file **INSTALL.txt**.

Compiling from source The source code of **grafgen** is distributed in the package. Instructions for compiling **grafgen** are given in the file **COMPILING.txt** located in the **source** directory.

If you successfully install **grafgen** on your system, we would be happy if you let us know by e-mail to grafgen@moulon.inra.fr.

3 Computations Principle

Before drawing the *Precision Graphical Genotype* of an individual, **grafgen** infers the genotype at each scanned point on the genome. This inference is based on the computation of the probabilities of all possible genotypes at the scanned point conditional on pedigree information (*i.e.* genotype at markers, breeding scheme and ancestors' genotypes). These probabilities are computed using the core computation function of the **mdm** program previously developped in our lab. If you want to learn more about **mdm** you can visit the **mdm** web site at: <http://moulon.inra.fr/~fred/mdm>.

This section is aimed at describing the way **grafgen** performs its computations. Although understanding how the program works is not necessary, it can help you to write input files. If you just want a description of the input files format, jump to section 4.

3.1 Genotypes

grafgen considers individuals described by their genotypes at loci of known positions on chromosomes (typically mapped marker loci). In practice, the genotyping of an individual produces an observation (*i.e.* 'phenotype') that poorly reflects its true genotype. Indeed, usually the marker phenotypes do not provide the gametic linkage phase of the chromosomes (which allele originates from which parent), for example in the case of a double or multiple heterozygote. Furthermore, genotyping data may not be fully informative, because of missing or incomplete data (*e.g.* in the case of dominant markers). So, the program distinguishes between 'observed genotypes' (OG), allowing missing or incomplete genotyping data and 'true genotypes' (TG), where all alleles at all loci as well as the gametic phase are assumed to be known. The core computation function of **grafgen** uses the recursion equation from Hospital *et al.*, 1996, which are implemented to compute frequencies of genotypes known without ambiguity (*i.e.*, TG). However, the user actually input genotypes that may include missing or incomplete genotyping information (*i.e.*, OG). Hence, the program needs to know the relationships between OG and TG. These relationships are given in the **codsys** file (see section 4.1). According to the coding system, OG at each generation are converted into all possible sets of corresponding TG. Then, the probabilities of transition between all possible sets of TG are computed according to the recursion equations of Hospital *et al.* (1996). Finally, these probabilities are summed to provide the probability of the OG at the next generation.

3.2 Breeding Scheme

grafgen computes expected frequencies of *offspring* individuals issuing from a *breeding scheme*. The breeding scheme is a succession of generations of mating between *ancestors*. For each generation of the breeding scheme, the input consists of:

- The genotypes of the ancestors (possibly missing or unknown)
- The *mating system* used to mate ancestors

grafgen can cope with different mating systems: hybridization, backcrossing, selfing, fullsib mating or doubled haploids. It can cope with some particular case of random mating, when all individuals in the population at a given generation are mated at random, *unconditional on their marker genotypes* (*i.e.* the corresponding ancestors genotypes for this generation are completely unknown).

3.3 Scanned Point

In order to infer the genotype at each scanned point, **grafgen** needs to include an additional locus on the genome, positionned on the scanned point. **grafgen** first computes the expected probabilities of the genotype at marker loci (m) (\mathcal{F}_m). Then **grafgen** includes the additional locus (non marker locus X), at the position of the scanned point. This allows

to obtain the joint expected probabilities of the genotype at markers plus the additional locus (\mathcal{F}_{m+X}). Finally, the conditional probabilities ($\mathcal{P}_m(x)$, where x is the position of the scanned point) of genotypes at this additional locus, given the genotype at markers are computed as:

$$\mathcal{P}_m(x) = \frac{\mathcal{F}_{m+X}}{\mathcal{F}_m}$$

It is possible to compute this probability for more than one genotype at the scanned point. **grafgen** will allocate in turn the different possible genotypes to the additional locus and compute \mathcal{F}_{m+X} for each of those.

grafgen can use all the information on the breeding scheme to infer the probabilities of different genotypes at a scanned point. When doing the genome scan, **grafgen** will change the position of the additional locus according to the step defined (see section 6) and compute the conditional probabilities on each of the scanned points.

4 How to write input files

The examples of input files provided in the **grafgen** package can help you to write your own input files (see the **examples** directory). **grafgen** gathers information on the breeding scheme from two input files. The first (herein called **codsys** file) is the file containing the coding system. The second file (herein called **infile**) contains the detailed description of the breeding scheme.

4.1 Writing a codsys file

We explain below how to write your own **codsys** file but in most cases, you can use the **codsys** file provided with the **grafgen** package (*i.e.* the file **codsys.ex** in the **examples** directory), for pedigrees involving two founders.

The **codsys** file contains the correspondence between OG and TG (see 3.1). It allows to work with any genotype-coding system used in a particular experiment. The **codsys** file is of the form:

```
nball
0       $i_{0,1}/j_{0,1}$     $i_{0,2}/j_{0,2}$    ...
1       $i_{1,1}/j_{1,1}$     $i_{1,2}/j_{1,2}$    ...
:
k       $i_{k,1}/j_{k,1}$     $i_{k,2}/j_{k,2}$    ...
:
n       $i_{n,1}/j_{n,1}$     $i_{n,2}/j_{n,2}$    ...
```

The first line of the file only contains the maximum number `nball` of alleles per locus, that is the number of different founders in the pedigree. The alleles are indexed from 0 to `nball-1`. The next lines define the OG codes, used to describe individuals in the other input file. Each OG is described by the list of its corresponding TGs. Each of these lines contains: first the OG code (say k), then the corresponding couples $\{(i_{k,l}/j_{k,l}), l \in [1, N_k]\}$. A couple $(i_{k,l}/j_{k,l})$ list the maternal / paternal allele defining the TG l corresponding to the OG k . N_k is the number of TG corresponding to the OG k . For example, an heterozygote for alleles 0 and 1 which gametic phase is unknown and arbitrarily coded 1 will be described by the line:

```
1      1/0   0/1
```

Appendix I contains an annotated example of `codsys` file in a biallelic breeding scheme.

4.2 Writing an infile

The `infile` contains all the information related to the breeding scheme, once the coding system is defined. The file is composed of different sections which we describe in turn. The structure of the file, and the syntax of the lines must be matched exactly (including line-breaks and blank spaces). If not, the program will not work and return error messages. Appendix II gives an example of `infile` corresponding to a simple breeding scheme leading to an F_3 population with two offspring studied. The codes used to describe individuals are the same as in the `codsys` file in Appendix I.

4.2.1 General Parameters

The first 4 lines of the `infile` describe the constant parameters of the breeding scheme: the number of generations (`GENER`), the number of individuals in the final population (`OFFSPRING`), and the number of genotypes to allocate in turn to the additional locus in the offspring (`EVALUATED GENOTYPES` *i.e.* the numbers of genotypes at the scanned point for which you want to compute $\mathcal{P}_c(x)$). Finally, this section contains the name of the `codsys` file (`CODING SYSTEM FILE`), include its full path if the file is not in the working directory.

4.2.2 Mating Systems

The section 'MATING SYSTEM FOR EACH GENERATION' of the `infile` allows to describe for each generation the mating system used to mate ancestors. Each line is composed of the index of the generation (noted `Gn:`, where `n` is the generation considered) followed by the type of mating system. The mating systems are described by keywords that `grafgen` can recognize:

- `hyb` or `bc` are the keywords to be used for hybridization or backcross which are in fact identical.
- `self` is the keyword to be used in case of selfing.

- **fs** is the keyword to be used in case of full sib mating and random mating.
- **hd** is the keyword to be used in case of doubled haploids.

Any other word will not be understood by **grafgen**, will bring an error message and stop the program.

4.2.3 Case of the additional locus

As we have seen in section (3.3), it is necessary to include a virtual additional locus positionned at the scanned point to infer the genotype of this scanned point. The section 'ADDITIONAL LOCUS INFORMATION' of the **infile** allows to define the genotype for the additional locus.

The genotype at the additional locus is typically only known for the founders of the pedigree. For the ancestors' generations this genotype is generally not known, in which case it should be coded as complete missing data (code 5 in the **codsys** file provided).

For each generation of ancestor, a line is composed of

- the index of the generation (noted **Gn**: where **n** is the index of the generation, as above)
- the genotype of the maternal ancestor at the additional locus
- then the genotype of the paternal ancestor at the additional locus, if exists.

Thus, in the example of Appendix II, as the mating system used at generation 1 is hybridization (keyword **hyb**), the line of this section of the **infile** contains the genotype of the maternal ancestor followed by the genotype of the paternal ancestor. For generations 2 and 3, as the mating systems used is selfing, each line contains only one genotype, corresponding to the selfed parent.

Finally the last line is the list of genotypes to allocate in turn to the additional locus in the offspring in the final generation (all on a single line). The beginning of the line is **0**:, allowing to distinguish it easily from the preceding lines. For each of these genotypes, the program will compute the corresponding $\mathcal{P}_c(x)$.

4.2.4 Describing individuals

The section 'GENETIC MAP AND MARKER GENOTYPES' of the **infile** allows to describe:

- the genetic map¹ used to describe individuals.
- the genotypes of the ancestors
- the genotypes of the offspring

The first line (capitalized words) contains the columns headings. Each following line describes a single marker locus.

The first column (heading: **CHROM**) contains the indexes of the chromosomes to which belong the markers, each index appearing as many times as there are mapped markers on

¹N.B.: The genetic map used by **grafgen** is constant during the whole breeding scheme. It is therefore assumed that recombination rates between loci are evaluated once (either on the population studied or on another one, *e.g.* if using a consensus or a joint map) and that they are not re-estimated during the breeding scheme.

the corresponding chromosome. The second column (heading: **MK_NAME**) contains the names of the markers. The third column (heading: **MK_POS**) contains the positions (understood as the distance to the telomere, in centiMorgans) of the markers on the chromosomes. The markers must be ordered from the first locus of the first chromosome to the last locus of the last chromosome. When these three columns are filled, the genetic map is defined.

Each next column contains the genotype of an individual in the breeding scheme. The columns whose headings begin with **MAT** contain the genotypes of maternal ancestors, those whose headings begin with **PAT** contain the genotypes of paternal ancestors and those whose headings begin with **OFF** contain the genotypes of the offspring. These columns must be ordered following three rules:

- the ancestors are described before the offspring (*i.e.* a column with a heading that begins with **OFF** is always after the columns whose headings begin with **MAT** or **PAT**)
- the ancestors are described from the first generation to the last (*e.g.* **MAT1** is always before **MAT2**)
- the paternal ancestors are always described after the maternal ancestors (*e.g.* **PAT2** is always after **MAT2**).

Note that only the beginning of the heading line is mandatory (*i.e.* **CHROM**). The further column headers are used to keep track more easily of the genotype data in the input file and are not used by the program.

4.2.5 Important Note

The breeding scheme should start from fully known genotypes (including the additional locus) so that all the allelic descents that follow have a common and unique starting point. Otherwise, all possible founders genotypes are considered equiprobable (which is most likely not what you want).

5 Outputs

Once the input files are composed, you may run **grafgen** in a terminal following the syntax: **grafgen [options] infile**. As the **codsys** file name is provided in the **infile**, there is no need to give it in the command line. The available options are described later in this manual (see section 6). For a short description of these options you can type **grafgen -h** in the terminal.

grafgen will then run and produce different output files which have to be read externally. The default prefix used for output files is **pgg** but can be defined by the user with an option (see below).

5.1 Numerical Output

grafgen writes the probabilities of all genotypes for each scanned point in a file called **pgg.dat** (or **PREFIX.dat**). This file is a CSV file and can be read with a spreadsheet program such as **gnnumeric** or **Excel** or a mathematical program such as **R** or **Splus**.

5.2 Graphical Output

The graphical output of **grafgen** is stored in a picture file in PNG or JPEG format. A PNG file is generally of better quality but bigger than a JPEG file. Depending on the type of information you want to get from the **grafgen** analysis of the data, you may want to produce different type of graphical outputs. First of all, **grafgen** can draw the genetic map of the genome. This sets the template of all other graphical representations. As an example, Figure 2 gives the graphical representation of the IBM maize genetic map of chromosome 9 (<http://www.maizemap.org>) produced by **grafgen**. Using this graphical map as a skeleton, **grafgen** will then paint the chromosomes according to the information to be graphically displayed : the color of a point on a chromosome then depends only on the criterion used to described the genotype at this scanned point. In the next examples, we show the possible ways to represent the genotype of an individual. We will use the map represented in Figure 2 as the skeleton for these representations.

TIP Running **grafgen** to produce a genetic map is a good way to test the format of your input files, as no computations are performed. If the program does not return any error, then your input file should be correct. You can also easily check that the map drawn matches your input.

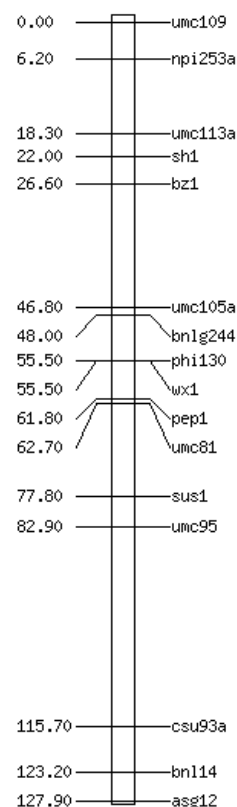


Figure 2: Maize IBM Genetic Map of Chromosome 9

5.2.1 Genotype Probability

In some cases it is interesting to show the probability that an individual is of a particular genotype at each position of the genome. This is the case for example in backcross breeding where only two genotypes are possible: homozygous for the recipient parent allele (genotype RR) or heterozygous donor parent / recipient parent (genotype DR). In this case plotting the probability of being of genotype RR allows to assess both the return to the recipient parent and the conservation of donor alleles in introgressed regions. Figure 3 shows a chromosome of a BC_3 individual carrying a donor segment to be introgressed between markers **bnlg244** (position 48 cM) and marker **pep1** (position 61.8 cM). The figure 3 shows that, as the markers **umc109**, **np1253a** and **csu93a** remain heterozygous (DR), the rest of the chromosome also remains heterozygous between the markers. It can also be interesting to plot the conditional probabilities of genotypes on the genome of individuals in segregating populations used for QTL detection: at each scanned position, one way to perform QTL detection is to regress the phenotype of individuals on the probability of being of genotype qq, qQ or QQ. It can thus be useful to have a graphical representation of individuals based on their probability of being of any of these three genotypes. Figure 4 shows three representations of the chromosome of a single individual colored according to the probabilities of being of genotype qq, qQ and QQ respectively for Figure 4a, Figure

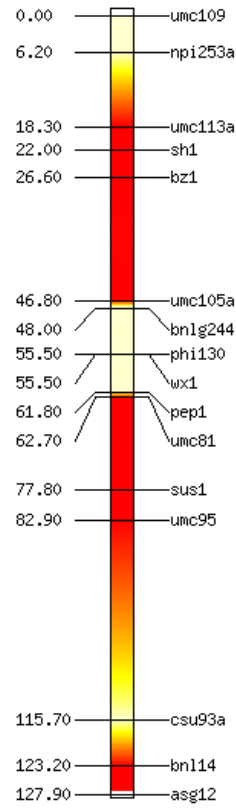


Figure 3: *Precision Graphical Genotype* of a BC_3 individual

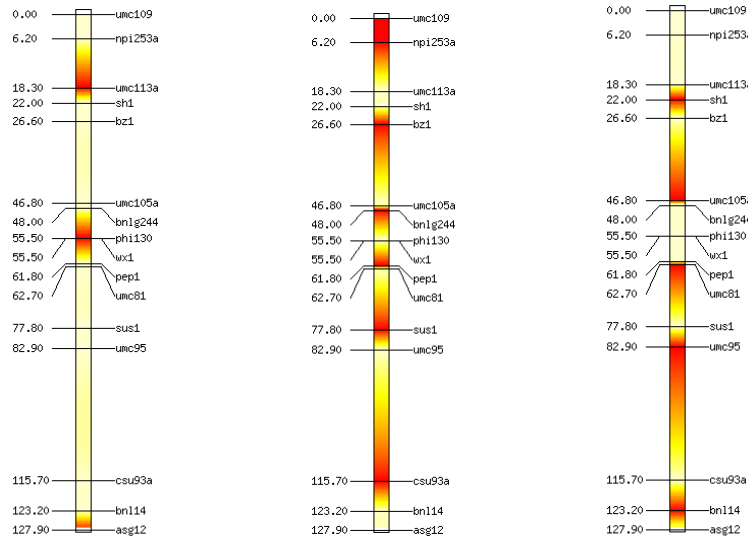


Figure 4a $P(0/0)$ **Figure 4b** $P(0/1)$ **Figure 4c** $P(1/1)$

Figure 4: Probabilities of 3 different genotypes in an F3

4b and Figure 4c.

5.2.2 Allele Dose

One way to summarize the information provided by the previous probabilities is to plot a single graphical representation based on the dose of one particular allele on the genome, which depends on the probabilities of every genotypes (TG) that contains at least one copy of this allele. The Figure 5 is the representation of the dose of allele q on the same data as Figures 4a, 4b and 4c.

5.2.3 Allele Frequency in a Population

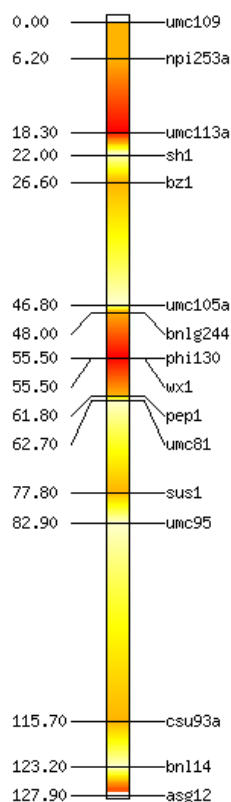


Figure 5: Dose of allele q in an F3 individual

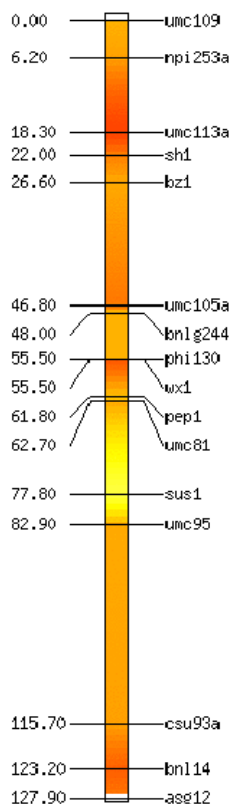


Figure 6: mean dose of one parental allele in an F3 population

When analysing a *population* of offspring, **grafgen** can be run to obtain graphical representations of each individual. However, it is sometimes interesting to display data concerning the population as a whole. This is for example the case when working on a segregating population (for marker mapping and QTL detection). By plotting the mean frequency of an allele in the population, it is possible to assess graphically the regions of

the genome which show a distortion of segregation. This feature is useful for example in a population that has been selected on the basis of the phenotypes of the ancestors. Indeed it is possible *a posteriori* to assess the genome regions that have been selected as they will present a distortion of segregation with a highest frequency of the favorable allele in the population.

The figure 6 represents the mean frequency of one of the segregating alleles in an F3 population derived from two inbred lines. The offspring is composed of 60 individuals which genotypes at markers have been chosen arbitrarily. The chromosome shows zones of normal frequency (orange zones), zones of high allele frequency (red zones) and zones of low allele frequency (yellow zones). Note that in this case the color contrast around 0.5 is increased as mean allele frequencies are close to 0.5. This leads to a fastest saturation of colors toward high or low values.

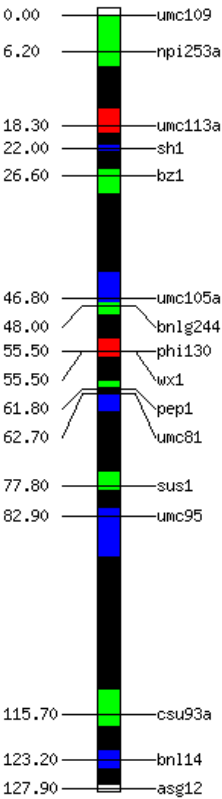
5.2.4 Zones of Highest Probability

The graphical representation that we have presented so far are nearly continuous. It can sometimes be useful to obtain a simpler representation of genotypes, for example to sort individuals in a population according to their genotypes. **grafgen** allows to discretize the values computed. In this case, the program will color in red, green or blue the zones where three different genotypes have a probability higher than a given threshold. The behaviour of **grafgen** is explicitly determined through the use of a very simple input file containing three columns. On each line is given : first the genotype to plot, then the threshold for the probability and finally the color to allocate to the zones where the probability of the genotype is higher than the threshold. For example with this input file:

#geno	threshold	color
0	0.8	red
1	0.8	green
2	0.8	blue

Figure 7:
Threshold
representation
of genotype
0/0 (red), 0/1
(green) and 1/1
(blue) at a 0.8
cutoff

The line beginning with a # is understood as a comment by **grafgen**. The genotype codes in the first column correspond to the **codsys** file in Appendix I. With this input file, the genome regions where an individual has a probability of being homozygote 1/1 greater than 0.8 are to be colored in blue, regions where the probability of genotype 0/0 is greater than 0.8 are to be colored in red and regions where the probability of genotype 0/1 is greater than 0.8 are to be colored in green. Regions that do not match any of these criteria are to be colored in black. An example of a resulting representation is given on figure 7.



6 Customizing the Behaviour of grafgen

6.1 Computations

In the examples above, we have always presented figures of a single chromosome. However, **grafgen** can draw precision graphical genotypes of the complete genome or of chromosomal segments. The precision on the computations of the genotype probabilities can be increased by increasing the density of the scanned points and / or by increasing the amount of information taken into account to infer genotypes probabilities at a given scanned point. The behaviour of **grafgen** can thus be customized with computation options that allow to define the range for which the *Precision Graphical Genotype* is to be plotted, the step to use for the scan (in cM) and the amount of information to be taken into account.

Plotting Range: -c -b -e options By default, **grafgen** will plot the *Precision Graphical Genotype* on the whole genome. This behaviour can be forced by using the **-g** option. It is possible to run the program on smaller segments of the genome :

- the **-c** option allows to scan a single chromosome defined by its index (*e.g.* **grafgen -c 2 ...** will scan the chromosome 2 only).
- It is possible to scan only a segment of a given chromosome, for example in particularly interesting regions (introgressions, QTL locations ...). This option is conditioned on the previous declaration of a chromosome to scan (**-c** option). The segment is defined by the beginning value (**-b** option) and the ending value (**-e** option). For example the command line **grafgen -c 2 -b 10 -e 40 ...** will scan the chromosome 2 at locations ranging from 10 cM to 40 cM.

Computation Step: -x option By default **grafgen** will compute the conditional probabilities of the different genotypes every centiMorgan. This behaviour can be changed by using the **-x** option to change the step used for the computations. The step is given in centiMorgans. Hence, the command line **grafgen -x 10 ...** will perform a scan every 10 cM. This option allows to reduce the time needed for the computations if using a large step (*e.g.* 10 centimorgans) or to increase the precision on the genomic composition estimate if using a small step (*e.g.* 0.2 cM).

Marker Information: -m option The program computes genotypes probabilities conditional on marker information. Generally, the more markers are taken into account in these computations, the more precise are the probabilities' estimates. The **-m** option allows to define the number of markers to be taken into account for the computations. The argument given to this option is the number of markers to be taken into account on each side of the scanned position. For example the command **grafgen -m 1 ...** will perform computations conditional on the genotype at flanking markers only, one on each side of the scanned position (this is the default behaviour). Note however that increasing the number of markers to be taken into account will increase the computation process exponentially.

6.2 Graphics

The graphical options allow to define the representation that **grafgen** will produce.

Output Format: -f option The option **-f** allows to define the type of format to be used for output, described by two strings : **jpg** or **png**. Hence the command **grafgen -f jpg ...** will produce a file in jpeg format. The default format is **png**.

Information displayed: -T -t options The **-T** option is used to define the type of representation that **grafgen** will produce on output. For some of these options, a second argument is needed and must be given to the program using the **-t** option.

- *No Output*: “**grafgen -T 0**” leads to performing computations and exiting, without producing any picture.
- *Genetic Map*: “**grafgen -T 1**” leads to produce a picture of the genetic map. This is the default behaviour. Thus the command **grafgen infile** or **grafgen -T 1 infile** provides a graphic representation of the genetic map in a file called **pgg_map.png**. In this case no computation is performed.
- *Conditional Probability*: “**grafgen -T 2 -t 0**” produces on output chromosomes colored according to the probability of being of genotype code 0 (*e.g.* 0/0 as defined in the **codsys** file in appendix I). The **-t** option is mandatory here to indicate the genotype to plot.
- *Allele Dose*: “**grafgen -T 3 -t 0**” produces on output chromosomes colored according to the dose of *allele* 0. Note that in this case, 0 does *not* mean the same thing as in the **-T 2** context (OG *vs.* allele index). The **-t** option is mandatory here to indicate the allele to plot.
- *Allele Frequency*: “**grafgen -T 4 -t 0**” will produce a single output file as shown in figure 6, representing the frequency of allele 0 in a population. The **-t** option is mandatory here to indicate the allele to plot.
- *Zones of high probability*: “**grafgen -T 5 -t thresholds.in**” will produce a representation bringing out the zones of highest probabilities of given genotypes. Using this option requires to create a simple input file (see 5.2.4 for an example) which name as to be given using the **-t** option (**thresholds.in** here). The **-t** option is mandatory here to indicate the file to use.

Project Title: -P option The **-P** option can be use to change the prefix used in the name of output files. The command line is then : **grafgen -P ''MyPrefix''** The name of output files will be prefixed with the supplied prefix. The default prefix is **pgg**.

References

- Hospital F, Dillmann C and Melchinger AE, 1996. A general algorithm to compute multilocus genotype frequencies under various mating systems. *Comput Appl Biosc.* 12: 455-462
- Servin B, Dillmann C, Decoux G and F Hospital, 2002. MDM: a program to compute fully informative genotype frequencies in complex breeding schemes. *J Hered* 93: 227-228.

APPENDICES

Appendix I

Example of a codsys file for a biallelic breeding scheme

```
2
0 0/0
1 0/1 1/0
2 1/1
3 0/0 0/1 1/0
4 0/1 1/0 1/1
5 0/0 0/1 1/0 1/1
6 0/1
7 1/0
```

OG-codes 0 and 2 correspond solely to genotypes homozygous for allele 0 and allele 1, respectively. Code 1 corresponds to an heterozygous locus which gametic phase is unknown. If the gametic phase is known, the different heterozygous loci can be coded differently, as exemplified by the codes 6 or 7. Using codes 6 or 7 increases the precision in the calculations (when combining genotypes over all loci) if the phase is known, for example when the genotype is an F_1 hybrid resulting from the cross between two completely homozygous parents. Codes 3 and 4 correspond to the case of dominance of allele 0 or 1, respectively. Finally code 5 corresponds to completely missing data.

Appendix II

Example of an infile

```
GENER = 3
OFFSPRING = 2
EVALUATED GENOTYPES = 3
CODING SYSTEM FILE = codsys.ex
```

MATING SYSTEMS FOR EACH GENERATION :

```
G1: hyb
G2: self
G3: self
```

ADDITIONAL LOCUS INFORMATION :

```
G1:    0 2
G2:    5
G3:    5
0:    0 1 2
```

GENETIC MAP AND MARKER GENOTYPES :

CHROM	MK_NAME	MK_POS	MAT1	PAT1	MAT2	MAT3	OFF1	OFF2
1	m1	0.0	0	2	1	5	0	1
1	m2	50.0	0	2	1	5	0	2
1	m3	100.0	0	2	1	5	0	2
2	m4	0.0	0	2	1	5	2	0
2	m5	75.0	0	2	1	5	2	0

This input file describes a pedigree spanning three generations (**GENER = 3**) and producing two offsprings (**OFFSPRING = 2**). The program will compute the probabilities of three genotypes along the genome (**EVALUATED GENOTYPES = 3**). The genotypes codes are provided in a file called **codsys.ex**, which is shown in Appendix I.

The first generation of the pedigree is an hybridation between two founders (**G1: hyb**) followed by two generations of selfing. The two founders of the pedigree are inbred lines:

- the first founder is of genotype 0 at all markers (column **MAT1**) and the second founder is of genotype 2 at all markers (column **PAT1**).

- the two founders are also of genotypes 0 and 2 respectively between the markers (*i.e.* at any additional locus). This is specified in the additional locus information at line 14 **G1:** 0 2.

The additional locus information is only known for the founders, and is missing at other generations (genotype code 5 at generations 2 and 3). The program will compute the probabilities of genotypes 0, 1 and 2 in the offsprings.

At generation 2 of the pedigree, an *F1* hybrid is obtained: this individual is heterozygote at all loci (column **MAT2**). This *F1* is selfed to produce an *F2* at generation 3. This *F2* was not genotyped: column **MAT3** is filled with missing genotypes (genotype code 5).

Finally two *F3* offsprings are obtained by selfing the *F2*. The genotypes at markers of the offsprings are indicated in columns **OFF1** and **OFF2**.

The markers are located on two chromosomes (1 and 2). The first chromosome has three markers (m1, m2 and m3) at positions (0cM, 50cM, and 100cM). The second chromosome has two markers (m4 and m5) at positions (0cM and 75 cM).