

Overview over the workflow for processing tiling array data: RNA hybridizations and finding new transcripts

Wolfgang Huber

May 22, 2005

Contents

1	Introduction	1
2	Reading the CEL files	2
3	Segmentation	2
4	Scoring the segments	2
5	Categorizing the segments	2
6	UTR mapping	2

1 Introduction

The data processing tasks in the following typically require a lot of CPU time (in the order of hours or days) and memory (several Gigabytes). Many of tasks will therefore usually be performed in scripts rather than in interactive sessions. Also, they are spliced in several smaller steps, to be able to examine intermediate results. Compute-heavy scripts (that in some cases can be run in parallel) are separated with visualization-oriented scripts for the analysis of (intermediate) results.

Intermediate results are stored and recovered with the *save* and *load* commands. The data types that are used are in many cases matrices, lists,

dataframes. The design is “good enough” to work, but as the package matures and is re-used in different projects, it is foreseen but these data structures will evolve, and some of them will undoubtedly benefit from being implemented as proper S4 classes.

This vignette is not complete. For detailed questions, please contact the author, or bear with me for a subsequent, more complete version.

2 Reading the CEL files

This is done by the script *readcels.R*. It uses the function *read.affybatch* from the *affy* package. It expects a sample annotation file (“phenoData”) table, and a number of CEL files. It creates the object of class *exprSet* *x*, which is saved in the file *x.Rdata*.

Reading several CEL files with 2560x2560 probes each can take tens of minutes to hours and require Gigabytes of RAM. Currently, the script does not allow for incremental reading, that is, if another CEL file is added to the project, the whole set of previous CEL files also needs to be imported again, to create a new object *x* and a new file *x.Rdata*.

3 Segmentation

This is done by the script *segment.R*.

4 Scoring the segments

This is done by the script *scoreSegments.R*.

5 Categorizing the segments

This is done by the script *tableSegments.R*.

6 UTR mapping

This is done by the script *utrmap.R*.