Title

ISOpureR: an R implementation of a computational purification algorithm of mixed tumour profiles.

Permalink
https://escholarship.org/uc/item/5wx285nw

Journal
BMC bioinformatics, 16(1)

ISSN
1471-2105

Authors
Anghel, Catalina V
Quon, Gerald
Haider, Syed
et al.

Publication Date
2015-05-01

DOI
10.1186/s12859-015-0597-x

Anghel *et al. BMC Bioinformatics* (2015) 16:156
DOI 10.1186/s12859-015-0597-x

BMC Bioinformatics

SOFTWARE **Open Access**

ISOpureR: an R implementation of a computational purification algorithm of mixed tumour profiles

Catalina V Anghel[1], Gerald Quon[2], Syed Haider[1,3], Francis Nguyen[1], Amit G Deshwar[4], Quaid D Morris[2,4,5,6*] and Paul C Boutros[1,7,8*]

Abstract

Background: Tumour samples containing distinct sub-populations of cancer and normal cells present challenges in the development of reproducible biomarkers, as these biomarkers are based on bulk signals from mixed tumour profiles. ISOpure is the only mRNA computational purification method to date that does not require a paired tumour-normal sample, provides a personalized cancer profile for each patient, and has been tested on clinical data. Replacing mixed tumour profiles with ISOpure-preprocessed cancer profiles led to better prognostic gene signatures for lung and prostate cancer.

Results: To simplify the integration of ISOpure into standard R-based bioinformatics analysis pipelines, the algorithm has been implemented as an R package. The *ISOpureR* package performs analogously to the original code in estimating the fraction of cancer cells and the patient cancer mRNA abundance profile from tumour samples in four cancer datasets.

Conclusions: The *ISOpureR* package estimates the fraction of cancer cells and personalized patient cancer mRNA abundance profile from a mixed tumour profile. This open-source R implementation enables integration into existing computational pipelines, as well as easy testing, modification and extension of the model.

Keywords: Tumour heterogeneity, mRNA abundance profile, Deconvolution

Background

Tumour heterogeneity provides both challenges and opportunities in the development of cancer biomarkers. Tumours are mixed populations of multiple cell-types. Currently, the molecular profiles of interest – those of cancer cells or of distinct sub-populations of cancer cells – are blurred by the mixed signal from all cell types in a sample [1,2]. However, characterizing the heterogeneity of a patient's tumour by identifying the sub-populations present, along with their proportions and molecular profiles, would provide a personalized cancer "fingerprint" that captures both cell-centred and whole-system information, opening up opportunities for targeted treatment [3,4]. The methods described in this article apply to mRNA expression data rather than to to DNA data (point mutations and copy number changes), which require other approaches [5-8].

As a first step, it is important to consider the two-population problem of normal and cancer cells. Even small fractions of contaminating normal cells can introduce noise in gene signatures [9,10], motivating the search for methods to deconvolve a mixed tumour profile by estimating the fraction of cancer cells and providing a personalized, purified mRNA abundance profile of the cancer cells.

Physical approaches for sample separation, such as laser capture micro-dissection [11] are costly, time-intensive, not always available and may degrade samples. Therefore, computational approaches to purification of tumour molecular profiles have become increasingly important.

*Correspondence: Quaid.Morris@utoronto.ca; Paul.Boutros@oicr.on.ca
[2] Department of Computer Science, University of Toronto, 10 King's College Road, Room 3303, M5S 3G4 Toronto, ON, Canada
[1] Informatics and Biocomputing Program, Ontario Institute for Cancer Research, 661 University Avenue, Suite 510, M5G 0A3 Toronto, ON, Canada
Full list of author information is available at the end of the article

Table 1 and Additional file 1 summarize the different methods currently available for deconvolving mRNA data.

Classical methods of profile deconvolution assume that a mixed profile is a linear combination of a predetermined number of pure constituent profiles. Written in matrix form, the measured, mixed profiles B are a product of A, a matrix of gene expression profiles of each constituent, weighted by the fractions X of each cell type in the mixture:

$$AX = B. \tag{1}$$

Equivalently, some algorithms start with the transpose of this equation. Different algorithms use different methods of deconvolution. Some assume that the fractions, X, are known [12-17]. Others use gene expression profiles [18], signatures [19-22] or markers [23-26] of the constituent profiles to recover X, and sometimes the expression profiles of other genes in the constituents [23]. (A gene marker is a set of genes assumed to be expressed solely in one cell subtype and in no other [4,27,28].) These approaches are limited in uncovering patient-specific variation in cancer, as they assume that all tumour profiles are mixtures of a small number of the same constituents. In addition, the expression data may be log-transformed [13,16], leading to a possible bias in the reconstruction of mixed tissue samples from constituent profiles [29].

Another approach to estimate both fraction of cancer content as well as patient-specific cancer profiles requires a matched normal profile for each patient [30,31]. In this case the normal profile is "electronically subtracted" from the bulk tumour profile. However, a matched normal profile may not always be available in existing datasets and may be difficult to obtain clinically. Furthermore, due to the biological variability of normal tissue, the provided normal profile may not match that of the tissue in the tumour sample.

Two algorithms, DeMix [32] and ISOpure [33] present statistical approaches for deconvolution of mixed tumour profiles given a set of unmatched normal samples. Cancer biomarkers generated from prostate and non-small cell lung cancer data purified using ISOpure were more effective at predicting survival relative to those generated using unpurified profiles [33].

Overview of ISOpure

In the following, we will use the term 'ISOpure' to refer to the algorithm in general, and *ISOpureR* to refer to the R package implementation. The next two sections will describe the statistical model and the algorithm in more detail, but we begin by providing a brief overview and example.

For the applications discussed here, the ISOpure algorithm is applied to microarray mRNA abundance data. The inputs to the model must be normalized (but not log-transformed) expression profiles. The following two sets of inputs are required: tumour mRNA abundance profiles, of the same cancer sub-type; and normal (*i.e.* healthy) mRNA abundance profiles, from the same tissue as the tumour.

The algorithm runs in two steps.

1. **Cancer Profile Estimation (CPE).** This step estimates and outputs an average cancer profile as well as a fraction of cancer in each tumour sample.
2. **Patient Profile Estimation (PPE).** This step estimates and outputs a cancer profile for each patient.These profiles are all similar to the average cancer profile, but contain patient-specific variations. The estimated cancer fraction for each tumour sample is fixed at the value calculated in the CPE step.

To run these two steps using *ISOpureR*, it suffices to apply the two functions `ISOpure.step1.CPE` and `ISOpure.step2.PPE`. The input expression data should be in matrix form, with samples along the columns and transcripts/features along the rows.

```
# For reproducible results, set the random
number generator seed set.seed(123);
# Run ISOpureR Step 1 - Cancer Profile
Estimation ISOpureS1model <- ISOpure.step1.
CPE(
    tumour.expression.data,
    normal.expression.data
    );

# For reproducible results, set the random
seed set.seed(456);
# Run ISOpureR Step 2 - Patient Profile
Estimation ISOpureS2model <- ISOpure.step2.
PPE(
    tumour.expression.data,
    normal.expression.data,
    ISOpureS1model
    );
```

The vector `ISOpureS1model$alphapurities` produced by the first step contains the proportion of cancer for each patient. The matrix `ISOpureS2model $cc_cancerprofiles` produced by the second step contains the patient-specific profiles, rescaled to be of the same scale as the tumour expression data. That is, these profiles are estimated as probabilities within the algorithm, but are scaled to represent microarray signal intensity data. A detailed example is given in Section 3 of the *ISOpureR* package vignette, included as Additional file 2.

In the statistical and algorithm descriptions of ISOpure, the notation $\mathbf{t}_n$ describes the matrix of tumour profiles, where the index n denotes the n-th patient. The fractions

Table 1 An overview of computational deconvolution algorithms for RNA profiles

Method	Ref.	Input	Output			Clinical data?			Availability			
			Prop.	Expr.	Individual profile	Cancer	Normal blood	Other	R	CellMix	MATLAB	Other
ISOpure (Quon)	[33]	tumour & unmatched normal	✓	✓	✓	✓			✓		✓	
DeMix (Ahn)	[32]	tumour & unmatched normal	✓	✓	✓				✓			
Clarke	[30]	paired mixed & pure profiles	✓		✓				✓			
Gosink	[31]	mixed profiles and known profile of one constituent	✓		✓							
DeconRNASeq (Gong)	[18]	profiles of constituents	✓						✓			
Gong	[19]	cell-type specific gene signatures	✓							✓		
Abbas	[20]	cell-type specific gene signatures	✓				✓			✓		
Wang M.	[21]	cell-type specific gene signatures	✓									
Lu	[22]	cell-type specific gene signatures	✓									*
PERT (Qiao)	[46]	reference profiles of constituents	✓	†	†		✓					✓
ESTIMATE (Yoshihara)	[47]	prior data used to derive cell-type specific gene signatures	✓			✓			✓			
DSection (Erkkilä)	[12]	prior knowledge of proportions	†	✓						✓	✓	
csSAM (Shen-Orr)	[13]	proportions of constituents		✓			✓		✓	✓		
Bar-Joseph	[14]	proportions of consitutents, one expression profile		✓		✓		✓				
Ghosh	[16]	proportions, tumour & unmatched normal		✓		✓			*			
Stuart	[17]	proportions of constitutents		✓		✓						
TEMT (Li)	[48]	prior knowledge of proportions, paired mixed-pure profiles			✓							✓
DSA (Zhong)	[23]	cell markers	✓	✓		✓			✓	✓		
ssNMF (Gaujoux)	[25]	cell markers	✓	✓			✓			✓		
PSEA (Kuhn)	[24]	cell markers	✓	✓				✓	✓			
deconf (Repsilber)	[26]	cell markers	✓	✓			✓		✓	✓		

Table 1 An overview of computational deconvolution algorithms for RNA profiles *(Continued)*

Tolliver	[49]	tumour profile, number of constituents	✓	✓	✓	
Roy	[50]	prior estimate of number of constituents	✓	✓		
Lähdesmäki	[15]	mixed expression profiles	†	✓		
Venet	[27]	mixed expression profiles, number of constituents	✓	✓	✓	
UNDO (Wang N.)	[51]	mixed expression profiles	✓	✓	✓	✓

Most of the algorithms are applied to microarray mRNA abundance data, although TEMP and ESTIMATE use high-throughput RNA-Seq data and ISOpure and DeconRNASeq can be applied to both [52]. The possible outputs of the algorithms are proportions of constituent cell-types (Prop.), average expression profiles (Expr.), or patient-specific expression profiles (Individual Profile) of constituent cell-types. The two main sources of clinical data were cancer-related gene expression data (including human Hodgkin's lymphomas) or normal blood expression data. PSEA was applied to expression data from patients with Huntington's disease, and Bar-Joseph also studied cell cycle synchronized foreskin fibroblast cells. In terms of availability, the summary package CellMix [28] is also an R package but is listed as a separate category. The only algorithms not available for either R or MATLAB are PERT (Octave) and TEMT (Python). Algorithms which were described as using built-in MATLAB or R functions were not included, as reproducible example code is not available for them. The currently available source code is summarized in Additional file 2.

Notes:

† Prior information about proportions or expressions is needed, but these values are re-estimated during the execution of the algorithm. For PERT, the individual profiles are adjusted (perturbed) versions of the reference profiles.

* The original code for Lu (Java-based) [22] and Ghosh [16] is no longer available.

α_n represent the cancer fractions. The $\mathbf{b}_r$ and $\mathbf{c}_n$ represent normal and purified cancer profiles, but they will be interpreted as probabilities, as described in more detail below. An overview of the algorithm workflow is given in Figure 1.

While we focus on microarray mRNA abundance data, ISOpure is generic when it comes to different species of RNA. For example, it is able to deconvolve both mRNA (non-coding RNAs inclusive) and microRNAs. This is true for both microarray as well as next-generation sequencing (RNA-Seq) data. Input data matrices can represent genes, isoforms, exons or microRNAs of samples. The data matrices should represent approximate counts of molecules (normalised but not log converted) for a given RNA profile, *e.g.* gene level mRNA. However, the input should not contain a mixture of mRNA and microRNA data. ISOpure has been applied to the RNA-Seq The Cancer Genome Atlas (TCGA) PRAD dataset as part of that project (manuscript in preparation), helping to demonstrate its wider applicability.

The ISOpure statistical model

A more complete explanation of the statistical model is given in the original article, [33], as well as in the *ISOpureR* package vignette (Additional file 2). As noted, ISOpure addresses the two-population problem, assuming that a patient's particular tumour mRNA abundance profile $\mathbf{t}_n$ can be decomposed into its cancer and healthy profile components. For patient n,

$$\mathbf{t}_n = \alpha_n \mathbf{c}_n + (1 - \alpha_n)\mathbf{h}_n + \mathbf{e}_n. \qquad (2)$$

Here, $\mathbf{c}_n$ is the personalized cancer profile, $\mathbf{h}_n$ the profile of the patient's healthy tissue, α_n the fraction of cancer cells ($0 \leq \alpha_n \leq 1$) and $\mathbf{e}_n$ the reconstruction error.

This linear system of equations (for patients 1 to N) is underdetermined, as the only known values are the $\mathbf{t}_n$'s. To

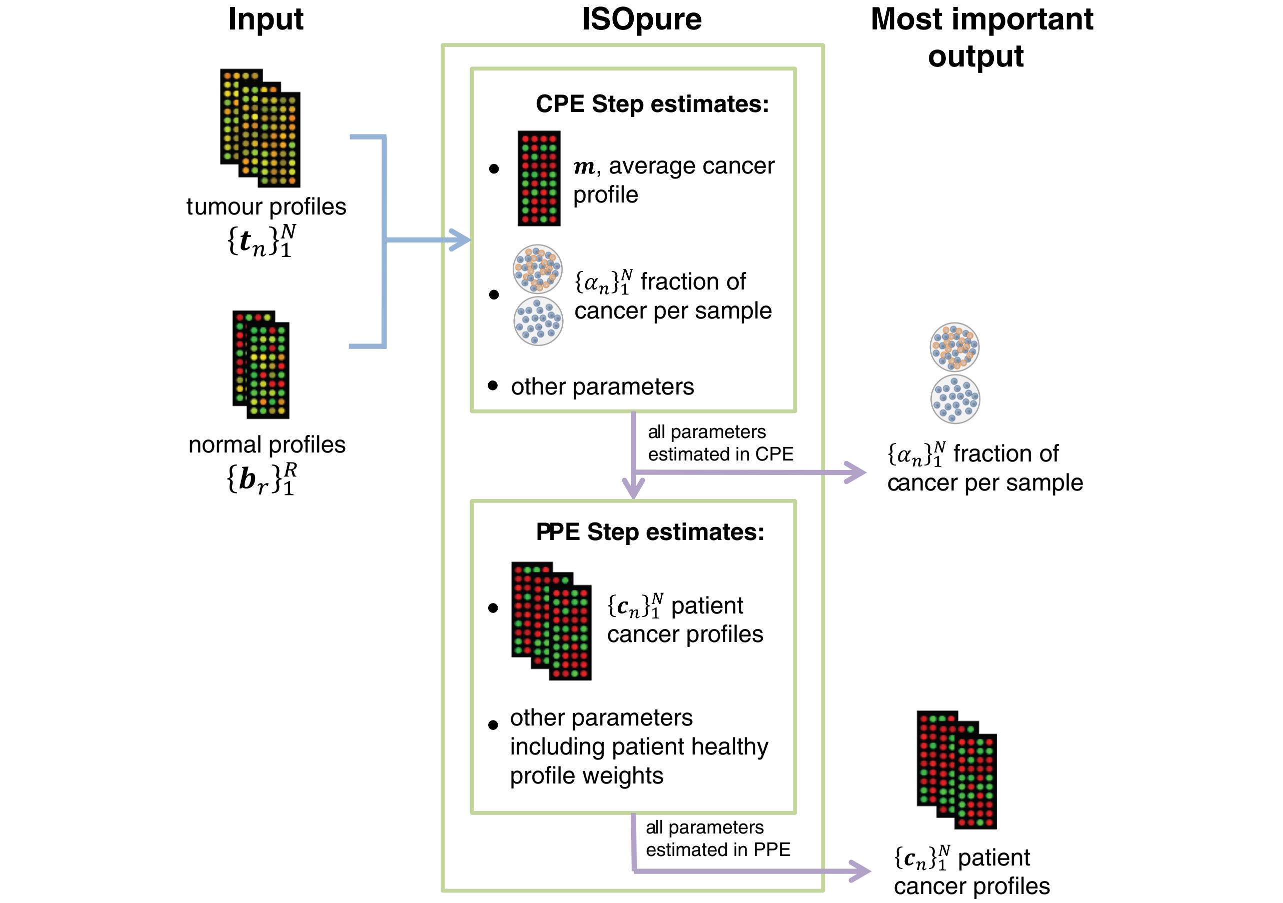
Input
ISOpure
Most important output
tumour profiles
$\{t_n\}_1^N$
normal profiles
$\{b_r\}_1^R$
CPE Step estimates:
• m, average cancer profile
• $\{\alpha_n\}_1^N$ fraction of cancer per sample
• other parameters
all parameters estimated in CPE
$\{\alpha_n\}_1^N$ fraction of cancer per sample
PPE Step estimates:
• $\{c_n\}_1^N$ patient cancer profiles
• other parameters including patient healthy profile weights
all parameters estimated in PPE
$\{c_n\}_1^N$ patient cancer profiles

Figure 1 ISOpure workflow. An overview of the ISOpure algorithm, illustrating the inputs and the most important outputs of the Cancer Profile Estimation (CPE) and the Patient Profile Estimation (PPE) steps. The CPE step estimates an average cancer profile over all patients and the proportion of cancer in each tumour, as well as the patient healthy profiles (as weights of input profiles). The healthy profile weights are re-estimated in the PPE step. This second step estimates the purified cancer profile for each patient. All estimated parameters from the CPE and PPE steps are output by the *ISOpureR* functions `ISOpure.step1.CPE` and `ISOpure.step2.PPE`.

reduce the number of parameters to be estimated, as well as to prevent overfitting, ISOpure employs two regularization techniques. First, the patient-specific healthy profile, $\mathbf{h}_n$ is assumed to be a convex combination of a reference set of known healthy profiles, $\mathbf{b}_1, \ldots, \mathbf{b}_n$:

$$\mathbf{t}_n = \alpha_n \mathbf{c}_n + \sum_{r=1}^{R} \theta_{n,r} \mathbf{b}_r + \mathbf{e}_n \tag{3}$$

where

$$\alpha_n + \sum_{r=1}^{R} \theta_{n,r} = 1. \tag{4}$$

This assumption is convenient for data availability (a paired normal sample is not always available in archived datasets), but is also motivated by biology; even a paired healthy sample may not correspond exactly with the healthy portion of the tumour sample, and may contain noise. The second regularization assumption is that the cancer profiles, $\mathbf{c}_n$ cluster near an estimated reference cancer profile $\mathbf{m}$, an assumption that is more accurate when the cancers are all of the same subtype [34,35].

For the statistical model, the cancer profiles, $\mathbf{m}$ and $\mathbf{c}_n$, and the healthy profiles, $\mathbf{b}_r$, are transformed into probability distributions. Thus,

$$\hat{\mathbf{x}}_n = \alpha_n \mathbf{c}_n + \sum_{r=1}^{R} \theta_{n,r} \mathbf{b}_r \tag{5}$$

becomes a probability vector. The tumour sample $\mathbf{t}_n$ is discretized to $\mathbf{x}_n$, and considered to be a sample from the multinomial distribution with probability vector $\hat{\mathbf{x}}_n$.

The following equations describe the full statistical model, which is also illustrated in the Bayesian network diagram in Additional file 3. To simplify notation, the vector $\boldsymbol{\theta}_n$ includes the entries $\theta_{n,1}, \theta_{n,2}, \ldots, \theta_{n,R}, \alpha_n$.

$$\mathbf{B} = [\,\mathbf{b}_1, \ldots, \mathbf{b}_R] \tag{6}$$

$$\hat{\mathbf{x}}_n = [\,\mathbf{B}\,\mathbf{c}_n]\,\boldsymbol{\theta}_n \tag{7}$$

$$p\left(\mathbf{x}_n|\mathbf{B}, \boldsymbol{\theta}_n, \mathbf{c}_n\right) = \text{Multinomial}\left(\mathbf{x}_n|\hat{\mathbf{x}}_n\right) \tag{8}$$

$$p\left(\boldsymbol{\theta}_n|\boldsymbol{\nu}\right) = \text{Dirichlet}\left(\boldsymbol{\theta}_n|\boldsymbol{\nu}\right) \tag{9}$$

$$p\left(\mathbf{c}_n|k_n, \mathbf{m}\right) = \text{Dirichlet}\left(\mathbf{c}_n|k_n\mathbf{m}\right) \tag{10}$$

$$p\left(\mathbf{m}|k', \mathbf{B}, \boldsymbol{\omega}\right) = \text{Dirichlet}\left(\mathbf{m}|k'\mathbf{B}\boldsymbol{\omega}\right) \tag{11}$$

Equation (7) is simply Equation (5) in matrix form, and Equation (8) summarizes the model described in the previous paragraphs.

The Dirichlet distributions are used for the parameters $\boldsymbol{\theta}_n$, $\mathbf{m}$ and $\mathbf{c}_n$, which are discrete probability distributions. The hyper-parameters $\boldsymbol{\nu}$, k_n, k' and $\boldsymbol{\omega}$ determine the mean and the concentration of the Dirichlet distributions.

The ISOpure algorithm

The goal of the algorithm is to maximize the complete likelihood function

$$\mathbb{L} = p\left(\mathbf{m}|k', \mathbf{B}, \boldsymbol{\omega}\right) \prod_{n=1}^{N} p\left(\mathbf{c}_n|k_n, \mathbf{m}\right) p\left(\boldsymbol{\theta}_n|\boldsymbol{\nu}\right) p\left(\mathbf{x}_n|\mathbf{B}, \boldsymbol{\theta}_n, \mathbf{c}_n\right). \tag{12}$$

The ISOpure algorithm splits this optimization into two steps. The Bayesian diagram and the flowchart of the algorithm for each step are given in Additional file 4. ISOpure uses block coordinate descent where all variables except one (or a 'block' of similar variables) are fixed, and the objective is minimized with respect to the one variable (or block).

Implementation

Motivation and software design

The main contribution of this paper is the implementation of ISOpure in the widely-used R statistical environment. R is one of the most popular programming languages in bioinformatics, in particular for the analysis and visualization of genomic data. It is freely available under the GNU General Public Licence (GPLv2/3) and can be extended by many open-source packages.

ISOpure was originally implemented using MATLAB [33,36]. The demand for an R version of ISOpure has been motivated both by the cost-effectiveness of R, as well as by the convenience and possible customization of the algorithm in a familiar language. *ISOpureR* enables the integration of the computational purification step within existing data-analysis pipelines. The code was designed using R version 3.1.1 (64-bit) on Ubuntu 12.04.4 LTS [37].

The organization of the R implementation closely follows the original MATLAB code. While the structure of the code remains consistent, the advantage of using an R package is that help files are easily accessible, as for any package (*e.g.* typing `help(package=ISOpureR)` after loading the library will list all functions). The most important help file is the vignette (Additional file 2), which gives details on the algorithm, the preprocessing steps for microarray data and an extended example of running ISOpure on a computationally convenient dataset included with *ISOpureR*, with visualizations of the output. The internal test cases for the package also use this small dataset to test the log likelihood and the derivative of the log likelihood functions for each parameter.

Comparison with the original code

Two main challenges in the translation of ISOpure from MATLAB to R were the differences in output of standard functions in the two languages, and differences in running time. Surprisingly, MATLAB and R outputs differ for two of the most basic operators: greaterthan (>)

and less than (<). In MATLAB, a comparison between a number and `NA` or `NaN` returns `FALSE`, while in R the output is `NA`. (For instance, `3 < NA` would return `0` in MATLAB, and `NA` in R.) The optimization function (`ISOpure.model_optimize.cg_code.rminimize.R` in *ISOpureR*, which copies the MATLAB ISOpure function) performs a line search using quadratic and cubic polynomial interpolations/extrapolations and the Wolfe-Powell stopping criteria. Intermediate iterates which fall outside the function's domain result in infinite function or derivative values, and `NaN` values in the succeeding iterate. In this case, the ISOpure algorithm outputs `FALSE` when testing the Wolfe-Powell stopping criteria and the search continues with an adjusted search point. Thus, alternative versions of the greater and less than operators in R (*e.g.* `ISOpure.util.matlab_greater_than.R`) ensure the correct performance of the minimizing function in *ISOpureR*.

Another difference in MATLAB and R is the behaviour of the logarithm function for negative real values. In R, `log(x)` outputs `NaN` for negative values of `x`, while in MATLAB the output is a complex number. In order to avoid underflow (the numerical error resulting from computing a number too small in magnitude to store in memory), ISOpure often performs calculations in the log domain. For some of the intermediate calculations, the logarithm of a negative value is calculated (*e.g.* the logarithm of a derivative which may have negative components).

In addition to code-level differences, R and MATLAB differ in running time. MATLAB takes advantage of multicore processing by default. While the average elapsed time for ISOpure in R was 1.73 to 2.61 times slower than the CPU-time in MATLAB, it was much slower than the elapsed time in MATLAB. To improve the runtime of *ISOpureR*, the current version (v.1.0.16) incorporates C++ code into the algorithm using RcppEigen [38], reducing the elapsed time two to three fold, from 8.9-13.4 to 3.9-4.8 times the elapsed MATLAB time. It is also useful to note that despite the slower time, for running large numbers of similar models (*e.g.* 50), the performance of *ISOpureR* was faster overall, as the jobs could be submitted simultaneously to a compute cluster, with no licence limitations.

The size of the dataset influences runtime most significantly. The running time seems to be linearly dependent on the number of transcripts/features when all other values (number of tumour samples, number of normal samples) are kept the same. Similarly, the time is also linearly dependent on the number of tumour samples and normal samples (Additional file 5).

Results and discussion

To verify the numerical equivalence of the MATLAB and R implementations of ISOpure, their performances on four datasets were compared. Two sets were of lung adenocarcinoma from Bhattacharjee [39] and Beer [40] and two were of prostate cancer from Wallace [41] and Wang [42] (Table 2). These four datasets are among the tumour datasets purified using ISOpure in [33], chosen because the cancer types are not yet known to have established subtypes. The array data processing was detailed in [33].

Each dataset was purified using both the MATLAB and R implementations of ISOpure and we compared the resulting parameters. The algorithms were run 50 times for each dataset, with different initial conditions. To minimize differences due to random number generation implementations, the initial values of parameters were loaded from a file, and the extra optimizations of parameters $\boldsymbol{\nu}$, $\boldsymbol{\omega}$, and k [33] in the CPE step, which included some random initializations, were omitted. The results of these models are very similar to results generated by the full, randomized version of *ISOpureR*; the motivation for minimizing randomness was simply to reduce differences in the performance comparisons between MATLAB and R.

The numerical differences in the parameter estimates produced by MATLAB and R are small enough to have no biological significance. We calculated the means of the parameters ($\bar{\mathbf{x}}_{MATLAB}$, $\bar{\mathbf{x}}_{R}$, for each of the estimated parameters such as $\boldsymbol{\nu}, \boldsymbol{\alpha}$, etc.) from the MATLAB and R models over the 50 iterations. A comparison of the entries of these mean parameter values produced by R and MATLAB shows that the two implementations are numerically analogous (Figure 2 and Additional file 6). In particular, β_1 and Spearman's ρ are 1 for both the estimated fraction of cancer cells in the tumour, α, and for the log of the individual cancer profiles $\mathbf{c}_n$.

Table 2 An summary of the datasets used to validate *ISOpureR*

Dataset	Ref.	Cancer type	Number of samples		Number of transcripts
			Tumour	Normal	
Beer	[40]	lung adenocarcinoma	86	10	5,151
Bhattacharjee	[39]	lung adenocarcinoma	139	17	8,383
Wallace	[41]	prostate cancer	69	18	12,140
Wang	[42]	prostate cancer	109	45	18,185

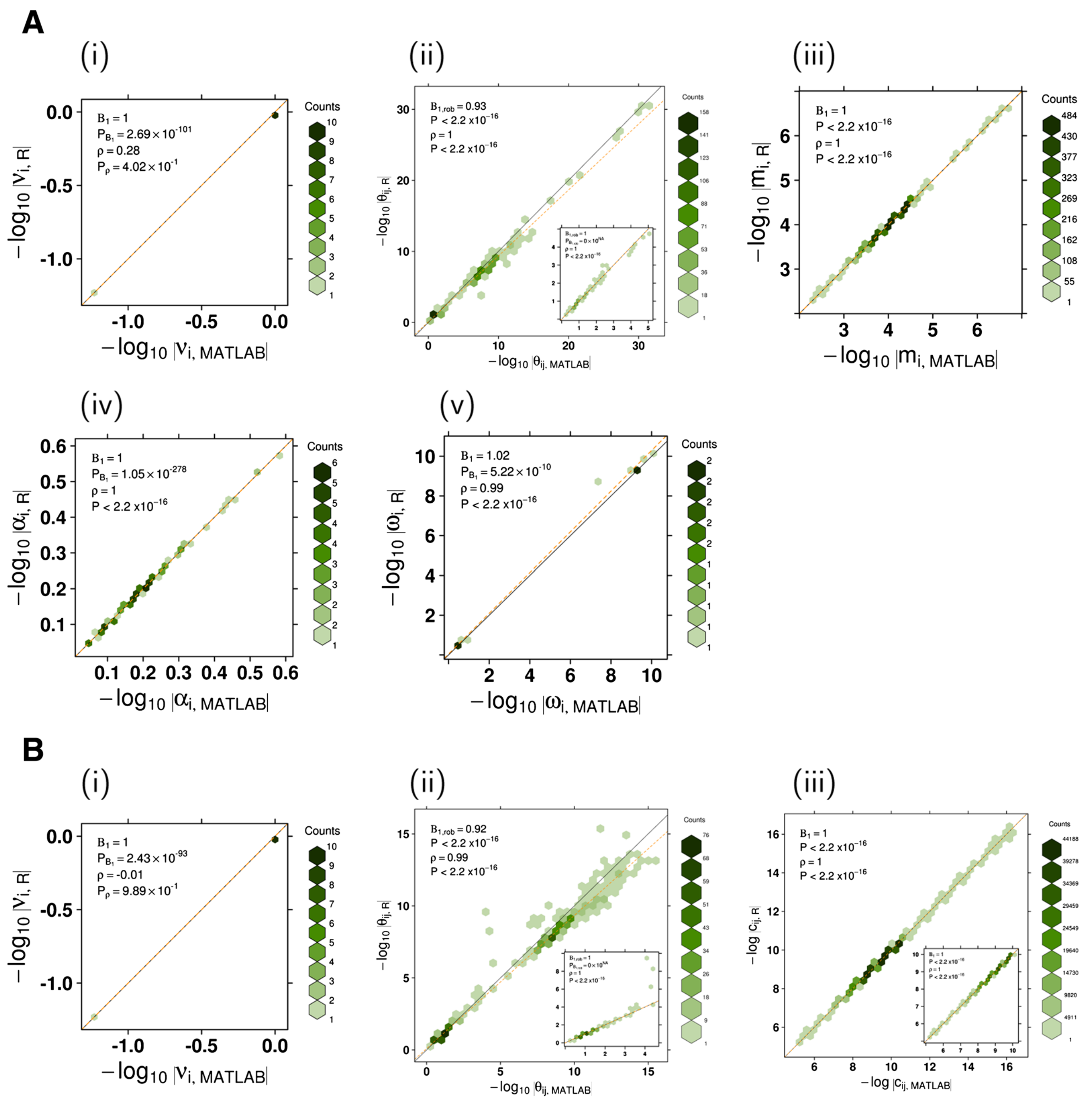

Figure 2 A comparison of parameters estimated by the MATLAB and R implementations of ISOpure for the Beer dataset. Each plot shows the entries of a parameter estimated using *ISOpureR* plotted against the corresponding entries estimated using the MATLAB code. The parameter is an average over 49 models run with different initial conditions (one MATLAB model for the Beer dataset resulted in a zero $\boldsymbol{\theta}$ value, and was dropped). The line $y = x$ is indicated in black, and the linear regression line, or robust regression line for $\boldsymbol{\theta}$, is dashed orange. **(A)** Parameters from the Cancer Profile Estimation step of ISOpure: (i) $\boldsymbol{\nu}$, the hyper-parameter for the Dirichlet distribution over $\boldsymbol{\theta}$, (ii) $\boldsymbol{\theta}$, the proportion of a patient sample from a known healthy-tissue profile, (iii) $\mathbf{m}$, the average mRNA abundance cancer profile, (iv) $\boldsymbol{\alpha}$, the fraction of cancer cells for every patient sample, (v) $\boldsymbol{\omega}$ a hyper-parameter for the Dirichlet distribution over $\mathbf{m}$. **(B)** Parameters from the Patient Profile Estimation step of ISOpure: (i) $\boldsymbol{\nu}$, the hyper-parameter for the Dirichlet distribution over $\boldsymbol{\theta}$, (ii) $\boldsymbol{\theta}$, the proportion of a patient sample from a known healthy-tissue profile, (iii) $\mathbf{c_n}$, the purified mRNA abundance cancer profile for each patient.

For the all datasets, the mean and median of the fractional difference between the entries of the parameter means are very close to zero for six of the nine parameters (Additional file 7). For instance, for both the Beer and the Bhattacharjee datasets the means and medians for the fractional difference of α and $\log \mathbf{c}_n$ are between 10^{-5} and 10^{-4}. The three parameters having larger differences, $\boldsymbol{\omega}$ in the CPE step and $\boldsymbol{\theta}$ in both CPE and PPE steps, contain very small entries which are not biologically significant. The entries of $\boldsymbol{\theta}$ represent the weights of known normal-tissue profiles adding up to a patient's particular normal profile; a weight of 10^{-5} essentially means that particular known normal profile is not present for that patient. When entries larger than a certain threshold, such as 10^{-5}, are compared, the fractional difference decreases.

Furthermore, the differences between the MATLAB and R algorithms are similar to the differences within each implementation, under different initial conditions (Additional file 8). In particular, smaller numbers are not as precisely predicted. Computationally, operations with smaller numbers are susceptible to floating point rounding errors; however differences in small numbers do not alter the model, as they are biologically unimportant.

Finally, the mean and median of the vector components of $(\bar{\mathbf{x}}_{MATLAB} - \bar{\mathbf{x}}_R)$ are very close to 0, and are sometimes positive and sometimes negative. The R algorithm does seem to under-estimate parameters compared to MATLAB, but this bias disappears when we compare parameter values larger than 10^{-5}.

Conclusions

The first stage in the development of *ISOpureR* focused on establishing the numerical equivalence of the results produced by the R and MATLAB code. Future steps include testing backward compatibility with ISOLATE [43], the precursor to ISOpure.

The translation of MATLAB code into R code was surprisingly challenging. Debugging took five times as long as the translation and initial testing, as differences in function performance appeared only for certain input values during the execution of the full algorithm rather than in the testing of individual functions. Perhaps some of the differences in basic MATLAB and R operators mentioned can be of help for others tackling a similar project. A list of key issues encountered is in Additional file 9.

One of the recommendations for the comparison of implementations would be to eliminate all sources of randomness in the algorithms and postpone the improvement of running time only once numerical results are consistent, and consistently reproducible given different initial random seeds.

Most importantly, the contribution of the R implementation of the ISOpure algorithm is that it can now be readily integrated into existing analysis pipelines. *ISOpureR* can easily be included in benchmark comparisons of different deconvolution algorithms. A promising upcoming project is the ICGC-TCGA DREAM Somatic Mutation Calling - Tumour Heterogeneity Challenge [44], a benchmarking competition of computational methods for determining the best sub-clonal reconstruction algorithms. The collaborative-competitive framework of the DREAM projects encourages transparency in algorithm development and the availability of open-source code. The R implementation of ISOpure provides increased flexibility and ease of parallelization, so that the algorithm may be easily modified, extended and tested by the community.

Availability and Requirements

The *ISOpureR* package is submitted to the Comprehensive R Archive Network (CRAN) which maintains an active package homepage for *ISOpureR*. The version of the code at the time of publication is included in Additional file 10. The package is written an implemented in the R programming language (version $\geq$ 3.1.1), with some C++ code incorporated using Rcpp [45] and RcppEigen [38]. It is platform-independent, but has been primarily tested on Linux, and is available under the GLP-2 license.

Additional files

Additional file 1: (Table) A summary of software availability for deconvolution of mRNA abunance data. A similar table is given in [52].

Additional file 2: The vignette of the *ISOpureR* package, ISOpureRGuide.pdf, including an extended description of the ISOpure model and algorithm, preprocessing steps for microarray files, and examples for applying *ISOpureR* to datasets. This file is also found within the *ISOpureR* package, in Additional file 10.

Additional file 3: (Figure) Bayesian network model for the ISOpure statistical model.

Additional file 4: (Figures) Bayesian network and flowchart diagrams for the cancer profile estimation and patient profile estimation steps of the ISOpure algorithm.

Additional file 5: (Figures) A comparison of running time for different dataset sizes (different number of transcripts or different number of tumour samples).

Additional file 6: (Figures) A comparison of parameters estimated by the MATLAB and R implementations of ISOpure for the Bhattacharjee, Wallace, and Wang datasets.

Additional file 7: (Tables) Statistics for differences in parameters estimated by the MATLAB and R implementations of ISOpure for the Beer, Bhattacharjee, Wallace, and Wang datasets.

Additional file 8: (Figure) A comparison of the parameter estimation in models with different initial conditions, one programming language.

Additional file 9: Recommendations for translating code from MATLAB to R.

Additional file 10: *ISOpureR* R package as a Linux-compatible file.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

PCB and QDM conceived of the project. The original ISOpure algorithm was designed by GQ and QDM. CVA and FN wrote the R implementation with input from SH, GQ, AGD and PCB. SH, AGD, FN and PCB collected and pre-processed the mRNA abundance profiles. CVA wrote the first draft of the manuscript, which all authors revised and approved.

Acknowledgements

The authors thank all members of the Boutros lab for helpful suggestions. This study was conducted with the support of the Ontario Institute for Cancer

Research to PCB through funding provided by the Government of Ontario. Dr. Boutros was supported by a Terry Fox Research Institute New Investigator Award, a Terry Fox Research Institute Program Project and a CIHR New Investigator Award. This work was supported by Prostate Cancer Canada and is proudly funded by the Movember Foundation – Grant #RS2014-01.

Author details

[1]Informatics and Biocomputing Program, Ontario Institute for Cancer Research, 661 University Avenue, Suite 510, M5G 0A3 Toronto, ON, Canada. [2]Department of Computer Science, University of Toronto, 10 King's College Road, Room 3303, M5S 3G4 Toronto, ON, Canada. [3]Department of Oncology, University of Oxford, Old Road Campus Research Building, Roosevelt Drive, OX3 7DQ Oxford, United Kingdom. [4]Edward S. Rogers Sr. Department of Electrical and Computer Engineering, University of Toronto, 10 King's College, Room SFB540, M5S 3G4 Toronto, ON, Canada. [5]Department of Molecular Genetics, University of Toronto, 1 King's College Circle, Room 4396, M4S 1A8 Toronto, ON, Canada. [6]The Donnelly Centre, 160 College Street, Room 230, M5S 3E1 Toronto, ON, Canada. [7]Department of Medical Biophysics, University of Toronto, 101 College Street, M5G 1L7 Toronto, ON, Canada. [8]Department of Pharmacology and Toxicology, University of Toronto, 1 King's College Circle, M5S 1A8 Toronto, ON, Canada.

Received: 21 November 2014 Accepted: 27 April 2015
Published online: 14 May 2015

References

1. Ein-Dor L, Kela I, Getz G, Givol D, Domany E. Outcome signature genes in breast cancer: is there a unique set? Bioinformatics. 2005;21(2): 171–8.
2. Ng CK, Weigelt B, A'Hern R, Bidard FC, Lemetre C, Swanton C, et al. Predictive performance of microarray gene signatures: impact of tumor heterogeneity and multiple mechanisms of drug resistance. Cancer Res. 2014;74(11):2946–61.
3. Fisher R, Pusztai L, Swanton C. Cancer heterogeneity: implications for targeted therapeutics. Br J Cancer. 2013;108(3):479–85.
4. Shen-Orr SS, Gaujoux R. Computational deconvolution: extracting cell type-specific information from heterogeneous samples. Curr Opin Immunol. 2013;25(5):571–8.
5. Yau C, Mouradov D, Jorissen RN, Colella S, Mirza G, Steers G, et al. A statistical approach for detecting genomic aberrations in heterogeneous tumor samples from single nucleotide polymorphism genotyping data. Genome Biol. 2010;11(9):92.
6. Van Loo P, Nordgard SH, Lingjærde OC, Russnes HG, Rye IH, Sun W, et al. Allele-specific copy number analysis of tumors. Proc Nat Acad Sci USA. 2010;107(39):16910–5.
7. Oesper L, Mahmoody A, Raphael BJ. THetA: inferring intra-tumor heterogeneity from high-throughput DNA sequencing data. Genome Biol. 2013;14(7):80.
8. Carter SL, Cibulskis K, Helman E, McKenna A, Shen H, Zack T, et al. Absolute quantification of somatic DNA alterations in human cancer. Nat Biotechnol. 2012;30(5):413–21.
9. de Ridder D, van der Linden CE, Schonewille T, Dik WA, Reinders MJ, van Dongen JJ, et al. Purity for clarity: the need for purification of tumor cells in DNA microarray studies. Leukemia. 2005;19(4):618–27.
10. Bachtiary B, Boutros PC, Pintilie M, Shi W, Bastianutto C, Li JH, et al. Gene expression profiling in cervical cancer: an exploration of intratumor heterogeneity. Clin Cancer Res. 2006;12(19):5632–40.
11. Emmert-Buck MR, Bonner RF, Smith PD, Chuaqui RF, Zhuang Z, Goldstein SR, et al. Laser capture microdissection. Science. 1996;274(5289):998–1001.
12. Erkkila T, Lehmusvaara S, Ruusuvuori P, Visakorpi T, Shmulevich I, Lahdesmaki H. Probabilistic analysis of gene expression measurements from heterogeneous tissues. Bioinformatics. 2010;26(20):2571–7.
13. Shen-Orr SS, Tibshirani R, Khatri P, Bodian DL, Staedtler F, Perry NM, et al. Cell type-specific gene expression differences in complex tissues. Nat Methods. 2010;7(4):287–9.
14. Bar-Joseph Z, Siegfried Z, Brandeis M, Brors B, Lu Y, Eils R, et al. Genome-wide transcriptional analysis of the human cell cycle identifies genes differentially regulated in normal and cancer cells. Proc Nat Acad Sci USA. 2008;105(3):955–60.
15. Lahdesmaki H, Shmulevich L, Dunmire V, Yli-Harja O, Zhang W. In silico microdissection of microarray data from heterogeneous cell populations. BMC Bioinformatics. 2005;6:54.
16. Ghosh D. Mixture models for assessing differential expression in complex tissues using microarray data. Bioinformatics. 2004;20(11):1663–9.
17. Stuart RO, Wachsman W, Berry CC, Wang-Rodriguez J, Wasserman L, Klacansky I, et al. In silico dissection of cell-type-associated patterns of gene expression in prostate cancer. Proc Nat Acad Sci USA. 2004;101(2): 615–20.
18. Gong T, Szustakowski JD. DeconRNASeq: a statistical framework for deconvolution of heterogeneous tissue samples based on mRNA-Seq data. Bioinformatics. 2013;29(8):1083–5.
19. Gong T, Hartmann N, Kohane IS, Brinkmann V, Staedtler F, Letzkus M, et al. Optimal deconvolution of transcriptional profiling data using quadratic programming with application to complex clinical blood samples. PLoS ONE. 2011;6(11):27156.
20. Abbas AR, Wolslegel K, Seshasayee D, Modrusan Z, Clark HF. Deconvolution of blood microarray data identifies cellular activation patterns in systemic lupus erythematosus. PLoS ONE. 2009;4(7):6098.
21. Wang M, Master SR, Chodosh LA. Computational expression deconvolution in a complex mammalian organ. BMC Bioinformatics. 2006;7:328.
22. Lu P, Nakorchevskiy A, Marcotte EM. Expression deconvolution: a reinterpretation of DNA microarray data reveals dynamic changes in cell populations. Proc Nat Acad Sci USA. 2003;100(18):10370–5.
23. Zhong Y, Wan YW, Pang K, Chow LM, Liu Z. Digital sorting of complex tissues for cell type-specific gene expression profiles. BMC Bioinformatics. 2013;14:89.
24. Kuhn A, Thu D, Waldvogel HJ, Faull RL, Luthi-Carter R. Population-specific expression analysis (PSEA) reveals molecular changes in diseased brain. Nat Methods. 2011;8(11):945–7.
25. Gaujoux R, Seoighe C. Semi-supervised Nonnegative Matrix Factorization for gene expression deconvolution: a case study. Infect Genet Evol. 2012;12(5):913–21.
26. Repsilber D, Kern S, Telaar A, Walzl G, Black GF, Selbig J, et al. Biomarker discovery in heterogeneous tissue samples -taking the in-silico deconfounding approach. BMC Bioinformatics. 2010;11:27.
27. Venet D, Pecasse F, Maenhaut C, Bersini H. Separation of samples into their constituents using gene expression data. Bioinformatics. 2001;17(Suppl 1):279–87.
28. Gaujoux R, Seoighe C. CellMix: a comprehensive toolbox for gene expression deconvolution. Bioinformatics. 2013;29(17):2211–2.
29. Zhong Y, Liu Z. Gene expression deconvolution in linear space. Nat Methods. 2012;9(1):8–9.
30. Clarke J, Seo P, Clarke B. Statistical expression deconvolution from mixed tissue samples. Bioinformatics. 2010;26(8):1043–9.
31. Gosink MM, Petrie HT, Tsinoremas NF. Electronically subtracting expression patterns from a mixed cell population. Bioinformatics. 2007;23(24):3328–34.
32. Ahn J, Yuan Y, Parmigiani G, Suraokar MB, Diao L, Wistuba II, et al. DeMix: deconvolution for mixed cancer transcriptomes using raw measured data. Bioinformatics. 2013;29(15):1865–71.
33. Quon G, Haider S, Deshwar AG, Cui A, Boutros PC, Morris Q. Computational purification of individual tumor gene expression profiles leads to significant improvements in prognostic prediction. Genome Med. 2013;5(3):29.
34. Parker JS, Mullins M, Cheang MC, Leung S, Voduc D, Vickery T, et al. Supervised risk predictor of breast cancer based on intrinsic subtypes. J Clin Oncol. 2009;27(8):1160–7.
35. Alizadeh AA, Eisen MB, Davis RE, Ma C, Lossos IS, Rosenwald A, et al. Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. Nature. 2000;403(6769):503–11.
36. MATLAB: Version 7.11.0.584 (R2010b) 64-bit. Natick, Massachusetts: The MathWorks Inc.; 2010.
37. R Core Team. R: A Language and Environment for Statistical Computing. Vienna, Austria: R Foundation for Statistical Computing; 2012. ISBN 3-900051-07-0. http://www.R-project.org/
38. Bates D, Eddelbuettel D. Fast and elegant numerical linear algebra using the RcppEigen package. J Stat Softw. 2013;52(5):1–24.
39. Bhattacharjee A, Richards WG, Staunton J, Li C, Monti S, Vasa P, et al. Classification of human lung carcinomas by mRNA expression profiling

reveals distinct adenocarcinoma subclasses. Proc Nat Acad Sci USA. 2001;98(24):13790–5.

40. Beer DG, Kardia SL, Huang CC, Giordano TJ, Levin AM, Misek DE, et al. Gene-expression profiles predict survival of patients with lung adenocarcinoma. Nat Med. 2002;8(8):816–24.
41. Wallace TA, Prueitt RL, Yi M, Howe TM, Gillespie JW, Yfantis HG, et al. Tumor immunobiological differences in prostate cancer between African-American and European-American men. Cancer Res. 2008;68(3): 927–36.
42. Wang Y, Xia XQ, Jia Z, Sawyers A, Yao H, Wang-Rodriquez J, et al. In silico estimates of tissue components in surgical samples based on expression profiling data. Cancer Res. 2010;70(16):6448–55.
43. Quon G, Morris Q. ISOLATE: a computational strategy for identifying the primary origin of cancers using high-throughput sequencing. Bioinformatics. 2009;25(21):2882–9.
44. Boutros PC, Ewing AD, Ellrott K, Norman TC, Dang KK, Hu Y, et al. Global optimization of somatic variant identification in cancer genomes with a global community challenge. Nat Genet. 2014;46(4):318–9.
45. Eddelbuettel D. Seamless R and C++ Integration With Rcpp. New York: Springer; 2013. ISBN 978-1-4614-6867-7.
46. Qiao W, Quon G, Csaszar E, Yu M, Morris Q, Zandstra PW. PERT: a method for expression deconvolution of human blood samples from varied microenvironmental and developmental conditions. PLoS Comput Biol. 2012;8(12):1002838.
47. Wang N, Gong T, Clarke R, Chen L, Shih IeM, Zang Z, et al. UNDO: a Bioconductor R package for unsupervised deconvolution of mixed gene expressions in tumor samples. Bioinformatics. Jan 2015;31(1):137–139.
48. Li Y, Xie X. A mixture model for expression deconvolution from RNA-seq in heterogeneous tissues. BMC Bioinformatics. 2013;14 Suppl 5:11.
49. Tolliver D, Tsourakakis C, Subramanian A, Shackney S, Schwartz R. Robust unmixing of tumor states in array comparative genomic hybridization data. Bioinformatics. 2010;26(12):106–14.
50. Roy S, Lane T, Allen C, Aragon AD, Werner-Washburne M. A hidden-state Markov model for cell population deconvolution. J Comput Biol. 2006;13(10):1749–74.
51. Wang N, Gong T, Clarke R, Chen L, Shih IM, Zhang Z, et al. UNDO: a Bioconductor R package for unsupervised deconvolution of mixed gene expressions in tumor samples. Bioinformatics. 2014.
52. Yadav VK, De S. An assessment of computational methods for estimating purity and clonality using genomic data derived from heterogeneous tumor tissue samples. Brief Bioinformatics. 2015;16(2):232–41.