Институт Биоинформатики

Научный руководитель:
Алексей Сазанов
ПСПбГМУ им. академика И. П. Павлова
Студент:
Фурменков Александр
Институт биоинформатики

Разработка диагностической системы предрасположенности к клинической депрессии на основе таргетного секвенирования

13.12.014

А зачем это нужно?

Патологии бывают разные, в нашем случае клиническая депрессия.

Возможные симптомы:

- Упадок производительности
- Утрата интересов и удовольствия от жизни
- Мрачное и пессимистическое видение будущего
- Идеи или действия, приводящие к самоповреждению или суициду

Иными словами, клиническая депрессия - это плохо.

Как вообще разрабатываются диагностические системы?

Обобщенный план в четыре шага:

1. Найти в базах данных гены и SNP связанные с предрасположенностью к той или иной патологии;
2. Выбирать из них те аллели, которые наиболее часто встречаются в требуемой популяции и вносят наибольший вклад в развитие заболевания;
3. Создать панели для таргетного секвенировнания;
4. Создать программу для выявления и аннотирования изучаемых SNP в полученных сиквенсах.

Как вообще разрабатываются диагностические системы?

Обобщенный план в четыре шага:

Уже сделано предшественником

1. ~~Найти в базах данных гены и SNP связанные с предрасположенностью к той или иной патологии;~~
2. ~~Выбирать из них те аллели, которые наиболее часто встречаются в требуемой популяции и вносят наибольший вклад в развитие заболевания;~~
3. Создать панели для таргетного секвенировнания;
4. Создать программу для выявления и аннотирования изучаемых SNP в полученных сиквенсах.

Постановка задачи

1. Составить панели для таргетного секвенирования;

2. Разработать программу для диагностики предрасположенности к клинической депрессии по результатам таргетного секвенирования. Программа должна искать известные патогенные аллели генов, наличие которых было выявлено у людей страдающих клинической депрессией.

Панели для таргетного секвенирования

Указываем интересующие нас диапазоны в геноме, которые должны содержать SNP. Стараемся минимизировать число диапазонов, которые нужно отсеквенировать.

SNP(SNV) name	Chromosome	Orientation/Strand
rs25532	Chr 17:30237152 +- 25	fwd/B
rs140700	Chr 17:30216371 +- 25	fwd/T
rs6355	Chr 17:30221792 +- 25	fwd/B
rs7137478	Chr 12:84855439 +- 25	fwd/B
rs11833800	Chr 12:84887749 +- 25	fwd/T
rs7959178	Chr 12:84900728 +- 25	fwd/B
rs2278361	Chr 12:98649429 +- 25	fwd/B
rs10745834	Chr 12:98666572 +- 25	fwd/T
rs2288726	Chr 12:98690609 +- 25	fwd/B

Программа

На вход 2 файла:

Файл с последовательностями SNP взятыми из референса

```
>Assembly:GRCh38 | SNVName:rs25532    | Functional:upstream | Minor:T
CCAGCATCCCCCCATGCACCCCCGG*ATCCCCCCTGCACCCCTCCAGCATT
>Assembly:GRCh38 | SNVName:rs140700   | Functional:intron   | Minor:A
TGTGATCTTTCTGCCACACCACCTC*CCCTCCTTTCTCAAGGTCTTCAAGA
>Assembly:GRCh38 | SNVName:rs6355     | Functional:missense | Minor:G
TGGGATAGAGTGCCGTGTGTCATCT*CCGCACCAGGACTTGGAACTGCTGA
>Assembly:GRCh38 | SNVName:rs7137478  | Functional:-        | Minor:T
TTAGCATTTTTTGTAGAAGAGGTCG*GTATGAATGAAATCTCTCAGCTTTT
>Assembly:GRCh38 | SNVName:rs11833800 | Functional:intron   | Minor:G
AGAAAGAATTATGACTGAACTCATT*GGGATTTTTTTCCAGTCATGTGTGG
>Assembly:GRCh38 | SNVName:rs7959178  | Functional:intron   | Minor:C
CATATCAGTTCTTCATCTATTCATA*GGAGTATCACATTCGTTGCTTTTCT
>Assembly:GRCh38 | SNVName:rs2278361  | Functional:intron   | Minor:C
TAGAACTTTGGAATAATTACTGAAG*AATACCATAAAAAACAAAGAGCAAA
>Assembly:GRCh38 | SNVName:rs10745834 | Functional:intron   | Minor:G
TCACTAACCTACGTGTTCTTTTTGT*GTCCAGGCCAATTTAGGATCCCGTA
>Assembly:GRCh38 | SNVName:rs2288726  | Functional:intron   | Minor:T
AATGGCCTAGTTTTACAGCTTTCTT*AACATTTGGCAACCTAATAATAATG
```

Файл с ридами от секвенатора

```
>SNVName:rs25532    | Contain allele T
AAAGCATCCCCCCATGCACCCCCGGTATCCCCCCTGCACCCCTCCAGCATTACT
>SNVName:rs140700   | Contain allele A
CCGGTTTGTGATCTTTCTGCCACACCACCTCACCCTCCTTTCTCAAGGTCTTCAAGA
>SNVName:rs6355     | Contain allele G
TGGGTTTAGAGTGCCGTGTGTCATCTGCCGCACCAGGACTTGGAACTGCTGA
>SNVName:rs7137478  | Contain allele C
TTAGCATTTTTTGTAGAAGAGGTCGCGTATGCTGGAAATCTCTCAGCTTTT
>SNVName:rs11833800 | Contain allele G
AGAAAGAATAATGACTGAACTCATTGGGGATTTTTTTCCAAACATGTGTGG
>SNVName:rs7959178  | Contain allele C
ATTCGTTGCATATCAGTTCTTCATCTATTCATACGGAGTATCACATTCGTTGCTTTTCTATTCGTTG
>SNVName:rs2278361  | Contain allele T
TAGAACTTTGGAATAATTACTGAAGTAATACCATAAAAAACAAAGAGCAAA
>SNVName:rs10745834 | Contain allele A
TCACTAACCTACGTTTTCTTTTTGTAGTCCAGGCCAATTTAGGATCCCGTACACTA
>SNVName:rs2288726  | Contain allele T
ATGCCAATGGCCTAGTTTTACAGCTTTCTTTAACATTTGGCAACCTAATAATAATGCCTAG
```

Программа

На выходе все найденные аллели и ссылка на их описание

```
        1) In DNA sequence "SNVName:rs25532    | Contain allele T" in
position 25 was found SNP(SNV) rs25532 with minor nucleotide "T"
You can read more about it SNP(SNV) on http://www.ncbi.nlm.nih.gov/SNP/
snp_ref.cgi?rs=25532

        2) In DNA sequence "SNVName:rs140700   | Contain allele A" in
position 31 was found SNP(SNV) rs140700 with minor nucleotide "A"
You can read more about it SNP(SNV) on http://www.ncbi.nlm.nih.gov/SNP/
snp_ref.cgi?rs=140700

        3) In DNA sequence "SNVName:rs6355     | Contain allele G" in
position 26 was found SNP(SNV) rs6355 with minor nucleotide "G"
You can read more about it SNP(SNV) on http://www.ncbi.nlm.nih.gov/SNP/
snp_ref.cgi?rs=6355

        4) In DNA sequence "SNVName:rs11833800 | Contain allele G" in
position 25 was found SNP(SNV) rs11833800 with minor nucleotide "G"
You can read more about it SNP(SNV) on http://www.ncbi.nlm.nih.gov/SNP/
snp_ref.cgi?rs=11833800
```

Как работает и почему именно так?

По сути задача сводится к поиску паттерна с ошибками, где паттерн это последовательность окружающая SNP, взятая из референса, а текст для поиска - риды полученные от секвенатора.

Почему с ошибками?

Вокруг интересующего нас SNP могут вплотную располагаться и другие снипы, на которые мы не рассчитываем.

Как работает и почему именно так?

Реальный пример.

Нам интересен SNP находящийся на месте *, его окружают нуклеотиды из референса:

TGGGATAGAGTGCCGTGTGTCATCT*CCGCACCAGGACTTGGAACTGCTGA

Однако там возможны и другие снипы:

TGGGATAGAGTGCCGTGTGTCATCT*CCGCACCAGGACTTGGAACTGCTGA

Как работает и почему именно так?

Проблема не только в неучтенных снипах, возможны и другие проблемы - вставки и удаления (indel’s) произвольного количества нуклеотидов вокруг.

Например последовательность из референса

TGGGATAGAGTGCCGTGTGTCATCT*CCGCACCAGGACTTGGAACTGCTGA

Может в реальности оказаться такой

TGGGATA____GCCGTGTGTCATCTC_GCACCAGGACTTGGAACTGCTGA

При этом инделы рядом с интересующим нас SNP самые плохие, тк из-за них становится почти невозможным различить интересующий нас SNP, от рядом стоящих нуклеотидов.

Как работает и почему именно так?

Для упрощения было решено искать так, как будто indel’ов нет (по сравнению со снипами они возникают гораздо реже).

Есть разные алгоритмы неточного поиска:

1. Основанные на сведение задачи к точному поиску, например алгоритм “Расширение выборки”
2. Основанные на использование расстояния Хэмминга (похожесть строк оценивается по числу не совпадений между ними)
3. Основанные на использование деревьев и других вспомогательных структур данных.

Как работает и почему именно так?

В итоге, тк объемы данных не очень большие, то было решено использовать расстояние Хэмминга с некоторыми проверками, для того чтобы не принять не правильный результат.

В итоге алгоритм стал выглядеть так:

1. Выбираем рид и ищем в нем по очереди все заданные снипы
2. В каждом риде сперва ищем позицию с минимальным расстоянием Хэмминга, для левой части паттерна текущего снипа, затем для правой. Если для одной части есть два результата - сохраняем оба для последующей проверки.
3. Проверяем все комбинации найденных позиций для левых и правых частей паттерна. Расстояние хэмминга для каждой части не должно превышать половину, от длины части. Суммарное расстояние Хэмминга для двух частей паттерна не должно быть меньше половины, от длины паттерна.
4. Проверяем, что между найденными левой и правой частью паттерна, находится ровно один нуклеотид - наш снип. Если это не так, или на данном этапе осталось несколько кандидатов, то заявляем о невозможности найти снип. Иначе сохраняем результат.

Что можно было бы еще сделать?

1. Добавить учет indel’ов.

2. Подобрать более эффективный алгоритм, или эффективнее реализовать имеющийся.

3. Формализовать проблему и попытаться использовать статистические оценки, для фильтрации результатов поиска.

Спасибо за внимание!