

MANUAL: snp5-probeset-genotype (SNP5-1.0)

Contents

- [Introduction.](#)
- [Quick Start - getting up and running.](#)
 - [Running BRLMM \(Mapping 500K and preceeding chips\)](#)
 - [Running BRLMM-P \(SNP5.0 chips\)](#)
 - [A note about the CHP file format](#)
- [Support.](#)
- [The Report File - explanation of contents.](#)
- [Program Options - command line options.](#)
- [FAQ - Frequently Asked Questions.](#)
- [Advanced Topics](#)
 - [CHP file format - Same format, different values to DM genotyping CHPs.](#)
 - [Memory \(RAM\) issues - a common source of difficulty.](#)
 - [Custom Analyses - how to vary the way calls are made.](#)
 - [Clustering transformations - different possible clustering spaces.](#)

Introduction

snp5-probeset-genotype is a program for making genotype calls from Affymetrix SNP microarrays. It currently implements two different genotype calling algorithms:

- BRLMM (pronounced "B-realm") a model based approach similar to the RLMM (pronounced "realm") model developed by Nusrat Rabbee and Terry Speed (<http://www.stat.berkeley.edu/users/nrabbee/RLMM/>) but with a Bayesian extension, hence the "B". Currently requires seed genotypes for every SNP and sample analyzed which it obtains via the DM algorithm, hence it requires the presence of MisMatch (MM) probes.
- BRLMM-P - a model-based approach which performs 1-dimensional clustering by fitting a Gaussian mixture model. No requirement for seed genotypes, hence can run on SNP chips without MM probes.

As brlmm and brlmm-p are model based algorithms they need to be run on multiple CEL files at once to estimate probe effect and SNP cluster parameters. For Mapping 500K data it is advisable to run on at least 50 distinct samples (excluding replicates) and ideally on about 100.

This version of the software has been developed and tested exclusively on SNP5.0 data. It is possible that it may also yield useful results on other chip types but that has not been tested, proceed with caution.

Quick Start

We illustrate the most basic way to run snp5-probeset-genotype with some examples.

The basic requirements for a run of `snp5-probeset-genotype` are:

- A collection of CEL files to process. For this example the CEL files are all located in a directory called 'cel_dir'. Note that the program will run *much faster* if the CEL files are located on a local disk as opposed to being read across a network. For better overall performance it is advisable to filter CEL files, running the analysis only on those passing a certain specification based on the per-sample QC metric. For Mapping 500K data the recommended per-chip spec is a call rate of 93% using DM at a threshold of 0.33.
- The CDF file corresponding to the array type of the CELs. CDF files can be downloaded from <http://www.affymetrix.com/support/technical/libraryfilesmain.affx>
- A chrX file consisting of the SNP IDs for all non-pseudo-autosomal chrX SNPs on the array. chrX files can be downloaded from the Affymetrix product page at <http://www.affymetrix.com>
- For BRLMM-P, a models file consisting of SNP-specific priors specifying for each SNP where the 3 genotype clusters are likely to be located.
- A directory where results will be stored. For this example it is called 'results_dir'

Running BRLMM (Mapping 500K and preceding chips)

On unix systems a basic command using the default parameters to do a run on Mapping 500K data would look like:

```
snp5-probeset-genotype \  
-o results_dir \  
-c Mapping250K_Sty.cdf \  
--chrX-snps Mapping250K_Sty.chrx \  
cel_dir/*.CEL
```

The output will consist of a report file with some summary statistics about each chip analyzed and a pair of tab-delimited text files with suffixes `.calls.txt` and `.confidences.txt` containing the genotype calls and their associated confidences. On windows the DOS prompt does not support wildcard expansion and the preferred method is to supply a text file with the path to the cel files via the '--cel-files' option (see below for details of file format). The windows DOS prompt also does not allow a continuation of a command with the '\ ' character, unlike unix.

On windows a command equivalent to the example above for Mapping 500K would look like:

```
snp5-probeset-genotype -c Mapping250K_Sty.cdf --chrX-snps  
Mapping250K_Sty.chrx -o results_dir --cel-files  
cel_file_list.txt
```

Note that there is a known bug with use of the --cel-files option in the windows version of the software. The "\" path separator is not being correctly handled, so for example an entry "C:\data\sample1.CEL" will result in an error about not being able to read the CEL file. This issue can be avoided by using "/" instead of "\" - so the example above could be handled correctly if specified as "C:/data/sample1.CEL"

For Mapping 500K chips snp5-probeset-genotype runs 100 CELs in 1-2 hours on a 3GHz 2Mb RAM machine using local disk.

Running BRLMM-P (GenomeWide SNP 5.0 chips)

On unix systems a basic command using the default parameters to do a run on SNP5.0 data would look like:

```
snp5-probeset-genotype \  
  -o results_dir \  
  -c GenomeWideSNP_5.cdf \  
  --chrX-snps GenomeWideSNP_5.chrx \  
  --read-models GenomeWideSNP_5.models \  
  -a brlmm-p \  
  cel_dir/*.CEL
```

Note in particular the use of the option "-a brlmm-p" which specifies that the BRLMM-P calling algorithm should be used (the default is brlmm, which won't work on a chip without MM probes such as the 5.0 chips).

WARNING: snp5-probeset-genotype will overwrite any existing output files it finds. If you wish to keep existing results make sure to specify a different output directory name.

Full details on the available options can be found [below](#); here we explain a few of the options used in the most common variants of the above workflow:

```
--txt-output      Output genotype calls and confidences in a  
                  directory named 'txt' under the specified  
                  output directory.  This option creates a  
                  single txt file per CEL file analyzed.  The  
                  txt file has 11 lines of header followed by  
                  tab-delimited text with one line per SNP and  
                  three columns:  
                  Column 1: SNP_ID  
                  Column 2: Genotype call  
                  (-1=NoCall, 0=AA, 1=AB, 2=BB)
```

```
Column 1: Genotype call confidence
-s, --probeset-ids Allows for specification of a subset of SNPs
to which analysis will be restricted. Run
time is directly proportional to the number
of SNPs so using this option can greatly
speed up the run time. Note that it is not
possible to create CHP files with only a
subset of SNPs and so if this option is used
CHP output options must be suppressed with
the
--no-chp-output option. The SNPs to analyze
are identified in a tab-delimited text file
which includes a column named 'probeset_id'
specifying the ids of the SNPs to be
analyzed.
```

Building upon the example above, here is an example in which only a subset of SNPs are analyzed and the results are written to a text table of genotype calls and a text table of call confidences. The subset of SNPs to be analyzed is specified in a tab-delimited text file called `subset_sty.txt`, which must contain a column named `'probeset_id'`.

```
snp5-probeset-genotype \  
-s subset_sty.txt \  
-c Mapping250K_Sty.cdf \  
--chrX-snps Mapping250K_Sty.chrx \  
-o results_dir \  
cel_dir/\*.CEL
```

A note about the CHP file format

In previous versions of `snp5-probeset-genotype` the default output format for genotype calls was the XDA CHP format (also known as GCOS CHP format). For the GenomeWide SNP 5.0 chips and subsequent products the use of the XDA CHP format is strongly discouraged, instead we recommend the newer AGCC CHP format. To help avoid accidental use of the XDA CHP format the defaults for output format have been changed to produce tab-delimited text tables of calls and confidences. The creation of the text table output can be suppressed with the `--no-table-output` option and the two CHP output formats can be selected with the `--xda-chp-output` and `--cc-chp-output` options.

The reason that the XDA CHP format is discouraged for the GenomeWide SNP 5.0 chips is that it doesn't contain entries for SNP IDs, the identity of a SNP is inferred from its order in the file. In the case of the GenomeWide SNP 5.0 chips

there are some SNPs that are not part of the default library file which some advanced users may choose to explore. This leads to the possibility of generating CHP files containing different SNP lists, something not well supported by the XDA CHP format. The AGCC CHP format has a slot for SNP IDs and thus is safer to use with chips for which users may be looking at different SNP lists.

Details on the XDA and AGCC CHP formats can be found at the [Affymetrix Developer's Network](#).

Support

Support for snp5-probeset-genotype is handled through the Affymetrix Developer Network. Specifically, questions, problems, feature requests, and other inquiries should be made through the Developer Network email address, devnet@affymetrix.com. snp5-probeset-genotype is not supported through the Affymetrix call center, Field Application Specialists, or the standard Affymetrix Technical support channels.

If you encounter an issue please make sure to collect the following information and report the problem to devnet@affymetrix.com

- Contents of program log file (a file called snp5-probeset-genotype.log is created in the output directory).
- Program output - cut and paste everything that the program reports to the screen.
- The specific command used.
- Type of machine (operating system, amount of memory).

The Report File

snp5-probeset-genotype creates a summary report file in the output directory with file name extension '.report.txt'. The report file contains some summary information about each chip analyzed and is useful in getting a quick overview of the CELs analyzed. The format of the file is tab-delimited text with a header line followed by a line for each CEL file analyzed. The columns are all explained below, most users will be mainly interested in the first few entries. The additional entries are provided as potentially useful metrics to track and identify outlier chips and are expected to be mainly of interest to advanced users. The column entries are:

1. The CEL file name.
2. The estimated gender (based upon DM chrX calls at confidence of 0.33).
3. The BRLMM or BRLMM-P call rate at the default or user-specified threshold.
4. The percentage of SNPs called AB (i.e. the heterozygosity).
5. The percentage of SNPs called AA.
6. The percentage of SNPs called BB.
7. The average of the raw PM probe intensities.
8. The standard deviation of the raw PM probe intensities.
9. The average of the allele signal estimates (log2 scale).


```

--read-models (BRLMM) File of precomputed snp specific models
                    or snp specific priors (BRLMM-P)
-s, --probeset-ids Allows for specification of a subset of SNPs
                    to which analysis will be restricted. Run
                    time is directly proportional to the number
                    of SNPs so using this option can greatly
                    speed up the run time. Note that it is not
                    possible to create XDA CHP files with only a
                    subset of SNPs and so this option cannot be
                    used at the same time as the --xda-chp-
output
                    option. The SNPs to analyze are identified
in
                    a tab- delimited text file which includes a
                    column named 'probeset_id' specifying the
ids
                    of the SNPs to be analyzed.

Output:
-o, --out-dir       Directory into which result files and
                    directories will be written. WARNING: any
                    previously-generated results in the
                    directory will be overwritten.
--table-output      Output genotype calls and confidences in a
                    pair of tab-delimited text matrices named
                    '*.calls.txt' and '*.confidences.txt' in the
                    specified output directory. The format of
                    each file is a few comment lines prefixed
                    with '#', a header line specifying the CEL
                    file names and then a line per SNP, the
                    first field is the SNP_ID and subsequent
                    fields contain genotype calls or
                    confidences. Calls are encoded as -
1=NoCall,
                    0=AA, 1=AB, 2=BB
--no-table-output   Don't output matrices of calls and
                    confidence values
--cc-chp-output     Output genotype calls and confidences
                    using Command Console CHP format in a
directory
                    called 'cc-chp' under out-dir.
[experimental:
                    CC CHP file output content/format will
change!]

```

file

Makes one CHP file per CEL file analyzed. [default: false]. A description of the CHP format can be found at <http://www.affymetrix.com/support/developer> (look for 'File Formats' section)

--xda-chp-output Create CHP format genotype calls and confidences in a directory called 'chp' under out-dir. Makes one CHP file per CEL file analyzed. [default: true]. A description of the CHP file format can be found at <http://www.affymetrix.com/support/developer>

--no-chp-output Don't output GCOS XDA or CC CHP files.

--table-output Output genotype calls and confidences in a pair of tab-delimited text matrices named '*.calls.txt' and '*.confidences.txt' in the specified output directory. The format of each file is a few comment lines prefixed with '#', a header line specifying the CEL file names and then a line per SNP, the first field is the SNP_ID and subsequent fields contain genotype calls or confidences. Calls are encoded as -

1=NoCall,
0=AA, 1=AB, 2=BB

--txt-output Output genotype calls and confidences in a directory named 'txt' under the specified output directory. This option creates a single txt file per CEL file analyzed. The txt file has 11 lines of header followed by tab-delimited text with one line per SNP and three columns:

Column 1: SNP_ID
Column 2: Genotype call
(-1=NoCall, 0=AA, 1=AB, 2=BB)
Column 1: Genotype call confidence

--summaries Output the allele signal estimates for each allele in a file with suffix .summary.txt located in the output directory. The format of the file is tab-delimited text with one row for each allele of each SNP. The first column is the SNP_ID and the allele, the subsequent columns contain the estimated allele signal for each CEL file.

```

--feat-effects Output the empirically-determined feature
(or probe) effects produced in the allele
summarization step. Results are written to
a file with suffix .plier.feature-
response.txt

located in the output directory. This file
can be used in future analyses with the
--use-feat-eff option. The format of the
file is tab-delimited text with one row per
PM probe or PM,MM probe pair. The columns
are:
    1 - probeset_id (i.e. SNP_id)
    2 - atom_id: identifier for PM probe or
        PM,MM probe pair, unique within a SNP.
    3 - probe_id: identifier for PM probe or
        PM,MM probe pair, unique within the
        chip.
    4 - feature_response: the estimated
        feature or probe effect.

--residuals Output the residuals from the quantification
method. Residuals are written as tab-
delimited text and since there is one
residual per PM probe per chip, analyzing
all the probesets will generate huge output.

--write-sketch Write the estimated quantile (or sketch)
normalization distribution to a file named
quant-norm.normalization-target.txt, so that
it can be re-used in future analyses with
the --target-sketch option. Format of the
file is text with the sorted probe
intensities of the target distribution
written one-per-line, with the first line
being 'intensities'.

--write-prior (BRLMM only) Write the estimated generic
prior information on SNP cluster centers and
variances to a file with suffix .prior.txt
in

the specified output directory. This file
can be used in future analyses with the
--prior-file option. The file format is
tab-separated text with entries as follows:
    1 - 6 comma-separated values corresponding
        to the cluster centers Values are
        (center_x,center_y) for AA,AB,BB.
    2 - 9 comma-separated values corresponding

```

```

to the variances estimated for each
genotype. Values are (var_x,cov_xy,
ar_y) for AA,AB,BB
3 - 36 comma-separated values
corresponding to the 6x6 variance
matrix estimated for the cluster
centers.
--write-models Should we write snp specific models out for
analysis? [experimental] This writes out the
snp specific cluster centers and variances
for use with another set of chips. Note that
the same normalization sketch and feature
effects should also be used.
--dm-out (BRLMM only) Output the initial DM generated
calls as a table. Note that these are the
calls
at p-val as determined by the --dm-thresh
call.
Other:
--rm-probes Do not use the probes specified in file for
delimited computing results. [experimental] Tab
coordinates file with an "x", "y", and "probeset_id"
columns corresponding to the x and y
of the features on the array to be removed
and the name of the probeset that the probe
belongs to. For example:
x y probeset_id
1314 1314 AFFX-2315060
1314 1313 AFFX-2315060
--block-size How many probesets to process at once,
useful when memory is limited. If set to 0
program attempts to guess available RAM and
set appropriately. For more info see the
notes on memory in the advanced topics
section below. [default: 0]
-v, --verbose How verbose to be with status messages:
0=quiet, 1=usual messages, 2=more messages.
--version Output program version and quit.
--explain Provide more detail on a particular analysis
operation. Usage is '--explain <method>'
where 'method' can be quant-norm, brlmm,
brlmm-p
plier or med-polish.
-h, --help This message.

```

The following options will change the genotype calls produced.

```
-a, --analysis      String specifying the kind of analysis to be
                    applied. The default value is 'brlmm' which
                    provides the default BRLMM implementation.
                    For SNP5.0 chips use 'brlmm-p'. Advanced
                    users may wish to explore custom analyses
with
                    this option. For further details see the
                    'custom analyses' section below. The values
                    'brlmm' and 'brlmm-p' are convenient aliases
                    for longer strings which fully specify the
for
                    analysis, see the end of this help message
                    the full strings they correspond to. Note
                    that this -a option will determine the
                    base name for many of the output files
                    generated, such as those generated by the
                    options --table-output, --summaries,
                    --feat-effects and --write-prior.
--qmethod-spec      Quantification Method to use for summarizing
                    alleles. Available methods are plier and
                    med-polish. Details on the options
                    available with each of these methods are
                    available by calling snp5-probeset-genotype
with
                    the option '--explain <method>' where
                    <method> is 'plier' or 'med-polish'.
                    [default: plier.optmethod=1]
--max-score         This option is deprecated. To get calls at
a
                    threshold X for a particular calling method
you
                    must now add 'MS=X' to the analysis string
(se
                    e
                    --analysis option). This change allows for
                    different calling methods to have different
                    default thresholds.
-f, --force         Ignore any disagreement between the array
                    types of the CEL files and the array type of
                    the CDF file.
--chrX-force        Perform analysis even without using
```

```

--chrX-snps option. Note that this may
yield sub-optimal calls on chrX snps.
--use-feat-eff File defining a precomputed feature (or
probe) effect to be used for each probe.
Note that precomputed effects should only be
used for an appropriately similar analysis
(i.e. feature effects for a pm-only analysis
may be different to feature effects for
pm-mm). See --feat-effects option for
details on file format.
--target-sketch File specifying a target distribution to
use for quantile or sketch normalization.
See --write-sketch option for details on
file format.
--genotypes File to read seed genotypes from. This file
has
conjunction
the
generate
BRLMM-P
string
that
it evaluates which are inconsistent with the
seeds.

The file should look like:
probeset_id celName1.cel ... celNameN.cel
SNP_1234 -1 0 1 2
SNP_1235 0 1 2 0
...
SNP_NNNN 0 -1 1 2

Where the headers are the names of the cel
files
and
the row names are the probeset names.

If you want, with BRLMM you can do a normal
run

```

example
option
used
--norm-size
--write-norm
a file
fit to
--list-sample
(BRLMM)
estimation
probeset-ids,
validation of
results

and use the new --dm-out flag to get an
of what the file should look like (This
writes out the initial DM seed calls and is
to test that --genotypes is working).
If nonzero, do contrast normalization using
a sample of this many SNPs.
If contrast normalization is performance
(--norm-size option) this option writes out
with the normalization parameters that are
each chip.
Only sample SNPs to be used in the prior
or contrast normalization parameter
(BRLMM-P) from list specified via --
not from the entire chip. Useful in
results from snp5-probeset-genotype with
from external implementations.

The following options will change the genotype calls produced and are relevant only to BRLMM.

--prior-size How many probesets to use for determining
prior. [default: 10000]
--prior-file File from which to load generic snp priors,
see --write-prior option for details on file
format.
--dm-thresh DM confidence threshold used for seeding
clusters. [default .17]
--dm-hetmult The DM algorithm's het dropout can be
ameliorated by adding a small constant to
the log likelihoods of all het calls, making
het calls more likely. This small constant
is called the 'DM het multiplier' and
setting it to something positive can help in
balancing het/hom performance for DM, which
is used to seed the clusters. [default = 0]

Frequently Asked Questions

Q. What do I do when I don't have enough memory to process all the data?

A. You can manually set the `--block-size` command to specify how many probesets will be run at once. The program will then reduce memory by only loading those probesets into RAM. If the `block-size` option is unset the program will attempt to figure out how much available RAM you have and run in that memory. To fit in memory the program will often need to read the original CEL files multiple times. Also, if doing a quantile normalization try using a sketch (or subset) of the chip for the normalization. Sketch normalization is the default so this would only apply if you are using non-default options.

Q. How can I make `snp5-probeset-genotype` run more probesets per iteration?

A. If you manually set the `--block-size` flag `snp5-probeset-genotype` will not try to guess the amount of probesets to be run per iteration and will use the supplied value instead.

Q. The program died with an error message like "Assertion failed: A->probes.size() == 2, file ...cpp." What does this mean?

A. This is symptomatic of trying to run BRLMM for a SNP with no MM probes. In its typical mode of running BRLMM relies on DM to generate initial seed calls, and the DM algorithm requires MM probes.

Q. The program died with an error message like "DmListener::getGenoCall() - Can't find genotypes for name: SNP_A-1780432". What does this mean?

A. This is symptomatic of having specified the wrong `chrX` file for the analysis. In order to reduce the likelihood of accidentally using the wrong `chrX` file `snp5-probeset-genotype` checks to make sure that all the SNPs specified in the `chrX` file are present on the chips being analyzed. If it finds a SNP present in the `chrX` file that is not identified in the CDF file it will die with the above message. Note that if you want to bypass the requirement of a `chrX` file you can use the `--chrX-force` option.

Q. The program died and I got an error message saying "Killed". What does this mean and what can I do?

A. Linux has a "feature" that it will promise more memory than it actually has in the hope that many programs won't actually be using all their memory at once. However, if linux does run short of memory it will start killing programs arbitrarily. You can read more about linux's OOM (out of memory) killer at at LWN.net.

Q. Why does `snp5-probeset-genotype` require information regarding SNPs on chromosome X?

A. The SNPs on chromosome X are evaluated separately for XX (female) and XY (male) individuals as the intensity estimates for the males will generally be lower on X due to one missing chromosome. The prior is also adjusted to remove the het center as XY individuals should only have hom calls on the X chromosome. For BRLMM analyses gender is estimated using the method employed in the GTYPE software: individuals are called male if less than 7.5% of the snps on X are called as hets by the initial DM calls using a .33 confidence threshold. For BRLMM-P gender is estimated by use of an Expectation Maximiation (EM) algorithm on the PM probes for chrX SNPs to estimate the het rate.

Advanced Topics

CHP File format.

The CHP files produced by the BRLMM algorithm are different than those from DM. Historically the genotyping CHP file is closely tied to the DM model and while BRLMM uses the same format for backward compatibility it is important to note that the interpretation of many fields are different. Below are the names of the fields and corresponding BRLMM values that are stored in them.

- AlleleCall: Same as DM ALLELE_NO_CALL, ALLELE_A_CALL, ALLELE_AB_CALL, ALLELE_B_CALL.
- Confidence: Value between 0 and 1. The ratio of the distance to the closest cluster to the second closest cluster. Lower values are more confident.
- pvalue_AA: Mahalanobis distance to AA cluster.
- pvalue_AB: Mahalanobis distance to AB cluster.
- pvalue_BB: Mahalanobis distance to BB cluster.
- pvalue_NN: Not valid for BRLMM, value FLT_MAX inserted as placeholder.

The following parameters are saved in the CCHPFileHeader object:

- het-mult: Het multiplier.
- iterations: Number of BRLMM iterations.
- iter-thresh: Maximum score used during BRLMM iterations.
- K: Scaling parameter in transformations.
- transform: Which transformation was used for A & B allele.
- prior-weight: Number of pseudocounts used as weight for prior.
- prior-mincall: Minimum number of observations for seeing each AA,AB,BB cluster to be used in prior. Minimum is 2 as can't get variance for less than 2.
- lowprecision: Were summary values rounded off before transforming. This is only used for regression testing to be compatible with R prototype.

A word about memory (RAM)

The most common challenge people have running snp5-probeset-genotype is with RAM. snp5-probeset-genotype will attempt to split up jobs into the amount of RAM that appears free on your computer. The job is split by subdividing the analysis

into blocks of probesets (or SNPs). Small block sizes will subdivide the job more and require less memory at the expense of having to read the CEL files more frequently. On the other hand, using a smaller number of large blocks will require more memory but will place minimal load on reading CEL files.

The default behaviour is for snp5-probeset-genotype to estimate the optimal block size based upon the amount memory that appears to be free at the time the job begins. This default behaviour can be overridden by the user with the use of the --block-size option, which specifies the number of probesets to be processed at one time. For example, specifying --block-size=20000 will analyze the data in batches of 20,000 probesets at a time.

A problem that may be encountered (especially on a multi-user or multi-processor system) is running out of memory when a run of snp5-probeset-genotype is initiated and then another big-memory process is started afterwards. In this circumstance the first instance of snp5-probeset-genotype sees substantial free memory and chooses a large block-size, but then the second process grabs more of the memory and the first run of snp5-probeset-genotype runs out of memory. This problem can be addressed by planning the work load on your machine and/or using an appropriately small block size with the --block-size option.

RAM usage (in bytes) for Mapping 500K data can be estimated by the following equation:

$$RAM \approx C \times [(B \times P \times (F + 1)) + (S \times F)] + (B \times N) + (B \times D) + K$$

- C is the number of CELs being analyzed
- B is the number of probesets being processed in each batch
- P is the average number of probes in a probeset (about 27 for Mapping 500K)
- F is the number of bytes in a float (4).
- S is the number of probes being used to approximate a full quantile normalization and is 50000 by default.
- N is the average size in bytes of SNP_IDs, for Mapping 500K it is about 25 bytes.
- D is the size in bytes for the probeset information (approximately 800 bytes, but dependent on pointer size).
- K is constant for hashes, book keeping etc. about 200MB on 32 bit system.

Below are some guidelines about how many probesets to run at once (i.e. the --block-size) in 1.9 Gig of RAM as a function of number of CEL files:

- 50 CELs -> 140,000 probesets (i.e. half the dataset for a Mapping 250K chip plus some for prior).
- 100 CELs -> 70,000 probesets
- 150 CELs -> 47,000 probesets
- 200 CELs -> 35,000 probesets

Note that the above recommendations won't use all of the 1.9 gigs of RAM. In addition to needing a relatively large amount of memory the program also needs

relatively large blocks of contiguous memory and as RAM usage approaches the maximum available these get harder and harder to find. If you've got memory to spare the amount of RAM to run all the data at once as a function of Chips:

- 50 CELs -> 1.9 Gb
- 100 CELs -> 3.3 Gb
- 150 CELs -> 4.6 Gb
- 200 CELs -> 6.0 Gb
- 250 CELs -> 7.3 Gb
- 300 CELs -> 8.7 Gb

Note that on most 32 bit (i.e. Pentium, Xeon, Windows) systems you can't use than ~2 Gig of RAM with a single process, even if there is more available.

Custom Analyses:

While aliases for common analysis such as `brlmm` with default parameters are provided it is possible to construct custom analyses on the command line. There are both program options and analysis parameters that can be set to affect the results. Most people are familiar with the standard method for setting program options, but the specification of the analysis method and its parameters in `snp5-probeset-genotype` works a little differently. The method for setting custom parameters to the analysis involves supplying a text representation of the analysis and parameters desired. This enables flexibility as each piece of an analysis is self-contained and they can be (almost) arbitrarily combined. Note that when using a custom analysis rather than an alias it is necessary to specify the entire analysis and not acceptable to pass custom parameters to the alias. For example, if you wanted to change the number of iterations `brlmm` performs you would have to specify `'quant-norm.sketch=50000,pm-only,brlmm.iterations=1'` rather than just typing `'brlmm.iterations=1'`

The current full default `brlmm` analysis is: `'quant-norm.sketch=50000,pm-only,brlmm'` where there can be multiple chipstream modules (in this case a single `quant-norm`) separated by commas and the last two entries are the `pm` adjuster (`pm-only`) and quantification method (`brlmm`). Parameters to a particular step in the analysis are supplied in `key=value` pairs and separated by periods. For example `'quant-norm.sketch=50000'` indicates that the chips should be quantile normalized and that a sketch (subset of total data) of size 50000 should be used to do the normalization. Using a sketch can significantly reduce the amount of memory needed with minimal impact on normalization values. To do quantile normalization with just the PM probes and resolve ties in the same manner as bioconductor's **RMA** version of quantile normalization you would specify `'quant-norm.sketch=50000.bioc=true.usepm=true'`. All of the parameters possible can be seen by using the `--explain` option in conjunction with the name of the module (i.e. `snp5-probeset-genotype --explain quant-norm`).

So a few examples custom analyses would be:

`'pm-only,brlmm.transform=rvt'` - No normalization, use `rvt` space for clustering in `brlmm`.

'med-norm,pm-mm,brlmm.het-mult=.9' - Do a median normalization, use a PM-MM adjustment for probes and a het multiplier of .9 to try and balance hom/het calls.

'rma-bg,quant-norm.sketch=50000.usepm=true.bioc=true,pm-only,blmm.K=4.transform=CCS' - Do an **RMA** style background subtraction followed by an **RMA** style quantile normalization using a subset of 50000 data points followed by brlmm in CCS (contrast centers space) space with K = 4.

For brlmm the parameters are:

```
snp5-probeset-genotype --explain brlmm
brlmm: Do genotyping calls using the BRLMM (Bayesian RLMM)
algorithm.

Parameters:
  'K'                Scale parameter used in CCS and CES.
                    transformations [default = 4].
  'het-mult'        Number to balance het calls with to balance
                    performance on het/hom calls [default = 1
                    (no effect)]
  'iter-thresh'     Maximum confidence score to use when doing
                    iterations internally [0,1] [default = .3].
  'iterations'      Number of times to iterate BRLMM classifier,
                    feeding in new calls from previous iteration
                    [default=0]
  'prior-mincall'   Minimum number of genotypes per cluster for
                    inclusion in prior estimation, must be >= 2
                    [default = 2]
  'prior-weight'    What psuedocount weight should the prior have?
                    [default = 40]
  'transform'       What transformation of initial data are we
                    feeding into the classifier? {'CCS','CES',
                    'MvA','RvT' } [default = 'CCS']
```

For brlmm the parameters are:

```
snp5-probeset-genotype --explain brlmm-p
brlmm-p
Do genotyping calls with BRLMM-P (perfect match) algorithm.

Parameters:
  'K'                Scale parameter used used in CCS and CES
                    transformations. [default = 4]
```

```

'transform'      Transformation of initial data are we feeding
into the
                 classifier? {'CCS', 'CES', 'MvA', 'RvT'} [default
=
                 'CCS']
'lowprecision'  R prototype uses summary values rounded to first
decimal
                 place. Use this flag to be simulate behavior.
                 [default = false]
'KX'            Prior strength for Homs
'KH'            Prior strength for Hets
'KXX'           Prior strength for hom covariance
'KXY'           Prior strength for hom-het covariance
'KAH'           Prior strength for A-H covariance
'KHB'           Prior strength for H-B covariance
'V'            Prior strength for variances
'AAM'           Prior location of AA mean
'ABM'           Prior location of AB mean
'BBM'           Prior location of BB mean
'AAV'           Prior variance of AA
'ABV'           Prior variance of AB
'BBV'           Prior variance of BB
'COMVAR'        Flag: common variance
'HARD'          Flag: type of hard shell
'SB'            Size of shell barrier
'CM'            Type of call method, CM=1 for posterior
'MS'            Threshold for no-calls
'bins'          Use efficient binning to speed up labeling
'hints'         Use reference genotype data to indicate clusters
'CP'            Penalty for contradicting reference genotype
'em_thresh'     set threshold for em gender routine
'em_cutoff'     set cutoff for em gender routine
'gender_cutoff' set cutoff for which gender in em gender call

```

Clustering Space Transformations:

There are a number of different transformations that are implemented for different spaces which can be specified via the transform parameter to brlmm and are detailed below. For all of these transformations **A** and **B** denote the intensity of the A and B alleles respectively as estimated by the quantification method (such as plier or **RMA**). **X** and **Y** denote the new coordinates that **A** and **B** will be transformed into.

- **CCS** = Contrast Centers Stretch:

$$X = a \sinh(K * (A - B) / (A + B)) / a \sinh(K)$$

$$Y = \log_2(A + B)$$
- **CES** = Contrast Extremes Stretch:

$$X = \sinh(K * (A - B) / (A + B)) / \sinh(K)$$

$$Y = \log_2(A + B)$$
- **MvA** = Minus Vs Average

$$X = (\log_2(A) + \log_2(B)) / 2$$

$$Y = \log_2(B) - \log_2(A)$$
- **RvT** = R vs Theta (polar coordinates)

$$X = \arctan(A/B)$$

$$Y = \ln(\sqrt{A^2 + B^2})$$

