BMC Bioinformatics

METHODOLOGY ARTICLE Open Access

A spatial simulation approach to account for protein structure when identifying non-random somatic mutations

Gregory A Ryslik[1*], Yuwei Cheng[2], Kei-Hoi Cheung[2,3], Robert D Bjornson[4], Daniel Zelterman[1], Yorgo Modis[5] and Hongyu Zhao[1,2*]

Abstract

Background: Current research suggests that a small set of "driver" mutations are responsible for tumorigenesis while a larger body of "passenger" mutations occur in the tumor but do not progress the disease. Due to recent pharmacological successes in treating cancers caused by driver mutations, a variety of methodologies that attempt to identify such mutations have been developed. Based on the hypothesis that driver mutations tend to cluster in key regions of the protein, the development of cluster identification algorithms has become critical.

Results: We have developed a novel methodology, *SpacePAC* (**S**patial **P**rotein **A**mino acid **C**lustering), that identifies mutational clustering by considering the protein tertiary structure directly in 3D space. By combining the mutational data in the Catalogue of Somatic Mutations in Cancer (COSMIC) and the spatial information in the Protein Data Bank (PDB), *SpacePAC* is able to identify novel mutation clusters in many proteins such as FGFR3 and CHRM2. In addition, *SpacePAC* is better able to localize the most significant mutational hotspots as demonstrated in the cases of BRAF and ALK. The R package is available on Bioconductor at: http://www.bioconductor.org/packages/release/bioc/html/SpacePAC.html.

Conclusion: *SpacePAC* adds a valuable tool to the identification of mutational clusters while considering protein tertiary structure.

Background

Cancer, at its most basic, is caused by the accrual of somatic mutations within oncogenes and tumor suppressors in the genome [1]. While mutations within tumor suppressors usually lower or completely disrupt the activity of genes that promote cell apoptosis or regulate the cell cycle, oncogenic mutations typically increase or destabilize the resulting protein output. As it is easier to disrupt protein function than restore it, there has been significant pharmacological research geared towards inhibiting oncogenic mutations as described in [2,3] and [4]. Coupled with the idea of "oncogene addiction", that a small set of "driver" genes promote uncontrolled cellular growth in a wide variety of cancers and that inactivation of these genes can significantly impair tumorigenesis [5,6], the identification of driver oncogenic mutations has become of key importance due to its potential translational benefit.

Due to the biological importance of this problem, a variety of methodologies have been proposed to identify regions where activating mutations may occur. One approach is based on the idea that compared to the background mutation rate, driver mutations will have a higher frequency of non-synonymous mutations [7,8]. Several improvements to this approach have been made such as normalizing for gene length [9] as well as accounting for different mutation rates due to features such as transitions versus transversions, location of *CpG* sites and tumor type [10]. Relatedly, instead of comparing the mutational frequency directly to the background rate, one can also compare the ratio of nonsynonymous (K_a) to synonymous (K_s) mutations per site [11]. Recently, [5,8,12]

*Correspondence: gregory.ryslik@yale.edu; hongyu.zhao@yale.edu
[1] Department of Biostatistics, Yale School of Public Health, New Haven, CT, USA
[2] Program of Computational Biology and Bioinformatics, Yale University, New Haven, CT, USA
Full list of author information is available at the end of the article

and [13] showed that somatic mutations appear to cluster within protein kinases while [14] and [15] demonstrated that mutational clustering can be a sign of positive selection for protein function and thus sites for protein engineering.

Alternatively, several machine learning methods have been developed to determine the nature of a specific mutation. For instance, *Polyphen-2* [16] attempts to discern whether a mutation is deleterious while *CHASM* [17] attempts to distinguish between driver and passenger mutations. These methods use a wide range of sequence and non-sequence features to build a set of rules that are then used to score each mutation with the score value determining the mutation classification. The rules are developed on a training data set via a variety of methods such as Random Forests [18], Bayesian Networks [19] and Support Vector Machines [20]. The features used often include information on the size and polarity of the substituted and original residues, available structural information as well as evolutionary conservation [21]. Some classifiers are optimized to use only a small set of features in prediction. For example, the *SIFT* classifier [22], only uses evolutionary conservation to predict whether the protein functional change is tolerated or damaging.

While all these methods have shown some success in identifying damaging or deleterious mutations, they nevertheless have limitations. Methods that rely upon differentiating the frequency of synonymous and non-synonymous mutations as compared to the background rate may fail to take into account that selection may occur upon only a small region of the gene and that signal loss may occur when the gene is considered in total. The methodologies proposed by [9] and [11] fail to distinguish between activating and non-activating non-synonymous mutations while the method developed by [10] may be biased if the background mutational rate is not accurately estimated. Furthermore, not only do machine learning classifiers require several sources of information that need to be periodically updated to account for new research, it is often the case that much of the requisite information needs to be collected for the first time at significant expense.

Building upon the hypothesis that driver mutations tend to cluster in functionally relevant protein regions, [23,24] and [25] recently developed several statistical methodologies to identify mutational clusters. Specifically, [23] developed Non-Random Mutational Clustering (*NMC*) which identifies clustering by testing against the null hypothesis that missense substitution locations are distributed uniformly. If two mutations are closer together than expected by chance, the null hypothesis that every amino acid has an equal probability of mutation is thus rejected. However, as the distance between every pair of mutations needs to be tested, a large multiple comparison penalty is incurred. *iPAC* [24] and *GraphPAC* [25] expanded upon *NMC* by taking into account protein tertiary structure via a MultiDimensional Scaling approach (MDS) [26] and a graph theoretical approach, respectively. While both of these methods improved over the linear *NMC* method, they nevertheless remap the protein to one dimensional space resulting in information loss.

In this article, we provide an improvement to *iPAC*, *GraphPAC* and *NMC* by considering the protein directly in three dimensional space and thereby avoid the information loss inherent in dimension reduction algorithms. Using this new approach, we are able to identify proteins with significant clusters, such as FGFR3 and CHRM2, which are otherwise missed by *iPAC*, *GraphPAC* and *NMC* (see Section "*SpacePAC* identifies additional proteins containing clusters"). In addition, *SpacePAC* provides better "localization" for mutational hotspots (see Section "*SpacePAC* improves cluster localization"). Finally, we show that many of the mutational hotspots identified by *SpacePAC* are categorized as activating mutations by *CHASM* and damaging mutations by *PolyPhen-2*. Overall, by avoiding the protein remapping step required by *iPAC* and *GraphPAC* as well as the multiple comparison penalty these methods incur for looking at every pairwise combination of mutations, we are better able to identify mutational hotspots that are indicative of driver mutations. For the rest of this paper, we refer to the set of *NMC*, *GraphPAC*, and *iPAC* as the "pairwise methods" as they consider every pairwise combination at the cost of an extra multiple comparison adjustment.

Methods

SpacePAC uses a three step process to identify mutational clusters. Step one is to obtain the mutational and structural data (see Sections "Obtaining mutational data" and "Obtaining 3D structural data"). Step two is to reconcile the databases so that the mutational information can be mapped onto the protein structure (see Section "Reconciling structural and mutational data"). The third step is to simulate the distribution of mutation locations over the protein tertiary structure and identify if any regions of the protein have observed mutational counts in the tail of the distribution (see Section "Identifying mutational hotspots"). Finally, although not part of the *SpacePAC* algorithm, we perform a multiple comparison adjustment to account for the multiple structures considered (see Section "Multiple comparison adjustment for structures").

Obtaining mutational data

The 65th Oracle version of the COSMIC database, the latest version as of when this paper was written, was used to obtain the mutation data. First, only missense substitution mutations recorded as "confirmed somatic variant" or "Reported in another cancer sample as somatic" were used. Further as both *SpacePAC* and the pairwise methods are tested against the null hypothesis that mutations are randomly distributed along the protein, only mutations from whole gene or whole genome screens were retained in order to avoid selection bias. Next, only genes labeled with a UniProt Accession Number [27] were kept as the UniProt identification was used to match the protein sequence to the protein structure. Finally, as several studies may report mutational data from a single cell line, mutation duplications were removed in order to avoid over counting specific mutations (see "Additional file 1: COSMIC query" for the SQL code). Mutations are then aggregated from various sources (cell lines, tumor samples, etc.) to help provide a complete mutational distribution for the protein during the analysis described in Section "Algorithm for identifying sphere positions".

Obtaining 3D structural data

Protein structures were obtained from the PDB. The spatial coordinates of the α-carbon atom in each amino acid were used to represent that amino acid's location in 3D space. Further, as multiple structures are often available for the same protein, all structures that matched the protein's UniProt Accession Number were analyzed and an appropriate multiple comparison adjustment applied afterwards (see Section "Multiple comparison adjustment for structures"). If multiple polypeptide chains within the same structure matched the UniProt Accession Number, the first matching chain shown in the file was used (commonly chain "A"). Similarly, if the structure determination provided more than one protein conformation, the first conformation listed in the pdb file was kept. We note that while we only used the first conformation displayed in the file, as done by [24] and [25], the *SpacePAC* software allows the user to select the desired conformation should an analysis of more than one alternative conformation be required. For a full listing of the 1,903 structure/side-chain combinations considered, see "Additional file 2: Structure files".

Reconciling structural and mutational data

As the residue numbering scheme often differs between the COSMIC and PDB databases, we reconciled the information in both sources. Similar to *iPAC* and *GraphPAC*, *SpacePAC* provides the user two possible reconciliation options. The first option is based upon a numerical reconstruction from the structural data available directly in the PDB file while the second performs a pairwise alignment as detailed in [28]. As the PDB file structure may change depending upon when the file was added to the database along with other technical difficulties, we used pairwise alignment to reconcile the mutational and positional information for each structure unless specifically noted. For further information on the pairwise alignmnent, please see the *iPAC* package available on Bioconductor. Successful alignment was obtained for 131 proteins corresponding to 1,110 individual structure/side-chain combinations. Note that structures that did not have tertiary data on at least two mutations were considered blank (as no clustering is possible) and dropped from the analysis. See "Additional file 2: Structure files" for full details of each combination.

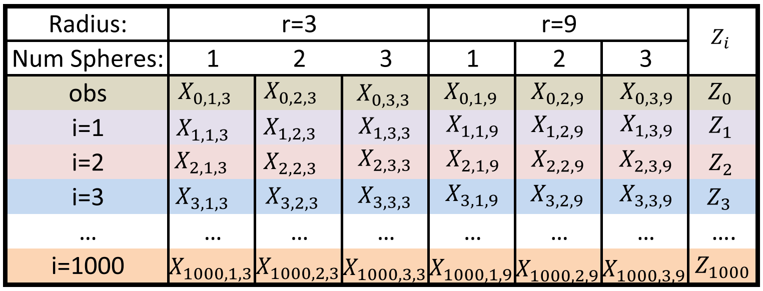
Identifying mutational hotspots

The general principle for *SpacePAC* is that we identify the one, two and three non-overlapping spheres that cover as many of the mutations as possible at various radii lengths. We then normalize the number of mutations covered by the spheres and find the maximum normalized value. This value is then compared to a simulated distribution to obtain a p-value. Specifically, we proceed as follows:

- Let s be the number of spheres we consider.
  $s \in \{1, 2, 3\}$.
- Let r be the radius considered. Here we consider,
  $r \in \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$.
- Simulate $T(\geq 1000)$ distributions of mutation locations over the protein structure. Specifically, for each simulation, every mutation is randomly assigned to a residue i where $1 \leq i \leq N$ and N is the total number of residues in the protein.

Next, let $X_{0,s,r}$ represent the number of mutations that occur within the s spheres over all samples. For instance, suppose we had three tumor samples, A, B and C, and were considering two spheres. Further, suppose that sample A had mutations at residues 5 and 20, sample B had mutations at residues 6 and 21, and sample C had only one mutation at residue 5. Thus, if sphere one overlapped residues 5 and 6 while sphere two overlapped residues 20, 21, then $X_{0,2,r} = 5$. Specifically, there are 3 mutations from samples A, B, C within the first sphere and 2 mutations within the second sphere.

Given, s we then identify the sphere centers in such a way that capture as many of the total mutations as possible (See Section "Algorithm for identifying sphere positions"). For instance, if $s = 3$, we then need to identify three sphere centers, p_1, p_2 and p_3. Let $X_{i,s,r}$ represent the same statistic but for simulation i. For given $\{s, r\}$, calculate $\mu_{s,r} = \underset{1 \leq i \leq T}{\text{mean}}\{X_{i,s,r}\}$ and $\sigma_{s,r} = \underset{1 \leq i \leq T}{\text{std. dev.}}\{X_{i,s,r}\}$. For each simulation, calculate $Z_i = \max_i\{(X_{i,s,r} - \mu_{s,r})/\sigma_{s,r}\}$. The p-value is then estimated as: $(\sum \mathbf{1}_{Z_0 > Z_i})/T$. Note that



Radius:	r=3			r=9			Z_i
Num Spheres:	1	2	3	1	2	3	
obs	$X_{0,1,3}$	$X_{0,2,3}$	$X_{0,3,3}$	$X_{0,1,9}$	$X_{0,2,9}$	$X_{0,3,9}$	Z_0
i=1	$X_{1,1,3}$	$X_{1,2,3}$	$X_{1,3,3}$	$X_{1,1,9}$	$X_{1,2,9}$	$X_{1,3,9}$	Z_1
i=2	$X_{2,1,3}$	$X_{2,2,3}$	$X_{2,3,3}$	$X_{2,1,9}$	$X_{2,2,9}$	$X_{2,3,9}$	Z_2
i=3	$X_{3,1,3}$	$X_{3,2,3}$	$X_{3,3,3}$	$X_{3,1,9}$	$X_{3,2,9}$	$X_{3,3,9}$	Z_3
...	...	...	...	...	...	...	....
i=1000	$X_{1000,1,3}$	$X_{1000,2,3}$	$X_{1000,3,3}$	$X_{1000,1,9}$	$X_{1000,2,9}$	$X_{1000,3,9}$	Z_{1000}

Figure 1 Statistic construction. In this example, we consider $r \in \{3, 9\}$ and up to 3 potential mutational hotspots in the protein. First, μ and σ are calculated over each column. Next, we normalize each entry in the column by calculating $Z_{i,s,r} = \frac{X_{i,s,r} - \mu_{s,r}}{\sigma_{s,r}}$. We then take the maximum over each row to get $Z_0, \ldots, Z_{1000}$. The percentage of times $Z_0 \geq Z_i$, where $i \in \{1, \ldots, 1000\}$, is the p-value of our observed statistic Z_0. Note that if Z_0 is greater than Z_i for all 1000 simulations, we report a p-value of <1.00 E-03.

while we identify up to three spheres (or "hotspots") that contain mutational clustering, only one p-value is necessary to reflect the statistical significance of all the hotspots. This process is best seen through Figure 1. For the rest of this paper, we will refer to "hotspot" and "cluster" interchangeably.

While our algorithm is generalizable to m spheres ($m << N$, where n is the total number of residues), we note that the worst case running time is $O\binom{N}{m}$. Due to the nature in which we traverse the sample space of possible sphere orientations (see Section "Algorithm for identifying sphere positions"), we rarely hit the worst case scenario, and even then, in the case of three spheres the running time is only $O\binom{N}{3}$ which is still computationally possible. Further, our algorithm was able to identify only 1 or 2 significant spheres for over 70% of the structures considered. Even more so, for the structures that did have 3 statistically significant spheres (hotspots), the three spheres covered on average more than 75% of all the mutations available. Given these empirical results, three spheres were selected to balance computational costs.

While sphere radii ranging from 1 to 10Å were considered in this study, the researcher can input larger values. However, over all the structures evaluated in this study, our methodology never required spheres of radius 10Å. Optimal sphere orientations were always found at smaller radii. Thus while larger sphere radii could of been considered, they would not have been selected by the algorithm.

Finally, due to our consideration of only non-overlapping spheres, additional constraints on the number of possible sphere orientations are placed which allows *SpacePAC* to efficiently solve for the optimal sphere orientation. For example, if two nearby residues are mutated, *SpacePAC* will incorporate them into a larger sphere and then use the remaining spheres to identify more distant clusters. If sphere overlap was permitted, not only would it become difficult to assign a mutation to a specific sphere but it could also lead to uninformative results. For instance, consider a worst case scenario where one residue has very many mutations and two spheres are being used to find the mutational hotspots. If the rest of the protein has no mutations, the two spheres will perfectly overlap which would in turn provide a meaningless result. If the mutations were spread over two nearby residues, the sphere overlap would again be almost perfect and not informative. By using the non-overlapping sphere constraint, *SpacePAC* will dynamically increase the sphere radius and capture the mutations in one sphere. On the whole, by considering only non-overlapping spheres we impose boundary constraints in order to find meaningful sphere configurations.

Algorithm for identifying sphere positions

In the approach described in Section "Identifying mutational hotspots", we find the 1, 2 and 3 non-overlapping spheres that cover the most mutations given a pre-specified radius length. Ignoring sphere overlap for the moment, if only one sphere is considered, the number of possible spheres is linear in the length of the protein (namely, a sphere centered at each residue). If two spheres are considered, there are $\binom{N}{2}$ possible sphere combinations if the protein is N residues long. Similarly, if three spheres are considered, there are $\binom{N}{3}$ such combinations (once again ignoring sphere overlap). For a medium-sized protein like PIK3Cα which is 1,068 residues long, considering three spheres allows for $\binom{1{,}068}{3} = 202{,}461{,}116$ possible positions. This makes it prohibitively expensive to perform a brute force approach. To quickly find the best sphere orientation, we execute Algorithm 1 below. Algorithm 1 is presented for 2 spheres but is trivially extendable to 3 or more spheres as well.

Algorithm 1 *SpacePAC* algorithm. We are interested in finding the two non-overlapping spheres that contain the most mutations. The algorithm takes as input a sorted vector v of mutation counts per amino acid where the amino acids are sorted from largest to smallest by mutation count. Amino acids with no mutations are not included in v. Note that the "cand" variable is an ordered list of 3-tuples.

```
Require: Sorted vector of counts $v$ with length $>= 2$
  starti = 2;
  startj = 1;
  k = length(v);
  cand = [(starti, startj,v[starti] + v[startj])]
  while (!is.empty(cand)) do
    index = max(cand) {#Max over the 3rd element in the 3-tuple.}
    i,j,s = cand[index]
    cand = cand[-index] {#Removes the current max.}
    if (No overlap between sphere i and j) then
      Return (i, j, v[i]+v[j]) {#Successful combination found.}
    end if
    if (j ==1) and ($i < k$) then
      cand.append[(i+1, j,v[i+1] + v[j])]
    end if
    if ($j + 1 < i$) then
      cand.append[(i, j+1,v[i] + v[j+1])]
    end if
  end while
  Return NULL {#No successful combination found.}
```

To help illustrate this algorithm, consider a protein N residues long with five mutated amino acid positions and suppose that we are interested in finding the two spheres that capture the most mutations. Without loss of generality and for ease of exposition, number the mutated residues 1 through 5. Further, suppose that after aggregating all the mutational counts from all samples the following mutational counts are recorded: residue 1 - 50 mutations, residue 2 - 40 mutations, residue 3 - 30 mutations, residue 4 - 20 mutations and residue 5 - 10 mutations. For clarity, we assume that the mutation counts are unique, but the algorithm is unaffected if there are identical mutation counts for some residues. We first construct the table shown in Figure 2, where the inside of the matrix is calculated to be the sum of the number of mutations at amino acid i and amino acid j. Observing, that the table is symmetric, we only need to evaluate the residues below the diagonal as the entries on the diagonal originate from residues that overlap each other perfectly. Thus for entry (i, j) we are considering two spheres, one centered at residue i and one centered at residue j.

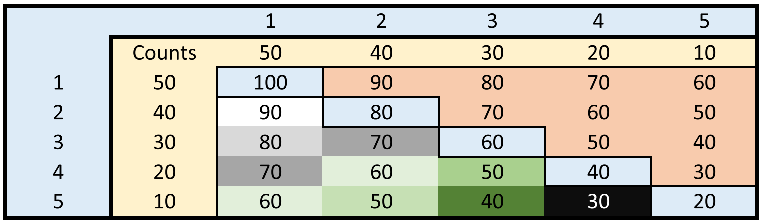
The algorithm proceeds by appending to the "candidate" stack the element directly below and the element directly to the right starting from the (2,1) entry. An "element" consists of a 3-tuple (i, j, s) where i represents the row, j represents the column and s represents the value in position (i, j). After every two potential appends to the stack, the maximum value over the 3rd position is found (with the third position corresponding to the number of mutations covered by both spheres). The two spheres that contribute to this max element are then checked for overlap. If the spheres do *not* overlap, then a successful case has been found and the algorithm completes. If the spheres *do* overlap, the next set of elements are appended and the process continues. By proceeding in the way described in Algorithm 1, at each iteration, the pair of spheres being considered contains the maximum number of mutations possible from the remaining set of sphere combinations. Hence, once the first pair of non-overlapping spheres is found, the optimal sphere combination has been found and the algorithm can terminate. To see this process, see Figure 3.

It is worth noting, that by proceeding as presented above, we are able to traverse the space of possible solutions in a manner that does not require every combination to be explicitly considered and yet at the same time results in a globally optimal solution. At each step, every subsequent combination of spheres will capture fewer mutations (or in very rare cases as many mutations). As soon as the first combination of non-overlapping spheres is found, it is immediately known that all the other remaining sphere combinations will capture fewer mutations. Therefore, there is no longer any need to explicitly consider them. In the worst-case scenario (where the spheres keep overlapping at every iteration), *SpacePAC* will still need to consider all $\binom{N}{3}$ combinations, just as the brute force method. However, by leveraging the fact that at small enough radii the algorithm quickly identifies a case where the spheres do not overlap, the worst-case scenario almost never occurs and the running time is drastically reduced.

Multiple comparison adjustment for structures

A multiple comparison adjustment was performed to account for testing 1,110 protein structures. Since many structures may pertain to one protein, a Bonferroni adjustment is too conservative and an FDR approach was used. Specifically, a rough FDR (rFDR) [29] approach, which is a good approximation to the standard FDR approach [30] when there are a large number of positively correlated or independent tests, was applied. In our case, the cutoff is set at:

$$rFDR = \alpha \left(\frac{k+1}{2k} \right)$$



		1	2	3	4	5
	Counts	50	40	30	20	10
1	50	100	90	80	70	60
2	40	90	80	70	60	50
3	30	80	70	60	50	40
4	20	70	60	50	40	30
5	10	60	50	40	30	20

Figure 2 Organizing the data example. In our example, the protein has 5 residues with mutations. The residues are sorted from largest to smallest (so residue 1 has the largest number of mutations, residue 2 the second largest number of mutations, etc.), and the inside of the table is calculated as the sum of the mutations on both residues using all the samples in the study. For instance, as residue 1 has 50 mutations and residue 2 has 40 mutations, there were 90 total samples that had a mutation either on residue 1 or 2. In the actual code, only the lower half of the table is considered and then only sequentially to decrease running time, but we present the whole table here for clarity.

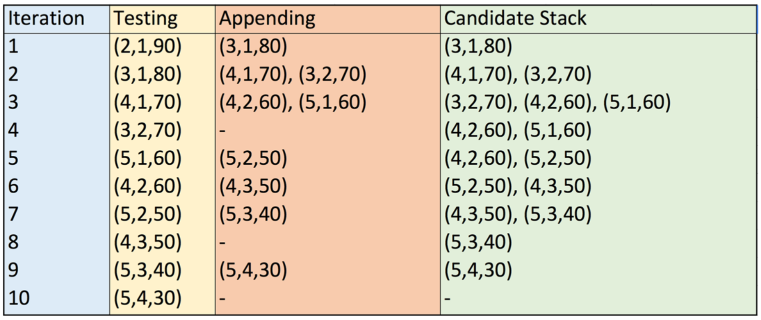
where $k = 1,110$, the total number of structures in the study. Using an $\alpha = 0.05$, the $rFDR \approx 0.025023$. To be conservative, we rounded down and deemed all clusters with a p-value less than or equal to 0.025 to be significant.

Results and discussion

Of the 131 proteins considered, *SpacePAC* identified 18 proteins with significant clustering as shown in Table 1. For a full list of which structures were found significant under *SpacePAC*, *GraphPAC* and *NMC*, see "Additional file 3: Results summary". We note that while Table 1 shows the p-values for the 18 proteins found significant by *SpacePAC*, there were 5 proteins that were found significant only by *GraphPAC*, 1 protein only by *iPAC* and 1 protein only by *NMC* (see "Additional file 4: Results table" for a complete list of what proteins were identified significant under each method). However, we also note that *SpacePAC* identified the largest number of proteins with significant clustering at the same false positive threshold[a]. We further note that several of the proteins identified only by *SpacePAC* have already been associated with cancer as shown in Section "*SpacePAC* identifies additional proteins containing clusters". Furthermore, as shown in Figure 4, 14 out of the 18 proteins identified by *SpacePAC* have their most significant hotspot overlap a biologically relevant region and three of the remaining four proteins (CTNNB1, FGFR3 and FSHR) have been implicated with cancer. For a full description regarding the overlap between *SpacePAC* identified hotspots and structurally significant regions, see "Additional file 5: Relevant sites".

Specifically, for CTNNB1, the *SpacePAC* identified hotspot covers mutations G34R and G34V, which are associated with hepatocellular carcinoma and hepatoblastoma [31,32], respectively. Further, FSHR has been shown to be expressed in the vascular endothelial tissue of a wide range of human tumors including lung, breast, prostate, colon and leiomyosarcoma [33]. For more detail on FGFR3, see Section "*SpacePAC* identifies additional proteins containing clusters".

Finally, we evaluated *SpacePAC* performance via two common machine learning methods, *PolyPhen-2* [16] and *CHASM* [17]. Before we summarize the results, we note that *PolyPhen-2* and *CHASM* utilize a large set of features when evaluating each mutation. The advantage of *SpacePAC* is that it is able to be run with vastly less a priori information. Out of the 38 mutated amino acids that fall within *SpacePAC* identified hotspots, *PolyPhen-2*



Iteration	Testing	Appending	Candidate Stack
1	(2,1,90)	(3,1,80)	(3,1,80)
2	(3,1,80)	(4,1,70), (3,2,70)	(4,1,70), (3,2,70)
3	(4,1,70)	(4,2,60), (5,1,60)	(3,2,70), (4,2,60), (5,1,60)
4	(3,2,70)	-	(4,2,60), (5,1,60)
5	(5,1,60)	(5,2,50)	(4,2,60), (5,2,50)
6	(4,2,60)	(4,3,50)	(5,2,50), (4,3,50)
7	(5,2,50)	(5,3,40)	(4,3,50), (5,3,40)
8	(4,3,50)	-	(5,3,40)
9	(5,3,40)	(5,4,30)	(5,4,30)
10	(5,4,30)	-	-

Figure 3 Algorithm execution example. This figure refers to the data in Figure 2. The first index, *i*, represents the row and the second index, *j*, represents the column. The third index, *s*, represents the total number of mutations at amino acids *i* and *j*. Beginning in position (2,1,90), we then add (3,1,80) to the stack, then {(4,1,70), (3,2,70)} and so forth. After each addition to the stack, we pick the element with the highest value in the third position of the 3-tuple.

Table 1 Summary of genes with significant clusters

	Method			
Gene	SpacePAC	iPAC	GraphPAC	NMC
AKT1	<1.00 E-03	4.48 E-04	5.54 E-04	5.55 E-04
ALK	<1.00 E-03	1.99 E-42	3.89 E-35	2.16 E-21
BRAF	<1.00 E-03	<2.23 E-308	<2.23 E-308	<2.23 E-308
CHRM2	1.10 E-02			
CTNNB1	2.00 E-03	6.69 E-03	1.09 E-03	1.09 E-03
DOCK2	2.20 E-02	9.29 E-03	3.39 E-03	
FGFR3	2.00 E-02			
FSHR	2.20 E-02			
HRAS	<1.00 E-3	1.55 E-23	1.87 E-32	2.68 E-15
IDE	2.00 E-02	3.60 E-03		
IGF2R	1.40 E-02		3.06 E-03	9.04 E-03
KIF18A	6.00 E-03	1.56 E-02	1.56 E-02	1.56 E-02
KRAS	<1.00 E-03	<2.23 E-308	<2.23 E-308	<2.23 E-308
NRAS	<1.00 E-03	1.53 E-75	6.65 E-77	6.77 E-77
PIK3CA	<1.00 E-03	4.73 E-118	4.73 E-118	4.73 E-118
PTEN	<1.00 E-03	5.71 E-03	7.60 E-04	1.68 E-04
SEC23A	1.70 E-02		1.18 E-02	
TP53	<1.00 E-3	1.78 E-134	6.22 E-169	1.08 E-88

This table shows the p-value of the most significant cluster for each of the 18 proteins identified by *SpacePAC* as well as the corresponding p-value under *iPAC*, *GraphPAC* and *NMC*. A blank entry in position (i, j) signifies that methodology j did not find any structures with significant clustering for protein i. Note that given n_i total mutations for protein i, the pairwise methodologies perform $\frac{n_i(n_i-1)}{2}$ comparisons, one for each pair of mutations. As such, the p-values shown for *iPAC*, *GraphPAC* and *NMC* have been multiplied by $\frac{n_i(n_i-1)}{2}$ in order to account for the multiple comparison and provide a number directly comparable to the *SpacePAC* p-value. Also, while the *GraphPAC* methodology was run under all three insertion methods (Cheapest, Nearest and Farthest) as described by [25], we display the minimum p-value over the three methods. Note that since 1000 simulations were used for *SpacePAC*, the most significant result we can report is $p < 1.00\text{E}-03$. Please see Figure 1 for further explanation.

identifies 36 (95%) as damaging while *CHASM* identifies 31 (82%) as driver mutations at a FDR of 20%. On the protein level, *PolyPhen-2* identifies all the 18 proteins identified by *SpacePAC* as significant while *CHASM* identifies 14 proteins as significant. Moreover, *SpacePAC* identifies several proteins with significant clustering that are missed by the machine learning methods. For instance, *SpacePAC* identifies FSHR as significant, which, as described above, has recently been associated with cancer. However, *CHASM* calculates a FDR of 0.45 for FSHR, which is above the significance threshold. See "Additional file 6: Performance evaluation" for more information.

SpacePAC identifies additional proteins containing clusters

As described in Section "Results and discussion", *SpacePAC* identified three proteins with significant clustering that were missed by *NMC*, *iPAC* and *GraphPAC*. We will now consider two of these proteins, Fibroblast Growth Factor Receptor 3 (FGFR3), and Muscarinic Acetylcholine Receptor M2 (CHRM2 or M2).

The CHRM2 structure (PDB ID: 3UON) [34] was identified by *SpacePAC* as having two significant mutational hotspots (p-value = 0.011) located at residues 52 and 144 (see Figure 5). CHRM2, essential for the physiological regulation of cardiovascular function [34] has been implicated in a variety of cardiovascular diseases. Recently, CHRM2 has also been associated with both autoimmune diseases and cancer [35]. Current research shows that M2 receptors are expressed in both glioblastoma cell lines and human samples. Moreover, the M2 agonist arecaidine strongly decreases cell proliferation in both primary cultures and cell lines in a dose and time dependent response profile. This suggests that M2 activation has an impor-

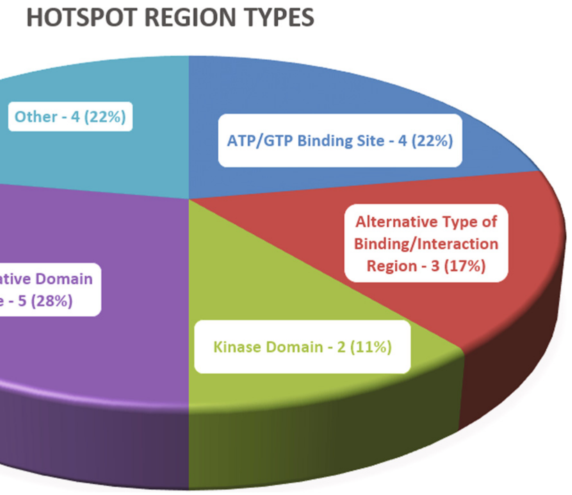


Figure 4 Results summary. A breakout of what biologically relevant regions are overlapped by the most significant cluster for each of the 18 proteins. Overall, 77% of the hotspots overlap a binding site or a protein domain.

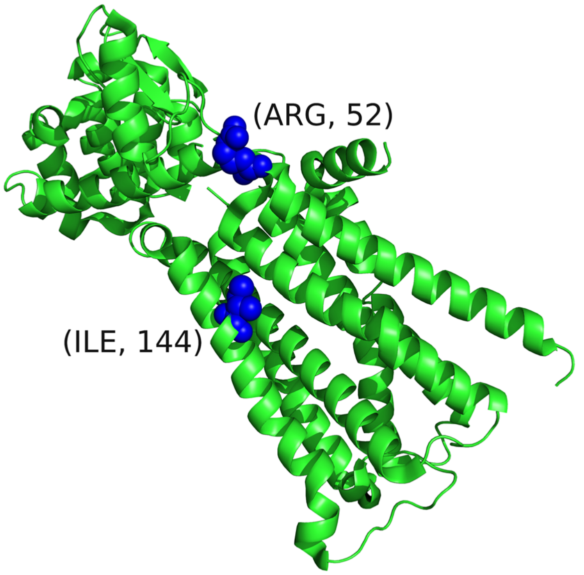


Figure 5 CHRM 2 clustering. The CHRM2 structure (PDB ID: 3UON) where residues 52 and 144 are highlighted.

tant role in suppressing glioma cell proliferation and can provide a novel therapeutic target [36]. Had the spatial structure not been taken into account, as under *NMC*, or if the structure was accounted for but only via remapping to 1D space followed by a pairwise multiple comparison adjustment, as under *iPAC* and *GraphPAC*, this cluster would have been missed.

SpacePAC identified the FGFR3 structure (PDB ID: 1RY7) [37] as having one significant hotspot (p-value = 0.020) centered at amino acid 248 (see Figure 6). FGFR3 is a tyrosine-protein kinase which plays a critical role in regulating cell differentiation, proliferation and apoptosis and is often associated with cancer and developmental disorders [38]. Mutation R248C occurs in the Ig-like domain and is a severe and lethal mutation associated with bladder cancer [39] along with a variety of other phenotypes such as thanatophoric dwarfism [40] and epidermal nevi [41]. This cluster represents a perfect example of signal loss when all pairwise mutations are considered. In the case of our data, as all the mutations occur on one residue, a cluster is formed at that one residue only. There is therefore no difference between any of the pairwise methods as the remapping step has no effect. However, since *iPAC*, *GraphPAC* and *NMC* need to account for all pairwise comparisons between mutations (all occurring on residue 248), the signal is lost under all three methods. On the other hand, as *SpacePAC* does not need to perform such a correction, it is successfully able to detect the cluster. Moreover, we note that [42] recently developed an anti-FGFR3 monoclonal antibody that interferes with FGFR3 binding and inhibits R248C [39].

SpacePAC improves cluster localization

For protein-structure combinations in which mutational clusters are detected by other methods, *SpacePAC* is often able to provide a smaller set of clusters compared to the pairwise methods while still covering the majority of mutations. To illustrate this point we consider two examples, the BRAF structure (PDB ID: 3Q96) [43] where *SpacePAC* identifies 3 mutational hotspots and the ALK structure (PDB ID: 2XBA) [44] where *SpacePAC* identifies 2 mutational hotspots.

BRAF is a well known oncogene that is part of the RAS-RAF-MEK-ERK-MAP kinase pathway which is often activated in human tumors. Further, it is estimated that approximately 90% of mutations in this gene are a substitution of a glutamate for a valine at residue 600 (V600E) [45]. In our mutational data, 187 (83.5%) mutations were on residue 600 with the remaining 37 mutations spread over 13 other residues. Mutations on V600 typically result in constitutively elevated kinase activity and have been found in a wide range of cancers such as metastatic melanoma, ovarian carcinoma and colorectal carcinoma

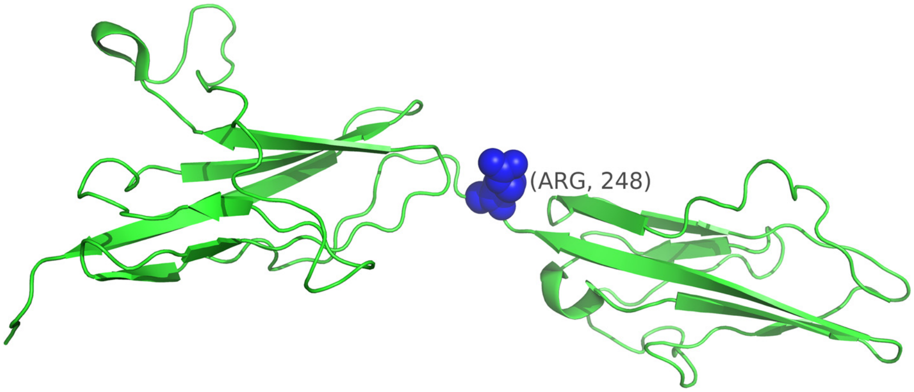


Figure 6 FGFR3 clustering. The FGFR3 structure (PDB ID: 1RY7) where residue 248 is highlighted blue.

[5,7,46-49]. Due to the large number of V600 mutations, *SpacePAC* and all the pairwise methods identified residue 600 as the most significant "cluster" in all structures where tertiary information was available for that residue. It is worth noting that BRAF V600 inhibitors, such as Vemurafenib, have already been developed, further supporting the hypothesis that mutational clusters may represent pharmaceutical targets [50].

As the signal presented by V600 is so strong, it may mask the signal from other mutations within the BRAF protein. As such, we considered structure 3Q96 which does not have tertiary information for residues 600 and 601. Of the remaining 28 mutations spread over 12 residues, *SpacePAC* identifies three hot spots with 7-8 mutations per cluster as shown in Table 2 (combined hot spot p-value <1.00 E-03). Moreover, each of the three regions identified have been associated with oncogenic elevated kinase activity as well as a variety of cancers such as lung adenocarcinoma, melanoma, colorectal adenocarcinoma and ovarian serous carcinoma [5,46,51]. See Figure 7 for a visual orientation of the clusters presented in Table 2.

Together, the three hot spots identified by *SpacePAC* cover 79% of the mutations for which tertiary information is available. Moreoever, while *NMC*, *iPAC* and the three *GraphPAC* methods report approximately 8 to 16 times as many clusters as *SpacePAC* (see Table 3), the additional clusters only cover the remaining 21% of mutations. Finally, all the residues that do not fall within *SpacePAC* hot spots are those which have only one mutation. These additional clusters stem from the fact that *NMC*, *iPAC* and *GraphPAC* must consider every pairwise combination of mutations resulting in many clusters that only differ from each other by a few residues. Further, by considering every pairwise combination, many smaller clusters are often combined into larger clusters with a less significant p-value. While technically still a "significant" cluster, these extra clusters provide little additional information. As *SpacePAC* does not have to consider every combination, it does not suffer from this issue.

We now consider the Anaplastic Lymphoma Kinase (ALK) protein for which *SpacePAC* identifies two mutational hotspots (combined hot spot p-value <0.001) (see Table 4). The ALK protein is a receptor-type tyrosine kinase that is preferentially expressed in neurons during the late embryonic stages [52]. Mutations in this protein have been associated with both neuroblastoma as well as non-small cell lung cancer [53,54]. Hotspots A and B in Table 4 both occur in the protein kinase. Further, mutations on F1174 and R1275 can cause constitutive activation which impairs receptor trafficking [55]. We note that *SpacePAC* perfectly identifies both hotspot locations. Moreover, it has recently been shown that activating mutations F1174L and R1275Q provide therapeutic targets in neuroblastoma [54], supporting the hypothesis

Table 2 BRAF clusters

Hotspot	Center	Within sphere	# Mutations
A	465	(464, 465, 466)	8
B	470	(469, 470, 471)	7
C	596	(595, 596, 597)	7

The three hot spots identified by *SpacePAC* for the BRAF structure (PDB ID: 3Q96) at an optimal radius of 4Å. See Figure 7 for a visual orientation.

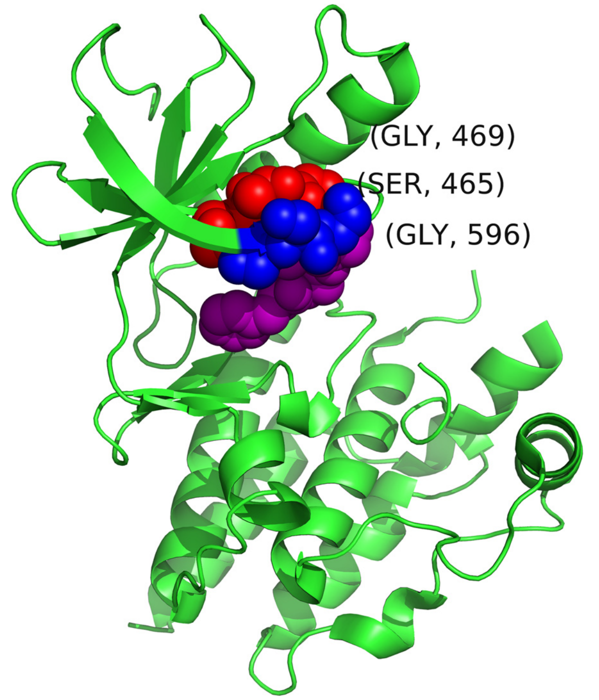


Figure 7 BRAF clustering. The BRAF structure (PDB ID: 3Q96) where cluster 464-466 is shown in blue, 469-471 is shown in red and 595-597 is shown in purple. The central residue in each cluster (465, 470 and 596 for the blue, red and purple clusters, respectively) is labeled.

Table 3 Methodology comparison for BRAF

	# Clusters/Hotspots	
Method/ Protein-structure	BRAF - 3Q96	ALK-2XBA
SpacePAC	3	2
NMC	22	11
iPAC	45	11
GraphPAC-Cheapest	36	7
GraphPAC-Nearest	45	8
GraphPAC-Farthest	47	6

The number of clusters found under each method. As *SpacePAC* does not need to consider every pairwise combination of mutations, the method provides a much smaller number of potential hotspots while still covering the majority of mutations.

Table 4 ALK clusters

Hotspot	Center	Within sphere	# Mutations
A	1174	(1173, 1174, 1175)	11
B	1275	(1274, 1275, 1276)	12

The two hot spots identified by *SpacePAC* as significant for the ALK Structure (PDB ID: 2XBA) [44] at an optimal radius of 4Å. See Figure 8 for a visual orientation.

that mutational clusters are indicative of activating mutations. See Figure 8 for a visual orientation of the clusters presented in Table 4.

The two hotspots identified by *SpacePAC* cover 88.5% of all the mutations in our data with the remaining mutations occurring on residues with only one mutation each. Further, as seen in Table 3, the pairwise methods have 3 to 5.5 times as many clusters as *SpacePAC*. As before, by not having to consider every pairwise combination and thus not reporting similar clusters, *SpacePAC* is able to better localize the critical mutational areas.

Conclusion

In this article we provide a novel algorithm to account for protein tertiary structure when identifying mutational clusters in proteins. By considering the protein structure directly in 3D space, we avoid the use of a dimension reduction algorithm and potential information loss. Further, by not considering every pair of mutations, we are able to reduce the multiple comparison penalty and identify additional clusters. We show several examples of clusters that are not identified by alternative methods as well as the ability to improve cluster localization while still covering the majority of mutations. Moreover, several of our examples identified clusters which overlap potential therapeutic targets, supporting the hypothesis that clusters may be indicative of activating driver mutations. Finally, since the *SpacePAC* methodology does not need to look at every pairwise combination of mutations it also runs much faster for proteins with many (> 400) mutations. In these situations, while the pairwise methods may take several days to complete, *SpacePAC* still finishes in a matter of hours. For proteins with fewer mutations, the running time of all the pairwise methods as well as *SpacePAC* is comparable, with the majority of protein/structure combinations terminating in under 10 minutes when executed on a consumer desktop with an Intel i7-2600k processor (at a frequency of 3.40 GHZ) and 16 GB of DDR3 RAM.

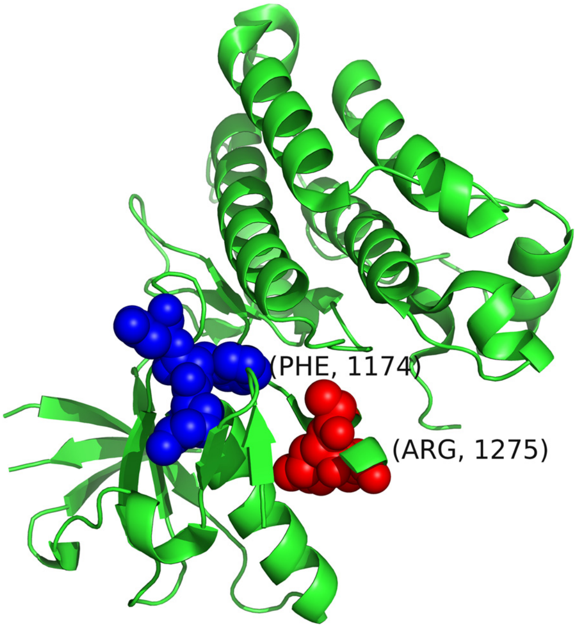


Figure 8 ALK clustering. The ALK structure (PDB ID: 2XBA) where cluster 1173-1175 is shown in blue and cluster 1274-1276 is shown in red. The central residue in each cluster (1174 and 1275 for the blue and red clusters respectively) is labeled.

SpacePAC, while presenting an important alternative to the one dimensionality restriction required by *iPAC* and *GraphPAC*, is nonetheless subject to several limitations. First, *SpacePAC* is currently limited to at most three mutational hotspots to save on computational time. While it is unlikely that a single structure will have more than three hotspots, the extension to allow *SpacePAC* to account for more than three hotspots is algorithmically simple. As *SpacePAC* was able to process our entire database of protein/structure combinations in under 5 hours (with all the structures evaluated in parallel), this restriction is minimal and will only grow smaller as computational power increases.

Second, to satisfy the uniformity assumption, the mutational status of each amino acid must be known. However, due to improvements in high-throughput sequencing, this is rapidly becoming a non-issue. Next, unequal rates of mutagenesis in specific genomic regions may violate the assumption that each residue has an equal probability of mutation. To help ensure that our data met this statistical assumption, we only considered missense mutations as many insertion and deletion mutations are sequence dependent. Relatedly, while the literature shows that CpG dinucleotides often have a mutational rate ten times or higher when compared to other locations [56], approximately only 14% of the clusters presented in section "*SpacePAC* identifies additional proteins containing clusters" and "*SpacePAC* improves cluster localization" overlapped CpG sites. Similarly, cigarette use typically causes transversion mutations within lung carcinomas [23] while colorectal carcinomas result in transition mutations [57]. In the case of KRAS however, the vast majority of mutations are located on residues 12, 13 and 61 for both cancers. This signifies that while the mutational type may differ, the impact on mutation location is minimal and does not violate the uniformity assumption. We also note that while we used a uniform mutational distribution for our simulations (as the goal of this paper was to present a novel technique that utilizes tertiary protein structure), this is not a requirement of the overall approach. Prior knowledge of varying mutation rates can be integrated into the simulation step described in Section “Identifying mutational hotspots”. This would allow for increased accuracy when identifying clusters by potentially increasing both the specificity and sensitivity of the statistic. Further research is required in this area to realize the full benefit of such an approach. Lastly, it is worth noting that as we obtained our mutational data from the COSMIC database, specific tissue types may be more represented than others. However, under this situation our analysis would be more conservative and the resulting findings even more significant. Specifically, aggregating over all tissues increases the total number of mutations, thus increasing the number of simulated mutations within any given segment of the protein. The resulting p-value of any observed hotspot would lose significance as more mutations would be simulated within the cluster. Overall, while this as well as previous studies are impacted by several external factors, it appears that selection of the cancer phenotype is the primary cause of clustering.

In conclusion, *SpacePAC* presents a novel approach to account for protein tertiary structure when identifying mutational hotspots. We show that *SpacePAC* identifies novel clusters of biological relevance, improves cluster localization and in several cases identifies pharmaceutical targets for which therapies are already in production. In turn, we further confirm the hypothesis that mutational hotspots are indicative of driver mutations and show that *SpacePAC* can be used to quickly locate such potential mutations as additional structures are published. Several novel areas for further research are directly applicable as well. First, while the algorithm developed here uses up to three spheres, the methodology can be expanded to use more spheres when identifying clustering. Further, the approach proposed here is not limited to identifying human mutational clusters in proteins, but can be extended to other species as well as DNA and RNA, once the requisite positional information is available.

Endnote

[a] The *GraphPAC* algorithm was run using each of the three insertion methods described in [25]. While the methodology is the same, the nature of the algorithm leads to different results when different insertion methods are considered. *SpacePAC* outperformed *GraphPAC* in comparison to each of the individual insertion methods.

Additional files

Additional file 1: Cosmic query. Shows the SQL query used to extract mutations from the COSMIC database.

Additional file 2: Structure files. A list of which protein-structure-chain combinations were considered for the study.

Additional file 3: Results summary. Shows the significant structures and their p-values under the *iPAC*, *SpacePAC*, *NMC* and *GraphPAC* methodologies.

Additional file 4: Results table. A summary of which proteins were identified as significant under *SpacePAC* and each of the pairwise methods.

Additional file 5: Relevant sites. A description of which of the *SpacePAC* hotspots fall within structurally relevant regions.

Additional file 6: Performance evaluation. In-depth analysis of the *SpacePAC* hotspots when evaluated using *PolyPhen-2* and *CHASM*.

Competing interests

The authors declare that they have no competing interests.

Authors’ contributions

GR, RB, DZ and HZ developed the *SpacePAC* methodology. KC and YC were responsible for obtaining the mutation data from the COSMIC database. GR and YC executed the methodology on the protein structures. GR drafted the original manuscript while KC, YC, YM, DZ and HZ were responsible for revisions. HZ finalized the manuscript. All authors have read and approved the final text.

Acknowledgements

This work was supported in part by NSF Grant DMS 1106738 (GR, HZ), NIH Grants GM59507 and CA154295 (HZ), and GM102869 (YM).

Author details

[1]Department of Biostatistics, Yale School of Public Health, New Haven, CT, USA. [2]Program of Computational Biology and Bioinformatics, Yale University, New Haven, CT, USA. [3]Yale Center for Medical Informatics, Yale School of Medicine, New Haven, CT, USA. [4]Department of Computer Science, Yale University, New Haven, CT, USA. [5]Department of Molecular Biophysics & Biochemistry, Yale University, New Haven, CT, USA.

Received: 15 November 2013 Accepted: 27 May 2014
Published: 3 July 2014

References

1. Vogelstein B, Kinzler KW: **Cancer genes and the pathways they control.** *Nat Med* 2004, **10**(8):789–799. [http://www.ncbi.nlm.nih.gov/pubmed/15286780]. [PMID: 15286780]
2. Faivre S, Kroemer G, Raymond E: **Current development of mTOR, inhibitors as anticancer agents.** *Nat Rev Drug Discov* 2006, **5**(8):671–688. [http://www.nature.com/doifinder/10.1038/nrd2062]
3. Hartmann JT, Haap M, Kopp HG, Lipp HP: **Tyrosine kinase inhibitors - a review on pharmacology, metabolism and side effects.** *Curr Drug Metab* 2009, **10**(5):470–481. [PMID:, 1968 9244]
4. Moreau P, Richardson PG, Cavo M, Orlowski RZ, San Miguel JF, Palumbo A, Harousseau JL: **Proteasome inhibitors in multiple myeloma: 10 years later.** *Blood* 2012, **120**(5):947–959. [http://www.bloodjournal.org/cgi/doi/10.1182/blood-2012-04-403733]
5. Greenman C, Stephens P, Smith R, Dalgliesh GL, Hunter C, Bignell G, Davies H, Teague J, Butler A, Stevens C, Edkins S, O’Meara S, Vastrik I, Schmidt EE, Avis T, Barthorpe S, Bhamra G, Buck G, Choudhury B, Clements J, Cole J, Dicks E, Forbes S, Gray K, Halliday K, Harrison R, Hills K, Hinton J, Jenkinson A, Jones D, Menzies A, Mironenko T, et al.: **Patterns of somatic mutation in human cancer genomes.** *Nature* 2007, **446**(7132):153–158. [http://www.nature.com/doifinder/10.1038/nature05610]
6. Weinstein IB, Joe AK: **Mechanisms of disease: Oncogene addiction–a rationale for molecular targeting in cancer therapy.** *Nat Clin Prac Oncol* 2006, **3**(8):448–457. [http://www.ncbi.nlm.nih.gov/pubmed/16894390]. [PMID: 16894390]

7. Sjöblom T, Jones S, Wood LD, Parsons DW, Lin J, Barber TD, Mandelker D, Leary RJ, Ptak J, Silliman N, Szabo S, Buckhaults P, Farrell C, Meeh P, Markowitz SD, Willis J, Dawson D, Willson JKV, Gazdar AF, Hartigan J, Wu L, Liu C, Parmigiani G, Park BH, Bachman KE, Papadopoulos N, Vogelstein B, Kinzler KW, Velculescu VE: **The consensus coding sequences of human breast and colorectal cancers.** *Science (New York NY)* 2006, **314**(5797):268–274. [PMID: 16959974]
8. Bardelli A, Parsons DW, Silliman N, Ptak J, Szabo S, Saha S, Markowitz S, Willson JKV, Parmigiani G, Kinzler KW, Vogelstein B, Velculescu VE: **Mutational analysis of the tyrosine kinome in colorectal cancers.** *Science (New, York, N.Y.)* 2003, **300**(5621):949. [http://www.ncbi.nlm.nih.gov/pubmed/12738854]. [PMID: 12738854]
9. Wang T: **Prevalence of somatic alterations in the colorectal cancer cell genome.** *Proc Nat Acad Sci* 2002, **99**(5):3076–3080. [http://www.pnas.org/cgi/doi/10.1073/pnas.261714699]
10. Youn A, Simon R: **Identifying cancer driver genes in tumor genome sequencing studies.** *Bioinformatics* 2010, **27**(2):175–181. [http://bioinformatics.oxfordjournals.org/cgi/doi/10.1093/bioinformatics/btq630]
11. Kreitman M: **Methods to detect selection in populations with applications to the human.** *Ann Rev Genom Hum Genet* 2000, **1:**539–559. [http://www.annualreviews.org/doi/abs/10.1146/annurev.genom.1.1.539]
12. Lynch TJ, Bell DW, Sordella R, Gurubhagavatula S, Okimoto RA, Brannigan BW, Harris PL, Haserlat SM, Supko JG, Haluska FG, Louis DN, Christiani DC, Settleman J, Haber DA: **Activating mutations in the epidermal growth factor receptor underlying responsiveness of non-small-cell lung cancer to gefitinib.** *New Engl J Med* 2004, **350**(21):2129–2139. [PMID: 15118073]
13. Torkamani A, Schork NJ: **Prediction of cancer driver mutations in protein kinases.** *Cancer Res* 2008, **68**(6):1675–1682. [http://www.ncbi.nlm.nih.gov/pubmed/18339846]. [PMID: 18339846]
14. Wagner A: **Rapid detection of positive selection in genes and genomes through variation clusters.** *Genetics* 2007, **176**(4):2451–2463. [http://www.genetics.org/cgi/doi/10.1534/genetics.107.074732]
15. Zhou T, Enyeart PJ, Wilke CO: **Detecting clusters of mutations.** *PLoS, ONE* 2008, **3**(11):e3765. [http://dx.plos.org/10.1371/journal.pone.0003765]
16. Adzhubei IA, Schmidt S, Peshkin L, Ramensky VE, Gerasimova A, Bork P, Kondrashov AS, Sunyaev SR: **A method and server for predicting damaging missense mutations.** *Nat Methods* 2010, **7**(4):248–249. [http://www.nature.com/doifinder/10.1038/nmeth0410-248]
17. Carter H, Chen S, Isik L, Tyekucheva S, Velculescu VE, Kinzler KW, Vogelstein B, Karchin R: **Cancer-specific high-throughput annotation of somatic mutations: computational prediction of driver missense mutations.** *Cancer Res* 2009, **69**(16):6660–6667. [http://cancerres.aacrjournals.org/cgi/doi/10.1158/0008-5472.CAN-09-1133]
18. Breiman L: **Random forests.** *Mach Learn* 2001, **45:**5–32. [http://dx.doi.org/10.1023/A]
19. Friedman N, Geiger D, Goldszmidt M: **Bayesian network classifiers.** *Mach Learn* 1997, **29**(2-3):131–163.
20. Cortes C, Vapnik V: **Support-vector networks.** *Mach Learn* 1995, **20**(3):273–297. [http://www.springerlink.com/index/10.1007/BF00994018]
21. Reva B, Antipin Y, Sander C: **Predicting the functional impact of protein mutations: application to cancer genomics.** *Nucleic Acids Res* 2011, **39**(17):e118–e118. [http://www.nar.oxfordjournals.org/cgi/doi/10.1093/nar/gkr407]
22. Ng PC, Henikoff S: **Predicting deleterious amino acid substitutions.** *Genome Res* 2001, **11**(5):863–874. [PMID: 11337480].
23. Ye J, Pavlicek A, Lunney EA, Rejto PA, Teng C: **Statistical method on nonrandom clustering with application to somatic mutations in cancer.** *BMC Bioinformatics* 2010, **11:**[http://www.biomedcentral.com/1471-2105/11/11]
24. Ryslik G, Cheng Y, Cheung kH, Modis Y, Zhao H: **Utilizing protein structure to identify non-random somatic mutations.** *pre-print* 2013. [ArXiv:1302.6977 [q-bio.GN]].
25. Ryslik GA, Cheng Y, Cheung KH, Modis Y, Zhao H: **A graph theoretic approach to utilizing protein structure to identify non-random somatic mutations.** *pre-print* 2013. [ArXiv:1303.5889]
26. Borg I, Groenen PJF: *Modern multidimensional scaling : theory and applications*. New York: Springer; 1997.
27. Consortium TU: **Reorganizing the protein space at the Universal Protein Resource (UniProt).** *Nucleic Acids Res* 2011, **40**(D1):D71–D75. [http://www.nar.oxfordjournals.org/cgi/doi/10.1093/nar/gkr981]
28. Pages H, Aboyoun P, Gentleman R, DebRoy S: **Biostrings: String objects representing biological sequences, and matching algorithms.** 2012. [R package version 2.24.1]
29. Gong Y, Kakihara Y, Krogan N, Greenblatt J, Emili A, Zhang Z, Houry WA: **An atlas of chaperone-protein interactions in Saccharomyces cerevisiae: implications to protein folding pathways in the cell.** *Mol Syst Biol* 2009, **5:**. [http://www.nature.com/doifinder/10.1038/msb.2009.26]
30. Benjamini Y, Hochberg Y: **Controlling the false discovery rate: a practical and powerful approach to multiple testing.** *J R Stat Soc Series B (Methodological)* 1995, **57:**289–300. [http://www.jstor.org/stable/2346101]
31. Legoix P, Bluteau O, Bayer J, Perret C, Balabaud C, Belghiti J, Franco D, Thomas G, Laurent-Puig P, Zucman-Rossi J: **Beta-catenin mutations in hepatocellular carcinoma correlate with a low rate of loss of heterozygosity.** *Oncogene* 1999, **18**(27):4044–4046. [PMID: 10435629]
32. Koch A, Denkhaus D, Albrecht S, Leuschner I, von Schweinitz D, Pietsch T: **Childhood hepatoblastomas frequently carry a mutated degradation targeting box of the beta-catenin gene.** *Cancer Res* 1999, **59**(2):269–273. [PMID: 9927029]
33. Siraj A, Desestret V, Antoine M, Fromont G, Huerre M, Sanson M, Camparo P, Pichon C, Planeix F, Gonin J, Radu A, Ghinea N: **Expression of follicle-stimulating hormone receptor by the vascular endothelium in tumor metastases.** *BMC, Cancer* 2013, **13:**246. [http://www.biomedcentral.com/1471-2407/13/246]
34. Haga K, Kruse AC, Asada H, Yurugi-Kobayashi T, Shiroishi M, Zhang C, Weis WI, Okada T, Kobilka BK, Haga T, Kobayashi T: **Structure of the human M2 muscarinic acetylcholine receptor bound to an antagonist.** *Nature* 2012, **482**(7386):547–551. [http://www.nature.com/doifinder/10.1038/nature10753]. [PDB ID: 3UON]
35. Ockenga W, Kühne S, Bocksberger S, Banning A, Tikkanen R: **Non-neuronal functions of the M2 muscarinic acetylcholine receptor.** *Genes* 2013, **4**(2):171–197. [http://www.mdpi.com/2073-4425/4/2/171/]
36. Ferretti M, Fabbiano C, Bari MD, Conte C, Castigli E, Sciaccaluga M, Ponti D, Ruggieri P, Raco A, Ricordy R, Calogero A, Tata AM: **M2 receptor activation inhibits cell cycle progression and survival in human glioblastoma cells.** *J Cell Mol Med* 2013, **17**(4):552–566. [http://doi.wiley.com/10.1111/jcmm.12038]
37. Olsen SK: **Insights into the molecular basis for fibroblast growth factor receptor autoinhibition and ligand-binding promiscuity.** *Proc Nat Acad Sci* 2004, **101**(4):935–940. [http://www.pnas.org/cgi/doi/10.1073/pnas.0307287101]. [PDB ID: 1RY7]
38. Hart KC, Robertson SC, Donoghue DJ: **Identification of tyrosine residues in constitutively activated fibroblast growth factor receptor 3 involved in mitogenesis, stat activation, and Phosphatidylinositol 3-Kinase activation.** *Mol Biol Cell* 2001, **12**(4):931–942. [http://www.molbiolcell.org/cgi/doi/10.1091/mbc.12.4.931]
39. Hadari Y, Schlessinger J: **FGFR3-targeted mAb therapy for bladder cancer and multiple myeloma.** *J Clin Investig* 2009, **119**(5):1077–1079. [http://www.jci.org/articles/view/38948]
40. Rousseau F: **Missense FGFR3 mutations create cysteine residues in thanatophoric dwarfism type I (TD1).** *Hum Mol Genet* 1996, **5**(4):509–512. [http://www.hmg.oupjournals.org/cgi/doi/10.1093/hmg/5.4.509]
41. Hafner C: **Mosaicism of activatingFGFR3 mutations in human skin causesepidermal nevi.** *J Clin Investig* 2006, **116**(8):2201–2207. [http://www.jci.org/cgi/doi/10.1172/JCI28163]
42. Qing J, Du X, Chen Y, Chan P, Li H, Wu P, Marsters S, Stawicki S, Tien J, Totpal K, Ross S, Stinson S, Dornan D, French D, Wang QR, Stephan JP, Wu Y, Wiesmann C, Ashkenazi A: **Antibody-based targeting of FGFR3 in bladder carcinoma and t(4;14)-positive multiple myeloma in mice.** *J Clin Investig* 2009, **119**(5):1216–1229. [http://www.jci.org/articles/view/38017]
43. Gould AE, Adams R, Adhikari S, Aertgeerts K, Afroze R, Blackburn C, Calderwood EF, Chau R, Chouitar J, Duffey MO, England DB, Farrer C, Forsyth N, Garcia K, Gaulin J, Greenspan PD, Guo R, Harrison SJ, Huang SC, Iartchouk N, Janowick D, Kim MS, Kulkarni B, Langston SP, Liu JX, Ma LT, Menon S, Mizutani H, Paske E, Renou CC, et al.: **Design and optimization**

of potent and orally bioavailable tetrahydronaphthalene raf i. *J Med Chem* 2011, **54**(6):1836–1846. [http://pubs.acs.org/doi/abs/10.1021/jm101479y]. [PDB ID: 3Q96]

44. Bossi RT, Saccardo MB, Ardini E, Menichincheri M, Rusconi L, Magnaghi P, Orsini P, Avanzi N, Borgia AL, Nesi M, Bandiera T, Fogliatto G, Bertrand JA: **Crystal structures of anaplastic lymphoma kinase in complex with ATP competitive inhibitors.** *Biochemistry* 2010, **49**(32):6813–6825. [http://pubs.acs.org/doi/abs/10.1021/bi1005514]. [PDB ID: 2XBA]
45. Tan YH, Liu Y, Eu KW, Ang PW, Li WQ, Salto-Tellez M, Iacopetta B, Soong R: **Detection of BRAF V600E mutation by pyrosequencing.** *Pathology* 2008, **40**(3):295–298. [http://content.wkhealth.com/linkback/openurl?sid=WKPTLP:landingpage&an=01268031-200840030-00009]
46. Davies H, Bignell GR, Cox C, Stephens P, Edkins S, Clegg S, Teague J, Woffendin H, Garnett MJ, Bottomley W, Davis N, Dicks E, Ewing R, Floyd Y, Gray K, Hall S, Hawes R, Hughes J, Kosmidou V, Menzies A, Mould C, Parker A, Stevens C, Watt S, Hooper S, Wilson R, Jayatilake H, Gusterson BA, Cooper C, Shipley J, et al: **Mutations of the BRAF gene in human cancer.** *Nature* 2002, **417**(6892):949–954. [http://www.ncbi.nlm.nih.gov/pubmed/12068308]. [PMID: 12068308]
47. Rajagopalan H, Bardelli A, Lengauer C, Kinzler KW, Vogelstein B, Velculescu VE: **Tumorigenesis: RAF/RAS oncogenes and mismatch-repair status.** *Nature* 2002, **418**(6901):934. [http://www.ncbi.nlm.nih.gov/pubmed/12198537]. [PMID: 12198537]
48. Hingorani SR, Jacobetz MA, Robertson GP, Herlyn M, Tuveson DA: **Suppression of BRAF(V599E) in human melanoma abrogates transformation.** *Cancer Res* 2003, **63**(17):5198–5202. [http://www.ncbi.nlm.nih.gov/pubmed/14500344]. [PMID: 14500344]
49. Andreu-Pérez P, Esteve-Puig R, de Torre-Minguela C, López-Fauqued M, Bech-Serra JJ, Tenbaum S, García-Trevijano ER, Canals F, Merlino G, Avila MA, Recio JA: **Protein arginine methyltransferase 5 regulates ERK1/2, signal transduction amplitude and cell fate through CRAF.** *Sci Signal* 2011, **4**(190):ra58. [http://www.ncbi.nlm.nih.gov/pubmed/21917714]. [PMID: 21917714]
50. Mao M, Tian F, Mariadason JM, Tsao CC, Lemos R, Dayyani F, Gopal YNV, Jiang ZQ, Wistuba II, Tang XM, Bornman WG, Bollag G, Mills GB, Powis G, Desai J, Gallick GE, Davies MA, Kopetz S: **Resistance to BRAF inhibition in BRAF-mutant colon cancer can be overcome with PI3K inhibition or demethylating agents.** *Clin Cancer Res* 2012, **19**(3):657–667. [http://clincancerres.aacrjournals.org/cgi/doi/10.1158/1078-0432.CCR-11-1446]
51. Lee JW, Yoo NJ, Soung YH, Kim HS, Park WS, Kim SY, Lee JH, Park JY, Cho YG, Kim CJ, Ko YH, Kim SH, Nam SW, Lee JY, Lee SH: **BRAF mutations in non-Hodgkin's lymphoma.** *Br J Cancer* 2003, **89**(10):1958–1960. [http://www.ncbi.nlm.nih.gov/pubmed/14612909]. [PMID: 14612909]
52. Motegi A: **ALK receptor tyrosine kinase promotes cell growth and neurite outgrowth.** *J Cell Sci* 2004, **117**(15):3319–3329. [http://jcs.biologists.org/cgi/doi/10.1242/jcs.01183]
53. Bang YJ: **Treatment of ALK-positive non-small cell lung cancer.** *Arch Pathol & Lab Med* 2012, **136**(10):1201–1204. [PMID: 23020724]
54. George RE, Sanda T, Hanna M, Fröhling S, Ii WL, Zhang J, Ahn Y, Zhou W, London WB, McGrady P, Xue L, Zozulya S, Gregor VE, Webb TR, Gray NS, Gilliland DG, Diller L, Greulich H, Morris SW, Meyerson M, Look AT: **Activating mutations in ALK provide a therapeutic target in neuroblastoma.** *Nature* 2008, **455**(7215):975–978. [http://www.nature.com/doifinder/10.1038/nature07397]
55. Mazot P, Cazes A, Boutterin MC, Figueiredo A, Raynal V, Combaret V, Hallberg B, Palmer RH, Delattre O, Janoueix-Lerosey I, Vigny M: **The constitutive activity of the ALK mutated at positions F1174 or R1275 impairs receptor trafficking.** *Oncogene* 2011, **30**(17):2017–2025. [http://www.nature.com/doifinder/10.1038/onc.2010.595]
56. Sved J, Bird A: **The expected equilibrium of the CpG dinucleotide in vertebrate genomes under a mutation model.** *Proc Nat Acad Sci* 1990, **87**(12):4692–4696. [http://www.pnas.org/content/87/12/4692.abstract]
57. Hollstein M, Sidransky D, Vogelstein B, Harris CC: **p53 mutations in human cancers.** *Science (New, York, N.Y.)* 1991, **253**(5015):49–53. [http://www.ncbi.nlm.nih.gov/pubmed/1905840]. [PMID: 1905840]

doi:10.1186/1471-2105-15-231
Cite this article as: Ryslik *et al.*: A spatial simulation approach to account for protein structure when identifying non-random somatic mutations. *BMC Bioinformatics* 2014 **15**:231.