Gmail Maciek Sykulski <macieksk@gmail.com>

Towards assignment algorithm

7 messages

Maciek Sykulski <macieksk@gmail.com> Sat, Dec 5, 2015 at 5:39 PM
To: Gregory Kucherov <Gregory.Kucherov@univ-mlv.fr>, Karel Břinda <karel.brinda@gmail.com>, Камиль Салихов <salikhov.kamil@gmail.com>

Hi Gregory, Hi Karel, Hi Kamil,

I think it will be easier to publish Ithaka paper, if it comes together with an assignment method. Thus, I've been thinking about it recently. Here are some results.

Generally I think about an EM algorithm which would solve two problems at once: abundance estimation, and assignment. This EM would maximize likelihood as a function of two types of variables:
1) abundances α_g such that $\sum_{g \in G} \alpha_g = 1$ correspond to proportion of reads from the whole sample which belong to genome g. Here G is the set of all genomes plus U - categories of unknown/unclassified reads.
2) categorical assignment indicator 0-1 variables $y_{f,g}$ which if equal to 1 mean that fragment (read) f comes from genome(category) g. Matrix of these variables is sparse – most of them are 0 – since we will only consider $y_{f,g}$ non zero if there are some hits from genome g in fragment f. $\sum_{g \in G} y_{f,g} = 1$

The total likelihood is proportional to

$$\left(\prod_{f \in F} \prod_{g \in G} P(f|g)^{y_{f,g}} \alpha_g^{y_{f,g}} \right) P_{prior}(\alpha)$$

$P_{prior}(\alpha)$ should be the conjugate prior distribution to categorical distribution, which is Dirichlet distribution.
Expectation step goes over $y_{f,g}$ variables (with α_g set)
Maximization step goes over α_g variables (with $y_{f,g}$ set).

P(f|g) is given, it's probability of obtaining fragment f assuming it comes from genome g, it depends on 1/kmer_richness_of_g, and my proposal for it depends also on $e_{f,g}$ -- a minimal number of errors/mutations in fragment f, given the coverage of the read f with hits from g. In other words, what is the minimal number of errors in fragment f to provide that the covarage looks how it looks (i.e. how it was found by ithaka query).

Computing $e_{f,g}$ turns out to be an instance of 0-1 integer programming -- an NP-complete problem. Thus, last week I programmed, using coin-Cbc integer programming library, a solution to this problem and in practice it works fast (it may work slower on seeds with many spaces, I am yet to test that).

Open problems:
1) what is the probability of one mutation_or_sequencing_error in a read? It may depend on sequencing technology.
2) what is the (approxmiate) k-mer richness of all organisms on Earth? How to estimate it? This one is needed to properly assign probability of assignment to unknown (unclassified read).

Maybe you can help with these problems?

Best regards,
Maciek

Gregory Kucherov <Gregory.Kucherov@univ-mlv.fr> Mon, Dec 7, 2015 at 7:20 PM
To: Maciek Sykulski <macieksk@gmail.com>
Cc: Karel Břinda <karel.brinda@gmail.com>, Kamil Salikhov <salikhov.kamil@gmail.com>

Hi Maciek,
Thanks for your mail.

Overall this is very interesting but unfortunately I could not understand everything in your mail.
It may well be that I am missing something (I am not a huge expert in EM).

For example, I didn't understand the idea of the likelihood formula. Do you really want to multiply \alpha_g as many times as there are reads coming from g? Sorry I may miss something... Could you give a few comments on your intuition about the formula?

The maximisation step looks like you simply compute the fractions \alpha_g from the information on the origin of each read, right?
I.e. you don't really "maximise" anything. Is this what is intended?

Concerning the NP-complete problem you talk about, wouldn't some simple heuristic do the job in practice ?

Sorry for my possible lack of insight.

Gregory

PS we will discuss your mail with Karel and Kamil at our next meeting

On 5 Dec 2015, at 17:39, Maciek Sykulski <macieksk@gmail.com> wrote:

> Hi Gregory, Hi Karel, Hi Kamil,
>
> I think it will be easier to publish Ithaka paper, if it comes together with an assignment method. Thus, I've been thinking about it recently. Here are some results.
>
> Generally I think about an EM algorithm which would solve two problems at once: abundance estimation, and assignment. This EM would maximize likelihood as a function of two types of variables:
> 1) abundances αg such that $\sum g \in G\ \alpha g = 1$ correspond to proportion of reads from the whole sample which belong to genome g. Here G is the set of all genomes plus U - categories of unknown/unclassified reads.
> 2) categorical assignment indicator 0-1 variables yf,g which if equal to 1 mean that fragment (read) f comes from genome(category) g. Matrix of these variables is sparse – most of them are 0 – since we will only consider yf,g non zero if there are some hits from genome g in fragment f. $\sum g \in G\ yf,g = 1$
>
> The total likelihood is proportional to
>
>
[Quoted text hidden]

Maciek Sykulski <macieksk@gmail.com> Mon, Dec 7, 2015 at 11:31 PM
To: Gregory Kucherov <Gregory.Kucherov@univ-mlv.fr>
Cc: Karel Břinda <karel.brinda@gmail.com>, Kamil Salikhov <salikhov.kamil@gmail.com>

Hi Gregory,

On Mon, Dec 7, 2015 at 7:20 PM, Gregory Kucherov <Gregory.Kucherov@univ-mlv.fr> wrote:

> Hi Maciek,
> Thanks for your mail.
> Overall this is very interesting but unfortunately I could not understand everything in your mail.
> It may well be that I am missing something (I am not a huge expert in EM).
>
> For example, I didn't understand the idea of the likelihood formula. Do you really want to multiply α_g as many times as there are reads coming from g? Sorry I may miss something... Could you give a few comments on your intuition about the formula?

Yes, $\alpha_g^{y_{f,g}}$ in principle shows up for every f (read) and every genome (g), but most of $y_{f,g} = 0$, precisely:
*) For each f there exists only one g for which $y_{f,g} = 1$.
(This is the case when interpreting the likelihood for a given assignment of variables. However, during the expectation step $y_{f,g}$ accept intermediate values between 0-1, and the above condition is changed so that $\sum_{g \in G} y_{f,g} = 1$. The task is to optimize with the best assignment of variables.).

The interpretation is as such:
*) vector α encodes categorical distribution among |G| categories (genomes),

$\sum_{g \in G} \alpha_g = 1$ and these are (sought) abundances of reads in the sample.
*) The total likelihood of obtaining a set of reads from a sample, is the product of likelihoods of obtaining each read independently.
The likelihood for obtaining one read is proportional to

$$P(f|g)^1 \alpha_g^1$$

this means: to obtain given read from genome g you randomly select it from |G| categories with probability given by α and then certain events happen:
the read is cut from somewhere in the genome g,
then sequencing errors happen.
Finally, the probability of these events is P(f|g).

Since I don't know a priori which genomes which fragments come from, I multiply all possibilities with $y_{f,g}$ 0-1 variables as powers.

> The maximisation step looks like you simply compute the fractions α_g from the information on the origin of each read, right?
> I.e. you don't really "maximise" anything. Is this what is intended?

During the maximization step many $y_{f,g}$ accept values between 0-1 (after the Expectation step). I compute (sum) effective exponents for each α_g.
The maximized expression (over α) is:

$$\text{const} * \prod_g \alpha_g^{e_g}$$

Maximization of this is the same as taking the mode of the appropriate Dirichlet distribution.

> Concerning the NP-complete problem you talk about, wouldn't some simple heuristic do the job in practice ?

I don't know, maybe, but the problem looks complicated, when the spaced seed is complicated. These spaces start to overlap with each other, or not, when shifting the seed, and the logical reasoning behind where a gap can be, or has to be, becomes difficult.
With the current approach processing HiSeq dataset (10000 reads) took approximately 10minutes with this seed: #######-###--#-##-#--##--########

> Sorry for my possible lack of insight.

Thank you for your input!

I still don't know what is the appropriate value for the probability of obtaining a sequencing error. I set it to 1/1000 for now, but this value must be published somewhere, at least for some sequencing technologies.

Best regards,
Maciek

> Gregory
>
> PS we will discuss your mail with Karel and Kamil at our next meeting
>
> On 5 Dec 2015, at 17:39, Maciek Sykulski <macieksk@gmail.com> wrote:
>
> > Hi Gregory, Hi Karel, Hi Kamil,
> >
> > I think it will be easier to publish Ithaka paper, if it comes together with an assignment method. Thus, I've been thinking about it recently. Here are some results.
> >
> > Generally I think about an EM algorithm which would solve two problems at once: abundance estimation, and assignment. This EM would maximize likelihood as a function of two types of variables:
> > 1) abundances αg such that ∑g ∈ G αg =1 correspond to proportion of reads from the whole sample which belong to genome g. Here G is the set of all genomes plus U - categories of unknown/unclassified reads.

> 2) categorical assignment indicator 0-1 variables yf,g which if equal to 1 mean that fragment (read) f comes from genome(category) g. Matrix of these variables is sparse – most of them are 0 – since we will only consider yf,g non zero if there are some hits from genome g in fragment f. ∑g ∈ G yf,g = 1
>
> The total likelihood is proportional to
>
>
> Pprior(α) should be the conjugate prior distribution to categorical distribution, which is Dirichlet distribution.
> Expectation step goes over yf,g variables (with αg set)
> Maximization step goes over αg variables (with yf,g set).
>
> P(f|g) is given, it's probability of obtaining fragment f assuming it comes from genome g, it depends on $1/kmer_{richness of g}$, and my proposal for it depends also on ef,g – a minimal number of errors/mutations in fragment f, given the coverage of the read f with hits from g. In other words, what is the minimal number of errors in fragment f to provide that the covarage looks how it looks (i.e. how it was found by ithaka query).
>
> Computing ef,g turns out to be an instance of 0-1 integer programming – an NP-complete problem. Thus, last week I programmed, using coin-Cbc integer programming library, a solution to this problem and in practice it works fast (it may work slower on seeds with many spaces, I am yet to test that).
>
> Open problems:
> 1) what is the probability of one $mutation_{or sequencing error}$ in a read? It may depend on sequencing technology.
> 2) what is the (approxmiate) k-mer richness of all organisms on Earth? How to estimate it? This one is needed to properly assign probability of assignment to unknown (unclassified read).
>
> Maybe you can help with these problems?
>
> Best regards,
> Maciek
>
>

Maciek Sykulski <macieksk@gmail.com> Tue, Dec 8, 2015 at 1:41 AM
To: Gregory Kucherov <Gregory.Kucherov@univ-mlv.fr>
Cc: Karel Břinda <karel.brinda@gmail.com>, Kamil Salikhov <salikhov.kamil@gmail.com>

Concerning the NP-complete problem of computing number of errors from a coverage.
This is just one proposal of P(f|g). If you have other ideas how to assign P(f|g) given the coverage of a spaced seed, please let me know.
Right now

$$P(f|g) = \frac{1}{\text{distinct kmers in g}} p_{err}^{k}(1-p_{err})^{\text{readlen}-k}$$

where k is the minimal number of errors for a given coverage.

An example coverage:
HS:Z:11111111111111111111111111111111111111111111100000010001101001011001100000

Best regards,
Maciek
[Quoted text hidden]

Gregory Kucherov <Gregory.Kucherov@univ-mlv.fr> Tue, Dec 8, 2015 at 12:23 PM
To: Maciek Sykulski <macieksk@gmail.com>
Cc: Karel Břinda <karel.brinda@gmail.com>, Kamil Salikhov <salikhov.kamil@gmail.com>

Maciek,
The sequencing error rate for Illumina is usually assumed to be 1.5%-2% (i.e. 0.015-0.02).
It is much higher for recent technologies like PacBio or Oxford Nanopore (about 15% for PacBio).

I'll write your more later.

Gregory

On 7 Dec 2015, at 23:31, Maciek Sykulski <macieksk@gmail.com> wrote:

> Hi Gregory,
>
> On Mon, Dec 7, 2015 at 7:20 PM, Gregory Kucherov <Gregory.Kucherov@univ-mlv.fr> wrote:
> Hi Maciek,
> Thanks for your mail.
> Overall this is very interesting but unfortunately I could not understand everything in your mail.
> It may well be that I am missing something (I am not a huge expert in EM).
>
> For example, I didn't understand the idea of the likelihood formula. Do you really want to multiply αg as many times
> as there are reads coming from g? Sorry I may miss something... Could you give a few comments on your intuition about the formula?
>
> Yes, αgyf,g in principle shows up for every f (read) and every genome (g), but most of yf,g = 0, precisely:
> *) For each f there exists only one g for which yf,g = 1.
> (This is the case when interpreting the likelihood for a given assignment of variables. However, during the expectation step yf,g accept intermediate values between 0-1, and the above condition is changed so that ∑g ∈ G yf,g = 1. The task is to optimize with the best assignment of variables.).
>
> The interpretation is as such:
> *) vector α encodes categorical distribution among |G| categories (genomes),
> ∑g ∈ G αg =1 and these are (sought) abundances of reads in the sample.
> *) The total likelihood of obtaining a set of reads from a sample, is the product of likelihoods of obtaining each read independently.
> The likelihood for obtaining one read is proportional to
>
> this means: to obtain given read from genome g you randomly select it from |G| categories with probability given by α and then certain events happen:
> the read is cut from somewhere in the genome g,
> then sequencing errors happen.
> Finally, the probability of these events is P(f|g).
>
> Since I don't know a priori which genomes which fragments come from, I multiply all possibilities with yf,g 0-1 variables as powers.
>
>
> The maximisation step looks like you simply compute the fractions αg from the information on the origin of each read, right?
> I.e. you don't really "maximise" anything. Is this what is intended?
>
> During the maximization step many yf,g accept values between 0-1 (after the Expectation step). I compute (sum) effective exponents for each αg.
> The maximized expression (over α) is:
>
>
>
[Quoted text hidden]

Gregory Kucherov <Gregory.Kucherov@univ-mlv.fr> Tue, Dec 8, 2015 at 11:25 PM
To: Maciek Sykulski <macieksk@gmail.com>
Cc: Karel Břinda <karel.brinda@gmail.com>, Kamil Salikhov <salikhov.kamil@gmail.com>

Maciek,
Thank you for the explanations, I understand your ideas much better now.
I think your approach is interesting and new.
Now my next questions.

The approach considers a set of genomes, without taking into account the taxonomic tree, right?
How do you plan to take into account the tree? Probably you can do this at the end of the procedure, after final y_{f,g} computed.

E.g. if y_{f,g1} and y_{f,g2} are both significant, then f might be assigned to a common ancestor of g1 and g2(?)

Now let us come to the computational aspects. The procedure computes the assignment of all reads simultaneously. You have to store all y_{f,g}'s, even if most of them are 0's, as you point out. Is this easy? Do you need some data structures for that?
What about the memory consumption? ...

Concerning the problem of computing the minimal number of errors from a coverage: I'll try to think about it, actually I am not completely sure that
it is NP-complete, as it is a very particular case of general NP-complete problems like set cover. Anyway, I assume that in practice
some simple greedy strategy should do a good job.

Best
Gregory

On 7 Dec 2015, at 23:31, Maciek Sykulski <macieksk@gmail.com> wrote:

> Hi Gregory,
>
> On Mon, Dec 7, 2015 at 7:20 PM, Gregory Kucherov <Gregory.Kucherov@univ-mlv.fr> wrote:
> Hi Maciek,
> Thanks for your mail.
> Overall this is very interesting but unfortunately I could not understand everything in your mail.
> It may well be that I am missing something (I am not a huge expert in EM).
>
> For example, I didn't understand the idea of the likelihood formula. Do you really want to multiply αg as many times
> as there are reads coming from g? Sorry I may miss something... Could you give a few comments on your intuition about the formula?
>
> Yes, αgyf,g in principle shows up for every f (read) and every genome (g), but most of yf,g = 0, precisely:
> *) For each f there exists only one g for which yf,g = 1.
> (This is the case when interpreting the likelihood for a given assignment of variables. However, during the expectation step yf,g accept intermediate values between 0-1, and the above condition is changed so that ∑g ∈ G yf,g = 1. The task is to optimize with the best assignment of variables.).
>
> The interpretation is as such:
> *) vector α encodes categorical distribution among |G| categories (genomes),
> ∑g ∈ G αg =1 and these are (sought) abundances of reads in the sample.
> *) The total likelihood of obtaining a set of reads from a sample, is the product of likelihoods of obtaining each read independently.
> The likelihood for obtaining one read is proportional to
>
> this means: to obtain given read from genome g you randomly select it from |G| categories with probability given by α and then certain events happen:
> the read is cut from somewhere in the genome g,
> then sequencing errors happen.
> Finally, the probability of these events is P(f|g).
>
> Since I don't know a priori which genomes which fragments come from, I multiply all possibilities with yf,g 0-1 variables as powers.
>
>
> The maximisation step looks like you simply compute the fractions αg from the information on the origin of each read, right?
> I.e. you don't really "maximise" anything. Is this what is intended?
>
> During the maximization step many yf,g accept values between 0-1 (after the Expectation step). I compute (sum) effective exponents for each αg.
> The maximized expression (over α) is:
>
>
>
[Quoted text hidden]

Maciek Sykulski <macieksk@gmail.com> Wed, Dec 9, 2015 at 1:46 AM
To: Gregory Kucherov <Gregory.Kucherov@univ-mlv.fr>
Cc: Karel Břinda <karel.brinda@gmail.com>, Kamil Salikhov <salikhov.kamil@gmail.com>

Hi Gregory,

On Tue, Dec 8, 2015 at 11:25 PM, Gregory Kucherov <Gregory.Kucherov@univ-mlv.fr> wrote:

> Maciek,
> Thank you for the explanations, I understand your ideas much better now.
> I think your approach is interesting and new.

The idea for α_g abundance vector is taken from kallisto algorithm, and the $y_{f,g}$ are based on categorical variables I had a chance to use in my phd disseration.

> Now my next questions.
>
> The approach considers a set of genomes, without taking into account the taxonomic tree, right?
> How do you plan to take into account the tree?

This is a good question, and I tried something like this: let's treat inner taxonomic nodes in exactly the same fashion as genomes. After all they have their kmer counts computed, and coverages for reads. Let's assign them $\alpha_{innernode}$ and $y_{f,innernode}$

> Probably you can do this at the end of the procedure, after final $y_{f,g}$ are computed.
> E.g. if $y_{f,g1}$ and $y_{f,g2}$ are both significant, then f might be assigned to a common ancestor of g1 and g2(?)

Yes, and in the approach I outlined above there is simply $y_{f,ga}$ variable, where ga is the common ancestor of g1,g2. If it turns out that $y_{f,ga} > y_{f,g1}$, and $y_{f,ga} > y_{f,g2}$ then the read f is assigned to ga.

> Now let us come to the computational aspects. The procedure computes the assignment of all reads simultaneously.

Yes.

> You have to store all $y_{f,g}$'s, even if most of them are 0's, as you point out. Is this easy? Do you need some data structures for that?
> What about the memory consumption? ...

I store a sparse matrix column wise, and I only store these variables which have a chance not to be zero and there is exactly as many elements of the matrix as rows in the output SAM/BAM file. (which indeed can be many: ex. 2.2 million for 50K HMP tongue dorsum sample, but this could be improved with some preprocessing filtering).
A good thing is that all computations can be easily parallelized.

> Concerning the problem of computing the minimal number of errors from a coverage: I'll try to think about it, actually I am not completely sure that
> it is NP-complete, as it is a very particular case of general NP-complete problems like set cover.

Yes, the case is particular. Maybe it's not NP-hard. It may be polynomial with the exponent of seed length (but this wouldn't help much).

> Anyway, I assume that in practice
> some simple greedy strategy should do a good job.

Possibly such a strategy should be implemented, or some other heuristic, because right now for 50K HMP sample this was the slowest part of the pipeline, because there are many hits/alignments with very low coverage (many 0) and this means that the integer programming problem has many variables, and solving it is slow. I actually stop it before it's solved and use the best achieved solution so far, but this still takes more than 0.01s for each read. Maybe we should filter such low coverage cases, but only if not all cases for a given read have low coverage.

I wonder how a greedy strategy for such integer programming problem looks like. It must be something like: select and set a variable to 0 (error) such that the most inequality constraints are being satisfied.

Best regards,
Maciek

> [Quoted text hidden]
> [Quoted text hidden]
> > With the current approach processing HiSeq dataset (10000 reads) took approximately 10minutes with this seed: ~~#######-###-#-##-#-##-########~~
> [Quoted text hidden]