

LoadIn-all of same file type

Lori Shepherd and Jonathan Dare

July 10, 2007

Contents

1 requirements	1
2 example load-in	1

This is meant as a detailed instruction/example for the load-in process. For details on requirements and a general manual see aCGHplus vignette.
Please download and utilize load.in.example1.

1 requirements

We have the following requirements for running this loadin example:

- Our flat files from image analysis software (Imagene,Feature Extraction, etc.) are in a directory 'arrays'
- We have an inventory of the samples indicating associated image.name, load.specs file, image.soft, design, default.map, orientation, and User. (see inventory help file/vignette for more details)
- We have a load.specs file located in a subdirectory config.files that will give details on how the sample should be readin. All our example samples came from the same software, however they utilized the mapping by block application. The arrays carry three different block dimensions 12 x 6 (72), 12 x 4 (48) and 12 x 2 (24), and there are therefore three different load.specs files. (see load.specs help file/vignette for more details)
- We have a mapping file located in the RData directory which maps the Probes/BACs to their genomic location. All our example samples were created off a 4.5K genomic build and therefore only one mapping file is needed.(see building map help file/vignette for more detail)

2 example load-in

Using this example data set, we will now take you through a step-by-step process of loading samples.

In the current directory which is above the subdirectory containing the flat files from image analysis software, for this example the subdirectory is called arrays, launch a R session at the command line by typing R, and load the aCGHplus package with the following command:

```
> library(aCGHplus)
```

This example set has 200 samples. Each sample must have a new object created for it, as well as be mapped and have design data stored. This can be done in one of three ways:

1. individually create the object, map the object, and store design information for each sample (NOT RECOMMENDED)
2. take all samples at once and create their objects, mappings, and store design information
3. Batch. This means breaking up the samples into group and creating their objects, mappings, and store design information

Loading the sample and creating the aCGHplus object aCGHroster may be done altogether, however it is recommended for larger sample sets to run Batch calls. For our example we will run 4 batch jobs, which means 50 samples will have objects created, be mapped, and have their design information stored per batch job. The following batch calls may be run in the same R session, or the user may open any number of additional R sessions in the same directory as the first, in the same fashion as the first, to run any one of the following calls:

```
> RosterBatch(array.dir = "arrays", inventory.file = "example.inv",  
+   image.dir = "array.images", inv.sep = "\t", overwrite = T,  
+   nbatchjobs = 4, ibatchjob = 1)  
> RosterBatch(array.dir = "arrays", inventory.file = "example.inv",  
+   image.dir = "array.images", inv.sep = "\t", overwrite = T,  
+   nbatchjobs = 4, ibatchjob = 2)  
> RosterBatch(array.dir = "arrays", inventory.file = "example.inv",  
+   image.dir = "array.images", inv.sep = "\t", overwrite = T,  
+   nbatchjobs = 4, ibatchjob = 3)  
> RosterBatch(array.dir = "arrays", inventory.file = "example.inv",  
+   image.dir = "array.images", inv.sep = "\t", overwrite = T,  
+   nbatchjobs = 4, ibatchjob = 4)
```

If additional R sessions were opened, once the function is finished the sessions may be exited. Keep one R session open. We will examine the third call:

```
RosterBatch(array.dir="arrays",
            inventory.file="example.inv",
            image.dir="array.images",inv.sep="\t",overwrite=T,
            nbatchjobs=4,ibatchjob=3)
```

Looking at this code we can say the following:

- The flat files for the samples may be found in the array.dir directory 'arrays'. This directory should be in the working directory of R.
- The inventory file for the data set in question is example.inv. All the samples in this inventory will be loaded. This inventory has entries separated by a tab.
- The results for each sample (i.e sample object, mapping, and design information) will be stored in a subdirectory image.dir 'array.images'. This directory will be located in the current working directory of R
- any existing files that existed will be re-created
- The total number of batch jobs is four and the current batch job being executed is three. The samples being processed, therefore, are the 100th to 150th entries listed in the inventory file

Only after all Batch jobs are finished running may we combined all output into an aCGHroster object. In this example all four batch jobs must have been run and have finished producing output.

We now use the call that would have loaded all the samples in one single group but specifying that if a file exists for the sample it will not be re-created.

```
> aCGHroster = create.roster.R(array.dir = "arrays", roster.file = "RData/aCGHrosterEx1.RData",
+   inventory.file = "example.inv", image.dir = "array.images",
+   inv.sep = "\t", overwrite = F, returnFlag = T, vrb = T)
```

Examining this code further, we may say the following:

- The flag files for the samples may be found in the array.dir directory 'arrays'. This directory should be in the working directory of R.
- The aCGHroster object created will be saved in a subdirectory RData, which will be located in the current working directory of R, with the file name aCGHrosterEx1.RData.
- The inventory file for the data set in question is example.inv. All the samples in this inventory will be loaded. This inventory has entries separated by a tab.
- The results for each sample (i.e sample object, mapping, and design information) will be stored in a subdirectory image.dir 'array.images'. This directory will be located in the current working directory of R

- If the files exist for a sample, they will not be recreated.
- The aCGHroster object will be returned as output of the function
- Status messages will be printed

In this example case, we have presumably run all samples using the batch calls. If the user forgot to run one of the batch calls or a sample, for one reason or another was not processed, this function will attempt to create the sample object, mapping, and get design information. Otherwise, it will assemble the needed information from the objects already created, and create the aCGHroster object.